41st Computational Complexity Conference

CCC 2026, August 3–6, 2026, Lisbon, Portugal

Edited by

Dana Moshkovitz





Editors

Dana Moshkovitz
University of Texas at Austin, TX, USA
danama@utexas.edu

ACM Classification 2012
Theory of computation

ISBN 978-3-95977-437-6

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at https://www.dagstuhl.de/dagpub/978-3-95977-437-6.

Publication date
July, 2026

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

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

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.CCC.2026.0

LIPIcs – Leibniz International Proceedings in Informatics

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

Editorial Board

- Christel Baier (TU Dresden, DE)
- Roberto Di Cosmo (Inria and Université Paris Cité, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Holger Hermanns (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)
- Daniel Král' (Leipzig University, DE and Max Planck Institute for Mathematics in the Sciences, Leipzig, DE – Chair)
- Sławomir Lasota (University of Warsaw, PL)
- Meena Mahajan (Institute of Mathematical Sciences, Chennai, IN)
- Chih-Hao Luke Ong (Nanyang Technological University, SG)
- Eva Rotenberg (IT University of Copenhagen, DK)
- Pierre Senellart (ENS, Université PSL, Paris, France)
- Alexandra Silva (Cornell University, Ithaca, US)

ISSN 1868-8969

https://www.dagstuhl.de/lipics

Contents

Preface
Dana Moshkovitz 0:ix

Conference Organization
..... 0:xi

External Reviewers
..... 0:xiii

List of Authors
..... 0:xv

Regular Papers

Tight Lower Bound for Approximating Parametrized Maximum Likelihood Decoding Under ETH
Rishav Gupta, Bingkai Lin, and Xin Zheng 1:1–1:17

Bounded-Independence Sampling of Edges for Combinatorial Graph Properties
Aaron Putterman, Salil Vadhan, and Vadim Zaripov 2:1–2:22

Improved Bounds on the Space Complexity of Circuit Evaluation
Yakov Shalunov 3:1–3:13

Probabilistically Checking Quantum Proofs, with Interaction
Baocheng Sun and Thomas Vidick 4:1–4:49

Condensing and Extracting Against Online Adversaries
Eshan Chattopadhyay, Mohit Gurumukhani, Noam Ringach, and Rocco A. Servedio 5:1–5:22

Improved Parallel Repetition for GHZ-Supported Games via Spreadness
Yang P. Liu, Shachar Lovett, and Kunal Mittal 6:1–6:21

The Log-Rank Conjecture: New Equivalent Formulations
Lianna Hambardzumyan, Shachar Lovett, and Morgan Shirley 7:1–7:9

Efficient Adversaries
Erfan Khaniki, Ján Pich, and Dmitry Sokolov 8:1–8:32

Hardness of Computing Nondeterministic Kolmogorov Complexity
Jinqiao Hu, Zhenjian Lu, and Igor C. Oliveira 9:1–9:50

The Rate-Immediacy Barrier in Explicit Tree Code Constructions
Gil Cohen, Leonard J. Schulman, and Piyush Srivastava 10:1–10:22

Beyond Bilinear Complexity: What Works and What Breaks with Many Modes?
Cornelius Brand, Radu Curticapean, Petteri Kaski, Baitian Li, Ian Orzel, Tim Seppelt, and Jiaheng Wang 11:1–11:23

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

Fine-Grained Complexity for Quantum Problems from Size-Preserving Circuit-To-Hamiltonian Constructions
Nai-Hui Chia, Atsuya Hasegawa, François Le Gall, and Yu-Ching Shen 12:1–12:35

Resolution Width Lifts to Near-Quadratic-Depth Res(⊕) Size
Dmitry Itsykson, Vladimir Podolskii, and Alexander Shekhovtsov 13:1–13:27

A Weak Regularity Lemma for Polynomials
Guy Moshkovitz and Dora Woodruff 14:1–14:24

Derandomised Tensor Product Gap Amplification for Quantum Hamiltonians
Thiago Bergamaschi, Tony Metger, Thomas Vidick, and Tina Zhang 15:1–15:22

Constant-Depth Circuits for Polynomial GCD over Any Characteristic
Somnath Bhattacharjee, Mrinal Kumar, Shanthanu S. Rai, Varun Ramanathan, Ramprasad Saptharishi, and Shubhangi Saraf 16:1–16:21

Rank Bounds and Polynomial-Time PIT for $\Sigma^k\Pi\Sigma\Pi^2$ Circuits
Abhibhav Garg, Rafael Oliveira, Akash Kumar Sengupta, Nir Shalmon, and Amir Shpilka 17:1–17:18

Multiplicative Pseudorandom Generators for Nondeterministic Circuits
Alon Dermer and Ronen Shaltiel 18:1–18:20

When Hilbert Approximates: A Strong Nullstellensatz for Approximate Polynomial Satisfiability
Sanyam Agarwal, Sagnik Dutta, Anurag Pandey, and Himanshu Shukla 19:1–19:18

On Factorization of Sparse Polynomials of Bounded Individual Degree
Aminadav Chuyoon and Amir Shpilka 20:1–20:20

Wide Replacement Products Meet Gray Codes: Toward Optimal Small-Bias Sets
Gil Cohen and Itay Cohen 21:1–21:27

Multilinear Algebraic Branching Programs and the Min-Partition Rank Method
Théo Borém Fabris, Nutan Limaye, Srikanth Srinivasan, and Amir Yehudayoff ... 22:1–22:20

Quantum–Classical Equivalence for AND-Functions
Sreejata Kishor Bhattacharya, Farzan Byramji, Arkadev Chattopadhyay, Yogesh Dahiya, and Shachar Lovett 23:1–23:24

All Polynomial Generators Preserve Distance with Mutual Correlated Agreement
Sarah Bordage, Alessandro Chiesa, Ziyi Guan, and Ignacio Manzur 24:1–24:18

Polynomial Identity Testing for Read-4 Arithmetic Formulas
Nimrod Kaplan and Amir Shpilka 25:1–25:18

A Simple Sub-Polynomial Degree Coboundary Expander
Max Hopkins and Arka Ray 26:1–26:41

Optimal Depth-Three Circuits for Inner Product
Mohit Gurumukhani, Daniel Kleber, Ramamohan Paturi, Christopher Rosin, and Navid Talebanfard 27:1–27:25

Plethysm is in #BQP
Matthias Christandl, Aram W. Harrow, Greta Panova, Pietro M. Posta, and Michael Walter 28:1–28:11

Frontier Space-Time Algorithms Using Only Full Memory
Petr Chmel, Aditi Dudeja, Michal Koucký, Ian Mertz, and Ninad Rajgopal 29:1–29:23

Bounds for Hardness Condensation in the Query Model
Chandrima Kayal, Rajat Mittal, Sai Soumya Nalli, Manaswi Paraashar, Karthikeya Polisetty, Jayalal Sarma, and Nitin Saurabh 30:1–30:20

Optimal Testing of Reed-Muller Codes with an Online Adversary
Esty Kelman, Uri Meir, and Kai Zhe Zheng 31:1–31:26

Fixed-Parameter Degree Bounds and Complexity of the Orbit Closure Intersection Problem for Tensors
M. Levent Doğan, John Maar, Rafael Oliveira, and Youming Qiao 32:1–32:21

Multilinear Formula Lower Bounds for Sparse Determinants
Pruthvi Boyapati, Suryajith Chillara, and Pratyush Vempati 33:1–33:21

Quantum Advantage in Tolerant Junta Testing
Avishay Tal and Weiqiang Yuan 34:1–34:17

Trace Hermitian Codes Have Vanishing Bias
Swastik Kopparty, Amnon Ta-Shma, and Kedem Yakirevitch 35:1–35:28

Asymptotic Rank Speedup Theorems, Revisited
Josh Alman and Baitian Li 36:1–36:42

Separations Above TFNP from Sherali-Adams Lower Bounds
Noah Fleming, Anna Gál, Deniz Imrek, and Christophe Marciot 37:1–37:23

Non-Clifford Gates Are Required for Long-Term Memory
Jon Nelson, Joel Rajakumar, and Michael J. Gullans 38:1–38:13

Faux Determinism
Shalev Ben-David and M. H. Ebtehaj 39:1–39:25

ETH-Hardness of Learning Monotone Circuits and Approximating Their Size
Bruno Cavalar, Susanna F. de Rezende, Matthew Gray, and Rahul Santhanam ... 40:1–40:25

Systematic Data Structure Lower Bounds via the Query-With-Sketch Model
Sumegha Garg, Songhua He, Yuanzhi Li, Periklis A. Papakonstantinou, and Xin Yang 41:1–41:44

Randomized Separations in Black-Box TFNP
Fedor Kiselev 42:1–42:17

Preface

The papers in this volume were accepted for presentation at the 41st Computational Complexity Conference (CCC 2026), held from August 3–6, 2026, in Lisbon, Portugal. The conference is organised by the Computational Complexity Foundation (CCF) in cooperation with the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT) and the European Association for Theoretical Computer Science (EATCS).

The call for papers sought original research papers in all areas of complexity theory. Of the 124 submissions, the program committee selected 42 for presentation at the conference.

The program committee would like to thank everyone involved in the conference, including all those who submitted papers for consideration as well as the reviewers (listed separately) for their scientific contributions; the board of trustees of the Computational Complexity Foundation for their advice and assistance; the Local Arrangements Committee Bruno Loff (chair), Diogo Poças and João Ribeiro; and Michael Wagner and Michael Didas for coordinating the production of the proceedings.

Dana Moshkovitz
Program Committee Chair, on behalf of the Program Committee

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

Conference Organization

Program Committee

Scott Aaronson, The University of Texas at Austin
Mitali Bafna, University of Washington
Lijie Chen, UC Berkeley
Karthik C.S., Rutgers University
Dean Doron, Ben-Gurion University
Mika Göös, EPFL
Tom Gur, University of Cambridge
Pavel Hrubeš, Institute of Mathematics, Czech Academy of Sciences
Dana Moshkovitz (Chair), The University of Texas at Austin
Jakob Nordström, University of Copenhagen and Lund University
Ben Rossman, Duke University
Makrand Sinha, University of Illinois at Urbana-Champaign
Nitin Saxena, IIT Kanpur
Ronald de Wolf, CWI, University of Amsterdam and Google research

Local Arrangements Committee

Bruno Loff (Chair), University of Lisbon
Diogo Poças, University of Lisbon
João Ribeiro, University of Lisbon

Board of Trustees

Mahdi Cheraghchi, University of Michigan
Pooya Hatami, Ohio State University
Antonina Kolokolova, Memorial University of Newfoundland
Shachar Lovett (President), University of California San Diego
Raghu Meka, University of California Los Angeles
Susanna de Rezende, Lund University
Benjamin Rossman, Duke University
Shubhangi Saraf, University of Toronto
David Zuckerman, The University of Texas at Austin

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

External Reviewers

Sanyam Agarwal
Aryan Agarwala
Omar Alrabiah
Prashanti Anderson
Robert Andrews
Noel Arteche
Albert Atserias
Zongbo Bao
Amik Raj Behera
Amey Bhangale
Vishwas Bhargava
Joshua Brody
Farzan Byramji
Ramya C
Clément Canonne
Bruno Cavalar
Sayak Chakrabarti
Diptarka Chakraborty
Abhranil Chatterjee
Prerona Chatterjee
Yanlin Chen
Yeyuan Chen
Nai-Hui Chia
Suryajith Chillara
Anirban Chowdhury
Sayak Ray Chowdhury
Gil Cohen
Roni Con
Arjan Cornelissen
Shalev Ben David
Ben Davis
Yotam Dikstein
Krishnamoorthy Dinesh
Yangjing Dong
Pranjal Dutta
Prateek Dwivedi
David Engström
Yuval Filmus
Samuel Fiorini
Noah Fleming
Abhibhav Garg
Yassine Ghannane
Sevag Gharibian
Prantar Ghosh
Uma Girish
Guy Goldberg
Halley Goldberg
Noam Goldgraber
Alexander Golovnev
Rohan Goyal
Stefan Grosser
Svyatoslav Gryaznov
Nikhil Gupta
Mohit Gurumukhani
Tal Herman
Shuichi Hirahara
John Hitchcock
Alexandros Hollender
Kaave Hosseini
Jun-Ting Hsieh
Pavel Hubáček
Yeongwoo Hwang
Christian Ikenmeyer
Rahul Ilango
Deniz Imrek
Siddarth Iyer
Vishnu Iyer
Siddhartha Jain
Duri Andrea Janett
Fernando Granha Jeronimo
Shunhua Jiang
Gorav Jindal
Valentine Kabanets
Erfan Khaniki
Swastik Kopparty
Oliver Korten
Arpita Korwar
Mrinal Kumar
Vinayak Kumar
Dmitriy Kunisky
Deepanshu Kush
Zehua Lai
Jiatu Li
Jiawei Li
Xin Li
Yuhao Li
Nutan Limaye
Yupan Liu
Bruno Loff
Zhenjian Lu

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

Pasin Manurangsi
Gal Maor
Christophe Marciot
Damian Markham
Noam Mazor
Ruta Mehta
Ian Mertz
Tony Metger
Dor Minzer
Tushant Mittal
Sidhanth Mohanty
Geoffrey Mon
Shivam Nadimpalli
Mikito Nanashima
Quynh T Nguyen
Ojas Parekh
Michal Parnas
Subhasree Patro
Arthur Pesah
Jan Pich
Aaron Potechin
Subin Pulari
Edward Pyne
Luowen Qian
Roshan Raj
Ninad Rajgopal
Varun Ramanathan
Hanlin Ren
Artur Riazanov
Kilian Risse
Robert Robere
Gregory Rosenthal
Rodrigo Rubio
Akhil S
Jayalal Sarma
Suronjona Sarma
Raghuvansh Saxena
Nikhil Shagrithaya
Suhail Sherif
Morgan Shirley
Devansh Shringi
Noah G. Singer
Amit Sinhababu
Dmitry Sokolov
Shashank Srivastava
Sathya Subramanian
Avishay Tal
Navid Talebanfard
Roei Tell
Anamay Tengse
Harsha Srimath Tirumala
Hans Raj Tiwary
Dante Tjowasi
Madhur Tulsiani
Quinten Tupker
Iddo Tzameret
Francisca Vasconcelos
Santhoshini Velusamy
Madhavan Venkatesh
S. Venkitesh
Agi Villanyi
Marc Vinyals
Emanuele Viola
Ben Lee Volk
Peter Vrana
Shuo Wang
Xinzhao Wang
Omri Weinstein
Kewen Wu
Pei Wu
Jeff Xu
Zhiyang Xun
Tianqi Yang
Penghui Yao
Eylon Yogev
Huacheng Yu
Weiqiang Yuan
Or Zamir
Hadas Zeilberger
Jiapeng Zhang
Peng Zhang
Rachel Yun Zhang
Tina Zhang
Zihan Zhang
Kai Zhe Zheng
Yan Zhong
Renfei Zhou
Jeroen Zuiddam

List of Authors

Sanyam Agarwal (19)
Universität des Saarlandes, Saarbrücken, Germany

Josh Alman (36)
Columbia University, New York, NY, USA

Shalev Ben-David (39)
Institute for Quantum Computing, Waterloo, Canada;
University of Waterloo, Waterloo, Canada

Thiago Bergamaschi (15)
University of California, Berkeley, CA, USA

Somnath Bhattacharjee (16)
University of Toronto, Canada

Sreejata Kishor Bhattacharya (23)
TIFR, Mumbai, India

Sarah Bordage (24)
EPFL, Lausanne, Switzerland

Pruthvi Boyapati (33)
CSTAR, International Institute of Information Technology, Hyderabad, India

Cornelius Brand (11)
University of Regensburg, Germany

Farzan Byramji (23)
UC San Diego, La Jolla, CA, USA

Bruno Cavalar (40)
University of Oxford, UK

Arkadev Chattopadhyay (23)
TIFR, Mumbai, India

Eshan Chattopadhyay (5)
Cornell University, Ithaca, NY, USA

Nai-Hui Chia (12)
Department of Computer Science, Rice University, Houston, TX, USA;
Ken Kennedy Institute, Rice University, Houston, TX, USA;
Smalley-Curl Institute, Rice University, Houston, TX, USA

Alessandro Chiesa (24)
EPFL, Lausanne, Switzerland

Suryajith Chillara (33)
CSTAR, International Institute of Information Technology, Hyderabad, India

Petr Chmel (29)
Computer Science Institute of Charles University, Prague, Czech Republic

Matthias Christandl (28)
University of Copenhagen, Denmark

Aminadav Chuyoon (20)
School of Mathematics, Tel Aviv University, Israel

Gil Cohen (10, 21)
Department of Computer Science, Tel Aviv University, Israel

Itay Cohen (21)
Tel Aviv University, Israel

Radu Curticapean (11)
University of Regensburg, Germany;
IT University of Copenhagen, Denmark

Yogesh Dahiya (23)
UC San Diego, La Jolla, CA, USA

Susanna F. de Rezende (40)
Lund University, Sweden

Alon Dermer (18)
Department of Computer Science, University of Haifa, Israel

M. Levent Doğan (32)
LMU Munich, Germany

Aditi Dudeja (29)
The School of Data Science, The Chinese University of Hong Kong, Shenzhen, China

Sagnik Dutta (19)
Max-Planck-Institut für Informatik, Saarbrücken, Germany

M. H. Ebtehaj (39)
Institute for Quantum Computing, Waterloo, Canada;
University of Waterloo, Waterloo, Canada

Théo Borém Fabris (22)
University of Copenhagen, Denmark

Noah Fleming (37)
Lund University, Sweden;
Columbia University, New York, NY, USA

Abhibhav Garg (17)
University of Waterloo, Canada

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

Sumegha Garg (41)
Rutgers University, New Brunswick, NJ, USA

Matthew Gray (40)
University of Oxford, UK

Ziyi Guan (24)
EPFL, Lausanne, Switzerland

Michael J. Gullans (38)
Joint Center for Quantum Information & Computer Science, University of Maryland, College Park, MD, USA;
National Institute of Standards and Technology, Gaithersburg, MD, USA

Rishav Gupta (1)
National University of Singapore, Singapore

Mohit Gurumukhani (5, 27)
Cornell University, Ithaca, NY, USA

Anna Gál (37)
University of Texas at Austin, TX, USA

Lianna Hambardzumyan (7)
The University of Copenhagen, Denmark

Aram W. Harrow (28)
Massachusetts Institute of Technology, Cambridge, MA, USA

Atsuya Hasegawa (12)
Graduate School of Mathematics, Nagoya University, Japan

Songhua He (41)
Rutgers University, New Brunswick, NJ, USA

Max Hopkins (26)
Institute for Advanced Study, Princeton, NJ, USA

Jinqiao Hu (9)
University of Warwick, UK

Deniz Imrek (37)
University of Texas at Austin, TX, USA

Dmitry Itsykson (13)
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Nimrod Kaplan (25)
Blavatnik School of Computer Science and AI, Tel Aviv University, Israel

Petteri Kaski (11)
Aalto University, Finland

Chandrima Kayal (30)
Université Paris Cité, CNRS, IRIF, Paris, France

Esty Kelman (31)
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA;
Boston University, Cambridge, MA, USA

Erfan Khaniki (8)
University of Oxford, UK

Fedor Kiselev (42)
MIPT, Moscow, Russia

Daniel Kleber (27)
Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA, USA

Swastik Kopparty (35)
Department of Mathematics and Department of Computer Science, University of Toronto, Canada

Michal Koucký (29)
Computer Science Institute of Charles University, Prague, Czech Republic

Mrinal Kumar (16)
Tata Institute of Fundamental Research, Mumbai, India

François Le Gall (12)
Graduate School of Mathematics, Nagoya University, Japan

Baitian Li (11, 36)
Columbia University, New York, NY, USA

Yuanzhi Li (41)
Carnegie Mellon University, Pittsburgh, PA, USA

Nutan Limaye (22)
IT University of Copenhagen, Denmark

Bingkai Lin (1)
State Key Laboratory for Novel Software Technology, Nanjing University, China

Yang P. Liu (6)
Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Shachar Lovett (6, 7, 23)
Department of Computer Science and Engineering, University of California, San Diego, CA, USA

Zhenjian Lu (9)
University of Victoria, Canada

John Maar (32)
Technical University Berlin, Germany

Ignacio Manzur (24)
EPFL, Lausanne, Switzerland

Christophe Marciot (37)
Lund University, Sweden

Uri Meir (31)
Blavatnik School of Computer Science, Tel Aviv University, Israel

Ian Mertz (29)
Computer Science Institute of Charles University, Prague, Czech Republic

Tony Metger (15)
ETH Zurich, Zurich, Switzerland; Simons Institute for the Theory of Computing, Berkeley, CA, USA

Kunal Mittal (6)
Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, NY, USA

Rajat Mittal (30)
Indian Institute of Technology Kanpur, India

Guy Moshkovitz (14)
City University of New York, NY, USA

Sai Soumya Nalli (30)
Microsoft Research India, Bangalore, India

Jon Nelson (38)
Joint Center for Quantum Information & Computer Science, University of Maryland, College Park, MD, USA; NIST, Gaithersburg, MD, USA

Igor C. Oliveira (9)
University of Warwick, UK

Rafael Oliveira (17, 32)
University of Waterloo, Canada

Ian Orzel (11)
University of Copenhagen, Denmark

Anurag Pandey (19)
Universität des Saarlandes, Saarbrücken, Germany

Greta Panova (28)
University of Southern California, Los Angeles, USA

Periklis A. Papakonstantinou (41)
Rutgers University, New Brunswick, NJ, USA

Manaswi Paraashar (30)
Indian Institute of Technology Hyderabad, India

Ramamohan Paturi (27)
Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA, USA

Ján Pich (8)
University of Oxford, UK

Vladimir Podolskii (13)
Tufts University, Medford, MA, USA

Karthikeya Polisetty (30)
Indian Institute of Technology Madras, India

Pietro M. Posta (28)
University of Copenhagen, Denmark

Aaron Putterman (2)
Harvard University, Cambridge, MA, USA

Youming Qiao (32)
University of Technology Sydney, Australia

Shanthanu S. Rai (16)
Tata Institute of Fundamental Research, Mumbai, India

Joel Rajakumar (38)
Joint Center for Quantum Information & Computer Science, University of Maryland, College Park, MD, USA; NIST, Gaithersburg, MD, USA; IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

Ninad Rajgopal (29)
Computer Science Institute of Charles University, Prague, Czech Republic

Varun Ramanathan (16)
University of Copenhagen, Denmark

Arka Ray (26)
Indian Institute of Science, Bengaluru, India

Noam Ringach (5)
Cornell University, Ithaca, NY, USA

Christopher Rosin (27)
Constructive Codes, USA

Rahul Santhanam (40)
University of Oxford, UK

Ramprasad Saptharishi (16)
Tata Institute of Fundamental Research, Mumbai, India

Shubhangi Saraf (16)
University of Toronto, Canada

Jayalal Sarma (30)
Indian Institute of Technology Madras, India

Nitin Saurabh (30)
Indian Institute of Technology Hyderabad, India

Leonard J. Schulman (10)
E&AS Division, California Institute of Technology, Pasadena, CA, USA

Akash Kumar Sengupta (17)
Rutgers University, Piscataway, NJ, USA

Tim Seppelt (11)
IT University of Copenhagen, Denmark

Rocco A. Servedio (5)
Columbia University, New York, NY, USA

Nir Shalmon (17)
Tel Aviv University, Israel

Ronen Shaltiel (18)
Department of Computer Science, University of Haifa, Israel

Yakov Shalunov (3)
Department of Computer Science, University of Chicago, IL, USA

Alexander Shekhovtsov (13)
EPFL, Lausanne, Switzerland

Yu-Ching Shen (12)
Department of Computer Science, Houston, Rice University, Houston, TX, USA

Morgan Shirley (7)
Lund University, Sweden

Amir Shpilka (17, 20, 25)
Tel Aviv University, Israel

Himanshu Shukla (19)
Universität Regensburg, Germany

Dmitry Sokolov (8)
Université de Montréal, Canada

Srikanth Srinivasan (22)
University of Copenhagen, Denmark

Piyush Srivastava (10)
Tata Institute of Fundamental Research, Mumbai, India

Baocheng Sun (4)
Weizmann Institute of Science, Rehovot, Israel

Amnon Ta-Shma (35)
Department of Computer Science, Tel Aviv University, Israel

Avishay Tal (34)
University of California, Berkeley, CA, USA

Navid Talebanfard (27)
University of Sheffield, UK

Salil Vadhan (2)
Harvard University, Cambridge, MA, USA

Pratyush Vempati (33)
CSTAR, International Institute of Information Technology, Hyderabad, India

Thomas Vidick (4, 15)
School of Computer and Communication Sciences and Center for Quantum Science and Engineering, École Polytechnique Fédérale de Lausanne, Switzerland;
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel

Michael Walter (28)
Ludwig-Maximilians-Universität München, Germany;
Munich Center for Quantum Science and Technology (MCQST), Germany;
Korteweg-de Vries Institute for Mathematics, Amsterdam, The Netherlands;
QuSoft, University of Amsterdam, The Netherlands

Jiaheng Wang (11)
University of Regensburg, Germany;
University of Helsinki, Finland

Dora Woodruff (14)
Massachusetts Institute of Technology, Cambridge, MA, USA

Kedem Yakirevitch (35)
Department of Computer Science, Tel Aviv University, Israel

Xin Yang (41)
Meta, Seattle, WA, USA

Amir Yehudayoff (22)
University of Copenhagen, Denmark;
Technion-ITT, Haifa, Israel

Weiqiang Yuan (34)
EPFL, Lausanne, Switzerland

Vadim Zaripov (2)
Harvard University, Cambridge, MA, USA

Tina Zhang (15)
Massachusetts Institute of Technology, Cambridge, MA, USA

Kai Zhe Zheng (31)
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA

Xin Zheng (1)
State Key Laboratory for Novel Software Technology, Nanjing University, China

Tight Lower Bound for Approximating Parametrized Maximum Likelihood Decoding Under ETH

Rishav Gupta
National University of Singapore, Singapore

Bingkai Lin
State Key Laboratory for Novel Software Technology, Nanjing University, China

Xin Zheng
State Key Laboratory for Novel Software Technology, Nanjing University, China

Abstract

We present a simple deterministic reduction which, assuming the Exponential Time Hypothesis (ETH), yields tight lower bounds for approximating the parameterized Maximum Likelihood Decoding problem (MLD) and the parameterized Nearest Codeword Problem (NCP) within some fixed constant factor. Our starting point is the ETH-based exponential-time hardness of (c, s)-Gap MAXLIN established in [7]. We transform a (c, s)-Gap MAXLIN instance into an instance of γ-Gap k-MLD via a novel combinatorial object that we call a *cover family*. We provide both a randomized construction of the required cover families and a subsequent derandomization. Prior to our work, $n^{\Omega(k)}$ hardness for constant-factor approximation was only shown under the randomized Gap Exponential Time Hypothesis Gap-ETH [16], which is a much stronger assumption than ETH. Under ETH, the strongest known lower bound was $n^{\Omega(k/\operatorname{poly}\log k)}$ due to [3]. Unlike previous approaches that rely on reductions from the hardness of approximating 2-CSP, our reduction provides a more direct and conceptually simpler route to achieving the optimal lower bounds.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Maximum Likelihood Decoding, Parameterized Complexity, Hardness of Approximation, Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.1

Related Version *Full Version*: https://arxiv.org/abs/2605.08797

Funding Supported by the "111 Cente" (No. B26023) and NRF grant NRF-NRFI09-0005.

Acknowledgements The authors would like to thank the anonymous reviewers for their comments and suggestions which help to improve the paper significantly. Special thanks to Reviewer B for simplifying the proof for Lemma 29.

1 Introduction

The study of error-correcting codes gives rise to various computational problems. In the context of channel coding, one of the most fundamental tasks is to recover the original message from a signal corrupted by noise. In this problem, which is known as the *Maximum Likelihood Decoding* (MLD) problem, we are given the parity check matrix $\boldsymbol{H} \in \mathbb{F}_q^{d\times n}$ of a linear code, a vector $\boldsymbol{t} \in \mathbb{F}_q^d$, and a parameter $k \in \mathbb{N}$. Our goal is to find a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ of Hamming weight at most k such that $\boldsymbol{Hx} = \boldsymbol{t}$. This problem is computationally equivalent to the *Nearest Codeword Problem* (NCP), where we are given the generator matrix $\boldsymbol{A} \in \mathbb{F}_q^{d\times n}$ and the goal is to find a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ such that $\|\boldsymbol{Ax} - \boldsymbol{t}\|_0 \le k$.


© Rishav Gupta, Bingkai Lin, and Xin Zheng;
licensed under Creative Commons License CC-BY 4.0
41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 1; pp. 1:1–1:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The computational hardness of these problems is well-established. They were proven to be NP-hard decades ago [5] and are known to be hard to approximate within any constant factor in polynomial time [2, 11]. Consequently, attention shifted to the framework of *parameterized complexity* [9], where the runtime is analyzed with respect to the parameter k.

In the parameterized setting, the hardness of the exact problem was settled at first. Downey and Fellows [10] proved that k-MLD parameterized by k is W[1]-hard, ruling out exact FPT algorithms, i.e., algorithms that run in $f(k) \cdot n^{O(1)}$ time, under W[1] $\neq$ FPT. However, the question of whether *approximation* could yield fixed parameter tractability remained a major open problem for over two decades.

This question was finally resolved in a series of breakthrough results. The authors of [6] established that k-NCP is W[1]-hard to approximate within any constant factor. They also proved the parameterized inapproximability of the *Minimum Distance Problem* (MDP) over $\mathbb{F}_2$ and the *Shortest Vector Problem* (SVP) in ℓ_p norm for every $p > 1$. Following this, Bennett, Cheraghchi, Guruswami, and Ribeiro [4] generalized these results, proving the inapproximability of k-MDP over all finite fields and k-SVP in general ℓ_p norms. These works painted a comprehensive picture of the FPT inapproximability of coding problems. With the W[1]-hardness established, the focus has shifted to the fine-grained complexity:

What is the precise running time required to approximate k-MLD when k is small?

Under the standard gap-free hypothesis, the *Exponential Time Hypothesis* (ETH), [6] ruled out $n^{O((\log k)^{1/2.01})}$-time algorithms for constant gap k-NCP. Their reduction incurred an exponential blow-up in the parameter, which results in the relatively weak lower bound. Very recently, Li, Lin, and Liu [15] developed a new gap-creating reduction with polynomial parameter growth, hence improving the lower bound under standard ETH, ruling out $n^{o(k^{1/3})}$-time algorithms that solve γ-Gap k-NCP for any $1 < \gamma < \frac{3}{2}$. Through a self-reduction, they further ruled out $n^{o(k^{\varepsilon})}$-time algorithms where $\varepsilon = \frac{1}{\text{poly}\log\gamma}$ for any constant gap $\gamma > 1$. While this was a major step forward, a substantial gap remains between the lower bound and the trivial $n^{O(k)}$ upper bound.

To date, the only tight lower bound comes from the work of Manurangsi [16], which showed that no approximation algorithm exists for these problems in time $n^{o(k)}$ for any constant factor. However, this result relies on the randomized *Gap-Exponential Time Hypothesis* (Gap-ETH), which is a significantly stronger assumption than ETH.

The above-mentioned results establish hardness for *all* constant approximation factors. However, even obtaining an optimal lower bound under ETH for *some* constant approximation factor remained open. Prior to our work, the strongest known ETH-based hardness for *some* approximation factor was established via a chain of reductions: first, by invoking the hardness of approximating k-variable 2-CSP [12, 3], then reducing k-variable 2-CSP to k-Exact Cover [13], and finally reducing k-Exact Cover to γ-Gap k-MLD$_q$ [2]. This approach yields suboptimal running-time lower bounds: namely $n^{k/2^{\Omega(\sqrt{\log k}\cdot\log\log k)}}$ via [12] and $n^{k/\log^{C}(k)}$ via [3].

Our Contribution. In this work, we prove that the tight $n^{\Omega(k)}$ lower bound holds from some constant approximation factor under the standard non-deterministic Exponential Time Hypothesis, removing the need for the stronger Gap-ETH assumption.

▶ **Theorem 1** (Main Result). *Assuming* ETH*, there exist constants $\gamma > 1$ and $\delta > 0$, such that for any algorithm that takes as input a matrix $\boldsymbol{H} \in \mathbb{F}_q^{d\times N}$, a vector $\boldsymbol{u} \in \mathbb{F}_q^d$ and a parameter $k \in \mathbb{N}$, it must take $N^{\delta k}$ time to distinguish the following two cases:*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{F}_q^N$ with $\|\boldsymbol{x}\|_0 \leq k$ such that $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{u}$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{F}_q^N$ with $\|\boldsymbol{x}\|_0 \leq \gamma k$, we have $\boldsymbol{H}\boldsymbol{x} \neq \boldsymbol{u}$.*

If we allow k to be some function of N, the proof of Theorem 1 actually gives an $N^{\delta k}$-time lower bound for γ-Gap k-MLD_q when $k \leq O((\log\log N)^{0.49})$. Assuming randomized ETH, we can obtain the same lower bound for $k \leq N^{\varepsilon}$ for some constant $\varepsilon > 0$.

▶ **Theorem 2.** *Assuming randomized* ETH*, there exist constants $\gamma > 1$ and $\delta, \varepsilon > 0$, such that for any algorithm that takes as input a matrix $\boldsymbol{H} \in \mathbb{F}_q^{d \times N}$, a vector $\boldsymbol{u} \in \mathbb{F}_q^d$ and an integer $k \in \mathbb{N}$ such that $2 \leq k \leq N^{\varepsilon}$, it must take $N^{\delta k}$ time to distinguish the following two cases:*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{F}_q^N$ with $\|\boldsymbol{x}\|_0 \leq k$ such that $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{u}$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{F}_q^N$ with $\|\boldsymbol{x}\|_0 \leq \gamma k$, we have $\boldsymbol{H}\boldsymbol{x} \neq \boldsymbol{u}$.*

Our Techniques. Our hardness proof builds upon the ETH-hardness of (c, s)-Gap MAXLIN_q established by [7]. In this problem, one must distinguish whether a system of linear equations is almost satisfiable (a c-fraction of equations hold), or is far from satisfiable (at most an s-fraction hold). Specifically, the result of [7] shows that assuming ETH, there is no algorithm that can solve (c, s)-Gap MAXLIN_q in time $2^{o(n)}$. By considering the dual of a hard (c, s)-Gap MAXLIN_q instance, we first obtain a hard instance of γ-Gap MLD_q, where the goal is to determine whether a target vector can be expressed as a linear combination of at most ℓ columns from a matrix, or it requires at least $\gamma\ell$ columns. This duality transformation implies that the non-parameterized version of γ-Gap k-MLD_q also cannot be solved in time $2^{o(n)}$.

The core of our contribution is a reduction from this non-parameterized γ-Gap MLD_q to its parameterized version γ-Gap k-MLD_q. We let each vector in the output k-MLD instance be the sum of ℓ/k vectors in the original MLD instance. Thus, a solution of size ℓ in the MLD instance can be represented by only k vectors in the k-MLD instance. However, the naive brute-force enumeration of all combinations of size ℓ/k produces an instance of size $N = \binom{n}{\ell/k}$, hence the reduction yields only an $N^{\Omega(k/\log k)}$ lower bound for γ-Gap k-MLD_q.

We achieve a tight lower bound via a novel combinatorial structure we term *cover families*. A cover family is a sufficiently small collection of subsets $\mathcal{S}$ over a universe $\mathcal{U}$, such that any *small* subset of $\mathcal{U}$ can be expressed as the union of k pairwise disjoint sets from $\mathcal{S}$, whereas no *large* subset of $\mathcal{U}$ can be expressed as the union of even γk sets from $\mathcal{S}$. To construct suitable cover families, we introduce an intermediate combinatorial object called *balanced partition families*. A balanced partition family is a collection of partitions $\mathcal{P}$ of a universe $\mathcal{U}$ such that every partition in the family is roughly balanced, and every subset of $\mathcal{U}$ of an appropriate size is (almost) equipartitioned by some partition $P \in \mathcal{P}$. We first show that independently sampling random partitions suffices to obtain the desired balanced partition family. We then give a deterministic construction of such balanced partition families using hypercube set systems. Finally we use a standard reduction from γ-Gap k-MLD_q to γ-Gap k-NCP_q to obtain the hardness of the latter.

Future Directions. Assuming ETH, we obtain optimal lower bounds for γ-Gap k-MLD_q and γ-Gap k-NCP_q for some constant approximation factor $\gamma > 1$. A natural question is whether this lower bound can be extended to an arbitrary constant factor assuming only ETH. In contrast, under Gap-ETH the corresponding conclusion is known to hold for every constant $\gamma > 1$ via [16]. Establishing such a result under ETH would close the current knowledge gap between ETH and Gap-ETH through the lens of parameterized coding-theoretic problems. It is worth mentioning that [3] rules out $n^{k/\log^C k}$-time algorithms that approximates k-NCP and k-CVP to any constant factor under ETH. We also wonder whether $n^{\Omega(k)}$ lower bound for γ-Gap k-MDP or other related problems can be established under ETH.

Another open direction is to understand what further consequences can be derived from the ETH-hardness of (c, s)-Gap MAXLIN_q established in [7]. Beyond our work, the only other result we are aware of that leverages this hardness result is [1]. Lastly, one can also explore other applications of the combinatorial objects, *cover family* and *balanced partition family* which are introduced in this paper.

Paper Organization. In Section 2 we define necessary notation and introduce useful tools from the literature. First in Section 3 we present a reduction from (c, s)-Gap MAXLIN_q to γ-Gap k-MLD_q using *cover family*. Then in Section 4 we give a randomized as well as a deterministic construction of a suitable *cover family* using an intermediate combinatorial object, *balanced partition family*.

2 Preliminaries

We begin by formally defining the computational problems that will be studied throughout the paper. For each problem, we specify its input, the underlying computational goal, and any associated promise conditions.

2.1 Computational Problems

We first define the standard k-SAT problem. For an integer $k \geq 2$, a k-SAT formula over n boolean variables is the conjunction of clauses, where each clause is the disjunction of k literals. That is, k-SAT formulas have the form $\bigwedge_{i=1}^{m} \bigvee_{j=1}^{k} b_{i,j}$, where $b_{i,j} = x_k$ or $b_{i,j} = \neg x_k$ for some boolean variable x_k.

▶ **Definition 3** (k-SAT). *For any $k \geq 2$, the decision problem k-SAT is defined as follows. The input is a k-SAT formula. It is a YES instance if there exists an assignment to the variables that makes the formula evaluate to true and a NO instance otherwise.*

We write k-SAT_C for a k-SAT instance where each variable x_i is contained in at most C clauses. We also define the corresponding optimization version Max-k-SAT.

▶ **Definition 4** (Max-k-SAT). *For any $k \geq 2$, the decision problem Max-k-SAT is defined as follows. The input is a k-SAT formula and an integer $S \geq 1$. It is a YES instance if there exists an assignment to the variables such that at least S of the clauses evaluate to true and a NO instance otherwise.*

▶ **Definition 5** ((c, s)-Gap MAXLIN_q). *A (c, s)-Gap MAXLIN_q instance $(\boldsymbol{A}, \boldsymbol{b})$ consists of an $m \times n$ matrix $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$ and a vector $\boldsymbol{b} \in \mathbb{F}_q^m$. The objective of the problem is to distinguish between the following cases.*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{F}_q^n$ such that $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_0 \leq (1 - c)m$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{F}_q^n$, $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_0 > (1 - s)m$.*

We need to mention that this problem is equivalent to the non-parameterized $\frac{1-s}{1-c}$-Gap MLD_q. Now we define the dual problem of MAXLIN, Maximum Likelihood Decoding Problem.

▶ **Definition 6** (γ-Gap MLD_q). *A γ-Gap MLD_q instance $(\boldsymbol{H}, \boldsymbol{u}, \ell)$ consists of a $d \times n$ matrix $\boldsymbol{H} \in \mathbb{F}_q^{d \times n}$, a target vector $\boldsymbol{u} \in \mathbb{F}_q^d$ and a value ℓ. The goal is to distinguish between the following cases.*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{F}_q^n$ with $\|\boldsymbol{x}\|_0 \leq \ell$ such that $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{u}$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{F}_q^n$ with $\|\boldsymbol{x}\|_0 \leq \gamma\ell$, we have $\boldsymbol{H}\boldsymbol{x} \neq \boldsymbol{u}$.*

We now define the parametrized version of γ-Gap MLD_q, which is almost the same as its non-parameterized version, just using $k := \ell$ as the parameter.

▶ **Definition 7** (γ-Gap k-MLD_q). *A γ-Gap k-MLD_q instance $(\boldsymbol{H}, \boldsymbol{u})$ consists of a $d \times n$ matrix $\boldsymbol{H} \in \mathbb{F}_q^{d\times n}$, a target vector $\boldsymbol{u} \in \mathbb{F}_q^d$ and a parameter $k \in \mathbb{N}$. The goal is to distinguish between the following cases.*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{F}_q^n$ with $\|\boldsymbol{x}\|_0 \le k$ such that $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{u}$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{F}_q^n$ with $\|\boldsymbol{x}\|_0 \le \gamma k$, we have $\boldsymbol{H}\boldsymbol{x} \neq \boldsymbol{u}$.*

Next we turn to the parameterized Nearest Codeword Problem, which is closely related to MLD.

▶ **Definition 8** (γ-Gap k-NCP_q). *A γ-Gap k-NCP_q instance $(\boldsymbol{A}, \boldsymbol{t})$ consists of a $n \times d$ matrix $\boldsymbol{A} \in \mathbb{F}_q^{n\times d}$, a target vector $\boldsymbol{t} \in \mathbb{F}_q^n$ and a parameter $k \in \mathbb{N}$. The goal is to distinguish between the following cases.*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{F}_q^d$ such that $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{t}\|_0 \le k$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{F}_q^d$, $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{t}\|_0 > \gamma k$.*

Similarly we define parametrized versions of Closest Vector Problem (CVP). For $\boldsymbol{x} \in \mathbb{Z}^d$ we will use the notation $\|\boldsymbol{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$, in the following definition.

▶ **Definition 9** (γ-Gap k-CVP_p). *A γ-Gap k-CVP_p instance $(\boldsymbol{A}, \boldsymbol{t})$ consists of a $n \times d$ matrix $\boldsymbol{A} \in \mathbb{Z}^{n\times d}$, a target vector $\boldsymbol{t} \in \mathbb{Z}^n$ and a parameter $k \in \mathbb{N}$. The goal is to distinguish between the following cases.*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{Z}^d$ such that $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{t}\|_p \le k$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{Z}^d$, $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{t}\|_p > \gamma k$.*

2.2 Fine-grained Hardness Assumptions

We introduce the following fine-grained hardness assumptions, Exponential time hypothesis ETH and its corresponding gap version $\mathsf{Gap\text{-}ETH}$ below.

▶ **Definition 10** (Exponential Time Hypothesis (ETH), [14]). *There exists $\delta > 0$ such that any algorithm which solves 3-SAT must take $2^{\delta n}$ time.*

▶ **Definition 11** ($\mathsf{Gap\text{-}ETH}$, [8]). *There exists $\delta > 0$ and $0 < \eta < 1$ such that given a 3-SAT instance with n variables and m clauses, any algorithm which can distinguish between the cases if all m clauses are satisfiable and one in which no assignment satisfies more than η-fraction of the clauses, must take $2^{\delta n}$ time.*

We now state the Sparsification Lemma.

▶ **Lemma 12** (Sparsification Lemma, [14]). *Let $\varepsilon > 0$, $k \ge 3$ be constants. There is a $2^{\varepsilon n} \cdot poly(n)$ time algorithm that takes a k-CNF F on n variables and produces $F_1, \ldots, F_{2^{\varepsilon n}}$, $2^{\varepsilon n}$ k-CNFs such that F is satisfied if and only if $\bigvee_i F_i$ is satisfied and each F_i has n variables and $n \cdot \left(\frac{k}{\varepsilon}\right)^{O(k)}$ clauses. In fact, each variable is in at most $\mathrm{poly}\left(\frac{1}{\varepsilon}\right)$ clauses, and the F_i are over the same variables as F.*

We will now state the following result from [7] which gives us a very strong starting point for establishing hardness results.

▶ **Theorem 13** ([7], Theorem 6.3). *For every finite field $\mathbb{F}_q$, there exists constants c and s where $0 < s < c < 1$, such that the following holds. There exists a polynomial time reduction which takes a 3-SAT_C instance with n variables, and outputs a (c, s)-Gap MAXLIN_q with $n' = O(n)$ variables and $m' = O(n)$ equations.*

The sparsification Lemma 12 and Tovey's reduction [17] together tell us that if ETH is true, then 3-SAT$_4$ over n variables must take $2^{\delta n}$ time for some $\delta > 0$. Together with Theorem 13, we find that if ETH holds, then for some $0 < s < c < 1$ and $C > 0$, any algorithm which solves (c, s)-Gap MAXLIN$_q$ with n variables and $m \leq Cn$ clauses, must take $2^{\delta n}$ time for some $\delta > 0$. We state it as the following corollary.

▶ **Corollary 14** ((c, s)-Gap MAXLIN$_q$ is ETH-Hard). *For every finite field $\mathbb{F}_q$, there exists constants $0 < s < c < 1$ and $C > 0$ such that unless* ETH *is false, any algorithm for (c, s)-Gap* MAXLIN$_q$ *with n variables and $m \leq Cn$ equations must take $2^{\delta n}$ time for some $\delta > 0$.*

3 Reduction from (c, s)-Gap MAXLIN to γ-Gap k-MLD$_q$

In this section, we present reductions from (c, s)-Gap MAXLIN to γ-Gap k-MLD$_q$. We first give a simple reduction that yields $N^{\Omega(k/\log k)}$ lower bound. Then we provide a reduction for the tight lower bound. The reduction contains two steps.

Step 1: Reduction from (c, s)-Gap MAXLIN to γ-Gap MLD$_q$.

▶ **Lemma 15.** *There exists a polynomial-time reduction that takes as input a (c, s)-Gap* MAXLIN *instance $(\boldsymbol{A}, \boldsymbol{b})$ where $\boldsymbol{A} \in \mathbb{F}_q^{m \times n}$, $\boldsymbol{b} \in \mathbb{F}_q^m$ and $0 < s < c < 1$. The reduction outputs a γ-Gap* MLD$_q$ *instance $(\boldsymbol{H}, \boldsymbol{u}, \ell)$ where $\boldsymbol{H} \in \mathbb{F}_q^{d \times m}$ $(d \leq m)$, $\boldsymbol{u} \in \mathbb{F}_q^d$, $\ell = (1-c)m$ and $\gamma = \frac{1-s}{1-c}$. This reduction satisfies the following properties:*

- ***Completeness:*** *If there exist $\boldsymbol{x} \in \mathbb{F}_q^n$ and $\boldsymbol{e} \in \mathbb{F}_q^m$ such that $\|\boldsymbol{e}\|_0 \leq (1-c)m$ and $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} = \boldsymbol{b}$, then there exists $\boldsymbol{x}' \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}'\|_0 \leq \ell$ such that $\boldsymbol{H}\boldsymbol{x}' = \boldsymbol{u}$.*
- ***Soundness:*** *If for every $\boldsymbol{x} \in \mathbb{F}_q^n$ and $\boldsymbol{e} \in \mathbb{F}_q^m$ satisfying $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} = \boldsymbol{b}$, we have $\|\boldsymbol{e}\|_0 > (1-s)m$, then there is no $\boldsymbol{x}' \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}'\|_0 \leq \gamma\ell$ such that $\boldsymbol{H}\boldsymbol{x}' = \boldsymbol{u}$.*

Proof. Let $\boldsymbol{H}$ denote any parity-check matrix for the code generated by $\boldsymbol{A}$, thus $\boldsymbol{H} \in \mathbb{F}_q^{d \times m}$ and $\boldsymbol{H}\boldsymbol{A} = \boldsymbol{0}$ where $d := m - \operatorname{rank}(\boldsymbol{A})$. Given $\boldsymbol{A}$, the matrix $\boldsymbol{H}$ can be computed in polynomial time via Gram–Schmidt orthogonalization applied to an appropriate basis of $\mathbb{F}_q^m$. We now convert the (c, s)-Gap MAXLIN instance $(\boldsymbol{A}, \boldsymbol{b})$ to its dual form by multiplying both sides of the equation $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} = \boldsymbol{b}$ by $\boldsymbol{H}$ on the left. Setting $\boldsymbol{u} := \boldsymbol{H}\boldsymbol{b}$, $\gamma := \frac{1-s}{1-c}$ and $\ell := (1-c)m$, we obtain the corresponding dual instance $(\boldsymbol{H}, \boldsymbol{u})$ satisfying:

- **Completeness.** If there exists $\boldsymbol{x} \in \mathbb{F}_q^n$ and $\boldsymbol{e} \in \mathbb{F}_q^m$ such that $\|\boldsymbol{e}\|_0 \leq \ell = (1-c)m$ and $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{e} = \boldsymbol{b}$, then $\boldsymbol{H}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{H}\boldsymbol{e} = \boldsymbol{H}\boldsymbol{b}$, hence $\boldsymbol{H}\boldsymbol{e} = \boldsymbol{u}$.
- **Soundness.** Assume for the sake of contradiction that there exists $\boldsymbol{e} \in \mathbb{F}_q^m$ with $\|\boldsymbol{e}\|_0 \leq \gamma\ell = (1-s)m$ such that $\boldsymbol{H}\boldsymbol{e} = \boldsymbol{u}$, then we have $\boldsymbol{H}(\boldsymbol{e} - \boldsymbol{b}) = \boldsymbol{0}$, hence $\boldsymbol{e} - \boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}$ for some $\boldsymbol{x} \in \mathbb{F}_q^n$.

Hence we get a polynomial-time reduction from (c, s)-Gap MAXLIN to γ-Gap MLD$_q$. ◀

Step 2: Reduction from γ-Gap MLD$_q$ to γ-Gap k-MLD$_q$. Next we will show a reduction from a non-parameterized MLD instance to a parameterized one. Consider a γ-Gap MLD$_q$ instance $(\boldsymbol{M}, \boldsymbol{u}, \ell)$. To obtain the parametrized instance, we need to "scale down" the solution size from ℓ to k by grouping the column vectors of $\boldsymbol{M}$ into appropriately sized blocks, and construct a new matrix in which each column vector behaves like an aggregated vector. This grouping ensures that, in the YES case, selecting k such aggregated vectors corresponds to selecting ℓ original vectors. For the reduction to be sound in the NO case, we additionally require that the size of each group be bounded by ℓ/k, so that any choice of at most γk aggregated vectors corresponds to at most $\gamma\ell$ original column vectors in $\boldsymbol{M}$.

▶ **Lemma 16.** *For every $\varepsilon > 0$, there exists a reduction that takes as input an integer $k \in \mathbb{N}$, a γ-Gap MLD_q instance $(\boldsymbol{M}, \boldsymbol{u}, \ell)$ where $\boldsymbol{M} \in \mathbb{F}_q^{d \times m}$, $\boldsymbol{u} \in \mathbb{F}_q^d$, and $\frac{k}{\varepsilon} < \ell < \frac{m}{\gamma}$. The reduction outputs a new matrix $\boldsymbol{M}_k \in \mathbb{F}_q^{d \times m'}$ in $(m')^{O(1)}$ time, which satisfies the following properties:*

- ***Size:*** *$m' = 2^{O(\frac{m \log k}{k})}$.*
- ***Completeness:*** *If there exists $\boldsymbol{x} \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}\|_0 \le \ell$ such that $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$, then there exists $\boldsymbol{y} \in \mathbb{F}_q^{m'}$ with $\|\boldsymbol{y}\|_0 \le k$ such that $\boldsymbol{M}_k\boldsymbol{y} = \boldsymbol{u}$.*
- ***Soundness:*** *If there is no $\boldsymbol{x} \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}\|_0 \le \gamma\ell$ such that $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$, then for $\gamma' = \gamma - \varepsilon$, there is no $\boldsymbol{y} \in \mathbb{F}_q^{m'}$ with $\|\boldsymbol{y}\|_0 \le \gamma' k$ such that $\boldsymbol{M}_k\boldsymbol{y} = \boldsymbol{u}$.*

Proof. Let $r := \lceil \frac{\ell}{k} \rceil$, $A := \{\boldsymbol{\alpha} \in \mathbb{F}_q^m : \|\boldsymbol{\alpha}\|_0 \le r\}$, and $m' := |A|$. We construct the new matrix $\boldsymbol{M}_k \in \mathbb{F}_q^{d \times m'}$ as follows. We establish a bijection between the column indices of $\boldsymbol{M}_k$ and A, and use $\boldsymbol{\alpha} \in A$ to denote a column index of $\boldsymbol{M}_k$. For each $\boldsymbol{\alpha} \in A$, we let the $\boldsymbol{\alpha}$-th column of $\boldsymbol{M}_k$ be

$$\boldsymbol{M}_k[\boldsymbol{\alpha}] = \boldsymbol{M}\boldsymbol{\alpha} \in \mathbb{F}_q^d.$$

Each column vector in $\boldsymbol{M}_k$ is therefore a linear combination of at most r column vectors from $\boldsymbol{M}$. Using the fact that $r = \lceil \frac{\ell}{k} \rceil < \frac{m}{k}$, the size of $\boldsymbol{M}_k$ satisfies

$$m' = \sum_{i=1}^{r} (q-1)^i \binom{m}{i} < r q^{\frac{m}{k}} \binom{m}{\frac{m}{k}} < r q^{\frac{m}{k}} (\mathrm{e}k)^{\frac{m}{k}} = 2^{O(\frac{m \log k}{k})}.$$

Completeness. Assume that there exists $\boldsymbol{x} \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}\|_0 \le \ell$ such that $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$. We construct k vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ such that $\|\boldsymbol{x}_i\|_0 \le r$ and $\sum_{i=1}^k \boldsymbol{x}_i = \boldsymbol{x}$. Such vectors exist and can be constructed in polynomial time. Let $\boldsymbol{y} \in \mathbb{F}_q^{m'}$ be such that $\boldsymbol{y}[\boldsymbol{x}_i] = 1$ for every $i \in [k]$, and $\boldsymbol{y}[\boldsymbol{\alpha}] = 0$ for any other $\boldsymbol{\alpha} \in A$. By construction, we have

$$\boldsymbol{M}_k\boldsymbol{y} = \sum_{i=1}^{k} \boldsymbol{M}_k[\boldsymbol{x}_i] = \sum_{i=1}^{k} \boldsymbol{M}\boldsymbol{x}_i = \boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}.$$

Soundness. Assume for the sake of contradiction that there exists $\boldsymbol{y} \in \mathbb{F}_q^{m'}$ with $\|\boldsymbol{y}\|_0 \le \gamma' k$ such that $\boldsymbol{M}_k\boldsymbol{y} = \boldsymbol{u}$. Let $X := \mathrm{supp}(\boldsymbol{y})$, then $|X| = \|\boldsymbol{y}\|_0 \le \gamma' k$, and we have

$$\boldsymbol{u} = \boldsymbol{M}_k\boldsymbol{y} = \sum_{\boldsymbol{\alpha} \in X} \boldsymbol{M}_k[\boldsymbol{\alpha}] \cdot \boldsymbol{y}[\boldsymbol{\alpha}] = \sum_{\boldsymbol{\alpha} \in X} \boldsymbol{M}\boldsymbol{\alpha} \cdot \boldsymbol{y}[\boldsymbol{\alpha}].$$

Let $\boldsymbol{x} := \sum_{\boldsymbol{\alpha} \in X} \boldsymbol{\alpha} \cdot \boldsymbol{y}[\boldsymbol{\alpha}]$, then we have $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$. Moreover, since $\|\boldsymbol{\alpha}\|_0 \le r$ for every $\boldsymbol{\alpha} \in A$, we have

$$\|\boldsymbol{x}\|_0 \le \sum_{\boldsymbol{\alpha} \in X} \|\boldsymbol{\alpha}\|_0 \le r|X| = \gamma' k \left\lceil \frac{\ell}{k} \right\rceil \le \gamma'(\ell + k).$$

Thus when $\ell > \frac{k}{\varepsilon}$, we have $\|\boldsymbol{x}\|_0 < (\gamma' + \varepsilon)\ell = \gamma\ell$, proving the soundness. ◀

By Lemma 15 and Lemma 16, we obtain a reduction from (c, s)-Gap MAXLIN with m equations to γ-Gap k-MLD_q with size $N \le 2^{O(\frac{m \log k}{k})}$ for some constant $\gamma > 1$, hence ruling out $N^{o(\frac{k}{\log k})}$-time algorithms for solving γ-Gap k-MLD_q.

Note that in our construction, when we group the vectors to form aggregated vectors, we chose all subsets of size at most $\lceil \ell/k \rceil$. This brute approach already gave us an almost tight lower bound for γ-Gap k-MLD_q. To get a tight lower bound we would want to get a

reduction where the number of aggregated vectors is upper bounded by $2^{O(m/k)}$. In the following section, we will make use of a combinatorial gadget, called the "cover family". This gadget will essentially enable us to do the grouping more cleverly and getting a tight lower bound.

3.1 Grouping using Cover Families

To get a tighter bound in the above reduction, we consider reducing the number of columns in $\boldsymbol{M}_k$. This requires us to find a smaller subset $A \subseteq \mathbb{F}_q^m$ that maintains the completeness and soundness requirements as follows:

- (Completeness) For every $\boldsymbol{x} \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}\|_0 \leq \ell$, there exist $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in A$ such that $\sum_{i=1}^k \boldsymbol{x}_i = \boldsymbol{x}$.
- (Soundness) For every $\boldsymbol{x} \in \mathbb{F}_q^m$, if there exist $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{\gamma' k} \in A$ such that $\sum_{i=1}^{\gamma' k} \boldsymbol{x}_i = \boldsymbol{x}$, then $\|\boldsymbol{x}\|_0 \leq \gamma\ell$.

To construct such set A, we will make use of a combinatorial gadget which we call "cover family". It produces a gap between the number of sets required to cover a small set and a large set. Let A be the set of all vectors whose support is in the cover family, then A meets all requirements above. The gadget is formally defined as follows.

▶ **Definition 17** ($(U, k, \alpha, \varepsilon)$-cover family)**.** *A $(U, k, \alpha, \varepsilon)$-cover family is a collection $\mathcal{S} \subseteq 2^U$ of subsets of U, such that*

- *(C1) For every $S \in \mathcal{S}$, $|S| \leq \frac{(1+\varepsilon)\alpha|U|}{k}$.*
- *(C2) For every $\widetilde{S} \subseteq U$ with $|\widetilde{S}| \leq \alpha|U|$, there exists $\{T_1, \ldots, T_k\} \subseteq \mathcal{S}$ such that $\bigcup_{i\in[k]} T_i = \widetilde{S}$. Moreover, $T_i \cap T_j = \emptyset$ for every $i \neq j \in [k]$.*

Note that property (C1) immediately implies that for any $\widetilde{S} \subseteq U$ with $|\widetilde{S}| \geq \beta|U|$, it requires at least $\frac{\beta}{(1+\varepsilon)\alpha}k$ sets in $\mathcal{S}$ to cover $\widetilde{S}$, hence producing our desired gap. We now present the improved reduction as follows.

▶ **Lemma 18.** *There exists a reduction that takes as input an integer $k \in \mathbb{N}$, an $([m], k, \alpha, \varepsilon)$-cover family $\mathcal{S} \subseteq 2^{[m]}$, and a γ-Gap MLD_q instance $(\boldsymbol{M}, \boldsymbol{u}, \alpha m)$ where $\boldsymbol{M} \in \mathbb{F}_q^{d\times m}$, $\boldsymbol{u} \in \mathbb{F}_q^d$. The reduction outputs a new matrix $\boldsymbol{M}_k \in \mathbb{F}_q^{d\times m'}$ in $(m')^{O(1)}$ time, which satisfies the following properties:*

- ***Size:*** *$m' = |\mathcal{S}| \cdot q^{\frac{(1+\varepsilon)\alpha m}{k}}$.*
- ***Completeness:*** *If there exists $\boldsymbol{x} \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}\|_0 \leq \alpha m$ such that $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$, then there exists $\boldsymbol{y} \in \mathbb{F}_q^{m'}$ with $\|\boldsymbol{y}\|_0 \leq k$ such that $\boldsymbol{M}_k\boldsymbol{y} = \boldsymbol{u}$.*
- ***Soundness:*** *If there is no $\boldsymbol{x} \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}\|_0 \leq \gamma\alpha m$ such that $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$, then there is no $\boldsymbol{y} \in \mathbb{F}_q^{m'}$ with $\|\boldsymbol{y}\|_0 \leq \frac{\gamma}{1+\varepsilon}k$ such that $\boldsymbol{M}_k\boldsymbol{y} = \boldsymbol{u}$.*

Proof. Let $A := \{\boldsymbol{\alpha} \in \mathbb{F}_q^m : \mathrm{supp}(\boldsymbol{\alpha}) \in \mathcal{S}\}$ and $m' := |A|$. We construct the new matrix $\boldsymbol{M}_k \in \mathbb{F}_q^{d\times m'}$ as follows. We establish a bijection between the column indices of $\boldsymbol{M}_k$ and A, and use $\boldsymbol{\alpha} \in A$ to denote a column index of $\boldsymbol{M}_k$. For each $\boldsymbol{\alpha} \in A$, we let the $\boldsymbol{\alpha}$-th column of $\boldsymbol{M}_k$ be

$$\boldsymbol{M}_k[\boldsymbol{\alpha}] = \boldsymbol{M}\boldsymbol{\alpha} \in \mathbb{F}_q^d.$$

The property (C1) of $\mathcal{S}$ guarantees that $|S| \leq \frac{(1+\varepsilon)\alpha m}{k}$, hence the size of $\boldsymbol{M}_k$ satisfies

$$m' = \sum_{S\in\mathcal{S}} (q-1)^{|S|} < |\mathcal{S}| \cdot q^{\frac{(1+\varepsilon)\alpha m}{k}}.$$

Completeness. Assume that there exists $\boldsymbol{x} \in \mathbb{F}_q^m$ with $\|\boldsymbol{x}\|_0 \le \alpha m$ such that $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$. Since $|\operatorname{supp}(\boldsymbol{x})| \le \alpha m$, by the property (C2) of $\mathcal{S}$, there exists $T_1, \ldots, T_k \in \mathcal{S}$ as a partition of $\operatorname{supp}(\boldsymbol{x})$.

For each $i \in [k]$, let $\boldsymbol{x}_i \in \mathbb{F}_q^m$ be the projection of $\boldsymbol{x}$ onto the coordinates in T_i, i.e., for every $j \in [m]$, we let

$$\boldsymbol{x}_i[j] = \begin{cases} \boldsymbol{x}[j], & j \in T_i, \\ 0, & j \notin T_i. \end{cases}$$

Then $\sum_{i=1}^k \boldsymbol{x}_i = \boldsymbol{x}$, and $\boldsymbol{x}_i \in A$ because $\operatorname{supp}(\boldsymbol{x}) = T_i \in \mathcal{S}$. Let $\boldsymbol{y} \in \mathbb{F}_q^{m'}$ be such that $\boldsymbol{y}[\boldsymbol{x}_i] = 1$ for every $i \in [k]$, and $\boldsymbol{y}[\boldsymbol{\alpha}] = 0$ for any other $\boldsymbol{\alpha} \in A$. By construction, we have

$$\boldsymbol{M}_k\boldsymbol{y} = \sum_{i=1}^k \boldsymbol{M}_k[\boldsymbol{x}_i] = \sum_{i=1}^k \boldsymbol{M}\boldsymbol{x}_i = \boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}.$$

Soundness. Assume for the sake of contradiction that there exists $\boldsymbol{y} \in \mathbb{F}_q^{m'}$ with $\|\boldsymbol{y}\|_0 \le \frac{\gamma}{1+\varepsilon}k$ such that $\boldsymbol{M}_k\boldsymbol{y} = \boldsymbol{u}$. Let $X := \operatorname{supp}(\boldsymbol{y})$, then we have

$$\boldsymbol{u} = \boldsymbol{M}_k\boldsymbol{y} = \sum_{\boldsymbol{\alpha}\in X} \boldsymbol{M}_k[\boldsymbol{\alpha}] \cdot \boldsymbol{y}[\boldsymbol{\alpha}] = \sum_{\boldsymbol{\alpha}\in X} \boldsymbol{M}\boldsymbol{\alpha} \cdot \boldsymbol{y}[\boldsymbol{\alpha}].$$

Let $\boldsymbol{x} := \sum_{\boldsymbol{\alpha}\in X} \boldsymbol{y}[\boldsymbol{\alpha}] \cdot \boldsymbol{\alpha}$, then $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$. By the property (C1) of $\mathcal{S}$, we have $|\operatorname{supp}(\boldsymbol{\alpha})| \le \frac{(1+\varepsilon)\alpha}{k}m$ for every $\boldsymbol{\alpha} \in A$. Since $|X| = \|\boldsymbol{y}\|_0 \le \frac{\gamma}{1+\varepsilon}k$, we have

$$|\operatorname{supp}(\boldsymbol{x})| \le \Big|\bigcup_{\boldsymbol{\alpha}\in X} \operatorname{supp}(\boldsymbol{\alpha})\Big| \le |X| \cdot \frac{(1+\varepsilon)\alpha}{k}m \le \gamma\alpha m.$$

Hence $\boldsymbol{x}$ satisfies $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{u}$ and $\|\boldsymbol{x}\|_0 \le \gamma\alpha m$, proving the soundness. ◀

Combining Lemma 15 and Lemma 18, we obtain a reduction which on input an $([m], k, 1-c, \varepsilon)$-cover family $\mathcal{S}$, it reduces a (c, s)-Gap MAXLIN instance with m equations to γ-Gap k-MLD$_q$ instance with size $N = |\mathcal{S}| \cdot 2^{O(m/k)}$ for $\gamma = \frac{1-s}{(1-c)(1+\varepsilon)}$. If $|\mathcal{S}| = 2^{O(m/k)}$, by Corollary 14 we get that assuming ETH, there exists a constant $\gamma > 1$ such that no algorithm running in time $N^{o(k)}$ can solve γ-Gap k-MLD$_q$.

To achieve the tight lower bound, we will construct a cover family over universe $[m]$ with size $|\mathcal{S}| = 2^{O(m/k)}$ using the following lemma, whose proof is postponed to Section 4.

▶ **Lemma 19.** *For every $m, k \in \mathbb{N}$ and $0 < \alpha, \varepsilon < 1$ such that $m \ge k^{4k^2/(\varepsilon^2\alpha)}$, there exists an $([m], k, \alpha, \varepsilon)$-cover family $\mathcal{S}$ with $|\mathcal{S}| = \lceil \log_k m \rceil \cdot k \cdot 2^{2m/k}$ that can be constructed deterministically in $|\mathcal{S}|^{O(1)}$ time.*

It follows that

$$N = 2^{O_q(m/k)}.$$

This improves upon the naive grouping bound of $2^{O(m \log k/k)}$ by removing the $\log k$ factor in the exponent. Consequently, under ETH, this reduction implies that γ-Gap k-MLD$_q$ cannot be solved in time $N^{o(k)}$, confirming the tightness of the lower bound for the parameterized problem.

Note that Lemma 19 requires $m \ge k^{\Omega_{\alpha,\varepsilon}(k^2)}$. To obtain the tight lower bound for larger k, we give a randomized construction of cover families.

▶ **Lemma 20.** *For every $m, k \in \mathbb{N}$, $0 < \alpha, \varepsilon < 1$ such that $m \ge \frac{6k \ln 2k}{\varepsilon^2\alpha}$, there exists an $([m], k, \alpha, \varepsilon)$-cover family $\mathcal{S}$ with $|\mathcal{S}| = \frac{12k^2}{\varepsilon^2\alpha} \cdot 2^{(1+\varepsilon)m/k}$ that can be constructed in $|\mathcal{S}|^{O(1)}$ time with probability $1 - o(1)$.*

3.2 Putting Everything Together

We now combine everything together to prove the main result and derive other conclusions.

▶ **Theorem 21** (Theorem 1 Restated). *Assuming* ETH*, there exist constants $\gamma > 1$ and $\delta > 0$, such that for any algorithm that takes as input a matrix $\boldsymbol{H} \in \mathbb{F}_q^{d\times N}$, a vector $\boldsymbol{u} \in \mathbb{F}_q^d$ and a parameter $k \in \mathbb{N}$, it must take $N^{\delta k}$ time to distinguish the following two cases:*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{F}_q^N$ with $\|\boldsymbol{x}\|_0 \le k$ such that $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{u}$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{F}_q^N$ with $\|\boldsymbol{x}\|_0 \le \gamma k$, we have $\boldsymbol{H}\boldsymbol{x} \neq \boldsymbol{u}$.*

Proof. We start with a (c, s)-Gap MAXLIN_q instance with n variables and $m \le Cn$ equations for some constant $0 < s < c < 1$ and $C > 0$. By Corollary 14, there exists some constant $\delta > 0$ such that deciding (c, s)-Gap MAXLIN_q must take $2^{\delta n}$ time assuming ETH.

We now invoke Lemma 19 with $\alpha \leftarrow 1 - c$, and a sufficiently small $\varepsilon > 0$ such that $\gamma := \frac{1-s}{(1+\varepsilon)(1-c)} > 1$. This gives us a $\big([m], k, 1-c, \varepsilon\big)$-cover family $\mathcal{S}$ with $|\mathcal{S}| = k \cdot 2^{O(m/k)}$, constructible in $|\mathcal{S}|^{O(1)}$ time. Using Lemma 15 and Lemma 18, we reduce this (c, s)-Gap MAXLIN_q instance to a γ-Gap k-MLD_q instance where the number of vectors is $N = |\mathcal{S}| \cdot q^{O(m/k)} = k \cdot 2^{O_q(n/k)}$.

Assume that γ-Gap k-MLD_q can be solved in $N^{\delta k}$ time for every $\delta > 0$, then for any $\delta' > 0$, there exists δ such that (c, s)-Gap MAXLIN_q can be solved in time $\big(k \cdot 2^{O_q(n/k)}\big)^{\delta k} < 2^{\delta' n}$. Therefore, there exist $\gamma > 1$ and $\delta > 0$ such that assuming ETH, any algorithm that decides γ-Gap k-MLD_q must take time $N^{\delta k}$. ◀

In fact, Theorem 21 also holds for larger k such that $k = O((\log\log N)^{0.49})$, since we can construct the $\big([m], k, 1-c, \varepsilon\big)$-cover family as long as $k^{\Omega(k^2)} \le m$. Following this observation, if we use the randomized construction of cover family instead of the deterministic one, we obtain the lower bound for any $k \le N^{\varepsilon}$ for some constant $\varepsilon > 0$ under randomized ETH.

▶ **Theorem 22** (Theorem 2 Restated). *Assuming randomized* ETH*, there exist constants $\gamma > 1$ and $\delta, \varepsilon' > 0$, such that for any algorithm that takes as input a matrix $\boldsymbol{H} \in \mathbb{F}_q^{d\times N}$, a vector $\boldsymbol{u} \in \mathbb{F}_q^d$ and an integer $k \in \mathbb{N}$ such that $2 \le k \le N^{\varepsilon'}$, it must take $N^{\delta k}$ time to distinguish the following two cases:*

- *YES Case: There exists $\boldsymbol{x} \in \mathbb{F}_q^d$ with $\|\boldsymbol{x}\|_0 \le k$ such that $\boldsymbol{H}\boldsymbol{x} = \boldsymbol{u}$.*
- *NO Case: For every $\boldsymbol{x} \in \mathbb{F}_q^d$ with $\|\boldsymbol{x}\|_0 \le \gamma k$, we have $\boldsymbol{H}\boldsymbol{x} \neq \boldsymbol{u}$.*

Proof. We still start with a $(1-\alpha, s)$-Gap MAXLIN_q instance with n variables and $m \le Cn$ equations. Pick a small enough constant $\varepsilon > 0$ such that $\gamma := \frac{1-s}{(1+\varepsilon)\alpha} > 1$. Let

$$N := \frac{12k^2}{\varepsilon^2\alpha} \cdot 2^{(1+\varepsilon)(1+\alpha\log_2 q)\frac{m}{k}} \quad \text{and} \quad \beta := \frac{\varepsilon^2\alpha}{6(1+\varepsilon)(1+\alpha\log_2 q)}.$$

When

$$k \le \Big(\frac{\varepsilon^2\alpha}{12}N\Big)^{\frac{\beta}{2(1+\beta)}},$$

which implies that

$$k\ln 2k \le k\log_2(2k) \le \frac{\varepsilon^2\alpha}{6}m,$$

by Lemma 20, we can construct a $\big([m], k, \alpha, \varepsilon\big)$-cover family $\mathcal{S}$ with size $|\mathcal{S}| = \frac{12k^2}{\varepsilon^2\alpha} \cdot 2^{(1+\varepsilon)m/k}$.

Using the $\big([m], k, \alpha, \varepsilon\big)$-cover family, we can reduce the MAXLIN instance to a γ-Gap k-MLD_q instance with size $N = |\mathcal{S}| \cdot q^{(1+\varepsilon)\alpha m/k} = k^2 \cdot 2^{O_q(n/k)}$. Assume that

γ-Gap k-MLD_q can be solved in $N^{\delta k}$ time for every $\delta > 0$, then for any $\delta' > 0$, there exists δ such that (c, s)-Gap MAXLIN_q can be solved in time

$$\left(k^2 \cdot 2^{O_q(n/k)}\right)^{\delta k} = 2^{\delta \cdot O_q(n + k \log k)} < 2^{\delta' n}.$$

This proves the $N^{\delta k}$-time lower bound for $k = O(N^{\varepsilon'})$ where $\varepsilon' := \frac{\beta}{2(1+\beta)}$. ◀

We state the following corollary which can be obtained by a standard reduction from γ-Gap k-MLD_q to γ-Gap k-NCP_q as in [15].

▶ **Corollary 23.** *Assuming* ETH*, there exists a constant $\gamma > 1$, such that no algorithm can decide γ-Gap k-NCP_q instance $(\boldsymbol{A}, \boldsymbol{t})$ where $\boldsymbol{A} \in F_q^{N \times M}, \boldsymbol{t} \in \mathbb{F}_q^N$ in $N^{o(k)}$ time.*

Using the reduction from γ-Gap NCP_2 to γ-Gap CVP_p as in [1], we can obtain optimal hardness for γ-Gap k-CVP_p for all $p > 1$.

▶ **Corollary 24.** *Assuming* ETH*, for every $p > 1$, there exists a constant $\gamma > 1$, such that no algorithm can decide γ-Gap k-CVP_p instance $(\boldsymbol{A}, \boldsymbol{t})$ where $\boldsymbol{A} \in \mathbb{Z}^{N \times M}, \boldsymbol{t} \in \mathbb{Z}^N$ in $N^{o(k)}$ time.*

4 Construction of Cover Families

In this section, we present the construction of cover families. Our approach relies on an intermediate combinatorial object which we call a *balanced partition family*. A balanced partition family is a family of partitions over a universe of size n, such that every subset of size αn is partitioned into almost equal size by some partition. We first formalize balanced partition families as follows, and then establish a reduction showing that any efficient balanced partition family yields a corresponding cover family.

▶ **Definition 25** ($(U, k, \alpha, \varepsilon, c)$-balanced partition family)**.** *A $(U, k, \alpha, \varepsilon, c)$-balanced partition family is a family $\mathcal{F} \subseteq [k]^U$ of functions from U to $[k]$, such that*

- *(P1) For every $f \in \mathcal{F}$ and $j \in [k]$, $|f^{-1}(j)| \leq \frac{c|U|}{k}$.*
- *(P2) For every $S \subseteq U$ with $|S| = \alpha|U|$, there exists $f \in \mathcal{F}$ such that $|S \cap f^{-1}(j)| \leq \frac{(1+\varepsilon)|S|}{k}$ for every $j \in [k]$.*

In the following theorem we present the construction of a *cover family*, assuming the existence of a *balanced partition family*.

▶ **Theorem 26.** *On input a $(U, k, \alpha, \varepsilon, c)$-balanced partition family $\mathcal{F}$, one can construct a $(U, k, \alpha, \varepsilon)$-cover family $\mathcal{S}$ with $|\mathcal{S}| = |\mathcal{F}| \cdot k \cdot 2^{c|U|/k}$ in $|\mathcal{S}|^{O(1)}$ time.*

Proof. Let $m := |U|$. We construct the cover family $\mathcal{S}$ as follows:

- For every function $f \in \mathcal{F}$ and every index $j \in [k]$, let $\mathcal{S}_{f,j}$ be the collection of all subsets of the j-th bucket $f^{-1}(j)$ that have size at most $\frac{(1+\varepsilon)\alpha m}{k}$. That is,

$$\mathcal{S}_{f,j} := \left\{ T \subseteq f^{-1}(j) : |T| \leq \frac{(1+\varepsilon)\alpha m}{k} \right\}.$$

- Define $\mathcal{S} := \bigcup_{f \in \mathcal{F}} \bigcup_{j \in [k]} \mathcal{S}_{f,j}$.

Size and Running Time. By property (P1) of the balanced partition family, for every $f \in \mathcal{F}$ and $j \in [k]$, the bucket size is bounded by $|f^{-1}(j)| \leq \frac{cm}{k}$. The number of subsets in $\mathcal{S}_{f,j}$ is bounded by the total number of subsets of the bucket, which is $2^{|f^{-1}(j)|}$. Therefore,

$$|\mathcal{S}| \leq \sum_{f \in \mathcal{F}} \sum_{j \in [k]} 2^{|f^{-1}(j)|} \leq |\mathcal{F}| \cdot k \cdot 2^{cm/k}.$$

The construction can be performed in time polynomial in $|\mathcal{S}|$.

Property (C1). By construction, the maximum size of any set $S \in \mathcal{S}$ is bounded by $\frac{(1+\varepsilon)\alpha m}{k}$.

Property (C2). Let $\widetilde{S} \subseteq U$ be any set with $|\widetilde{S}| \leq \alpha m$. We need to show that $\widetilde{S}$ can be exactly covered by k disjoint sets from $\mathcal{S}$. Pick an arbitrary set S' such that $|S'| = \alpha m$ and $\widetilde{S} \subseteq S'$. By property (P2) of the balanced partition family, there exists a function $f \in \mathcal{F}$ such that for all $j \in [k]$,

$$|\widetilde{S} \cap f^{-1}(j)| \leq |S' \cap f^{-1}(j)| \leq \frac{(1+\varepsilon)|S'|}{k} = \frac{(1+\varepsilon)\alpha m}{k}.$$

Define $T_j := \widetilde{S} \cap f^{-1}(j)$ for each $j \in [k]$.

- Since $T_j \subseteq f^{-1}(j)$ and $|T_j| \leq \frac{(1+\varepsilon)\alpha m}{k}$, we have $T_j \in \mathcal{S}_{f,j} \subseteq \mathcal{S}$.
- Since $\{f^{-1}(j)\}_{j \in [k]}$ forms a partition of U, the sets $T_1, \ldots, T_k$ are pairwise disjoint.
- $T_1, \ldots, T_k$ can cover $\widetilde{S}$, because $\bigcup_{j \in [k]} T_j = \widetilde{S} \cap \bigcup_{j \in [k]} f^{-1}(j) = \widetilde{S} \cap U = \widetilde{S}$.

Thus, $\{T_1, \ldots, T_k\} \subseteq \mathcal{S}$ is a exact cover for $\widetilde{S}$ and satisfies (C2). ◀

4.1 Randomized Construction

We will prove the existence of a balanced partition family with $O(k)$ size, by showing that randomly sampling $O(k)$ partitions and then rejecting all partitions that do not satisfy (C1) produces a balanced partition family with high probability.

▶ **Theorem 27.** *For every $m, k \in \mathbb{N}$ and $0 < \alpha, \varepsilon < 1$ such that $m \geq \frac{6k \ln 2k}{\varepsilon^2 \alpha}$, there exists an $([m], k, \alpha, \varepsilon, 1+\varepsilon)$-balanced partition family $\mathcal{F}$ with $|\mathcal{F}| = \frac{12k}{\varepsilon^2 \alpha}$.*

Proof. Let $t := \lceil \frac{12k}{\varepsilon^2 \alpha} \rceil$. We sample t functions $f_1, \ldots, f_t : U \to [k]$ independently and uniformly at random. We define $I \subseteq [k]$ as the indices of functions that satisfy (P1), formally, let

$$I := \left\{ i \in [t] : \forall j \in [k],\ |f_i^{-1}(j)| \leq \frac{(1+\varepsilon)m}{k} \right\}.$$

For any subset $S \subseteq U$, let $B(S)$ be the "bad event" that no function f_i with $i \in I$ evenly splits S. That is, for every $i \in I$, there exists some bucket $j \in [k]$ such that $|S \cap f_i^{-1}(j)| > \frac{(1+\varepsilon)|S|}{k}$. We will show that with high probability, $B(S)$ does not occur for every subset $S \subseteq [m]$ of size αm.

Consider a fixed index $i \in [t]$ and a fixed subset $S \subseteq U$ with $|S| = \alpha m$. We define two bad events for f_i and analyze the probability of these events:

- Let $E_1(i)$ be the event that $i \notin I$. Since f_i is a uniform random function, $f(x)$ is drawn uniformly and independently from $[k]$ for each $x \in [m]$. Hence for a fixed bucket $j \in [k]$, $x \in f_i^{-1}(j)$ with probability $\frac{1}{k}$ for every $x \in [m]$, and thus $|f_i^{-1}(j)|$ is a sum of independent Bernoulli trials with mean m/k. By the Chernoff bound,

$$\mathbf{Pr}\left[|f_i^{-1}(j)| > \frac{(1+\varepsilon)m}{k} \right] < \exp\left(-\frac{\varepsilon^2 m}{3k} \right).$$

Taking a union bound over all k buckets,

$$\mathbf{Pr}[E_1(i)] < k \exp\left(-\frac{\varepsilon^2 m}{3k}\right).$$

- Let $E_2(i,S)$ be the event that f_i fails the property (P2) for the specific set S. This means there exists some bucket $j \in [k]$ such that $|S \cap f_i^{-1}(j)| > \frac{(1+\varepsilon)\alpha m}{k}$. For a fixed $j \in [k]$, the random variable $|S \cap f_i^{-1}(j)|$ follows a binomial distribution $\mathrm{Bin}(\alpha m, 1/k)$. Similarly, by the Chernoff bound and a union bound over k buckets, we have

$$\mathbf{Pr}[E_2(i,S)] < k \exp\left(-\frac{\varepsilon^2 \alpha m}{3k}\right).$$

Let $E(i,S) := E_1(i) \vee E_2(i,S)$ be the combined bad event where f_i either violates (P1) or fails to balance S. Since $\alpha < 1$, we have

$$\mathbf{Pr}[E(i,S)] \le \mathbf{Pr}[E_1(i)] + \mathbf{Pr}[E_2(i,S)] < 2k \exp\left(-\frac{\varepsilon^2 \alpha m}{3k}\right).$$

Since the functions are sampled independently, we have

$$\mathbf{Pr}[B(S)] = \prod_{i=1}^{t} \mathbf{Pr}[E(i,S)] < \left(2k \exp\left(-\frac{\varepsilon^2 \alpha m}{3k}\right)\right)^t.$$

Take a union bound over all possible subsets S of size αm. Since the total number of such subsets is $\binom{m}{\alpha m} < \mathrm{e}^m$, the probability that there exists a bad set S is at most

$$\begin{aligned}\sum_{S \in \binom{[m]}{\alpha m}} \mathbf{Pr}[B(S)] &< \mathrm{e}^m \left(2k \exp\left(-\frac{\varepsilon^2 \alpha m}{3k}\right)\right)^t \\ &= \exp\left(m - t\left(\frac{\varepsilon^2 \alpha m}{3k} - \ln 2k\right)\right).\end{aligned}$$

When $m \ge \frac{6k \ln 2k}{\varepsilon^2 \alpha}$ and $t = \left\lceil \frac{12k}{\varepsilon^2 \alpha} \right\rceil$, the total probability is at most e^{-m}.

We let the final family $\mathcal{F}$ contain all functions that satisfy property (P1), i.e., let $\mathcal{F} := \{f_i : i \in I\}$. By definition, every function in $\mathcal{F}$ satisfies (P1). Furthermore, we have proved that with probability $1 - \mathrm{e}^{-m}$, for every subset $S \in \binom{U}{\alpha m}$, there exists $f \in \mathcal{F}$ that evenly splits S, hence $\mathcal{F}$ also satisfies (P2). Thus, $\mathcal{F}$ is a valid balanced partition family with high probability. ◀

Using Theorem 26, we obtain an $([m], k, \alpha, \varepsilon)$-cover family $\mathcal{S}$ with $|\mathcal{S}| = \frac{12k}{\varepsilon^2 \alpha} \cdot k \cdot 2^{(1+\varepsilon)m/k}$ in $|\mathcal{S}|^{O(1)}$ time, hence proving Lemma 20.

4.2 Derandomization

This section introduces the hypercube partition system as a derandomization of the balanced partition family. We define the hypercube partition system as follows.

▶ **Definition 28** (Hypercube Partition System). *A (k,d)-hypercube partition system is a pair $(U, \mathcal{F})$ where $U := [k]^{[d]}$ and $\mathcal{F} := \{f_1, \ldots, f_d\}$ is a collection of funtions from U to $[k]$. We view U as the set of all points in a d-dimensional hypercube where each coordinate takes a value in $[k]$. The i-th partition $f_i : U \to [k]$ is defined as for every $x \in U$*

$$f_i(x) = x(i).$$

In other words, for any $x \in U$, f_i groups x according to its i-th coordinate, into the $x(i)$-th set.

We show that a (k, d)-hypercube partition system $(U, \mathcal{F})$ is a balanced partition family for sufficiently large d. Informally, it suffices to show the following proposition:

(⋆) For any subset $S \subseteq U$, if on every coordinate $i \in [d]$, there exists some $j \in [k]$ such that at least $\frac{(1+\varepsilon)|S|}{k}$ elements $x \in S$ satisfy $x(i) = j$, then $|S| < \alpha|U|$.

By the contrapositive of (⋆), if S is large enough, then at least one coordinate $i \in [d]$ must evenly split S. This indicates that $\mathcal{F}$ is a balanced partition family.

To show (⋆), we consider a uniform distribution p over the subset S. Let p_i denote the marginal distribution of p on the i-th coordinate. The condition in (⋆) guarantees that for each coordinate i, p_i concentrates on some value j, which implies that p_i has small entropy. Then we add up the entropy of p_i over all coordinates, which provides a upper bound on the entropy of p. Hence $|S|$ must be small.

We prove this formally in Lemma 29.

▶ **Lemma 29.** *Let $d, k \in \mathbb{N}$, $0 < \alpha, \varepsilon < 1$, and $U := [k]^d$. For every $i \in [d]$, let $f_i : U \to [k]$ be such that $f_i(x) = x_i$. If $d \geq \frac{4k}{\varepsilon^2 \alpha}$, then for every $S \subseteq U$ with $|S| = \alpha k^d$, there exists $i \in [d]$, such that $|S \cap f_i^{-1}(j)| \leq (1+\varepsilon)\alpha k^{d-1}$ for every $j \in [k]$.*

Proof. Let X be a random variable uniformly distributed over the subset S. We have $H(X) = \ln |S| = d \ln k + \ln \alpha$. Let p_i be the distribution of X_i, i.e., for every $j \in [k]$,

$$p_i(j) = \frac{|\{x \in S : x_i = j\}|}{|S|} = \frac{|S \cap f_i^{-1}(j)|}{\alpha k^d}.$$

Now we will prove that there exists some $i \in [d]$, such that $p_i(j) \leq \frac{1+\varepsilon}{k}$ for every $j \in [k]$, which immediately implies that $|S \cap f_i^{-1}(j)| \leq (1+\varepsilon)\alpha k^{d-1}$.

Assume for the sake of contradiction that for every $i \in [d]$, there exists some $j_i \in [k]$ such that $p_i(j_i) > \frac{1+\varepsilon}{k}$. Fix any $i \in [d]$ and let $\rho_i := p_i(j_i)$. Then, by Jensen's inequality,

$$\begin{aligned} H(X_i) &= p_i(j_i) \ln \frac{1}{p_i(j_i)} + \sum_{j \neq j_i} p_i(j) \ln \frac{1}{p_i(j)} \\ &\leq \rho_i \ln \frac{1}{\rho_i} + (1-\rho_i) \ln \frac{k-1}{1-\rho_i}. \end{aligned}$$

Let $h(x) := x \ln \frac{1}{x} + (1-x) \ln \frac{k-1}{1-x}$. We have $h'\left(\frac{1}{k}\right) = 0$ and $h''(x) = -\frac{1}{x(1-x)}$. By Taylor's theorem with the Lagrange's form of remainder, we expand $h(x)$ at $x = 1/k$, then there exists ξ with $\frac{1}{k} < \xi < \rho_i$ such that

$$\begin{aligned} h(\rho_i) &= h\Big(\frac{1}{k}\Big) + h'\Big(\frac{1}{k}\Big)\Big(\rho_i - \frac{1}{k}\Big) + \frac{h''(\xi)}{2}\Big(\rho_i - \frac{1}{k}\Big)^2 \\ &= \ln k - \frac{1}{2\xi(1-\xi)}\Big(\rho_i - \frac{1}{k}\Big)^2. \end{aligned}$$

Recall that $\rho_i > \frac{1+\varepsilon}{k}$. Since $\xi(1-\xi) < \xi < \rho_i$, and by the fact that $f(x) := x + \frac{1}{x} - 2$ is increasing when $x > 1$, we have

$$\ln k - h(\rho_i) > \frac{1}{2\rho_i}\Big(\rho_i - \frac{1}{k}\Big)^2 = \frac{f(k\rho_i)}{2k} > \frac{f(1+\varepsilon)}{2k} > \frac{\varepsilon^2}{4k},$$

thus

$$H(X_i) \leq h(\rho_i) < \ln k - \frac{\varepsilon^2}{4k}.$$

By the subadditivity of entropy, when $d \geq \frac{4k}{\varepsilon^2\alpha}$, we have

$$H(X) \leq \sum_{i\in[d]} H(X_i) \leq d\ln k - \frac{\varepsilon^2 d}{4k} \leq d\ln k - \frac{1}{\alpha} < d\ln k + \ln\alpha,$$

which contradicts $H(X) = \ln|S| = d\ln k + \ln\alpha$. ◀

Lemma 29 implies that a (k,d)-hypercube partition system is a $([k]^d, k, \alpha, \varepsilon, 1)$-balanced partition family of size d if d is large enough. A limitation of this construction is that we require the size of the universe to be an integer power of k. To address this, we define a subset $U \subseteq [k]^d$ of size exactly m by taking a "diagonal slice" of the hypercube. This slice preserves the balanced property of projections: fixing one coordinate to any value leaves the remaining coordinates to cycle through values uniformly, ensuring the slice is spread evenly across buckets.

▶ **Theorem 30.** *For every $m, k \in \mathbb{N}$ and $0 < \eta, \varepsilon < 1$ such that $m \geq k^{4k^2/(\varepsilon^2\eta)}$, there exists an $([m], k, \eta, \varepsilon, 2)$-balanced partition family $\mathcal{F}$ of size $|\mathcal{F}| = \lceil \log_k m \rceil$ that can be constructed in $m^{O(1)}$ time.*

Proof. Let $d := \lceil \log_k m \rceil$ and $c := \left\lceil \frac{m}{k^{d-1}} \right\rceil$. We have $(c-1)k^{d-1} < m \leq ck^{d-1}$ and $2 \leq c \leq k$. Define a partition $U_0, \ldots, U_{k-1}$ of $[k]^d$ based on the modular sum of coordinates:

$$U_\ell := \left\{ x \in [k]^d : \left(\sum_{j=1}^{d} x_j \right) \bmod k = \ell \right\}.$$

Let $U' := U_0 \cup \cdots \cup U_{c-1}$. Then $|U'| = ck^{d-1}$ because $|U_\ell| = k^{d-1}$ for every ℓ. We set the universe U to be an arbitrary subset of U' of size $m \leq |U'|$. Define $\mathcal{F} = \{f_1, \ldots, f_d\}$ as the projections on each coordinate: $f_i(x) = x(i)$.

Property (P1). For any $i \in [d], j \in [k]$ and $0 \leq \ell < k$, the number of $x \in U_\ell$ such that $x_i = j$ is exactly k^{d-2}. To prove this, we take an arbitrary $i' \neq i$. If we fix x_t for every $t \in [d] \setminus \{i, i'\}$, then $x_{i'}$ must be $x_{i'} = \ell - \sum_{t\in[d]\setminus\{i'\}} x_t$. The number of ways to fix x_t is k^{d-2}, hence there are k^{d-2} many $x \in U_\ell$ with $x_i = j$, i.e.,

$$|U_\ell \cap f_i^{-1}(j)| = k^{d-2}.$$

Since $m > (c-1)k^{d-1}$, for any $i \in [d]$ and $j \in [k]$,

$$|U \cap f_i^{-1}(j)| \leq |U' \cap f_i^{-1}(j)| = ck^{d-2} \leq 2(c-1)k^{d-2} < \frac{2m}{k}.$$

Hence $\mathcal{F}$ satisfies property (P1).

Property (P2). Let $\alpha := \frac{\eta m}{k^d}$. Since $m > k^{d-1}$, we have $\eta < \alpha k$, hence

$$d \geq \log_k m \geq \frac{4k^2}{\varepsilon^2\eta} > \frac{4k}{\varepsilon^2\alpha}.$$

Applying Lemma 29, for any $S \subseteq [k]^d$ of size $\alpha k^d = \eta m$, there exists $f_i \in \mathcal{F}$ such that every bucket $S \cap f_i^{-1}(j)$ has size at most

$$(1+\varepsilon)\alpha k^{d-1} = \frac{(1+\varepsilon)\eta m}{k} = \frac{(1+\varepsilon)|S|}{k}.$$

Hence $\mathcal{F}$ also satisfies property (P2). ◀

Using Theorem 26, we obtain an $([m], k, \eta, \varepsilon)$-cover family $\mathcal{S}$ with $|\mathcal{S}| = \lceil \log_k m \rceil \cdot k \cdot 2^{2m/k}$ in $|\mathcal{S}|^{O(1)}$ time, hence proving Lemma 19.

References

1 Divesh Aggarwal, Rishav Gupta, Aditya Morolia, and Chuanqi Zhang. Mind the gap? not for svp hardness under eth!, 2026. arXiv:2504.02695.

2 Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.

3 Mitali Bafna, Karthik C. S., and Dor Minzer. Near optimal constant inapproximability under ETH for fundamental problems in parameterized complexity. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 2118–2129. ACM, 2025. doi:10.1145/3717823.3718257.

4 Huck Bennett, Mahdi Cheraghchi, Venkatesan Guruswami, and João Ribeiro. Parameterized inapproximability of the minimum distance problem over all fields and the shortest vector problem in all p norms. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 553–566. ACM, 2023. doi:10.1145/3564246.3585214.

5 Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inf. Theory*, 24(3):384–386, 1978. doi:10.1109/TIT.1978.1055873.

6 Arnab Bhattacharyya, Édouard Bonnet, László Egri, Suprovat Ghoshal, Karthik C. S., Bingkai Lin, Pasin Manurangsi, and Dániel Marx. Parameterized intractability of even set and shortest vector problem. *J. ACM*, 68(3):16:1–16:40, 2021. doi:10.1145/3444942.

7 Nir Bitansky, Prahladh Harsha, Yuval Ishai, Ron D. Rothblum, and David J. Wu. Dot-product proofs and their applications. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 806–825. IEEE, 2024. doi:10.1109/FOCS61266.2024.00057.

8 Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electron. Colloquium Comput. Complex.*, TR16-128, August 2016. URL: https://eccc.weizmann.ac.il/report/2016/128.

9 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0515-9.

10 Rodney G. Downey, Michael R. Fellows, Alexander Vardy, and Geoff Whittle. The parametrized complexity of some fundamental problems in coding theory. *SIAM J. Comput.*, 29(2):545–570, 1999. doi:10.1137/S0097539797323571.

11 I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, 2003. doi:10.1109/TIT.2002.806118.

12 Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, and Kewen Wu. Almost optimal time lower bound for approximating parameterized clique, csp, and more, under ETH. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 2136–2144. ACM, 2025. doi:10.1145/3717823.3718130.

13 Venkatesan Guruswami, Xuandi Ren, and Sai Sandeep. Baby PIH: Parameterized Inapproximability of Min CSP. In Rahul Santhanam, editor, *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:17, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2024.27.

14 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, March 2001. doi:10.1006/jcss.2000.1727.

15 Shuangle Li, Bingkai Lin, and Yuwei Liu. Improved lower bounds for approximating parameterized nearest codeword and related problems under ETH. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, Tallinn, Estonia, July 8-12, 2024*, volume 297 of *LIPIcs*, pages 107:1–107:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.107.

16 Pasin Manurangsi. Tight running time lower bounds for strong inapproximability of maximum k-coverage, unique set cover and related problems (via t-wise agreement testing theorem). In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 62–81. SIAM, 2020. doi:10.1137/1.9781611975994.5.

17 Craig A. Tovey. A simplified np-complete satisfiability problem. *Discret. Appl. Math.*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7.

Bounded-Independence Sampling of Edges for Combinatorial Graph Properties

Aaron Putterman
Harvard University, Cambridge, MA, USA

Salil Vadhan
Harvard University, Cambridge, MA, USA

Vadim Zaripov
Harvard University, Cambridge, MA, USA

Abstract

Random subsampling of edges is a commonly employed technique in graph algorithms, underlying a vast array of modern algorithmic breakthroughs. Unfortunately, using this technique often leads to randomized algorithms with no clear path to derandomization because the analyses rely on a union bound over exponentially many events. In this work, we revisit this goal of *derandomizing* randomized sampling in graphs.

We give several results related to bounded-independence edge subsampling, and in the process of doing so, generalize several of the results of Alon and Nussboim (FOCS 2008), who studied bounded-independence analogues of random graphs (which can be viewed as edge subsamples of the complete graph). Most notably, we show:

1. $O(\log(m))$-wise independence suffices for preserving connectivity when sampling at rate $1/2$ in a graph with minimum cut $\geq \kappa \log(m)$ with probability $1 - 1/\text{poly}(m)$ (for a sufficiently large constant κ).
2. $O(\log(m))$-wise $(1/\text{poly}(m))$-almost independence suffices for ensuring cycle-freeness when sampling at rate $1/2$ in a graph with minimum cycle length $\geq \kappa \log(m)$ with probability $1-1/\text{poly}(m)$ (for a sufficiently large constant κ).
3. If we relax to *arbitrary* distributions, we show there is an explicit distribution with marginals $\leq 1/2$ generated using $O(\log(m) \log\log(m))$ random bits such that in a graph with minimum cut $\geq \kappa \log(m)$ (for a sufficiently large constant κ), a sample from the distribution has is still connected with probability $1 - 1/\text{poly}(m)$.

To demonstrate the utility of our results, we revisit the classic problem of using parallel algorithms to find graphic matroid bases, first studied in the work of Karp, Upfal, and Wigderson (FOCS 1985). In this regime, we show that the optimal algorithms of Khanna, Putterman, and Song (arxiv 2025) can be *explicitly* derandomized while maintaining near-optimality.

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Graphs, random sampling

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.2

Related Version *Full Version*: https://arxiv.org/abs/2603.25095

Funding *Aaron Putterman*: Supported in part by the Simons Investigator awards of Madhu Sudan and Salil Vadhan, and AFOSR award FA9550-25-1-0112.
Salil Vadhan: Supported by a Simons Investigators Award.

© Aaron Putterman, Salil Vadhan, and Vadim Zaripov;
licensed under Creative Commons License CC-BY 4.0

1 Introduction

1.1 Background

Suppose that one wishes to run an algorithm on an Erdős–Rényi random graph $\mathcal{G}(n,p)$. There are exponentially many graphs in the support of $\mathcal{G}(n,p)$, so even storing a graph requires resources polynomial in n. A common approach for derandomizing such an algorithm is to replace $\mathcal{G}(n,p)$ with some random looking distribution $\mathcal{G}_n$ whose support is much smaller (for example, $\text{poly}(n)$), so that all the properties of random graphs on which the algorithm relies are still preserved. This line of work was initiated by Goldreich, Goldwasser, and Nussboim [16], who studied pseudorandom graphs that are indistinguishable from $\mathcal{G}(n,p)$ by any oracle machine that makes $\text{poly}(\log n)$ adjacency queries to a graph. Goldreich et al. considered several properties of random graphs, such as connectedness and Hamiltonicity, and presented pseudorandom graph constructions that preserve these properties while having support polynomial in n.

Alon and Nussboim [4] studied the specific strategy of using k-wise independence for sampling a pseudorandom graph. These can be sampled with support size $n^{\Theta(k)}$ and stored and accessed with $\widetilde{O}(k \log n)$ space and time. Similarly to [16], [4] show that many typical properties and thresholds that are enjoyed by truly random graphs *still* hold true with k-wise independent sampling for $k = O(\log n)$: for instance, edge connectivity, vertex connectivity, jumbledness, and the existence of matchings all mimic the behavior of truly independent sampling.

We continue this line of study by focusing on random subsampling of edges in graphs: instead of looking at $\mathcal{G}(n,p)$, we start with a given graph and randomly take each of its edges in our sample with some fixed probability, getting a sparser subgraph. This is a fundamental technique which underlies a huge portion of modern graph algorithms, including sparsification and its many applications [8, 28], coloring [5], vast arrays of sublinear algorithms [25], and many more. However, for these graph algorithms, the gold standard is often the design of *deterministic* algorithms, thereby bypassing any uncertainty in the algorithm's output. Motivated by this, the focus of one line of study, including this work, is on the ability to *derandomize* graph sampling.

Subsequent works have studied the feasibility of k-wise independent subsampling in arbitrary graphs: as an example, the work of Doron, Murtagh, Vadhan, and Zuckerman [11] studied the ability of k-wise independent sampling to produce *spectral sparsifiers* of graphs, again showing that mild independence (in fact, $k = O(\log(n))$) suffices for preserving the graph's spectrum.

However, these aforementioned results rely on a shared insight: namely that bounded independence sampling suffices in good spectral expanders for concentration of the *eigenvalues* of a graph's Laplacian. Alon and Nussboim [4] leverage the fact that a complete graph is a good expander, and therefore has eigenvalues well-separated from 0, and Doron et al. [11] assign different sampling probabilities to each edge proportional to their effective resistances, which was a procedure known to preserve graph's eigenvalues with fully independent sampling [27].

Thus, in our work we seek to understand:

> *Does k-wise independent subsampling for small k also preserve* combinatorial *properties of graphs that are not good expanders?*

By "combinatorial" properties, we mean those where we do not know how to use the spectrum directly to analyze fully independent subsampling.

As we shall see, we show that for certain properties the answer is strongly affirmative. In the following section, we discuss these results in more detail and subsequently explain how they lend themselves to new explicit derandomized graph algorithms.

1.2 Our Graph Sampling Results

As an illustration of the challenges we face, consider a graph $G = (V, E)$ on n vertices with minimum cut $\lambda \approx 100 \log(n)$. Karger's [18] cut-counting bound establishes that in such a graph, the number of cuts of size $\leq \alpha\lambda$ is at most $n^{2\alpha}$. An immediate corollary of this is that in such a graph G, if one samples each edge independently with probability $1/2$, then the resulting graph is connected with probability at least $1 - 1/\text{poly}(n)$. To see why, we bin our cuts to those of size in the interval $[\lambda, 2\lambda)$, $[2\lambda, 4\lambda)$, etc. Each cut in the bin $[\alpha\lambda, 2\alpha\lambda)$ has at least one edge in the surviving sample of edges with probability $1 - 1/2^{\alpha\lambda} = 1 - 1/n^{100\alpha}$. We can then take a union bound over all $\leq n^{4\alpha}$ cuts in this bin and then a further union bound over all $\leq n$ bins to conclude that *every* cut has at least one surviving edge in the sample with high probability, and thus the graph is connected.

As mentioned above, a natural approach to derandomizing the above property is to use k-wise independent sampling. Unfortunately, if we revisit the argument above, we can see that an essential step in arguing that the graph is connected under random subsampling is to take a union bound over an *exponential* number of cuts in the graph. This relies on having "success" probabilities for individual cuts that are as large as $1 - 2^{-(n-1)}$, since there are 2^{n-1} cuts in the graph. In general, no pseudorandom algorithm with a sample space smaller than 2^{n-1} can guarantee such high success probabilities, since for such an algorithm, the individual probabilities are multiples of $2^{-(n-1)}$.

Furthermore, a graph having a minimum cut of size $100 \log(n)$ provides *no meaningful bound* on its expansion or the effective resistances of its underlying edges, meaning that the techniques of [4] and [11] do not directly lend themselves to this setting. Thus, it may seem that we are unable to conclude this relatively benign connectivity property under bounded-independence sampling.

Despite the appearance of the above barriers, we show that for $k = O(\log(n))$, k-wise independent sampling still yields connected graphs with high probability.

▶ **Theorem 1.1.** *Let $G = (V, E)$ be a graph with m edges and minimum cut $\lambda \geq \kappa \log(m)$ for some absolute constant κ. Let $\widetilde{G}$ denote the result of subsampling edges of G in a $2\kappa \log(m)$-wise independent manner with marginals $1/2$. Then, $\widetilde{G}$ is connected with probability $\geq 1 - 1/\text{poly}(m)$.*

This result serves as a strong generalization of some of the results of [4]. Indeed, [4] studies which properties of *the complete graph* are preserved under k-wise independent subsampling. Naturally, working with such a structured graph enables simpler analysis. Nevertheless, in our much less structured regime where graphs only have a lower bound on their minimum cut, we are still able to show that $O(\log(m))$-wise independent sampling suffices for preserving connectivity.

Sampling from a $O(\log m)$-wise independent distribution requires $\Theta(\log^2(m))$ random bits; indeed the support must be of size $2^{\Omega(\log^2(m))}$ [3]. Thus, as an alternative, we construct another sampling procedure that maintains connectivity while having almost-polynomial support size:

▶ **Theorem 1.2.** *There exists an explicit distribution Y on $\{0,1\}^m$ that can be sampled using $O(\log m \cdot \log\log m)$ random bits and has marginals $\leq 1/2$ such that the following holds. For every graph $G = (V, E)$ with m edges and minimum cut $\geq \kappa \log(m)$ for some absolute constant κ, subsampling the edges of G using Y yields a connected subgraph with probability $1 - 1/\text{poly}(m)$.*

In a similar manner, we also study *cycle-free-ness* when randomly sampling edges with large girth. In more detail, consider a graph $G = (V, E)$ whose shortest cycle is of length $\geq 100 \log(n) = \lambda$. A bound due to Subramaniam [29] (and rediscovered in several other works, including [13]) states that in such a graph, the number of cycles of length $\leq \alpha \cdot \lambda$, is bounded by $n^{2\alpha}$. Thus, if one samples the edges in this graph uniformly at rate $1/2$, the exact same argument as in the cut setting above will show that the resulting sample of edges is *cycle free* with high probability. However, this argument again relies on exponential concentration in the tail, and thus does not immediately translate into a proof of the same property holding under bounded independence. Despite this, we are able to show that bounded independence suffices:

▶ **Theorem 1.3.** *Let $G = (V, E)$ be a graph with m edges and minimum cycle length $\lambda \geq \kappa \log(m)$ for some absolute constant κ. Let $\widetilde{G}$ denote the result of subsampling edges of G in a $(1/m^{200})$-almost $2\kappa \log(m)$-wise independent manner with marginals $1/2$. Then, $\widetilde{G}$ is cycle free with probability $\geq 1 - 1/\text{poly}(m)$.*

Unlike Theorem 1.1, this theorem allows for the use of an *almost* $O(\log m)$-wise independent distribution, which can be sampled using $O(\log m)$ random bits.

Together, these results show that bounded-independence sampling *does suffice* for preserving basic combinatorial properties in weakly-structured graphs (i.e., non-expanders). As we elaborate upon more in the Technical Overview (Section 1.4), our results rely on very careful analyses of graph structure. Furthermore, this improved understanding of graph sampling under bounded-independence immediately lends itself to better deterministic parallel algorithms for finding matroid bases.

1.3 Applications to Parallel Matroid Algorithms

1.3.1 Matroid Background

Matroids are fundamental and ubiquitous objects in combinatorial optimization, generalizing a number of basic structures like spanning forests in graphs, and bases of vector spaces. More formally, a matroid is modelled as a set system $\mathcal{M} = (E, \mathcal{I})$, where E is the *ground set*, consisting of m elements, and $\mathcal{I} \subseteq 2^E$ is the *set of independent sets*, satisfying:

1. $\emptyset \in \mathcal{I}$ (non-empty property).
2. If $S \in \mathcal{I}$ and $S' \subseteq S$, then $S \in \mathcal{I}$ (downward closure property).
3. If $S, T \in \mathcal{I}$ and $|S| > |T|$, then $\exists e \in S \setminus T : T \cup \{e\} \in \mathcal{I}$ (exchange property).

Within the study of matroids, a critical notion is that of a *basis*, or namely a set $S \in \mathcal{I}$ that is *maximal* under inclusion. Note that because of the exchange property above, we know that for a given matroid $\mathcal{M} = (E, \mathcal{I})$, all bases of $\mathcal{M}$ are of the same size.

For instance, one can take the ground set E to be the set of edges in a graph $G = (V, E)$, and let a set of edges $S \subseteq E$ be independent if and only if S is *cycle-free* in the graph G. This yields the family of so-called *graphic matroids*, with a maximal independent set (i.e., basis) being a spanning forest of G.

1.3.2 Parallel Basis Finding

Despite their prevalence in combinatorial optimization (see e.g., [12, 17, 15, 24, 6, 9]), our understanding of matroids is incomplete, even in basic directions. One such direction is in the study of *parallel algorithms* for finding matroid bases. In this model, first introduced by Karp, Upfal, and Wigderson [19, 20], one wishes to find a basis of some matroid $\mathcal{M} = (E, \mathcal{I})$.

However, because the number of matroids on n elements grows as 2^{2^n} [7], it is too expensive to store the entire description of the matroid. Instead, the algorithm is given access to the matroid only in the form of an *independence oracle*; i.e., a function $\mathrm{Ind} : 2^E \to \{0,1\}$, such that $\mathrm{Ind}(S) = \mathbf{1}[S \in \mathcal{I}]$. Under this form of access, the works of Karp, Upfal, and Wigderson [19, 20] asked:

> *How many adaptive rounds of independence queries are required to find a basis of an arbitrary matroid, when each round is allowed only* $\mathrm{poly}(n)$ *many queries to the independence oracle?*

In the seminal work [19], it was proven that in the setting of arbitrary matroids, there is a *deterministic* algorithm which finds a basis of any matroid in just $O(\sqrt{n})$ rounds, and that there is a lower bound of $\Omega(n^{1/3})$ many rounds for this task. Despite providing a narrow gap for the parallel complexity of this basic problem, the ensuing four decades saw no improvements to these bounds. Only recently, in the work of Khanna, Putterman, and Song [21, 23], were these bounds improved, with the establishment of an $\widetilde{O}(n^{3/7})$ round *randomized* algorithm for finding bases in arbitrary matroids. Nevertheless, the true complexity of this problem is yet to be established.

Due to the complex structure of arbitrary matroids, the work of [19] proposed studying the parallel complexity of *concrete classes* of matroids. Among these, the most notable class studied by [19] is the class of *graphic matroids*:

▶ **Definition 1.4.** *Given a graph $G = (V, E)$, the* graphic matroid *induced by G is the matroid $\mathcal{M} = (E, \mathcal{I})$ such that for a set $S \subseteq E$, $S \in \mathcal{I}$ if and only if S is cycle-free.*

For this class of matroids, [19] provided deterministic algorithms that, on graphs with m edges, achieve either (a) for any constant $d \in \mathbb{N}$, $m^{1/d}$ rounds and $m^{O(d)}$ queries per round, or (b) $O(\log(m))$ rounds with $m^{O(\log(m))}$ queries per round. This was later improved in the work of Khanna, Putterman, and Song [22], which provided an algorithm that finds bases of graphic matroids with only $\mathrm{poly}(m)$ queries per round and $O(\log(m))$ many rounds.

Note that, as mentioned in their work, the algorithms of [22] can be derandomized *non-explicitly*, in an argument akin to Adleman's proof of $\mathbf{BPP} \subseteq \mathbf{P/poly}$ [1]. This of course answers the natural question of the deterministic "query complexity" of finding a matroid basis, but does not yield an efficiently implementable uniform algorithm.

1.3.3 Our Results

Our first result uses Theorem 1.3 to deterministically find bases in *graphic matroids*. In this regime, we show the following:

▶ **Theorem 1.5.** *There is a uniform, deterministic algorithm that, for any graphic matroid on m elements, finds a basis in $O(\log m \cdot \log\log m)$ many rounds of $\mathrm{poly}(m)$ queries.*

This theorem shows that the result of [22] *can* be explicitly derandomized with only a minor increase in the number of rounds (from $\log(m)$ to $\log(m)\log\log(m)$), while still retaining a polynomial number of queries per round. The proof of this result relies intimately on Theorem 1.3 (in fact, even on a strengthening of Theorem 1.3).

Next, as an application of Theorem 1.2, we show that basis finding in *cographic* matroids can also be derandomized. A set of edges $S \subseteq E$ in a cographic matroid for a graph G is defined to be independent if and only if S does not completely contain any non-empty cut C of G, and so a basis of a cographic matroid is a set of edges that is the *complement* of a spanning forest.

▶ **Theorem 1.6.** *There is a uniform, deterministic algorithm that, for any cographic matroid on m elements, finds a basis in only $O(\log m \cdot \log\log m)$ many rounds of $m^{O(\log\log(m))}$ queries.*

Note that, prior to this work, there was no non-trivial uniform, deterministic algorithm for basis finding in cographic matroids. The best algorithm was only known to use $O(\sqrt{m})$ rounds and poly(m) queries per round - due to [19] for *arbitrary matroids* (thus including cographic ones as a special case). We view Theorem 1.6 as further evidence that one can likely obtain polylog(m) rounds and poly(m) queries, though we leave this as an open question.

As we elaborate upon in the next section, both Theorem 1.5 and Theorem 1.6 rely on a new algorithmic framework for finding bases of matroids which, unlike the algorithms of [22], uses both deletion and contraction of elements. However, to remove the randomness in this algorithmic framework, our results rely on the concrete derandomization results for basic graph sampling tasks that we discussed above.

1.4 Technical Overview

In this overview, we discuss some of the intuition that underlies both our derandomized graph sampling results as well as our improved explicit deterministic matroid algorithms.

1.4.1 Derandomizing Graph Sampling

We start by providing intuition for Theorem 1.1. Before doing so, we briefly recap the discussion in Section 1.2. In this setting, we are given a graph $G = (V, E)$ with minimum cut λ of size $\geq \kappa \log(m)$ for some constant κ. Our goal is to subsample the edges of this graph at rate $1/2$ using a *bounded-independence* distribution while still ensuring that the resulting sampled graph is connected with high probability. As mentioned in Section 1.2 however, a simple union bound over all of the cuts in the graph *does not* suffice for arguing connectivity when using bounded-independence distributions. Indeed, there are 2^{n-1} cuts in a graph on n vertices, and thus any union bound requires that most such cuts have at least one surviving edge with probability $1 - \Omega\left(2^{-n}\right)$; a probability which is too close to 1 to be possible under k-wise independent sampling with $k = o(n)$.

For comparison, [4] and [11] have studied bounded-independence sampling for small k (generally, $k = O(\log(n))$) and showed that in these settings, many spectral properties of graphs which are good expanders can be preserved. Here, spectral properties of a graph $G = (V, E)$ refer to properties of the graph's Laplacian $L_G = \sum_{e=(u,v)\in E} w_e \cdot \chi_e \chi_e^T$, where $\chi_e \in \mathbb{R}^n$ is the vector such that $\chi_e = \mathbf{1}_u - \mathbf{1}_v$ when $e = (u, v)$. In this direction, [11] provides a more general result than [4]: indeed, it follows from [11] that if the *leverage scores* in a graph G are all smaller than $\approx \frac{\varepsilon^2}{2\log(n)}$, then $O(\log(n))$-wise independent sampling with marginals $1/2$ yields a *spectral sparsifier* of the graph G. In this context, the leverage score of an edge $e = (u, v)$ is defined as $\ell(u, v) = w_e \cdot \chi_e^T L_G^\dagger \chi_e$ and provides a measure of how important the edge (u, v) is for preserving connectivity between u and v. A spectral sparsifier of a graph G is simply a subgraph $\widetilde{G}$ such that $(1-\varepsilon)L_G \preceq L_{\widetilde{G}} \preceq (1+\varepsilon)L_G$, where $\preceq$ is the Loewner order on PSD matrices ($A \preceq B$ if $x^T A x \leq x^T B x$ for all x). If $\widetilde{G}$ is a spectral sparsifier of G, then this implies that for every $x \in \{0, 1\}^n$, $x^T L_{\widetilde{G}} x \in (1 \pm \varepsilon) x^T L_G x$, which means that $x^T L_{\widetilde{G}} x = 0$ if and only if $x^T L_G x = 0$. This means that $\widetilde{G}$ and G will have the same number of connected components (since these equal the multiplicity of 0 as an eigenvalue of the Laplacian), and in fact the connected components will be identical.

This provides a natural first attempt towards showing that $O(\log(m))$-wise independent sampling in our graph G preserves connectivity. I.e., can we try to argue the *stronger* property that the resulting sampled edges constitute a *spectral sparsifier* of G? Unfortunately,

this turns out not to be the case. This is because a graph having a large minimum cut is very strongly *not* sufficient for the graph to have small leverage scores (see [14] for more discussion). Indeed, even if we tried sampling the edges of G *uniformly* at random, the resulting graph would not be guaranteed to be a spectral sparsifier with high probability.

To overcome this, we rely on a key observation: the graph G that we are working with so far is *unweighted*. But, edge weights *can* alter the behavior of the graph's Laplacian, and in turn, the leverage scores of the edges. Thus, one may wonder: is it possible to re-weight the edges of the graph such that all the leverage scores become smaller? Indeed, one of our key contributions is to show that this is true, and in doing so, we provide a new connection between leverage scores and a graph's minimum cut:

▶ **Theorem 1.7.** *Given an unweighted graph $G = (V, E)$ with minimum cut c, there exists a weighting $w : E \to \mathbb{R}_{\geq 0}$ such that for the weighted graph $G' = (V, E, w)$ and for each $e \in E$, the leverage score of e in G' is $\ell(e) \leq O(1/c)$.*

Moreover, the converse is also true: if there exists a weighting such that all leverage scores are at most $1/c$, the minimal cut of the original graph is $\geq c$.

With this theorem in hand, we are then immediately able to show that connectivity is preserved whenever our graph G has minimum cut $\geq \kappa \log(m)$: indeed, given the graph $G = (V, E)$, we implicitly analyze the *re-weighted* graph $\hat{G}$. On $\hat{G}$, we sample each edge using our $O(\log(m))$-wise independent distribution, and are guaranteed via [11] that the resulting sampled edges are a spectral sparsifier of $\hat{G}$, which in particular means that the sampled edges form a connected graph. Then, one must only observe that the sampling procedure on G and $\hat{G}$ is identical, as both have the same set of edges.

Proving Theorem 1.7 is more subtle; our proof relies on a "bounded effective resistance diameter" decomposition theorem of [2], along with a careful, recursive weight assignment scheme. We omit the details from the technical overview for brevity, and direct the reader to Section 3.1 for more details.

In a separate direction, our proof of Theorem 1.3 foregoes a spectral argument entirely. Indeed, in this setting where we must argue *cycle-freeness*, we instead show that the existence of *any cycles* in the graph can instead be modeled via the existence of *short paths* in the graph. This allows us to transition from an event space of nearly-exponential size (i.e., the set of all cycles) to instead work with an event space of only polynomial size (pairs of vertices with short paths between them). We present this argument in Section 4.

The proof Theorem 1.2 relies on different techniques. We construct our distribution Y on $\{0,1\}^m$ by combining $O(\log m)$ independent copies of any distribution Y_0 on $\{0,1\}^{m \times \log\log m}$ that fools read-once DNF formulas on $\widetilde{O}(\log m)$ (out of the m possible) variables. By the results of De, Etesami, Trevisan, and Tulsiani [10], such a distribution Y_0 can be sampled using $O(\log\log m)$ random bits, meaning that we can generate Y using $O(\log m \cdot \log\log m)$ random bits. Interestingly, the way we combine the copies of Y_0 to set Y is also using a read-once DNF formula, namely the Tribes function.

Together, these results provide a robust toolkit for reasoning about graph properties under bounded-independence sampling when the underlying graph is not a good expander. As we shall see below, this has direct applications to derandomized matroid basis computation.

1.4.2 Derandomized Matroid Basis Finding

Once we have our improved derandomized graph sampling results, there are still barriers towards implementing better explicit parallel basis finding algorithms. To illustrate these barriers, we first revisit the algorithm of [22].

Indeed, let us consider the setting of graphic matroids; recall that here there is an underlying graph $G = (V, E)$, and the goal of the algorithm is to find a *spanning forest* of G. The algorithms in this setting only have access to an *independence oracle*, meaning that the algorithm can query a set $S \subseteq E$ of edges, and the oracle reports whether the set S has any cycles. The intuition for the algorithm of [22] is that each round of the algorithm *deletes* more edges from the graph without altering the connected components of the graph. Then, after $O(\log(n))$ many rounds of computation, [22] shows that the graph has no cycles remaining, and thus the leftover edges must be a spanning forest of the original graph G. To implement this argument more carefully, [22] relies on two key invariants:

1. After i rounds of the algorithm, the remaining edges $E^{(i)} \subseteq E$ induce the same connected components as E.
2. After i rounds of the algorithm, there are no cycles in $E^{(i)}$ of length $\leq 1.01^i$.

Clearly then, one can see that after $O(\log(n))$ rounds, the remaining edges constitute a spanning forest.

To actually ensure these invariants hold, [22] relies on random sampling of edges. Indeed, in an iteration i where the minimum cycle length in $E^{(i)}$ is $\lambda = 1.01^i$, [22] shows that *uniformly random sampling* of edges can be used to isolate any near-minimum length cycle. Formally, [22] shows:

▶ **Lemma 1.8** ([22]). *Let $G = (V, E)$ be a graph with m edges and minimum cycle length λ. Let $C \subseteq E$ be a cycle in G of length $\leq 1.01\lambda$. Now, let $\widetilde{G}$ be the result of uniformly randomly sampling the edges of G at rate $p = 1/m^{100/\lambda}$. Then, $C \subseteq E$ is the* unique *surviving cycle in $\widetilde{G}$ with probability at least $1/\mathrm{poly}(m)$.*

With such a lemma in hand, [22] performs random sampling a large polynomial number times, ensuring that every single near-minimum length cycle is the unique surviving cycle in some sampled graph. This then enables [22] to recover the identities of *all* near-minimum length cycles (and in so doing, also enables their removal without altering connectivity).

Our goal is to adopt this algorithmic framework, but to use bounded-independence sampling instead of truly random sampling. Ultimately, this then allows us to *enumerate* all possible samples of edges, thereby obtaining explicit, deterministic results.

Unfortunately, directly using our derandomized graph sampling results *does not* work inside this algorithmic framework. There are two primary reasons why:

1.4.2.1 Unique Cycle Survival

First, as currently stated, Theorem 1.3 only governs the probability that a sampled graph is *cycle-free*, *not* the probability that there is a single, unique cycle which survives the bounded-independence sampling. Fortunately, Theorem 1.3 can be modified to ensure unique cycle survival in the regime where the minimum cycle length is $\lambda = \Theta(\log(m))$. To see why, fix a single cycle C of length $\leq 1.01\lambda$. For this cycle, if one uses (almost) k-wise independent sampling with $k \geq 10\lambda$, then C's survival probability under bounded-independence sampling is in fact *the same* as under a uniformly random distribution. Now, conditioned on C's survival, we can re-invoke Theorem 1.3 on the remaining graph to ensure that *no other cycles* survive sampling. This argument requires care, and appears in Section 4.2.

1.4.2.2 Large Cycle Lengths

The second barrier is that, as stated, Theorem 1.3 requires a minimum cycle length of at least $\kappa \log(m)$, along with sampling at rate $1/2$. Not only this, but in the previous barrier, the argument we just discussed relies on using k-wise (almost) independent sampling for $k \geq 10\lambda$, where λ is the current minimum cycle length.

Unfortunately, the algorithmic framework of [22] requires a spectrum of different cycle lengths and sampling rates, starting with a minimum cycle length of 1, and slowly increasing as edges are deleted between rounds. Thus, there are even rounds where the minimum cycle length $\lambda = \Omega(m)$, which in our above argument requires $\Omega(m)$-wise independent sampling, which is far too costly.

To overcome this barrier, we employ a much more careful analysis than [22]: rather than continuing to remove short cycles until the minimum cycle length is $> m$, we instead invest $O(\log\log(m))$ rounds until the minimum cycle length becomes $\approx \kappa \log(m)$. Note that in these rounds where the minimum cycle length is $o(\log(m))$, we even require a modification to Theorem 1.3 which shows that one can sample *at smaller marginal rates* (instead of $1/2$) while still ensuring unique cycle survival. We then progressively alter these sampling rates until the minimum cycle length becomes $\approx \kappa \log(m)$.

Once the minimum cycle length becomes $\approx \kappa \log(m)$, instead of trying to enumerate short cycles in the graph, we instead *directly invoke* Theorem 1.3 to find *large sets of cycle-free edges* (i.e., $\Omega(m)$ many edges without any cycles). Once we have such a set of edges, we then *commit* to including these edges in our final spanning forest. Thus, from a graph on n vertices, we now have collected $\Omega(m)$ of the edges we need for our ultimate spanning forest.

The key invariant now is that every $O(\log\log(m))$ rounds, we recover an $\Omega(1)$ fraction of the remaining edges we need for our spanning forest. Thus, after only $O(\log(m)\log\log(m))$ rounds, we recover a spanning forest of our starting graph.

We use the same strategy in the cographic setting. Namely, within each recursive iteration we invest $O(\log\log(m))$ rounds to eliminate small cuts, so that the minimum cut becomes $\approx \kappa \log(m)$ (to achieve this, we rely on a modification of Theorem 1.2 that allows sampling at smaller marginal rates). We then directly invoke Theorem 1.2 to find a large independent set and commit to including it in the basis, moving to the next iteration.

1.5 Organization

We start with preliminaries in Section 2. In Section 3 and Section 4, we provide tight guarantees on how graph structure behaves under bounded independence sampling: Section 3 shows that graphs with logarithmic min-cut remain connected under bounded-independence sampling, while Section 4 shows that graphs with large minimum cycle length become cycle-free.

In the full version, we present a general framework that underlies our basis finding algorithms for both graphic and cographic matroids. We then use the cycle-freeness characterization to design near-optimal explicit, deterministic parallel algorithms for finding bases of graphic matroids, and use the aforementioned connectedness guarantee to provide explicit, deterministic parallel algorithms for finding bases of cographic matroids.

2 Preliminaries

We begin with providing tools and definitions that are necessary for the remainder of this paper.

An *unweighted undirected graph* is defined as $G = (V, E)$, where V is the set of vertices and E is the set of edges. We always use n, m to be the number of vertices and edges of a graph, respectively: $n = |V|, m = |E|$. A *weighted* graph $G = (V, E, w)$ also has a function $w : E \to \mathbb{R}^+$, weighting the edges. For any subset of edges $A \subseteq E$, $w(A)$ denotes the sum of the weights of all edges in A. For any $v \in V$, $\delta(v) \subseteq V$ denotes the set of *neighbors* of v in G. For any subset of vertices $V' \subseteq V$, $G[V'] = (V', E', w')$ denotes the *subgraph of G induced*

by V', while $E(V') \subseteq E$ denotes the set of edges with both endpoints in V'. We say that a set of edges $C \subseteq E$ is a *cut* in G if and only if there is a set $S \subseteq V$ such that C is the set of edges between S and $\bar{S}$.

For any subset of vertices $S \subseteq V$, G/S denotes the result of *contracting* G on a set S, i.e. identifying all the vertices of S. Specifically, $G/S = ((V \setminus S) \cup \{s^*\}, E')$, where $E' = \{(u,v) : (u,v) \in E \text{ and } u,v \notin S\} \cup \{(u,s^*) : (u,s) \in E \text{ for some } s \in S\}$. Note that G/S is a multigraph: if u is connected to multiple edges in S, we keep all these edges as parallel edges in G/S.

Importantly, contracting a graph on a subset of vertices $S \subseteq V$ does not decrease its minimum cut, since it only limits the set of all cuts to those that do not cut through S.

We say that $f : \mathbb{N} \to \mathbb{N}$ is $\text{poly}(n)$ if there exist $n_0, c \in \mathbb{N}$ such that $\forall n \geq n_0$, $f(n) \leq n^c$.

Finally, we use $\tilde{O}$ notation to omit polylogarithmic factors: $\tilde{O}(f(n)) = O(f(n) \cdot \text{poly}(\log n))$.

2.1 Bounded-Independence Sampling

We start by recalling the notion of a δ-almost k-wise independent distribution:

▶ **Definition 2.1.** *We say that a distribution* $X = (x_1, \dots x_n)$ *is* δ-almost k-wise independent *with marginals* p *if for all subsets* $S \subseteq [n], |S| \leq k$, *we have*

$$d_{\text{TV}}(U(S), X(S)) \leq \delta,$$

where d_{TV} *is the* total variation distance *and* $U(S)$ *denotes the uniform distribution (with marginals* p*) over the set* S. *When* $\delta = 0$, *i.e.* $X(S) = U(S)\ \forall S : |S| \leq k$, *we say that* X *is* k*-wise independent.*

The work of Naor and Naor [26] explicitly constructed such δ-almost k-wise independent random variables of small size:

▶ **Theorem 2.2** ([26])**.** *There is an explicit construction of a* δ*-almost* k*-wise independent distribution* $X = (x_1, \dots x_n)$ *with marginals* $1/2$ *and seed length* $O(k + \log\log(n) + \log(1/\delta))$.

By "explicit" construction of a distribution X on $\{0,1\}^n$, we mean that there is a polynomial time algorithm $G : \{0,1\}^r \to \{0,1\}^n$ (where r is the *seed length* or the *number of random bits* used to generate X), such that $X = G(U_r)$.

When $\delta = 0$ (exact k-wise independence) we have the following space bounds:

▶ **Theorem 2.3** ([30])**.** *There is an explicit construction of a* k*-wise independent distribution* $X = (x_1, \dots x_n)$ *with marginals* $1/2$ *and seed length* $O(k \log(n))$.

The following reduction allows us to construct bounded-independence distributions with arbitrarily small marginal probabilities:

Algorithm 2.1 SampleLowerMarginals.

input : $n, k \in \mathbb{N}, \delta \geq 0, p$ s.t. $\log(1/p) \in \mathbb{Z}$;
$(y_1, \dots y_{n\log(1/p)})$ sampled from a δ-almost $k\log(1/p)$-wise independent distribution Y with marginals $1/2$.

output : $X = (x_1, ..., x_n)$, where $x_i = \prod_{j=(i-1)\log(1/p)+1}^{i\log(1/p)} y_j$ for every $i \in [n]$.

▶ **Lemma 2.4.** *For any $n, k \in \mathbb{N}$, $\delta \geq 0$, p such that $\log(1/p)$ is an integer, and δ-almost $k\log(1/p)$-wise independent distribution $Y = (y_1, \ldots y_{n\log(1/p)})$ with marginals $1/2$, the distribution $X = (x_1, \ldots x_n)$ constructed using Algorithm 2.1 is δ-almost k-wise independent with marginals p.*

The proof of this lemma is deferred to the full version of the paper.

The support size of the distribution X obtained from Algorithm 2.1 is the same as the support size of the original distribution Y with marginals $1/2$. Hence, for $\delta > 0$, using Theorem 2.2 with Algorithm 2.1 we get a δ-almost k-wise independent distribution with marginals p and seed length $O(k\log(1/p) + \log\log(n\log(1/p)) + \log(1/\delta))$. Similarly, for $\delta = 0$, using Theorem 2.3 with Algorithm 2.1 we get a k-wise independent distribution with marginals p and seed length $O(k\log(1/p)\log(n\log(1/p)))$.

2.2 Electric Flow and Effective Resistance

▶ **Definition 2.5** (Laplacian Matrix). *Given an undirected weighted graph $G = (V, E, w)$ on n vertices, the* Laplacian matrix $\mathbf{L}_G \in \mathbb{R}^{n\times n}$ *of G is defined as:*

$$\mathbf{L}_G = \mathbf{D}_G - \mathbf{A}_G,$$

where $\mathbf{D}_G \in \mathbb{R}^{n\times n}$ is the diagonal matrix of weighted degrees of G, and $\mathbf{A}_G$ is the weighted adjacency matrix of G: $(\mathbf{A}_G)_{ij} = (\mathbf{A}_G)_{ji} = w_{ij}$.

It is easy to verify that for any graph G, $\mathbf{L}_G$ is positive semidefinite. Indeed, for any $\mathbf{x} \in \mathbb{R}^n$, the Laplacian quadratic form is: $\mathbf{x}^T\mathbf{L}_G\mathbf{x} = \sum_{(u,v)\in E} w(u,v)\cdot(\mathbf{x}(u) - \mathbf{x}(v))^2 \geq 0$ since the weights are non-negative.

Since $\mathbf{L}_G$ is symmetric, it has a basis of eigenvectors. Hence, we use $\lambda_1, \lambda_2, ..., \lambda_n$ to denote the eigenvalues of $\mathbf{L}_G$ in increasing order ($\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$). Since $\mathbf{L}_G\mathbf{1}^n = 0$, $\lambda_1 = 0$ for any graph G. It can further be verified that $\lambda_2 = 0$ iff G is not connected.

▶ **Definition 2.6** (Spectral Approximation). *For $\varepsilon \geq 0$, a graph H is an* ε-spectral approximation *of a graph G if for all $\mathbf{x} \in \mathbb{R}^n$ we have that $\mathbf{x}^T\mathbf{L}_H\mathbf{x} \in (1 \pm \varepsilon)\cdot\mathbf{x}^T\mathbf{L}_G\mathbf{x}$. In this case we write $\mathbf{L}_H \cong_\varepsilon \mathbf{L}_G$.*

▶ **Definition 2.7** (Network Flow). *Given a graph $G = (V, E, w)$, some external flows $\mathbf{i}_{ext} \in \mathbb{R}^n$, and an arbitrary orientation of edges $E^\pm$, a* flow *is a function $\mathbf{f} \in \mathbb{R}^{E^\pm}$ such that $\mathbf{f}(u,v) = -\mathbf{f}(v,u)\ \forall (u,v) \in E^\pm$ and for any $v \in V$:*

$$(\mathbf{i}_{ext})_v + \sum_{u\in\delta(v)} \mathbf{f}(u,v) = 0 \qquad \text{(flow conservation)}$$

The flow conservation law simply implies that no flow disappears or is created at any vertex. In contrast to arbitrary network flows, an electric flow also has to obey Ohm's law:

▶ **Definition 2.8** (Electric Flow). *Given a graph $G = (V, E, w)$, external flows $\mathbf{i}_{ext} \in \mathbb{R}^V$, and an arbitrary orientation of edges $E^\pm$, a network flow $\mathbf{f} \in \mathbb{R}^{E^\pm}$ (as in Definition 2.7) is an* electric flow *if there exists a potential function $\mathbf{p} \in \mathbb{R}^V$ such that $\forall (u,v) \in E$:*

$$\mathbf{f}(u,v) = w(u,v)(\mathbf{p}(u) - \mathbf{p}(v)) \qquad \text{(Ohm's law)}$$

An important property that distinguishes an electric flow from just an arbitrary network flow is that it minimizes the electrical energy:

▶ **Definition 2.9** (Electrical Energy). *Given a graph $G = (V, E, w)$ and some network flow $\mathbf{f} \in \mathbb{R}^{E^{\pm}}$ (as in Definition 2.7) in this graph, the* electrical energy *of $\mathbf{f}$ is defined as:*

$$\mathcal{E}(\mathbf{f}) = \sum_{e \in E^{\pm}} \frac{\mathbf{f}(e)^2}{w(e)}$$

▶ **Fact 2.10.** *Given a graph $G = (V, E, w)$ and external currents $\mathbf{i}_{ext} \in \mathbb{R}^V$, an electric flow $\mathbf{f} \in \mathbb{R}^{E^{\pm}}$ is the unique network flow (with the given external currents) minimizing $\mathcal{E}(\mathbf{f})$.*

▶ **Definition 2.11** (Effective Resistance). *Given a weighted undirected graph $G = (V, E, w)$, the* effective resistance *between $u, v \in V$ is the potential difference between u and v induced by the unit u-v electric flow (i.e. the unique electric flow induced by setting $\mathbf{i}_{ext}(u) = 1$, $\mathbf{i}_{ext}(v) = -1$, and $\mathbf{i}_{ext}(s) = 0$ for $s \notin \{u, v\}$):*

$$\mathrm{Reff}_G(u, v) = \mathbf{p}(u) - \mathbf{p}(v)$$

▶ **Fact 2.12** (Thomson's Principle). *Given a graph $G = (V, E, w)$ and a unit u-v electric flow $\mathbf{f}_{uv} \in \mathbb{R}^{E^{\pm}}$, $\mathrm{Reff}_G(u, v) = \mathcal{E}(\mathbf{f}_{uv})$.*

▶ **Definition 2.13** (Effective Resistance Diameter). Effective resistance diameter *of a graph $G = (V, E, w)$ equals the maximum effective resistance among all pairs of vertices:*

$$\mathrm{R}_{\mathrm{diam}}(G) = \max_{u,v \in V} \mathrm{Reff}_G(u, v)$$

▶ **Definition 2.14** (Leverage Score). *Given a graph $G = (V, E, w)$, the* leverage score *of an edge $(u, v) \in E$, denoted as $\ell(u, v)$, is the ratio between the effective resistance between its endpoints $\mathrm{Reff}_G(u, v)$ and the actual resistance of an edge $1/w(u, v)$:*

$$\ell(u, v) = w(u, v) \cdot \mathrm{Reff}_G(u, v)$$

▶ **Fact 2.15.** *Consider a connected graph $G = (V, E, w)$. If we sample a spanning tree of G so that the probability of sampling a given tree is proportional to the product of the weights of edges in it, then the probability that a particular edge $(u, v) \in E$ is included in the sampled tree equals $\ell(u, v)$.*

2.3 Graph Clustering using Effective Resistance

Alev et al. [2] show that any graph can be partitioned into components with small effective resistance diameter:

▶ **Theorem 2.16** ([2]). *Given a weighted graph $G = (V, E, w)$, and a large enough parameter $\delta > 1$, there is an algorithm with time complexity $\tilde{O}\left(m \cdot n \cdot \log\left(\frac{w(E)}{\min_e w(e)}\right)\right)$ that finds a partition $V = \bigcup_{i=1}^{h} V_i$ satisfying*

1. $w\left(E - \cup_{i=1}^{h} E\left(V_i\right)\right) = O\left(w(E)/\delta\right)$,
2. $\mathrm{R}_{\mathrm{diam}}(G\left[V_i\right]) = O\left(\delta^3 \cdot \frac{|V|}{w(E)}\right)$ *for all $i = 1, ..., h$.*

Making a constant δ large enough gives us the following corollary:

▶ **Corollary 2.17.** *There exists an absolute constant α such that given a weighted graph $G = (V, E, w)$, there is an algorithm with time complexity $\tilde{O}\left(m \cdot n \cdot \log\left(\frac{w(E)}{\min_e w(e)}\right)\right)$ that finds a partition $V = \bigcup_{i=1}^{h} V_i$ satisfying:*

1. $w\left(E - \cup_{i=1}^{h} E\left(V_i\right)\right) \leq w(E)/2$
2. $\mathrm{R}_{\mathrm{diam}}(G\left[V_i\right]) \leq \alpha \frac{|V|}{w(E)}$ *for all $i = 1, ..., h$.*

2.4 Spectral Sparsification via Bounded Independence Sampling

Doron et al. [11] give an algorithm for spectral sparsification via bounded independence sampling:

Algorithm 2.2 Sparsify$(G = (V, E, w), \{\tilde{R}_{ab}\}_{(a,b)\in E}, k, \varepsilon, \delta)$.

1. Initialize H to be the empty graph on $n = |V(G)|$ vertices.
2. Set $s \leftarrow \frac{18e \log n}{\varepsilon^2} \cdot \left(\frac{n}{\delta}\right)^{2/k}$.
3. For every edge $(a, b) \in E$, set $p_{ab} \leftarrow \min\left\{1, w_{ab} \cdot \widetilde{R}_{ab} \cdot s\right\}$
4. For every edge $(a, b) \in E$, add (a, b) to H with weight w_{ab}/p_{ab} with probability p_{ab}. Do this sampling in a k-wise independent manner.
5. Return H.

▶ **Theorem 2.18** ([11] Theorem 3.1). *Let $G = (V, E, w)$ be an undirected connected weighted graph on n vertices with Laplacian $\mathbf{L}_G$ and effective resistances $R = \{R_{ab}\}_{(a,b)\in E}$. Let $0 < \varepsilon < 1$, $0 < \delta < 1/2$, and let $k \le \log n$ be an even integer. Let H be an output of* Sparsify$(G, R, k, \varepsilon, \delta)$ *and let $\mathbf{L}_H$ be its Laplacian. Then, with probability at least $1 - 2\delta$ we have:*

1. $\mathbf{L}_H \cong_\varepsilon \mathbf{L}_G$,
2. *H has $O\left(\frac{1}{\delta^{1+2/k}} \cdot \frac{\log n}{e^2} \cdot n^{1+\frac{2}{k}}\right)$ edges.*

Note that if G is connected and $\mathbf{L}_H \cong_\varepsilon \mathbf{L}_G$ for some $\varepsilon \ge 0$, then H must also be connected. Thus, setting $\varepsilon = 1/2$, $k = \Theta(\log n)$, and $\delta = 1/\text{poly}(n)$, get that $s = \Theta(\log n)$ and $p_{ab} = \Theta(w_{ab} \cdot R_{ab} \cdot \log n)$, giving us the following corollary:

▶ **Corollary 2.19.** *Let $G = (V, E, w)$ be an undirected connected weighted graph on n vertices with effective resistances $\{R_{ab}\}_{(a,b)\in E}$. Then, for any $\delta = 1/\text{poly}(n)$, sampling edges of G in a $\Theta(\log n)$-wise independent manner with marginals $\Theta(w_{ab} \cdot R_{ab} \cdot \log n)$ for each $(a, b) \in E$ yields a connected graph H with probability $1 - 2\delta$.*

3 Connectedness Under Bounded-Independence Edge Subsampling

3.1 Leverage Scores and Minimal Cut Duality

In this section, we prove the following theorem:

▶ **Theorem 3.1.** *Given an unweighted graph $G = (V, E)$ with m edges and minimum cut c, there exists a weighting $w : E \to \mathbb{R}$ such that the weighted graph $G' = (V, E, w)$ satisfies the following:*

- *for each $e \in E$, the leverage score is $\ell(e) \le 4\alpha/c$, where α is an absolute constant from Corollary 2.17,*
- $w_{\max}/w_{\min} = m^{O(\log m)}$.

Moreover, the converse is also true: if there exists a weighting such that all leverage scores are at most $1/c$, the minimal cut of the original graph is $\ge c$.

We prove the theorem by recursively decomposing our graph into components with high effective resistance diameter, reweighting the edges within each component, and then contracting the graph on these components. We start by establishing the following useful lemma:

▶ **Lemma 3.2.** *Given a graph $G = (V, E, w)$ of effective resistance diameter R, and two disjoint vertex sets $s_1, s_2, ..., s_k$ and $t_1, t_2, ..., t_\ell$, let $\mathbf{f}$ be the unique electric flow induced by injecting current α_i to s_i and extracting β_j from t_j so that $\sum_i \alpha_i = \sum_j \beta_j = 1$. Then $\mathcal{E}(\mathbf{f}) \leq R$.*

Proof. Let $\mathbf{f}_{ij}$ be a unit electric flow from s_i to t_j. Then:

$$\mathcal{E}(\mathbf{f}_{ij}) = \mathrm{Reff}(s_i, t_j) \leq R$$

Now, consider the distribution of pairs (I, J) such that $P[I = i, J = j] = \alpha_i \beta_j$. If we pick the source s_I and the sink s_J according to this distribution, and induce a current as above, we will get some flow $\mathbf{f}_{IJ}$, which in expectation will be:

$$\mathbb{E}\left[\mathbf{f}_{IJ}\right] = \sum_{i,j} \alpha_i \beta_j \mathbf{f}_{ij} = \tilde{\mathbf{f}}$$

Importantly, $\tilde{\mathbf{f}}$ is not necessarily an electric flow, since it might not obey Ohm's law. However, the energy of $\tilde{\mathbf{f}}$ bounds the energy of the true electric flow with the same external currents.

Finally, note that by definition, $\mathcal{E}(\mathbf{f}) = \sum_e \mathbf{f}(e)^2 / w_e = \mathbf{f}^T \mathbf{R} \mathbf{f}$, where $\mathbf{R}$ is the diagonal matrix of resistances. Since $\mathbf{R}$ is positive semi-definite, then $\mathcal{E} : \mathbb{R}^n \to \mathbb{R}, \mathbf{f} \mapsto \mathcal{E}(\mathbf{f})$ is convex, meaning that by Jensen's inequality:

$$\mathcal{E}(\tilde{\mathbf{f}}) \leq \mathbb{E}_{(I,J)}\left[\mathcal{E}(\mathbf{f}_{IJ})\right] = \sum_{i,j} \alpha_i \beta_j \mathcal{E}(\mathbf{f}_{ij}) \leq R \sum_{i,j} \alpha_i \beta_j = R$$

Moreover, note that the external currents of $\tilde{\mathbf{f}}$ match exactly those of $\mathbf{f}$: the current injected at s_i is $\sum_j \alpha_i \beta_j = \alpha_i$ and the current extracted from t_j is $\sum_i \alpha_i \beta_j = \beta_j$. Since the actual electrical flow minimizes the energy, then the energy of $\mathbf{f}$ is $\mathcal{E}(\mathbf{f}) \leq \mathcal{E}(\tilde{\mathbf{f}}) \leq R$. ◀

We use this lemma to show that when a graph is contracted on some subgraphs, the effective resistance diameter of the contracted graph can be combined with effective resistance diameters of contracted components to bound the diameter of the original graph.

▶ **Lemma 3.3.** *Given a graph $G = (V, E, w)$ and its partitioning $V = \sqcup_{i=1}^{h} V_i$, let the contracted graph $G' = (V', E', w')$ be constructed by identifying vertices in the same partition set. Assume $\mathrm{R}_{\mathrm{diam}}(G') \leq R_1$ and $\mathrm{R}_{\mathrm{diam}}(G[V_i]) \leq R_0$ for all i. Then $\mathrm{R}_{\mathrm{diam}}(G) \leq R_1 + hR_0$.*

Proof. Consider any $s, t \in V$. We will show that $\mathrm{Reff}(s, t) \leq R_1 + hR_0$. Note that if $s, t \in V_i$ for some i, then the effective resistance between s and t in G is bounded by the effective resistance in $G[V_i]$, meaning that $\mathrm{Reff}(s, t) \leq R_0 \leq R_1 + hR_0$. Hence, assume $s \in V_i$, $t \in V_j$ for some $i \neq j$.

Note that the effective resistance equals the minimal energy of a unit flow passing from s to t. Hence, it is sufficient to construct some unit flow (not necessarily an electric flow) $\mathbf{f}$ such that $\mathcal{E}(\mathbf{f}) \leq R_1 + hR_0$.

Let $s', t' \in V'$ be the vertices of G' that correspond to contracted components V_i, V_j in G. Consider an electric unit flow $\mathbf{f}'$ from s' to t' in G': since $\mathrm{Reff}_{G'}(s', t') \leq \mathrm{R}_{\mathrm{diam}}(G') \leq R_1$, then $\mathcal{E}_{G'}(\mathbf{f}') = \sum_{e \in E'} \mathbf{f}'(e)^2 / w'(e) \leq R_1$. It remains to lift this flow to a flow in G.

Our lifted flow $\mathbf{f}$ would match $\mathbf{f}'$ on the flows of all edges crossing between partitions: namely, for each edge $e' \in E'$ of G', if the corresponding edge in G is $e \in E$, then we would force $\mathbf{f}(e) = \mathbf{f}'(e')$. Then, consider any subgraph $G[V_i]$ of G. The currents across the edges going between partitions force particular external flows to $G[V_i]$: specifically, if we limit our view to $G[V_i]$, any vertex $v \in V_i$ has external flow equaling the sum of all flows going to v

through these crossing edges (with the only exception of vertices s and t, where the external currents are 1 and -1, respectively). These external currents force a unique electric flow $\mathbf{f}_i$ through $G[V_i]$. Since the total external current flowing in $G[V_i]$ equals the total current flowing out of $G[V_i]$ and is bounded by 1 (the total current injected in G), then by Lemma 3.2 we get that the energy of this flow is $\mathcal{E}_{G[V_i]}(\mathbf{f}_i) \leq \mathrm{R}_{\mathrm{diam}}(G[V_i]) \leq R_0$.

Hence, we get the flow $\mathbf{f}'$ fixing the flow across the edges going in between partitions, and in each subgraph $G[V_i]$ we get a flow $\mathbf{f}_i$ consistent with the currents external to $G[V_i]$. Combining these flows, we get a unit s-t flow $\mathbf{f}$ in the original graph G. Finally, since the energy of the flow only depends on the flow across each edge and edge weights, we can safely combine the eneries of each of these flows to get the energy of $\mathbf{f}$:

$$\mathcal{E}(\mathbf{f}) = \mathcal{E}(\mathbf{f}') + \sum_{i=1}^{h} \mathcal{E}(\mathbf{f}_i) \leq R_1 + hR_0$$

Finally, since the effective resistance between s and t in G equals the minimal energy of a unit flow from s to t, the above expression bounds it from above: $\mathrm{Reff}_G(s,t) \leq \mathcal{E}(\mathbf{f}) \leq R_1 + hR_0$. ◀

With this lemma, we are finally ready to prove Theorem 3.1:

Proof of Theorem 3.1. The converse is fairly straightforward to see. Consider the weighting for which the leverage scores are bounded by $1/c$ and take any cut of G that has k edges. Then sample a spanning tree of G' with the probabilities proportional to the product of the edge weights in the tree. It is known that the probability of any edge being included in the spanning tree equals its leverage score, meaning that each cut edge will be included with probability at most $1/c$. Then, by the union bound, the probability that at least one cut edge is included is at most k/c. But one edge of any cut is definitely included in any spanning tree, meaning that $k/c \geq 1$, thus giving that any cut has size k is $\geq c$.

Now let us prove the main direction. Let $\Delta \in \mathbb{R}$ be the parameter to be chosen later. We will reweight the graph using Algorithm 3.1.

Algorithm 3.1 GraphReweight.

1. Set $i = 0$ and $G_0 = G$
2. On recursive step i, find a partition of $G_i = (V_i, E_i)$ into subgraphs $V_i = \sqcup_{j=1}^{h} V_{ij}$ according to Corollary 2.17, meaning that the number of edges going between partition sets is $\leq |E_i|/2$, and effective resistance diameter of $G_i[V_{ij}]$ is $\leq \alpha \cdot |V_i|/|E_i|$ for all j (where α is a constant from Corollary 2.17).
3. For each edge e in each $G_i[V_{ij}]$, set $w(e) = 1/\Delta^i$.
4. Contract each V_i in a single vertex, obtaining a multigraph $G_{i+1} = (V_{i+1}, E_{i+1})$, where $E_{i+1} = \{(u,v) \in E_i : u \in V_{ij}, v \in V_{ik} \text{ for } j \neq k\}$
5. If $|V_{i+1}| > 1$, return to step 2 for $i+1$, otherwise output w.

Consider any step i of the algorithm and any subgraph $G_i[V_{ij}]$ of G_i that would be contracted on this step. By a slight abuse of notation, write $G[V_{ij}]$ to be the subgraph of the original graph G induced by all the vertices in V that are contracted to vertices V_{ij} on steps 0 to $i-1$.

We will prove by induction that on each step i, for each subgraph $G[V_{ij}]$ of the original graph, the effective resistance diameter (with respects to the weights w assigned) is:

$$\mathrm{R}_{\mathrm{diam}}\left(G[V_{ij}]\right) \leq 2\alpha \cdot \Delta^i \cdot \left(1 + \frac{n}{\Delta}\right)^i \cdot \frac{1}{c},$$

where α is a constant from Corollary 2.17.

Note that since the minimal cut of G is c, then each vertex has degree at least c, meaning that the total number of vertices in G is at least $c|V|/2$. Then $|V|/|E| \leq |V|/(c|V|/2) = 2/c$, so by our choice of partitioning on step 2:

$$\mathrm{R}_{\mathrm{diam}}\left(G[V_{0j}]\right) = \mathrm{R}_{\mathrm{diam}}\left(G_0[V_{0j}]\right) \leq \alpha\frac{|V|}{|E|} = 2\alpha \cdot \frac{1}{c},$$

proving the base case of induction.

Then, consider any step $i \geq 1$ and any subgraph $G_i[V_{ij}]$ of G_i. Notice that the minimal cut of G_i is at least c, since contracting on subgraphs only limits the set of all cuts to those which don't cut through contracted subgraphs, meaning that it does not decrease the min cut. Hence, $|V_i|/|E_i| \leq 2/c$, so the effective resistance diameter of a subgraph found on Step 2 of Algorithm 3.1 is $\leq 2\alpha/c$. However, note that when we set the weights of all edges of $G_i[V_{ij}]$ to $1/\Delta^i$, the effective resistance diameter given above (which is the diameter for unit weights) will be scaled by Δ^i (since the resistance of each edge is scaled by Δ^i), becoming:

$$\mathrm{R}_{\mathrm{diam}}(G_i[V_{ij}]) \leq 2\alpha \cdot \Delta^i \cdot \frac{1}{c}$$

Then, consider any vertex $v_k \in V_{ij} \subseteq V_i$. This vertex was created by contracting some component $V_{(i-1)k}$ on the previous recursive step of the algorithm, so by the inductive hypothesis we get that:

$$\mathrm{R}_{\mathrm{diam}}\left(G[V_{(i-1)k}]\right) \leq 2\alpha \cdot \Delta^{i-1} \cdot \left(1 + \frac{n}{\Delta}\right)^{i-1} \cdot \frac{1}{c}$$

Note that $G_i[V_{ij}]$ of effective resistance diameter $2\alpha\Delta^i/c$ is obtained from $G[V_{ij}]$ by contracting at most n such subgraphs $G[V_{(i-1)k}]$ of effective resistance diameter $2\alpha \cdot \Delta^{i-1} \cdot \left(1 + \frac{n}{\Delta}\right)^{i-1} \cdot \frac{1}{c}$ each, meaning that by applying Lemma 3.3 we can bound $\mathrm{R}_{\mathrm{diam}}(G[V_{ij}])$ as:

$$\mathrm{R}_{\mathrm{diam}}(G[V_{ij}]) \leq 2\alpha\Delta^i \cdot \frac{1}{c} + 2\alpha n \cdot \Delta^{i-1} \cdot \left(1 + \frac{n}{\Delta}\right)^{i-1} \cdot \frac{1}{c} \leq 2\alpha \cdot \Delta^i \cdot \left(1 + \frac{n}{\Delta}\right)^{i} \cdot \frac{1}{c},$$

which concludes our inductive proof for $\mathrm{R}_{\mathrm{diam}}$.

Finally, consider any edge that was contracted on step i: $e \in G_i[V_{ij}]$. Then, clearly the effective resistance of e in G is upper bounded by the effective resistance of e in a subgraph of G, $G[V_{ij}]$, which in turn is upper bounded by $\mathrm{R}_{\mathrm{diam}}(G[V_{ij}])$. Then, since $w(e) = 1/\Delta^i$, we get that the leverage score of e in G is upper bounded by:

$$\ell_G(e) = w(e)\,\mathrm{Reff}_G(e) \leq w(e)\,\mathrm{Reff}_{G[V_{ij}]}(e) \leq \frac{\mathrm{R}_{\mathrm{diam}}(G[V_{ij}])}{\Delta^i} \leq 2\alpha \cdot \left(1 + \frac{n}{\Delta}\right)^{i} \cdot \frac{1}{c}$$

To conclude the proof, we observe that on each recursive step $|E_i|$ is reduced by a factor of 2, meaning that the total number of steps the algorithm takes to complete is $\log m$. Then, picking $\Delta = \Theta(n \log m)$ is sufficient to ensure that $\left(1 + \frac{n}{\Delta}\right)^{i} \leq \left(1 + \frac{n}{\Delta}\right)^{\log m} \leq 2$, resulting in all leverage scores being $\leq 4\alpha/c$.

For such Δ, get that $w_{\max}/w_{\min} = \Delta^{O(\log m)} = m^{O(\log m)}$. ◀

3.2 Concluding Connectedness Under Subsampling

Finally, we show that Theorem 3.1 can be combined with Theorem 2.18 to show that a graph with a large minimum cut stays connected even when edges are subsampled in a k-wise independent manner:

▶ **Theorem 3.4.** *Given an undirected unweighted graph $G = (V, E)$ on m edges, if the minimum cut of G is $c \log m$, then sampling the edges of G in a $O(\log m)$-wise independent manner with marginals $O(1/c)$ yields a connected graph with probability $1 - 1/\text{poly}(m)$.*

Proof. Theorem 3.1 tells us that there is a weighting w of G such that for each $(a, b) \in E$, the leverage score is $w_{ab} \cdot \tilde{R}_{ab} \leq 4\alpha/c \log m$. Then, by Corollary 2.19, $O(\log m)$-wise independent sampling with marginals $p_{ab} = \Theta(w_{ab} \cdot \tilde{R}_{ab} \cdot \log n) = \Theta(1/c)$ gives a connected graph with probability $1 - 1/\text{poly}(m)$. ◀

We also have the following immediate corollary:

▶ **Corollary 3.5.** *Given an undirected unweighted graph $G = (V, E)$ on m edges, there is a choice of constant κ such that if the minimum cut of G is $\kappa \log m$, then sampling the edges of G in a $O(\log m)$-wise independent manner with marginals $1/2$ yields a connected graph with probability $1 - 1/\text{poly}(m)$.*

In the full version, we show that these results can be extended to show that by choosing the marginal probabilities correctly, we can ensure that there is a non-negligible probability that *any* near-minimum cut is the *unique* surviving cut in our sample. Additionally, we construct a different distribution that satisfies the same connectedness results while needing only $O(\log m \cdot \log \log m)$ random bits.

4 Cycle-Freeness Under Almost Bounded-Independence Sampling

In this section, we consider using (almost) bounded-independence when sampling graphs with large girth. For these graphs, we show that even this bounded-independence sampling yields subsampled graphs with *no cycles* with high probability.

4.1 Cycle-Freeness Under High Girth

To start, we recall the work of Karp, Upfal, and Wigderson [19], who showed the following claim that characterizes when cycles appear under *uniform sampling* of edges:

▷ Claim 4.1. Let $G = (V, E)$ be a graph with m edges and shortest cycle length $> 20 \log(m)$. A uniformly random subsample of E at rate $1/2$ will (with high probability):

1. Contain at least $m/4$ edges.
2. Not have any cycles.

We include the proof here for completeness:

Proof. To start, observe that because the shortest cycle is of length $> 20 \log(m)$, this means that for any pair of vertices $i, j \in V$, there can be at most one path of length $10 \log(m)$ between i, j.

Now, let us consider sampling E at rate $1/2$ to yield $\hat{E}$. If $\hat{E}$ contains a cycle, then it must be a cycle of length $> 20 \log(m)$, as E itself had no cycles of shorter length, and $\hat{E} \subseteq E$. At the same time, if $\hat{E}$ contains a cycle C of length $> 20 \log(m)$, C must contain some path of length $10 \log(m)$. Thus, in order to show that $\hat{E}$ does not contain any cycles, it suffices to show that $\hat{E}$ does not contain any paths of length $10 \log(m)$. To see this, we recall that there is at most one path of such length for each pair of $i, j \in V$, and thus there are at most

$$\binom{|V|}{2} \leq \binom{2|E|}{2} \leq 2m^2$$

such paths. Because each path survives with probability $\leq \left(\frac{1}{2}\right)^{10\log(m)}$, we see that there is no such path with probability $\geq 1 - \frac{2m^2}{m^{10}} \geq 1 - \frac{2}{m^8}$. A simple Chernoff bound yields that the number of edges sampled is $\geq m/4$ with exponentially high probability, and thus conditions (1) and (2) are satisfied with probability $\geq 1 - \frac{1}{m^7}$. ◁

Now, we show how to use δ-almost k-wise independent distributions to derandomize the above sampling procedure:

▷ Claim 4.2. Let $G = (V, E)$ be a graph with m edges and shortest cycle length $> 20\log(m)$. Consider sampling the edges of G using a $(1/m^{100})$-almost $10\log(m)$-wise independent distribution with marginals $1/2$. Then, there is a sample from the distribution which:

1. Contains at least $m/10$ edges.
2. Does not have any cycles.

Proof. As in the proof of Claim 4.1, it suffices to prove that none of the $\leq 2m^2$ paths of length $10\log(m)$ survive the sampling. For each such path, let us denote the constituent edges by S. Because our sample space is δ-almost $10\log(m)$-wise independent, we know that over a random sample $X = (x_1, \ldots x_m)$ from our space,

$$d_{\mathrm{TV}}(X(S), U(S)) \leq \frac{1}{m^{100}}.$$

In particular, this means that

$$\Pr[X(S) \neq 1^S] \geq \Pr[U(S) \neq 1^S] - \frac{1}{m^{100}} = 1 - \frac{1}{m^{10}} - \frac{1}{m^{100}} \geq 1 - \frac{2}{m^{10}},$$

which means that

$$\Pr[\text{path } S \text{ survives}] \leq \frac{2}{m^{10}}.$$

Taking a union bound over all $\leq 2m^2$ paths then yields that with probability $\geq 1 - \frac{4}{m^8}$, no path of length $10\log(m)$ survives, and thus that no cycle survives the sampling.

Now, because we use a δ-almost $10\log(m)$-wise independent distribution with marginals $1/2$, we know that $\mathbb{E}[\sum_{i=1}^m X_i] = m/2$. By a simple Markov bound, we know then that $\Pr[\sum_{i=1}^m X_i \geq m/10] \geq 0.4$. So, by a union bound, we know that with probability $\geq 0.4 - \frac{4}{m^8}$, a sample from our distribution will have at least $m/10$ edges and not have any cycles. Because this probability is > 0, this yields the stated claim. ◁

We also can extend the result above to graphs with smaller shortest cycle length provided that we decrease the marginals accordingly:

▶ **Corollary 4.3.** *Given an undirected unweighted graph $G = (V, E)$ on m edges, if the shortest cycle length of G is $\ell \leq 20\log(m)$, then sampling the edges of G using Algorithm 2.1 with $p = (1/2)^{\lceil 40\log(m)/\ell \rceil}$ and $(1/m^{200})$-almost ℓ-wise independence yields a connected graph with probability $1 - 1/\mathrm{poly}(m)$.*

Proof. We construct an auxiliary graph G', whereby we replace each edge in G with a path of length $s = \lceil 40\log(m)/\ell \rceil$ (adding $s - 1$ new vertices for each edge). Immediately, we can see that the shortest cycle in G' is of length $\ell' \geq 40\log(m)$. At the same time, the number of edges in G' is $m' = sm < m^2$, since $s \leq m$ for sufficiently large m. Hence, we get that $\ell' \geq 40\log(m) > 20\log(m')$ and $\log(m') \leq 2\log(m)$, meaning that we can apply

Claim 4.2 to G'. Namely, if one samples the edges of G' using $(1/m^{2\cdot 100})$-almost $(2\cdot 10\log m)$-wise independent random bits with marginals $1/2$, the resulting graph is connected with probability $1 - 1/\mathrm{poly}(m)$.

Now, observe that our implementation of δ-almost ℓ-wise independent sampling with marginals $(1/2)^{\lceil 40\log(m)/\ell\rceil}$ from Algorithm 2.1 exactly corresponds to the above approach. ◀

4.2 Unique Cycle Survival Under Careful Sampling Rates

In this section, we extend the argument from the previous section to also address the setting of *cycles uniquely surviving* under sampling:

▷ Claim 4.4. Let $G = (V, E)$ be a graph with shortest cycle length ℓ, and let $C \subseteq E$ denote the edges participating in a cycle of length $\in [\ell, 1.01\ell]$. Now, let G/C denote the result of contracting the graph G on the edges C. Then, the shortest cycle length in G/C is $\geq 0.2\ell$.

Proof. Note a set of edges $C' \subseteq G/C$ forms a cycle in G/C if and only if $C' \cup C$ contains a cycle different from C in G. Thus, we instead focus on cycles $\hat{C} \subseteq G$, $\hat{C} \neq C$, and lower bound $|\hat{C} \setminus C|$.

It is easy to verify that for any pair of cycles $\hat{C} \neq C$, their symmetric difference $\hat{C} \oplus C = (\hat{C} \cup C) \setminus (\hat{C} \cap C)$ also contains a cycle (if we consider the edge-induced subgraph, any vertex in the symmetric difference would have an even degree).

Assume $|\hat{C} \setminus C| < 0.2\ell$. Then, since $|\hat{C}| \geq \ell$ (ℓ is the minimum cycle length), we have that $|\hat{C} \cup C| > 0.8\ell$. Then, since $|C| \leq 1.01\ell$, $|C \setminus C'| < 1.01\ell - 0.8\ell = 0.21\ell$. But then $|\hat{C} \oplus C| = |\hat{C} \setminus C| + |C \setminus \hat{C}| < 0.2\ell + 0.21\ell < \ell$. Since $\hat{C} \oplus C$ contains a cycle, and the minimum cycle length is ℓ, this is a contradiction. Hence, $|\hat{C} \setminus C| \geq 0.2\ell$, and so the shortest cycle length in G/C is $\geq 0.2\ell$. ◁

Now, we have the following claim:

▷ Claim 4.5. Let $G = (V, E)$ be a graph with shortest cycle length $\ell \leq 20\log(m)$, and let C be an arbitrary cycle in G of length $[\ell, 1.01\ell]$. Consider sampling the edges of G using a $(1/m^{500})$-almost 2ℓ-wise independent distribution, with marginals $p = (1/2)^{\lceil 200\log(m)/\ell\rceil}$. Then, with probability > 0, the resulting sample will:

1. Sample all edges involved in the cycle C.
2. Not contain any other cycle C'.

Proof. Because C is a cycle of length $[\ell, 1.01\ell]$, $\ell \leq 20\log(m)$, and $p = (1/2)^{\lceil 200\log(m)/\ell\rceil} \geq 1/(2m^{200/\ell})$, the probability C survives sampling is at least:

$$\Pr[C \text{ survives}] \geq \left(\frac{1}{2m^{200/\ell}}\right)^{1.01\ell} - \frac{1}{m^{500}} = \frac{1}{2^{1.01\ell}\cdot m^{202}} - \frac{1}{m^{500}} \geq \frac{1}{m^{250}},$$

where the last inequality follows since $2^{1.01\ell} \leq m^{20.2}$ for $\ell \leq 20\log(m)$.

At the same time, conditioned on C surviving, the only way for another cycle C' to survive is if the contracted graph G/C contains a cycle.

Consider the probability distribution of edges in G/C conditioned on C surviving sampling. Take any event A in this distribution (event depending on the edges in $E \setminus C$) that depends on $\leq 0.2\ell$ edges. Then:

$$\Pr[A | C \text{ survives}] = \frac{\Pr[A \wedge C \text{ survives}]}{\Pr[C \text{ survives}]} = \frac{\Pr_U[A \wedge C \text{ survives}] + e_1}{\Pr_U[C \text{ survives}] + e_2},$$

where $\Pr_U$ denotes the probabilities under fully independent sampling and e_1, e_2 are error terms. Since the event of C surviving depends on $|C| \leq 1.01\ell$ edges, then both of the events above depend only on $\leq 2\ell$ edges. Our (unconditional) distribution is δ-almost 2ℓ-wise independent, meaning that $|e_1|, |e_2| \leq \delta$. Moreover, for the independent distribution, $\Pr_U[A \wedge C \text{ survives}] = \Pr_U[A] \cdot \Pr_U[C \text{ survives}]$. Then note that:

$$\left|\Pr[A|C \text{ survives}] - \Pr_U[A]\right| = \left|\frac{\Pr_U[A] \cdot \Pr_U[C \text{ survives}] + e_1}{\Pr_U[C \text{ survives}] + e_2} - \Pr_U[A]\right|$$

$$\leq \frac{|e_1| + \Pr_U[A] \cdot |e_2|}{\Pr[C \text{ survives}]} \leq \frac{2\delta}{\Pr[C \text{ survives}]}$$

Hence, since $\delta = 1/m^{500}$ and $\Pr[C \text{ survives}] \geq 1/m^{250}$, we get that the distribution of edges in G/C conditioned on C surviving is $(1/m^{200})$-almost 0.2ℓ-wise independent.

Since the shortest cycle length in G/C is at least $\ell/5$ (as per Claim 4.4), by Corollary 4.3 we know that subsampling the edges of G/C using Algorithm 2.1 with $(1/m^{200})$-almost 0.2ℓ-wise independence and marginals $(1/2)^{\lceil 40\log(m)/0.2\ell \rceil}$ yields a cycle-free graph with probability $1 - 1/\operatorname{poly}(m)$[1]. Therefore:

$$\Pr[C \text{ survives} \wedge \nexists C' : C' \text{ survives}] = \Pr[C \text{ survives}] \cdot \Pr[G/C \text{ is cycle-free } |C \text{ survives}] \geq$$

$$\geq \frac{1}{m^{250}} \cdot (1 - 1/\operatorname{poly}(m)) > 0.$$

This yields the claim. ◁

References

1 Leonard Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*, pages 75–83. IEEE Computer Society, 1978. doi:10.1109/SFCS.1978.37.

2 Vedat Levi Alev, Nima Anari, Lap Chi Lau, and Shayan Oveis Gharan. Graph clustering using effective resistance. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 41:1–41:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ITCS.2018.41.

3 Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Inf. Process. Lett.*, 88(3):107–110, November 2003. doi:10.1016/S0020-0190(03)00359-4.

4 Noga Alon and Asaf Nussboim. K-wise independent random graphs. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 813–822. IEEE, 2008. doi:10.1109/FOCS.2008.61.

5 Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786. SIAM, 2019. doi:10.1137/1.9781611975482.48.

[1] Note that the number of edges in G/C is actually $m - |C| < m$; however, by using m instead of $m - |C|$ we only decrease the marginal probabilities $(1/2)^{\lceil 40\log(m)/0.2\ell \rceil}$ and error term $\delta = 1/m^{500}$, meaning that cycle-freeness is also preserved with these parameters. We also note that for $\ell \leq 20\log(m)$, we have $0.2\ell \leq 20\log(m - 1.01\ell) \leq 20\log(m - |C|)$ for sufficiently large m, and so our invocation of Corollary 4.3 is valid.

6 Eric Balkanski, Aviad Rubinstein, and Yaron Singer. An optimal approximation for submodular maximization under a matroid constraint in the adaptive complexity model. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 66–77, 2019. doi:10.1145/3313276.3316304.

7 Nikhil Bansal, Rudi A Pendavingh, and Jorn G Van der Pol. On the number of matroids. *Combinatorica*, 35:253–277, 2015. doi:10.1007/S00493-014-3029-Z.

8 András A. Benczúr and David R. Karger. Approximating s-t minimum cuts in $\tilde{O}(n^2)$ time. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 47–55. ACM, 1996. doi:10.1145/237814.237827.

9 Joakim Blikstad, Sagnik Mukhopadhyay, Danupon Nanongkai, and Ta-Wei Tu. Fast algorithms via dynamic-oracle matroids. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1229–1242, 2023. doi:10.1145/3564246.3585219.

10 Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. In *Proceedings of the 13th International Conference on Approximation, and 14 the International Conference on Randomization, and Combinatorial Optimization: Algorithms and Techniques*, APPROX/RANDOM'10, pages 504–517, Berlin, Heidelberg, 2010. Springer-Verlag. doi:10.1007/978-3-642-15369-3_38.

11 Dean Doron, Jack Murtagh, Salil P. Vadhan, and David Zuckerman. Spectral sparsification via bounded-independence sampling. *Electron. Colloquium Comput. Complex.*, TR20, 2020. URL: https://api.semanticscholar.org/CorpusID:211506132.

12 Jack Edmonds. Matroid intersection. In *Annals of discrete Mathematics*, volume 4, pages 39–49. Elsevier, 1979.

13 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-nc. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 754–763. ACM, 2016. doi:10.1145/2897518.2897564.

14 Wai Shing Fung, Ramesh Hariharan, Nicholas J. A. Harvey, and Debmalya Panigrahi. A general framework for graph sparsification. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 71–80. ACM, 2011. doi:10.1145/1993636.1993647.

15 Harold N Gabow and Robert E Tarjan. Efficient algorithms for a family of matroid intersection problems. *Journal of Algorithms*, 5(1):80–131, 1984. doi:10.1016/0196-6774(84)90042-7.

16 Oded Goldreich, Shafi Goldwasser, and Asaf Nussboim. On the implementation of huge random objects. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '03, page 68, USA, 2003. IEEE Computer Society.

17 Per M Jensen and Bernhard Korte. Complexity of matroid property algorithms. *SIAM Journal on Computing*, 11(1):184–190, 1982. doi:10.1137/0211014.

18 David R. Karger. Global min-cuts in rnc, and other ramifications of a simple min-cut algorithm. In Vijaya Ramachandran, editor, *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas, USA*, pages 21–30. ACM/SIAM, 1993. URL: http://dl.acm.org/citation.cfm?id=313559.313605.

19 Richard M. Karp, Eli Upfal, and Avi Wigderson. The complexity of parallel computation on matroids. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 541–550. IEEE Computer Society, 1985. doi:10.1109/SFCS.1985.57.

20 Richard M. Karp, Eli Upfal, and Avi Wigderson. The complexity of parallel search. *J. Comput. Syst. Sci.*, 36(2):225–253, 1988. doi:10.1016/0022-0000(88)90027-X.

21 Sanjeev Khanna, Aaron Putterman, and Junkai Song. On the parallel complexity of finding a matroid basis. *arXiv preprint, IEEE 66th Annual Symposium on Foundations of Computer Science, FOCS 2025, to appear*, 2025. arXiv:2507.08194,.

22 Sanjeev Khanna, Aaron Putterman, and Junkai Song. Optimal parallel basis finding in graphic and related matroids. *arXiv preprint arXiv:2511.04826*, 2025. doi:10.48550/arXiv.2511.04826.

23 Sanjeev Khanna, Aaron Putterman, and Junkai Song. An $\widetilde{O}(n^{3/7})$ round parallel algorithm for matroid bases. *arXiv preprint arXiv:2605.03979*, 2026.

24 Robert Kleinberg and Seth Matthew Weinberg. Matroid prophet inequalities. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 123–136, 2012.

25 Andrew McGregor. Graph stream algorithms: a survey. *ACM SIGMOD Record*, 43(1):9–20, 2014. doi:10.1145/2627692.2627694.

26 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 213–223. ACM, 1990. doi:10.1145/100216.100244.

27 Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. doi:10.1137/080734029.

28 Daniel A. Spielman and Shang-Hua Teng. Spectral sparsification of graphs. *SIAM J. Comput.*, 40(4):981–1025, 2011. doi:10.1137/08074489X.

29 Ashok Subramanian. A polynomial bound on the number of light cycles in an undirected graph. *Inf. Process. Lett.*, 53(4):173–176, 1995. doi:10.1016/0020-0190(94)00202-A.

30 Salil P Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1-3):1–336, 2012. doi:10.1561/0400000010.

Improved Bounds on the Space Complexity of Circuit Evaluation

Yakov Shalunov
Department of Computer Science, University of Chicago, IL, USA

Abstract

Williams (STOC 2025) recently proved that time-t multitape Turing machines can be simulated using $O(\sqrt{t \log t})$ space using the Cook-Mertz (STOC 2024) tree evaluation procedure. As Williams notes, applying this result to fast algorithms for the circuit value problem implies an $O(\sqrt{s} \cdot \operatorname{polylog} s)$ space algorithm for evaluating circuits with s gates.

In this work, we provide a direct reduction from circuit value to tree evaluation without passing through Turing machines, simultaneously improving the bound to $O(\sqrt{s \log s})$ space and providing a proof with fewer layers of abstraction.

This result can be thought of as a "sibling" result to Williams' for circuit complexity instead of time; in particular, using the fact that time-t Turing machines have size $O(t \log t)$ circuits, we can recover a slightly weakened version of Williams' result, simulating time-t machines in space $O(\sqrt{t} \log t)$.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Communication complexity

Keywords and phrases circuit value problem CVP, space complexity, tree evaluation problem

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.3

Supplementary Material *Software (Formalization of key lemma in Lean)*: `https://github.com/Yakov-Shalunov/space-complexity-circuiteval-formalization`
archived at `swh:1:dir:a0330b101611eb4eccc32994641173be36fed69e`

Acknowledgements I would like to thank William Hoza for suggesting this research direction, helping review this article, and general guidance. I am also grateful to Alexander Razborov for feedback on presentation.

1 Introduction

Recently, Williams proved an extremely counterintuitive result:

▶ **Theorem 1** (Williams [15]). *All time-$t(n) \geq n$ multitape Turing machines can be simulated using $O(\sqrt{t \log t})$ bits of space.*

This holds no matter how they use the $\Omega(t)$ cells of space they can touch and improves dramatically on the 50-year-old best-known simulation using space $O(t/\log t)$ due to Hopcroft, Paul, and Valiant [9].

This simulation can be composed with Turing machine algorithms for evaluating circuits to obtain space-efficient algorithms for circuit evaluation. In particular, the best known algorithm for circuit simulation on Turing machines is due to Pippenger [11] (Theorem 7) and yields a space $O(\sqrt{s} \cdot \log^{3/2} s)$ algorithm for evaluating arbitrary size-s circuits.

Motivated by the fact that circuits themselves are an appealing model of fine-grained complexity, we directly prove an improved bound on the space complexity of circuit evaluation:

▶ **Theorem 2** (Main theorem). *Given a size s circuit C on $n \leq s$ inputs and an input $x \in \{0,1\}^n$, the output $C(x)$ can be evaluated in space $O(\sqrt{s \log s})$. Further, a logspace algorithm for the tree evaluation problem would immediately imply a space $O(\sqrt{s})$ algorithm for circuit evaluation.*

© Yakov Shalunov;
licensed under Creative Commons License CC-BY 4.0

The uniform formulation immediately implies a nonuniform version:

▶ **Corollary 3.** *If $f : \{0,1\}^n \to \{0,1\}$ can be computed by a size s circuit then f can be computed by a size-$2^{O(\sqrt{s \log s})}$ branching program.*

Additionally, Theorem 2 together with standard results for simulating Turing machines with circuits (see Theorem 9) gives $\mathsf{TIME}[t] \subseteq \mathsf{SPACE}\left[\sqrt{t}\log t\right]$, re-proving Williams' result up to log factors.

Technique outline

Similarly to Williams', our result is based on a reduction to the tree evaluation problem. Roughly speaking, the tree evaluation problem (Definition 13) is the task of evaluating a height h, d-ary tree where each leaf is a b-bit value and each internal node is a function $\{0,1\}^{db} \to \{0,1\}^b$, given as an explicit table of 2^{db} b-bit entries.

The natural depth-first approach requires space $O(hdb)$ to evaluate the tree, storing db bits at each of h recursive levels. Though this depth-first approach intuitively seems "inherent" to the problem (indeed, the problem was originally posed to attempt to separate L from ¶), Cook and Mertz came up with an $O(h \log db + db)$-space algorithm for the problem, improving the space complexity quadratically when $h \approx db$ [4].[1]

Williams' result reduces Turing machine computation to a tree evaluation instance of arity $d = O(1)$ with height $h = \Theta(t/b)$ and value-size b for some parameter b – evaluating this instance naively earns nothing, since $\Theta\left(\frac{t}{b}\right) \cdot b = \Omega(t)$, but balancing t/b and b and then applying the Cook-Mertz tree evaluation procedure yields quadratic savings.

We provide a direct reduction from circuit evaluation to tree evaluation which bypasses reasoning about the motion of Turing machine heads.

For intuition, we first sketch a warm-up result:

▶ **Proposition 4.** *Size s circuits can be evaluated in space $\tilde{O}(s^{2/3})$.*

Proof sketch. Similar to the approach used for Turing machines, we partition our circuit into blocks of size b. This, in particular, ensures that the *number* of blocks (and thus depth) is $\frac{s}{b}$. Our "blocks" will be intervals of consecutive vertices under a topological ordering, thus ensuring that the quotient graph[2] is a DAG.

We can view this quotient DAG as a "circuit" over values from $\{0,1\}^b$ instead of over individual bits: the value at each vertex in the quotient graph is the list of values of all b gates in the corresponding block of the original circuit and each wire carries all b of those bits.

Since the quotient DAG connects blocks together when any pair of underlying gates have a connection, all gates in the block have sufficient information to be evaluated and our "quotient circuit" is able to simulate the original circuit.

Now we observe that in much the same way that one can expand a circuit into a formula, we can expand this quotient circuit into a tree evaluation instance (i.e., duplicating each vertex once for every path to reach it from the root). This increases the size exponentially (which is fine, because we'll never write down the whole tree, instead computing it on-demand) but, crucially, does not affect the depth.

---

[1] Several years prior to this $O(\log n \cdot \log\log n)$ result, Cook and Mertz first showed that the problem can be solved in space $O(\log^2 n/\log\log n)$ [2, 3]. (Here, $n = \tilde{\Theta}(d^h \cdot 2^{db})$ so $\log n = \Theta(h \log d + db)$.)

[2] Here, when we say "quotient" we mean a graph with a vertex for each part of the partition and a single edge between any two parts whose components have any edges in the original graph. A more formal definition can be found in the preliminaries in Section 2.

Since the depth of the original quotient circuit was bounded by the number of blocks in it, which in turn was $\frac{s}{b}$, the height of our tree evaluation instance is $h = \frac{s}{b}$, and since each block has b gates, the value-size parameter is exactly b.

In order to get $\tilde{O}(\sqrt{s})$, we ultimately need an arity of $d = O(1)$; unfortunately the naive construction given here only allows a trivial bound on the in-degree: given a block B, each wire coming into B (i.e., edges (u, v) with $v \in B$) in the original circuit connects to one vertex and thus adds at most one block (the block $B(u)$ containing u) to the list of blocks with wires to B. Since there are b gates and each has two inputs (we work with binary circuits), there are at most $2b$ total incoming wires, and thus at most $2b$ connected blocks.

This gives us $d \approx b$ and so the Cook-Mertz procedure gives us space $O(\frac{s}{b} \log b^2 + b^2)$. Setting $b \approx \sqrt[3]{s}$ gives $\tilde{O}(s^{2/3})$. ◀

In order to get $\tilde{O}(\sqrt{s})$, we will need the key technical contribution of this work: a trick for reducing the in-degree of the quotient graph, expressed (in slightly simplified form[3]) in the following graph-theoretic lemma:

▶ **Lemma 5** (Low-degree DAG partition (simplified)). *For any directed acyclic graph G of size s and maximum in-degree $d > 1$ and any integer choice of $b \in [d, s]$, there is a subdivision[4] G' of G and a partition P of G' such that:*

- *Every part in P has at most b vertices.*
- *Every vertex in the quotient graph G'/P has in-degree at most 2.*
- *All directed paths in G'/P have length less than $d \cdot \frac{s}{b}$.*

Since we consider binary circuits, our application will have $d = 2$, and $b = \tilde{\Theta}(\sqrt{s})$, yielding a depth of $2 \cdot \frac{s}{b} - 1 = \tilde{O}(\sqrt{s})$ and giving us the desired result.

Intuitively, we are "buying" a lower depth by "paying" with increased block size – were we doing a naive depth-first evaluation, this would get us nothing, but using Cook-Mertz tree evaluation procedure allows us to have increased block size "for free" (up to the point where it exceeds depth).

▶ Remark 6. When we reduce the depth of the graph in the proof of the lemma, we do so by "brute-force," of a sort: we first create a quotient graph on the original graph G which has a total *size* of $d \cdot \frac{s}{b}$ (thereby bounding the depth) and then use careful subdivision to reduce the in-degree without increasing the depth. This allows us to reduce the depth of an *arbitrary* graph without care for its structure since the depth of a graph cannot exceed its size.

Outline

In Section 2, we give some preliminaries on the circuit and tree evaluation problems. In Section 3, we prove the Partition Lemma (Lemma 5). Finally, in Section 4, we apply the Partition Lemma to reduce circuit evaluation to an appropriate TreeEval instance.

[3] The full version is stated as Lemma 15; it provides additional complexity guarantees and allows for trading off between the final in-degree and depth. (In particular, a final in-degree $d + 1$ allows a depth of s/b.)

[4] A subdivision of a graph subdivides some edges by inserting vertices into them (i.e., replacing $e = (u, v)$ with a vertex v_e and edges $(u, v_e), (v_e, v)$). In the context of circuits specifically, these will be additional identity gates spliced into some wires, which trivially does not affect behavior.

2 Preliminaries

Model relationships. First, the following pair of results together illustrate the close connection between circuits and Turing machine time:

▶ **Theorem 7** (Pippenger [11]). *Given a size s circuit C and an input x, the output $C(x)$ can be evaluated in multitape Turing machine time $O(s \log^2 s)$.*

▶ Remark 8. Pippenger's proof of this result does not appear to be available online; correspondingly, we have included an exposition of the algorithm in appendix A. See also Williams' discussion of the result [15].

▶ **Theorem 9** (Pippenger, Fischer [12]; Koucký, cf [1]). *For every $t(n) \geq n$ and $L \in \mathsf{TIME}[t]$, the* log*-space-uniform circuit complexity of L is $O(t \log t)$.*

▶ Remark 10. While this result is originally due to Pippenger and Fischer (proven by passing through an oblivious Turing machine simulation), Koucký developed a simpler direct proof [1]; while Koucký's original proof does not appear to be digitally available, an exposition of this result by Viola can be found in Theorem 1.6 of [14].

Additionally, we follow Williams and remark that unlike for time complexity (where best-known simulations incur polylogarithmic or even quadratic overhead) the case of space complexity is much cleaner: all standard models (e.g., single tape, multitape, and random access Turing machines as well as register-based random access machines) can be mutually simulated with only constant-factor space overhead [6, 13, 15]. Correspondingly, we do not worry about the exact computational model underlying our algorithms.

As is typical for sublinear space algorithms, we assume a read-only input tape and write-only output tape are provided to the machine in addition to its bounded workspace.

Circuits. Formulations of the circuit evaluation/circuit value problem differ. However, all that we are aware of reduce trivially to the FullCircuitEval formulation below. The following lemma justifies working with the simplified CircuitEval.

▶ **Definition 11** (Circuit Evaluation). *FullCircuitEval instances take the form (x, C) where x is an m-bit string and C is an s-gate circuit in the full binary basis on m inputs, provided as a list of triples of the form (ℓ_i, r_i, φ_i) where $\varphi_i : \{0,1\}^2 \to \{0,1\}$ is a binary function and each of ℓ_i and r_i is either a gate index $j \neq i$ or a variable index k for some $k \leq m$.*

The objective is to output the value v_i of each gate. For simplicity, assume $s \geq m$.

A CircuitEval instance C is an s-gate circuit in the full binary basis provided as a list of triples of the form (ℓ_i, r_i, φ_i) where $\varphi_i : \{0,1\}^2 \to \{0,1\}$ is a binary function and each of ℓ_i and r_i are either a gate index $j < i$ (i.e., the input is in topologically sorted order) or a constant bit.

The objective is to output the value of the last gate.

▶ **Lemma 12** (CircuitEval is good enough). *A space S algorithm for CircuitEval on circuits of size s implies a space $S + \log^2 s$ algorithm for FullCircuitEval.*

Proof. First, observe that given an instance of FullCircuitEval, we can replace every reference to variable k with the value x_k in space $\log s$ by simply scanning over the gates of C and for each one, if ℓ_i is a gate index, outputting it as is, and if it is a variable index k then storing the current gate in $\log s$ bits and the variable index in $\log m \leq \log s$ bits and scanning back to find x_k, outputting that value instead of k. Similarly for r_i.

Next, observe that it is possible to topologically sort a graph in $O(\log^2 s)$ space [5].

Finally, suppose M computes CircuitEval in space S. We use space-efficient composition and the above procedures to evaluate the values v_i bit-by-bit: run M on gates $\{0, \ldots, i\}$ for each i, sorting with respect to the sink i, then write v_i to the output tape. Between computations we only need to store which gate we are on using $\log s$ bits and everything else can be reused. ◀

Graphs. When we refer to a quotient graph with respect to a partition, we refer to a graph which collapses together all vertices inside a part of the partition and then removes all duplicate edges that result. That is, if $G = (V, E)$ is a graph and $P : V \to [k]$ is a partition into k parts, we say that for $i, j \in [k]$, parts i, j are connected in G/P if there exist $u, v \in V$ such that $P(u) = i$, $P(v) = j$, and $(u, v) \in E$. There is at most one edge between any pair of parts and we will discard self-loops. We will also use the term "block" to refer to parts of the partition.

Given a graph $G = (V, E)$, an "edge subdivision at e" is the operation of taking an edge $e = (u, v) \in E$ and removing it, adding in its place a vertex v_e and edges (u, v_e) and (v_e, v). The subdivided graph then has $V' = V \cup \{v_e\}$ and $E' = E \cup \{(u, v_e), (v_e, v)\} \setminus \{e\}$. A subdivision G' of G is any graph created by 0 or more edge subdivisions.

For the purposes of the complexity of operations, we restrict the representations of objects: partitions are specified as an explicit function mapping vertices to the labels of their part and the set of labels of vertices in a subdivided graph is a superset of the labels in the original graph. This allows us to maintain the gate information of the circuit and to identify all subdivision vertices to assign them identity gates.

Tree evaluation. Formally, the tree evaluation problem is defined as follows:

▶ **Definition 13** (Tree Evaluation). *TreeEval instances are full d-ary trees of height h where each leaf $\ell \in \{0, \ldots, d-1\}^h$ is labeled with a b-bit value v_ℓ and each internal node $u \in \{0, \ldots, d-1\}^{<h}$ is labeled with an explicit function (provided as a table of values) $f_u : \{0,1\}^{db} \to \{0,1\}^b$.*

We recursively define the value v_u at each internal node u in the natural way to be

$$v_u := f_u(v_{u0}, \ldots, v_{u(d-1)}).$$

The objective is to evaluate $v_{\text{root}} \in \{0,1\}^b$, the value of the root.

(For the history and broader significance of the tree evaluation problem, see Cook and Mertz work [4]. We use the formulation of the problem exposited by Goldreich [7].)

▶ **Theorem 14** (Cook-Mertz [4, 7]). *TreeEval instances of height h, arity d, and value-size b can be evaluated in space $O(h \log db + db)$.*

3 Graph partitioning

First, let us state the lemma in its general form:

▶ **Lemma 15** (Low-degree DAG partition). *For any directed acyclic graph G of size s and maximum in-degree $d > 1$, for any integer choice of parameters $d' \in [2, d+1]$ and $b \in [d/(d'-1), s]$, there is a subdivision G' of G and a partition P of G' such that:*

- *Every part in P has size at most b.*
- *Every vertex in the quotient graph G'/P has in-degree at most d'.*

- *G'/P is a layered DAG with layer count at most $\frac{d}{d'-1} \cdot \frac{s}{b}$.*
- *Given G in topologically sorted order, d', and b, the subdivision G' and partition P can be computed in $O(\log s)$ space.*

▶ Remark 16. By allowing an additional "vertex subdivision"[5] operation, one can obtain a slightly stronger version of the lemma, obtaining optimal depth s/b when $d = d' = 2$. However, the factor-of-2 savings on depth do not change the bottom line result (Theorem 2), the construction is somewhat more convoluted, and it is not immediately clear how it generalizes to arbitrary d and d'.

We also believe it should be possible to improve the $\frac{d}{d'-1}$ factor to either $\frac{d-1}{d'-1}$ or $\frac{d}{d'}$ without the introduction of a new operation, but this seems to significantly increase the complexity of the proof. This constant-factor improvement would similarly not change the bottom line result.

▶ Remark 17. Theorem 37 in Mertz's thesis [10] provides a way to control arity of TreeEval instances directly. Discussion of the relevant differences appears after the following proof.

We will first describe the construction, then prove that it has the necessary properties. As in the statement of the lemma, let G be a size-s DAG of max in-degree d and let $d' \in [2, d+1]$, $b \in [d/(d'-1), s]$ be parameters.

Symbols and indices. Throughout the proof: s refers to the size of the input graph G; d is the max in-degree of G; d' is the target in-degree of the quotient graph; b is the maximum size of blocks and the value-size in the tree evaluation instance; b_0 refers to the size of the initial blocks; i and j refer to indices of blocks or directly comparable values; ℓ is used to index incoming edges in the quotient graph; finally t refers to the index of the final initial block and is the depth of the graph.

3.1 Construction

Initial partition. Though ultimately we produce a subdivision and a partition of that subdivision, we start with a partition of the original graph. We will use this initial partition to describe the subdivision and ultimate partition. In particular, the mappings of the vertices in G in the final partition will be the same as the initial partition: we will simultaneously add subdivision vertices and blocks to contain them.

While blocks in the ultimate partition are allowed size up to b, the initial blocks have to be somewhat smaller: specifically, they will be size $b_0 = \left\lfloor \frac{d'-1}{d} \cdot b \right\rfloor$. Let $V = (v_0, \ldots, v_{s-1})$ be a topological ordering of G and let $t = \lceil s/b_0 \rceil - 1$. Then for $i < t$, we define

$$B_i := (v_{ib_0}, \ldots, v_{(i+1)b_0-1})$$

where for the final block B_t we truncate to $v_{tb_0}, \ldots, v_{s-1}$. In the notation of a mapping of vertices to parts, this is the mapping $V = \{0, 1, \ldots, s-1\} \to \{0, 1, \ldots, t\}$ given by $k \mapsto \lfloor k/b_0 \rfloor$. Note that the resulting quotient graph is a DAG and the ordering B_i is a topological sort since if $i < j$, $v_k \in B_i$ and $v_{k'} \in B_j$ then $k < k'$. Thus, there is no edge (k', k) and so there can be no edge (j, i).[6]

[5] That is, replacing a v with two vertices v_{in} and v_{out} which are connected by an edge $(v_{\text{in}}, v_{\text{out}})$ and replacing all edges (u, v) with edges (u, v_{in}) and edges (v, w) with (v_{out}, w). In the context of circuits, v_{in} is the original gate and v_{out} is an identity gate.

[6] Note that *any* partition P of G such that G/P is a DAG must have the property that each block in P represents an interval in *some* topological sort.

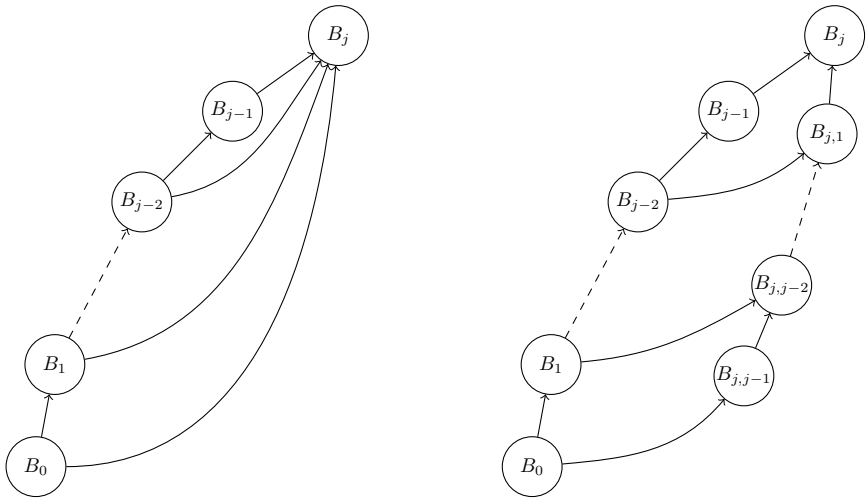


Figure 1 Addition of "cable" blocks to reduce in-degree of block B_j in the $d' = 2$ case. In the full construction, the other B_i blocks have their own cables, which are omitted here for clarity.

As in the sketch, we have successfully created a quotient DAG where the maximum depth is $t+1 = \frac{s}{b_0} = \frac{d}{d'-1} \cdot \frac{s}{b}$ and all parts have size at most b. Now we must address the key issue: the in-degree. The source vertices of those $b \cdot (d'-1)$ edges have no reason to be confined to d' blocks.[7]

Observe that we have chosen b_0 such that each block has at most $b_0 \cdot d = b \cdot (d'-1)$ incoming edges in the original graph. Intuitively, in order to manage the in-degree of the quotient graph, we will subdivide incoming edges and group the newly created vertices into $d'-1$ blocks which funnel them into the block (with the final d'th block being the immediately preceding block).

Cables. In order to resolve this, we will introduce "cables" to "gradually gather together" the incoming edges into a small number of blocks. For every edge $e = (u, v)$ with $u \in B_i$ and $v \in B_j$ for $i < j$, we will subdivide e $j - i - 1$ times, creating vertices $v_{e,k}$ for $1 \le k < j - i$. This gives us our graph G'.

Now we need to partition the newly-created vertices. For simplicity, let us first consider the case where $d' = 2$. To get the desired result: for every edge e starting outside B_j and terminating in B_j, we will add the vertex $v_{e,k}$ (if it exists) to block called $B_{j,k}$ for $k < j$. Since, in this case, there are at most $db_0 = b \cdot (d'-1) = b$ edges e terminating in B_j, the size of each block $B_{j,k}$ is at most b.

As suggested by the notation $B_{j,k}$, the blocks $B_{j,1}$ through $B_{j,j-1}$ form a sort of "tail" or "cable" connected to B_j; they run along the preceding blocks of the graph, connecting to each one at the appropriate point in the cable (specifically, block B_i is connected to the cable at block $B_{j,j-i-1}$). The cable carries all the edges terminating in block B_j from earlier blocks (see Figure 1).

[7] In fact, it is possible to construct graphs where this procedure will yield max in-degree of $\min(d' \cdot b, t-1)$ (i.e., every incoming edge leads to a different block, as long as enough blocks exist for that). As an explicit example, consider $d = 2, d' = 3$, and $s = b^2$, with edges $(k, k+1)$ for all $0 \le k < s-1$ and edges $(k, tb + k/b)$ for $k < t$ where $b|k$. That is, a graph where a "spine" of edges enforces a specific topological ordering and then the first vertex of each block B_i connects to vertex i in block B_t. Then B_t has in-degree $t - 1 \gg d'$.

Now we consider the case where $d' > 2$. The "bandwidth" of the cable is b, and that becomes insufficient when there are up to $(d'-1)b$ incoming edges. We solve this bandwidth problem in a very natural way: add more cables. Formally: in the case where $d' > 2$, we will instead have blocks $B^{\ell}_{j,k}$ where $k < j$ indexes the "layer" as before and $1 \leq \ell < d'$ indexes the cable. We can enumerate the edges $\{e_1, \ldots, e_{b\cdot(d'-1)}\}$ coming into B_j. Then $v_{e_r,k}$ will be placed into block $B^{\lfloor r/b \rfloor}_{j,k}$.

For ease of computation, we will define this enumeration of the edges to be indexed by first the index of the terminal vertex and then by which of $\leq d$ incoming edges it corresponds to (sorted by whatever order they appear in in the input representation). We will allow indices to be skipped: for example, e_d will be reserved for the dth incoming edge of vertex v_{jb}, even if the in-degree of v_{jb} is strictly less than d and that edge does not exist. This completes the description of the partition P of G'.

3.2 Analysis

Proof of Lemma 15. Recall that we want a subdivision G' and a partition P such that:

- Every part in P has size at most b.
- Every vertex in the quotient graph G'/P has in-degree at most d'.
- G'/P is a strictly layered DAG with depth (i.e., layer count) at most $\frac{d}{d'-1} \cdot \frac{s}{b}$.
- Given G in topologically sorted order, d', and b, the subdivision G' and partition P can be computed in $O(\log s)$ space.

We show that the above construction satisfies these properties.

Size The first condition is trivially satisfied for the initial blocks which have size $b_0 = \frac{d'-1}{d} b \leq b$. It is satisfied for the cable blocks since each cable block gets at most b gates assigned to it because the map $[b \cdot (d'-1)] \to [d'-1]$ given by $r \mapsto \lfloor r/b \rfloor$ is b-to-1.

In-degree The second condition is satisfied for the cable blocks since each $B^{\ell}_{j,k}$ has incoming edges only from $B^{\ell}_{j,k+1}$ and B_{j-k-1} and is satisfied by construction for the initial blocks since B_j has incoming edges only from $B^{\ell}_{j,1}$ for $\ell \in [d'-1]$ and from B_{j-1}.

Layering The third condition can be seen similarly to the second: there are $t+1$ layers $L_0, \ldots, L_t$ where B_j belongs to L_j and $B^{\ell}_{j,k}$ belongs to L_{j-k}. Since each block $B_j \in L_j$ has incoming edges from only $B^{\ell}_{j,1} \in L_{j-1}$ and $B_{j-1} \in L_{j-1}$ and each block $B^{\ell}_{j,k} \in L_{j-k}$ connects only to $B_{j-k-1} \in L_{j-k-1}$ and $B^{\ell}_{j,k+1} \in L_{j-k-1}$. Thus, blocks in L_i connect only to blocks in L_{i+1}, and so the partition $\{L_0, \ldots, L_t\}$ represents a "strict layering." In particular, this trivially implies that the maximum directed path length is at most $t = \frac{s}{b_0} - 1 = \frac{d}{d'-1} \cdot \frac{s}{b} - 1$.

Complexity First, note that the parameter b_0 can be computed from b, d, and d'. Since d is not given, it needs to be computed from the graph G but this can be done by iterating over all vertices and keeping a running max of their in-degrees (which in turn can be computed by iterating over all other vertices and counting how many have an edge to the given vertex).

The computation of the subdivision is arithmetic: for each edge $e = (v_m, v_n)$ we compute $i = \lfloor m/b_0 \rfloor$ and $j = \lfloor n/b_0 \rfloor$ and then subdivide that edge $j - i - 1$ times. Since this is all arithmetic on indices, it is logspace computable.

The computation of the partition is similarly efficient: v_m is assigned to $B_{\lfloor m/b_0 \rfloor}$ and $v_{e,k}$ for $e = (v_m, v_n)$ is assigned to $B^{\ell}_{j,k}$ where $\ell = \lfloor r/b \rfloor$ and $r = d \cdot (n \mod b_0) + p$ where p is the index of e among the incoming edges of v_n (which is logspace computable since we define the ordering on the edges for a given sink vertex to be whatever order they appear in in the input representation).

When outputting the subdivision, we can trivially output all vertices of the original graph first followed by the subdivision vertices, satisfying the condition that labels of vertices be preserved. ◀

▶ Remark 18. Mertz's result [10] allows for controlling the arity of a TreeEval instance by separating layers in the instance. One might hope to stop at G/P, expand this to a TreeEval instance, and then do something similar. However, because the nodes of a TreeEval instance are arbitrary functions, this approach to decreasing the arity requires increasing the value-size proportionally to the decrease in arity – that is, the product of value-size and arity which governs the space usage of the Cook-Mertz algorithm remains unchanged. (Working with arbitrary TreeEval instances also requires a logarithmic increase in depth to decrease the arity.)

In our use case of $d = d' = 2$, G/P would become a TreeEval instance of height $\Theta(\frac{s}{b})$, value-size b, and arity $\Theta(b)$. Attempting to reduce the arity to $d = 2$ at this point would require increasing the value-size to $\Omega(b^2)$ and the height to $\Omega(\frac{s}{b} \cdot \log b)$.

By decreasing the in-degree before we "forget" that the TreeEval instance came from the quotient of a DAG with small in-degree, we are able to leverage the additional structure to reduce arity from $\Theta(b)$ to 2 without significant penalty in value-size or height.

4 Circuit evaluation reduction

Our reduction to tree evaluation can be stated as follows:

▶ **Theorem 19.** *Given a topologically sorted circuit of size s and any $b \in [2, s]$, we can produce an equivalent TreeEval instance*[8] *with arity $d = 2$, height $h = O(\frac{s}{b})$ and word-size b as chosen. Further, this reduction is locally computable in space $O(b + \log s)$.*

That is, there is a space $O(b + \log s)$ procedure which, on input (C, b, u, x, y) (where C is a circuit, $b \in [2, s]$, $u \in \{0,1\}^{\leq h}$ specifies a node in the tree, $x, y \in \{0,1\}^b$) computes $f_u(x, y)$ where f_u is from the TreeEval instance equivalent to C.

In particular, choosing $b = \sqrt{s \log s}$ and applying Theorem 14 immediately gives the first half of Theorem 2; further, a space $O(h + b)$ procedure for TreeEval would immediately imply a space $O(\sqrt{s})$ procedure for CircuitEval by choosing $b = \sqrt{s}$, giving the second half of Theorem 2.

▶ Remark 20. While Stoeckl showed that TreeEval can be solved in space $o(\log n \cdot \log \log n)$ (Theorem 21), the specific parameters do not give us a better result than $\sqrt{s \log s}$. To improve upon it, we must have $h + b = o(\sqrt{s \log s})$. Note that $h + b = \Omega(\sqrt{s})$ unconditionally. Then:

$h + b = O(\sqrt{s \log s})$ implies $b = \Omega\left(\sqrt{s \log^{-1} s}\right)$ and so $\log b = \Omega(\log s - \log \log s) = \Omega(\log s)$. Similarly, we must have $b = O(\sqrt{s \log s})$ so $\log b = O(\log s)$. Thus,

$$(h + b) \cdot \frac{\log b}{\log \log b} = \Omega(\sqrt{s}) \cdot \frac{\Omega(\log s)}{O(\log \log s)} = \Omega(\sqrt{s \log s}).$$

▶ **Theorem 21** (Stoeckl, cf. [8])**.** *There exists an algorithm for evaluating binary TreeEval instances of height h and value-size b which runs in space $\Theta\left((h + b) \cdot \frac{\log b}{\log \log b}\right)$.*

[8] Technically, the TreeEval instance outputs b bits while the circuit outputs 1 bit; we actually produce a TreeEval instance where the output of the circuit is bit $s \bmod b_0$ of the output for an efficiently computable b_0.

Proof of Theorem 19. At a high level, we apply the DAG Partition Lemma (vertices introduced by subdivision become identity gates) to C and expand C'/P into a tree, letting the function $f_B(x, y)$ computed at a node $B \in C'/P$ be simply the subcircuit B evaluated with the external input wires taken from x, y. Note that B may be smaller than size b (they can even be size 0) and may have fewer than $2b$ external input wires. We can embed each part into a value by saying that the values of gates fill the bits in the $\{0,1\}^b$ values in topologically sorted order.

To ensure that we get a full binary tree, we will always make nominal connections $B_{j-1} \to B_j$, $B_{j,1} \to B_j$, and $B_{j,i} \to B_{j,i-1}$. If there are no wires between any of those pairs of blocks in the original circuit then f_B will simply ignore those inputs. To ensure the blocks $B_{j,j-1}$ are binary, we will add empty $B_{j,j}$ blocks.

Further, the part of computing $f_B(x, y)$ corresponding to evaluating the subcircuit B can be done by naively evaluating each gate in topological order from previous ones and writing it down, yielding the $O(b)$ component of the space.

It turns out that there are no pitfalls, and the above description simply works. Nonetheless, it is stated somewhat more rigorously below for completeness.

Proof. First, note that we can efficiently identify the children of a given block (e.g., by iterating over all blocks and checking whether they are connected, which can in turn be done by iterating over all pairs of vertices in the two blocks being considered). We can identify the root block by finding which block the vertex v_{s-1} maps to. We will consider the "left" child of a block to be the child block whose label is lexicographically first and the "right" child to be the one whose label is lexicographically second.

Given as input a circuit C in the full binary basis, the parameter b, and (u, x, y), using space-efficient composition, we:

1. Apply the DAG Partition Lemma with b as given and $d' = 2$. Initialize $B \leftarrow \mathsf{root}$.
2. For each bit u_i of u, identify the left or right child of B based on whether $u_i = 0, 1$ and update B to that child.
3. Once we have found the block B corresponding to the tree node u, we must evaluate it. If it is a leaf block, it must have no inputs, so we compute it by naive dynamic programming.[9]

   Otherwise, since we defined values to fill the b-bit values in topologically sorted order, for each input $v \in B_c$ (for $c \in \{x, y\}$) to a gate $w \in B$, we iterate over G' and count how many gates we find in B_c before reaching v (call it ℓ); we then take bit c_ℓ as the corresponding input to w. This allows us to locally compute the circuit B with external inputs substituted in.
4. Having computed (or rather, expressed a procedure by which we can space-efficiently compute bits of) the circuit B and which bit of x, y each input to B corresponds to, we compute the values of B via naive dynamic programming.

In order to extract the value of the circuit from the reduction, we then ultimately take bit $s \mod b_0$ of v_{root}, since that is the location of the last gate in the final block.

The correctness of the reduction is straightforward: the behavior of the circuit C' is the same as C since we just inserted some identity gates into wires. Further, the value of a block $B \in C'/P$ can be computed using the values of all the gates some gate in B takes input from, which by construction are just the 2 children of B.

[9] The space complexity of the reduction could be improved to $O(\sqrt{b \log b} + \log s)$ by recursing, but this would not improve the bottom line space complexity of circuit evaluation since the space is already dominated by the evaluation of the tree evaluation instance.

For complexity, observe that on-demand computing bits of G' and P as needed takes space $O(\log s)$; step 2 requires holding a label B and index i into u, which also takes space $O(\log s)$; iterating over G' in step 3 requires a single log-sized counter and counting gates in B_c similarly requires logspace since we just compute for each gate which block it belongs to (which we are given explicitly) and maintain a counter of how many have been in B_c; step 4 is done via naive dynamic programming, which requires space $O(b)$ to evaluate a size-b circuit.

Finally, observe that using space-efficient composition routines, we can then compose this with the Cook-Mertz algorithm using $O(b+h)$ bits of overhead. Naively: we can iterate over all (u, x, y) to print the whole tree, which is $O(2^h \cdot 2^{2b})$ in size, and then use standard space-efficient composition techniques which use log-sized indices of size $O(b+h)$ to keep track of head locations on synthetic read/write tapes.

We can do much better[10] but any TreeEval algorithm will necessarily use at least $\Omega(b+h)$ space to index its input, so this is good enough. ◀

References

1. Markus Bläser, Valentine Kabanets, Ronen Shaltiel, and Jacobo Torán. Algebraic and Analytic Methods in Computational Complexity (Dagstuhl Seminar 22371). *Dagstuhl Reports*, 12(9):41–59, 2023. doi:10.4230/DagRep.12.9.41.
2. James Cook and Ian Mertz. Catalytic approaches to the tree evaluation problem. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 752–760, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384316.
3. James Cook and Ian Mertz. Encodings and the tree evaluation problem. *Electron. Colloquium Comput. Complex.*, TR21, 2021. URL: https://eccc.weizmann.ac.il/report/2021/054/.
4. James Cook and Ian Mertz. Tree evaluation is in space $O(\log n \cdot \log \log n)$. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1268–1278, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649664.
5. Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64(1):2–22, 1985. International Conference on Foundations of Computation Theory. doi:10.1016/S0019-9958(85)80041-3.
6. Peter van EMDE BOAS. Chapter 1 - machine models and simulations. In JAN VAN LEEUWEN, editor, *Algorithms and Complexity*, Handbook of Theoretical Computer Science, pages 1–66. Elsevier, Amsterdam, 1990. doi:10.1016/B978-0-444-88071-0.50006-0.
7. Oded Goldreich. On the Cook-Mertz tree evaluation procedure. *Electron. Colloquium Comput. Complex.*, TR24, 2024. URL: https://api.semanticscholar.org/CorpusID:271770168.
8. Oded Goldreich. *Computational Complexity and Local Algorithms : On the Interplay Between Randomness and Computation*, chapter Solving Tree Evaluation in $o(\log n \cdot \log \log n)$ Space, pages 113–118. Springer Nature Switzerland, Cham, 2025. doi:10.1007/978-3-031-88946-2_7.
9. John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. *J. ACM*, 24(2):332–337, 1977. doi:10.1145/322003.322015.
10. Ian Mertz. *The Complexity of Composition: New Approaches to Depth and Space*. PhD thesis, University of Toronto, 2022.
11. Nicholas Pippenger. *Fast simulation of combinational logic networks by machines without random-access storage*. IBM Thomas J. Watson Research Division, 1977.

[10] This algorithm already makes local computations and so we simply need to convert a given tape index i into u, x, y via arithmetic. We can further improve it by observing that the Cook-Mertz algorithm already implicitly divides its seeks into u, x, y.

12 Nicholas Pippenger and Michael J. Fischer. Relations among complexity measures. *J. ACM*, 26(2):361–381, April 1979. doi:10.1145/322123.322138.

13 Cees Slot and Peter van Emde Boas. The problem of space invariance for sequential machines. *Information and Computation*, 77(2):93–122, 1988. doi:10.1016/0890-5401(88)90052-1.

14 Manu Viola. Mathematics of the impossible. https://web.archive.org/web/20250716031147/https://www.ccs.neu.edu/home/viola/papers/moti.pdf, May 2025. [Online; accessed 2025-07-16].

15 R. Ryan Williams. Simulating time with square-root space. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 13–23, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3717823.3718225.

A Fast circuit evaluation

Because it is not digitally available, we provide an exposition of Pippenger's algorithm below. We make no claims of originality. Note that this exposition is not intended to be rigorous so much as explanatory, so we have omitted a proof of correctness or runtime analysis.

▶ **Theorem 22** (Pippenger [11]). *Given a size s circuit C and an input x, the output $C(x)$ can be evaluated in multitape Turing machine time $O(s \log^2 s)$.*

We will describe the high-level recursive algorithm; in order to implement it on a multitape Turing machine, one simply needs to create a tape for each "local variable" and treat this tape as a stack, pushing the new value of each local variable to its stack when recursing. By, for example, using a marker to separate layers of the stack, any "well-behaved" recursive algorithm (i.e., one which does not try to read back up into the caller's stack frames) can be expressed using a constant number of tapes, since the "current level" of each stack can function as an arbitrary (one-sided) Turing machine tape.

We will work with lists of "records," which will be various tuples of indices (i.e., all entries in the list have the same size which is $O(\log s) = O(\log(\text{input length}))$). We will use the following high-level operations:

Classify Given a list of n records and a linear-time predicate, "classify" the records into two lists based on whether they satisfy the predicate. On a multitape Turing machine, we can perform this operation in time $O(n \log s)$ trivially if the source list and two destination lists are all on separate tapes.

Merge Given a linear-time comparison operation and two lists of lengths n and m records sorted according to the comparator, "merge" the two lists into one sorted list of length $n + m$. On a multitape Turing machine, we can perform this operation in time $O((n + m) \log s)$ if the source lists and destination list are all on different tapes.

Sort Given a list of n records and a linear-time comparator, sort the list of n records according to the comparator. On a multitape Turing machine, this can be done recursively with a constant number of auxiliary tapes in time $O(n \log n \log s)$.

Much like in the FullCircuitEval problem, we will assume that the input is a pair (x, C) of input and circuit, with the circuit encoded as a list of s records. However, we will assume that C is provided in topologically sorted order. We will adopt the convention that $0 < 1 < x_1 < \cdots < x_n < v_1 < \cdots < v_s$. Note that unlike in the space-bounded computation case, we cannot simply assume that the caller only wants the last bit, since rerunning the algorithm for each prefix of the circuit would make the running time quadratic. Correspondingly, the algorithm outputs the value of all gates.

The first step is to eliminate the input x and bake it into the circuit as constants: first, we copy x and C to separate tapes. In the process of copying C, we will replace each record (ℓ_i, r_i, φ_i) with the record $(\ell_i, r_i, \varphi_i, i)$ (which we can do by keeping the counter i on an auxiliary tape).

Then we sort C by the field ℓ. Then scan C and x in parallel, advancing the x head whenever ℓ increases, substituting the appropriate constants for the left inputs of each gate which takes a variable input. We can then repeat sorting by field r to eliminate the rest of the inputs and, finally, sort C by the index field to restore the original order.

At this point, we can copy C back to the original input tape and discard the contents of all other tapes.

Now, we will convert the circuit to (a variant of) Pippenger's "alternate representation" – instead of a list of gate records which each store their inputs, we will have a list of gate records, which are now just the index and gate function, together with a second list of "wire"/"value" records of the form (u, v, d) where u is either a gate index (wire) or constant (value), v is a gate index, and $d \in \{L, R, O\}$ indicates whether this wire is the left input to v, the right input, or an "output record." In the case of an output record, u will be either $*$ (wire) or a constant (value), representing either a place holder for the value of the gate or the computed value of the gate.

We can do this by simply scanning over the circuit and for each gate $v = (\ell, r, \varphi, i)$ outputting wire records (ℓ, i, L), (r, i, R), and $(*, i, O)$ and gate record (φ, i).

Given sets of gates I, J, let $W_{I\to J}$ represent the set of wires starting in I and terminating in J. Let $V_{I\to J}$ be the set of value records corresponding to those wires. In both cases let $I \to$ and $\to J$ refer to wires/values starting in I and terminating in J respectively. Let G_I denote the gate records of I.

In the following procedure, each expression like G_K or $V_{\to I}$ should be interpreted as the name of a local variable. Performing the Turing machine construction by replacing each variable with a tape interpreted as a stack of tape yields a multitape Turing machine with around 20 tapes (including a couple work tapes in addition to the stacks).

We now describe a recursive procedure which, given an interval K, $G(K)$, $V_{\to K}$, and $W_{K\to}$ computes $V_{K\to}$. If $|K| = 1$, we simply evaluate the single gate in K using the two value records in $V_{\to K}$ and then substitutes that value into each record in $W_{K\to}$. Otherwise:

- Split K into I and J, each half the size of K.
  - Classify G_K into G_I and G_J.
  - Classify $W_{K\to}$ into $W_{I\to}$ and $W_{J\to}$.
  - Classify $V_{\to K}$ into $V_{\to I}$ and $V_{\to J} \setminus V_{I\to J}$.
- Letting $K \leftarrow I$, recurse to compute $V_{I\to}$ (which we move from the output stack $V_{K\to}$ to the stack $V_{I\to}$).
- Classify $V_{I\to}$ into $V_{I\to J}$ and $V_{I\to} \setminus V_{I\to J}$.
- Merge $V_{I\to J}$ with $V_{\to J} \setminus V_{I\to J}$ to produce $V_{\to J}$.
- Letting $K \leftarrow J$, recurse to compute $V_{J\to}$ (which we again move to stack $V_{J\to}$).
- Merge $V_{I\to}$ and $V_{J\to}$ into $V_{K\to}$

In order to compute our final output, we ensure $V_{K\to}$ is sorted by output gate, and then scan over it and extract the bits b from the records of the form (b, v, O).

Probabilistically Checking Quantum Proofs, with Interaction

Baocheng Sun
Weizmann Institute of Science, Rehovot, Israel

Thomas Vidick
School of Computer and Communication Sciences and Center for Quantum Science and Engineering, École Polytechnique Fédérale de Lausanne, Switzerland
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel

Abstract

The model of interactive oracle proofs (IOP) generalizes the notion of probabilistically checkable proof (PCP), in which a static proof is verified probabilistically by querying a small number of bits, to the interactive setting: a polynomial-time verifier interacts with an unbounded prover, but is restricted to only reading a small number of bits, in total, from the messages sent by the prover. IOPs provide a relaxed setting in which to study local probabilistic verification. They have proved instrumental in devising efficient methods for verification through subsequent compilation into non-interactive or succinct protoocls.

We study a quantum analogue of interactive oracle proofs (qIOP) in which the verifier and communication are both allowed to be quantum; yet the verifier is restricted to perform measurements only on a small number of qubits received from the prover. Our main result is a qIOP for any language in QMA, in which the total communication is polynomial but the verifier only reads a polylogarithmic number of qubits in total. The protocol has completeness parameter exponentially close to 1 and soundness bounded away from 1 by a constant. In the absence of a quantum PCP theorem, this provides the first information-theoretically sound local and robust characterization of QMA, albeit interactive. Previous works in the information-theoretic setting either considered two isolated but entangled quantum provers or quantum verifiers whose effort in a single round is small but remains polynomial when aggregated across all rounds of the protocol.

Our protocol combines the use of a quantum locally testable code (LTC) with classical techniques, notably probabilistically checkable proofs of proximity (PCPP). We avoid the necessity for complex multi-qubit tests employed in other settings by leveraging the local indistinguishability property of the quantum LTC.

2012 ACM Subject Classification Theory of computation → Interactive proof systems; Theory of computation → Quantum complexity theory

Keywords and phrases quantum complexity theory, quantum probabilistically checkable proofs, interactive oracle proofs, quantum locally testable codes, QMA

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.4

Acknowledgements Both authors were supported by AFOSR Grant No. FA9550-22-1-0391 and ERC Consolidator Grant VerNisQDevS (101086733).

1 Introduction

The PCP theorem shows that any problem which has efficiently verifiable proofs, has proofs that in addition can be verified with high confidence by probing only a small, in fact constant, number of bits. Based on the standard conjecture that QMA ≠ NP it is widely believed that there exist problems which admit efficiently verifiable *quantum* proofs, but no classical proofs. Do all such problems have (quantum) local verifiers, which only probe a constant number of qubits from the proof? This is the content of the *quantum PCP conjecture*, which

© Baocheng Sun and Thomas Vidick;
licensed under Creative Commons License CC-BY 4.0

has been open for more than two decades. The main difficulty, colloquially, is that due to the presence of entanglement information in a quantum proof can be encoded globally and is not necessarily accessible to local probes.

In this work we give a new characterization of QMA in terms of (quantum) efficient *interactive* and local verification procedures. We study a recently introduced [33] quantum analogue of *interactive oracle proofs* (IOP) and show that every language in QMA has an efficient quantum interactive oracle proof (qIOP) where the verifier runs in quantum polynomial time and "reads" only a small (polylogarithmic, or even constant) number of qubits from the proof.[1] In addition, in our proof system the honest prover is relatively efficient and can be executed in quantum polynomial time given access to the right witness for the QMA language instance.

1.1 Results

To introduce our results in more detail we first need to sketch the model of quantum interactive oracle proof (qIOP). Informally, a qIOP (for a language L) is an interactive protocol between a quantum polynomial-time verifier and a quantum unbounded-time verifier in which the verifier "reads only a small number of qubits, in total, that were sent by the prover." Here and henceforth, we use "small" to refer to a quantity that is constant or at most polylogarithmic in the instance size. The protocol should satisfy the usual completeness and soundness conditions: for any instance that is in the language, there should exist a prover that is accepted with probability at least $2/3$;[2] and for any instance that is not in the language, any prover, even computationally unbounded, should be accepted with probability at most $1/3$.

Of course, the subtlety in the definition lies in the precise formalization of the intuitive condition "reads only a small number of qubits." Even for the case of a quantum probablistically checkable proof (qPCP), which corresponds to the non-interactive variant of a qIOP, the notion is not trivial to formalize and some work needs to be done to show that different natural formulations are equivalent (when thay are); see e.g.[1, 9] or even [28].

A formal definition of qIOP appears in [33]. We give an adaptation of the same definition in Section 4, removing variations that are not needed for this paper; we refer the reader to that section for the formalities. For the purpose of the introduction we proceed by giving a semi-formal definition that is tailored to our protocol construction and yet can be seen to fit the formal definitions from Section 4 in a straightforward manner.

Given a language L and functions (of the input length) ℓ, q, r, an (ℓ, q, r)-qIOP for L is given by the description of a quantum polynomial-time verifier which interacts with a quantum prover through a $(2r+1)$-message interaction. Without loss of generality, the first message comes from the prover and consists of an ℓ_0-qubit register W, which we think of as containing a suitably encoded quantum proof. Each subsequent round $i = 1, \ldots, r$ is a 2-message interaction which proceeds as follows. The verifier may perform three actions: measure up to q_i qubits from the prover; send M_i, a subset of qubits from W that is deterministically chosen before the protocol begins, to the prover; and send a classical message C_i, possibly depending on some private coin tosses, to the prover. The prover applies an arbitrary unitary on all his registers. He performs a measurement and returns a classical register C'_i to the

[1] Here, we put quotes because the notion of "reading" a qubit can take different formalizations; we adopt a natural one which is described later on.

[2] For this case, we may in addition require the prover to be efficient given a witness. Whenever this holds, it should be stated explicitly.

verifier.[3] (Let ℓ_i be the combined length, in qubits, of M_i, C_i and C'_i; the parameters ℓ for the qIOP is defined as $\sum \ell_i$.) At the last stage, i.e. after having received C'_r from the prover, the verifier flips some private coins and reads at most $q - \sum q_i$ bits from all classical registers he has in his possession (i.e. private classical registers, and all the C'_i registers). Based on the bits read, and any measurement outcomes and classical random bits generated by the verifier earlier in the protocol, the verifier makes his decision.

We note that this formulation precludes some adaptations that would be natural. For example, one could allow the verifier to adaptively determine which qubits to send back in each round. The verifier could also send qubits that were not directly obtained from the prover. The general model of qIOP naturally allows such operations; because our protocol does not require them we omit them from the preceding discussion, for simplicity.

For the protocol to be considered a qIOP, it is required that q is small; whereas ℓ is allowed to be an arbitrary polynomial. While in principle r may also be polynomial, for us it is sufficient to take r to be small as well (polylogarithmic). Using this formalism our main result can be informally stated as follows.

▶ **Theorem 1.1** (Main result, informal). *Every language in* QMA *has a* (poly, polylog, polylog)*-qIOP. In addition, the qIOP has completeness exponentially close to* 1 *and the honest prover is efficient, i.e. it can be executed in quantum polynomial time provided it is initially given access to the right quantum witness.*

We note a few desirable features of our protocol. Firstly, the verifier's additional classical bits sent to the prover are restricted to a single bit for all but the last message, which is polynomial (but could be made logarithmic using a stronger QMA-complete problem as starting point, see Section 1.2.1 below).

Secondly, the polylogarithmic query and round complexity are due to the sub-optimal parameters known for the best quantum locally testable code (qLTC) construction known [13], a key ingredient in our protocol. If there was a good qLTC construction, with positive rate, constant relative distance and soundness, and constant-weight parity checks, then the query and round complexity of our protocol would also both be improved to constant.

Thirdly, the total communication in the protocol is essentially quasi-linear, *except* for the length of the various (classical) PCPP proofs that are sent by the prover to the verifier, which to the best of our knowledge may be super-linear, e.g. quadratic in length. If linear PCPPs were obtained then the total communication in our protocol would be linear, up to polylogarithmic factors. Moreover, we note that the only quantum communication consists of (i) the quantum witness sent from prover to verifier in the first round, and (ii) successive blocks of the same witness sent back from the verifier to the prover in each of the r rounds of the protocol.

Nevertheless, we remark that the parameters as stated in the theorem may appear quite lose to a reader familiar with the literature on classical IOPs. Indeed, in the latter case there is a focus on the proof length (the parameter ℓ here) which is usually desired to be quasi-linear. This is because one already has access to the PCP theorem, which provides a polynomial-size proof, and the focus for IOPs is on strong efficiency. For us, there is no known quantum PCP and so any polynomial length proof is interesting. Furthermore, the size of the prover's first quantum message is actually linear in the QMA proof length (for

[3] The prover may be asked to return the register M_i, but in our protocol this is not needed by the verifier. Moreover, note that in general it cannot be enforced what qubits the prover sends back; only how many qubits are sent.

the specific language that we choose), while subsequent message registers W_i sent from the (honest) prover to the verifier can be longer but are "classical" in the sense that they are obtained by performing a classical computation in superposition on the register M_i.

1.2 Proof ideas

We now discuss the proof of our result, namely the construction of our quantum IOP for QMA and the completeness and soundness arguments.

1.2.1 Amplified XZ Hamiltonians

Our starting point is the QMA-complete local Hamiltonian problem [23]. For reasons that will become clear later on, it will be convenient for us to start with a local Hamiltonian H that decomposes as an average of two local Hamiltonians $H = \frac{1}{2}(H^Z + H^X)$, one of which, H^Z, is diagonal in the computational basis and the other, H^X, in the Hadamard basis. The QMA-completeness of such "XZ Hamiltonians" was already shown in [7].[4]

The associated completeness and soundness parameters (expressed as the smallest eigenvalue of H for YES and NO instances respectively), however, are only inverse polynomially separated. To obtain a qIOP with constant completeness-soundness gap this needs to be amplified before any protocol is attempted.[5] The usual tensoring amplification as used in e.g. [27] leads to mixed terms (with components in both bases), which is inconvenient for us. Instead, we need to make two modifications.

First, we need to start with a complete problem that has the desired structure, in terms of separation of the local terms by X or Z basis, but such that in addition each local term is a projection, the completeness is exponentially close to 0 (since we are talking about the minimum eigenvalue), and the soundness is inverse polynomially large. This will be needed for the subsequent amplification step. The result of [7] does not immediately give this. A close result is the QMA-completeness proof of the Clifford Hamiltonian problem from [8], which has the required parameters but is based on a richer set of bases than X and Z. Inspired by their work, we developed a bespoke reduction from QMA to a local Hamiltonian $H = \frac{1}{2}(\Pi^Z + \Pi^X)$ where Π^Z and Π^X are normalized averages of local projections diagonal in the computational and Hadamard basis respectively. The same reduction has been independently been obtained by Natarajan and Ma [24], with similar motivation. Since their work appeared before ours, we directly use their result.[6]

We then amplify the completeness-soundness gap of H while keeping the two-bases structure intact, e.g. we obtain $H^{amp} = \frac{1}{2}(\Pi^{Z,amp} + \Pi^{X,amp})$ where $\Pi^{Z,amp}$ and $\Pi^{X,amp}$ are normalized averages of non-local (but still efficiently described as tensor products of local terms, up to a subtraction of the identity) projections diagonal in the Z and X basis respectively. Furthermore, the smallest eigenvalue of H is exponentially close to 0 in the case of a positive instance, and larger than a universal constant in the case of a negative instance. This amplification is achieved using a simple method due to Anshu and explained in Section 5. We note that the amplification leads to $\Pi^{Z,amp}$ and $\Pi^{X,amp}$ that are averages of an exponential number of non-local terms; yet a given non-local term can still be efficiently

[4] In that reference such Hamiltonians, when in addition they are 2-local, are called "$ZZXX$ Hamiltonians." We use the terminology that was subsequently used in e.g. [18, 27].

[5] This is because e.g. sequential amplification of the protocol itself would require polynomially many repetitions, ruining any chances at keeping the query complexity small.

[6] And, obviously, do not claim credit for it.

sampled and described. This leads directly to the length of the last classical message sent by the verifier in our protocol to be polynomial. It is possible to make that message logarithmic by using the more advanced, derandomized amplification technique in [6]. Since at the moment we do not know if the length of that message is important (see Section 1.4 on open questions), we do not include details on this modification.

1.2.2 Trusted measurement provers

Given an amplified XZ Hamiltonian $H = \frac{1}{2}(\Pi^Z + \Pi^X)$ on B_q qubits provided as input, we warm up by describing a simple verification protocol in the case of a "trusted measurement prover." This is a fictional prover (considered only for the purpose of this introduction) who holds an arbitrary quantum state $|\varphi\rangle_{\mathsf{W}}$ of M qubits and returns M-bit measurement outcomes of $|\varphi\rangle_{\mathsf{W}}$ in the computational (Z) or Hadamard (X) basis whenever asked to do so.

Then the following simple, classical protocol would work. Since the verifier is only allowed to access a small number of bits of any outcome reported by the prover, and we do not assume any promise on the degree or locality of H (so a small number of bit or phase flips can potentially affect the energy by a lot), it is sensible to require $|\varphi\rangle_{\mathsf{W}}$ to be an encoding of a ground state $|\Gamma\rangle$ of H using a suitable quantum error correcting code. Thus let C_q (q for "quantum") be an $[M, B_q, d_q]$ quantum error correcting code with linear distance $d_q \propto M$, such that C_q is furthermore a CSS code with parity check matrices (H_X, H_Z). Let $\mathtt{E}_q$ the unitary encoding map of C_q, and $|\varphi\rangle_W = \mathtt{E}_q(|\Gamma\rangle_{\mathsf{W}} |0\rangle_{\mathsf{S}})$ where $|0\rangle_{\mathsf{S}}$ are ancilla qubits required for the encoding process.

Now suppose the verifier chooses a basis $W \in \{X, Z\}$ at random and requests measurement outcomes obtained from measuring $|\varphi\rangle_W$ in the basis W. The trusted measurement prover obliges, sending outcomes $w \in \{0,1\}^M$. The verifier then samples a projection Π_j^W in such a way that $\Pi^W = \mathbb{E}_j \Pi_j^W$. Since it is diagonal in basis W, the constraint Π_j^W can be represented by a boolean function $f_j : \{0,1\}^{B_q} \to \{0,1\}$ (we suppress the dependence on W for clarity) such that $\Pi_j^W = \sum_x f_j(x) S_W |x\rangle\langle x| S_W^\dagger$, where $S_W = \mathsf{I}^{\otimes B_q}$ if $W = Z$ and $S_W = H^{\otimes B_q}$ if $W = X$. Furthermore, we can represent the function f_j as a "logical" function $f'_j : \{0,1\}^M \to \{0,1\}$ by using logical operators for the code C_q.[7] For reasons that will soon become clear, define $\widetilde{f}_j = f'_j \circ \mathtt{E}_W \circ \mathtt{D}_W$, where $\mathtt{E}_W$ and $\mathtt{D}_W$ are the encoding and decoding maps associated with the linear code $C_W = \ker H_W$ respectively.[8]

The goal of the verifier is to check that $f'_j(w) = 0$. If $w \in C_W$, as it would be in case $|\varphi\rangle_{\mathsf{W}}$ is the intended witness, this is equivalent to checking that $\widetilde{f}_j(w) = 0$. So, it is enough for the verifier to check that $w \in C_W$ AND $\widetilde{f}_j(w) = 0$. Since it cannot do either task by itself (in particular, $\widetilde{f}_j$ is nonlocal) it will enlist the prover's help.

To explain this first recall that a *PCP of proximity* (PCPP) for an (efficiently computable) statement of the form "$g(y) = 0$" is a proof π that guarantees that y is *close* to a string y' such that $g(y') = 0$, and such that verifying π only requires reading a constant number of bits of y and π.

---

[7] Somewhat informally, we map a string $x \in \{0,1\}^M$ to $x' \in \{0,1\}^{B_q}$ by evaluating the logical operators, i.e. $x' = Kx$ where the rows of K represent the support of logical operators for C_q in basis W. We refer to the definition of $\widetilde{f}$ from f in Section 3 for the precise methodology.

[8] Note that C_W does not necessarily have distance. The only property we will need is that the decoder $\mathtt{D}_W$ decodes a small syndrome to an error whose size is not much bigger. This property is satisfied by the decoder in basis W for the code we use, see Section 1.2.5 and Section 2.4.

The prover is then asked to provide a PCPP for the statement: "$\widetilde{f_j}(w) = 0$ AND $H_W w = 0$". Here, the role of y is played by w, and the role of g is played by the function $(1 \oplus \widetilde{f_j}(\cdot)) \wedge 1_{H_W(\cdot)=0}$.[9] By the PCPP guarantee, if the PCPP proof is accepted with high probability then w is *close* to a string w' such that $H_W w' = 0$ and $\widetilde{f_j}(w') = 0$. We claim that it must be that $\widetilde{f_j}(w) = 0$ as well. This is because by the soundness property of the PCPP, we must have $w = w' + a$ where a is a small error. Using that $\mathtt{D}_W$ is a local decoder (see footnote 8), $a' = \mathtt{E}_W \circ \mathtt{D}_W(a)$ is a small element of $\mathrm{Im}(H_{W'}^T)$ (where W' is the complementary basis to W). Thus for every logical operator, $K_i \cdot a' = 0$ where K_i is the support of the ith logical operator in basis W (see Lemma 6.5). It follows from the definition that $\widetilde{f_j}(w) = \widetilde{f_j}(w')$, as desired. This concludes the analysis of the situation where the prover is a "trusted measurement prover."

1.2.3 Verifying measurement outcomes

Now of course the main difficulty remains: we need to devise a query-efficient procedure by which the verifier can determine that the prover correctly reports a string of measurement outcomes in the X or Z basis, where in either case the measurement has been performed on the same quantum state $|\varphi\rangle_{\mathsf{W}}$ (which need not be pure).

Interactive procedures for verifying measurement outcomes in these bases have been developed in other settings, and form the cornerstone of delegated computation protocols. For example, in the two-prover setting it is by now well-understood that building on non-local games such as the CHSH or Magic Square games, and combining with techniques from classical locally testable codes, one can devise games that force the prover to measure in the required bases; see e.g. [22, 10]. Similarly, Mahadev's delegated computation protocol [25], with a single quantum prover and a classical verifier, makes use of computational assumptions to essentially force the prover to make measurements in two mutually unbiased bases; see e.g. [20] for more on this. We also note that the two-prover multi-qubit tests have been ported to the single-prover, classical-verifier setting through compilation, based on quantum fully homomorphic encryption; see e.g. [31, 27].

In our setting, we have access to a single quantum prover and no computational assumptions. However, we may leverage quantum communication. A recent paper [33] introduces a direct analogue of the two-prover multi-qubit tests by using interaction and hiding the verifier's choice of test in entanglement between the verifier and prover. However, in that paper they were only able to achieve a qIOP with exponential communication. Our model and theirs are closely related, yet not entirely comparable.

Here we use a different idea, which leverages the properties of quantum error correcting codes and quantum locally testable codes.

1.2.4 Leveraging local indistinguishability

We first observe that there is a very classic single-qubit measurement test that leverages quantum communication. The verifier prepares an EPR pair and sends one qubit to the prover. The verifier asks the prover to measure their qubit in a randomly chosen basis $W \in \{X, Z\}$ and report the outcome x. The verifier measures their own half-EPR pair to

[9] Note that in the actual protocol, the verification that w is a codeword of the code C_W is performed by running the local testing algorithm of C_W, which is locally testable (see Definition 2.20). In the protocol, we also need to perform a codeword test for a classical ECC; however, this part is unrelated to the current section. The underlying intuition for that part is explained in Section 1.2.5.

obtain an outcome y, and accepts if and only if $x = y$. It is easily verified that the only possibility for winning perfectly in this game is for the prover to measure the qubit he received from the verifier in the correct basis.

The main limitation of this protocol is that the qubit measured is a half-EPR pair; whereas what we really want is a measurement of the quantum witness. A similar difficulty in the two-prover setting was resolved by the use of teleportation in [18]. Here, we employ a different idea, which is inspired by the local indistinguishability property of quantum error-correcting codes. Recall that in Section 1.2.2 we already required the witness to be encoded using a quantum code C_q. If the code has distance d_q, then it is well-known that the reduced density matrix ρ_S of any codeword on a subset S of $|S| < d_q$ qubits must be independent of the specific codeword. Suppose for simplicity that ρ_S is totally mixed. Then this means that, up to unitary rotations, the qubits in S and the qubits in $\overline{S}$ are equivalent to a number of EPR pairs! This suggests that a suitable adaptation of the simple test exposed in the previous paragraph could lead to certifying the prover's measurements on any sufficiently small portion of the code.

Concretely, we consider the following procedure. Divide the M qubits of an encoded state into r blocks such that each block has less than d_q qubits. For any $i \in \{1, \ldots, r\}$, the verifier sends to the prover the qubits in the ith block only. The verifier requests a measurement in basis W of those qubits, leading to outcomes x. The prover reports x. The verifier measures all other qubits in basis W, leading to an outcome x'. The verifier checks that together, (x, x') satisfy the code stabilizers in basis W, i.e. $H_W(x, x') = 0$.

It is not hard to show that a prover succeeding with probability 1 in this test must indeed perform honest measurements on the qubits of the codeword that it is given. This idea forms the basis for our measurement extraction procedure, described in full in Section 8. However, we need to perform a number of modifications that introduce additional technical difficulties.

Firstly, the description we gave requires the verifier to perform a large measurement on the witness, which is not allowed by the qIOP model. To alleviate this we rely on a quantum code that not only has low-weight parity checks but furthermore is locally testable. This ensures that the verifier (together with the prover's outcomes) will be able to assess the distance of the quantum witness from the quantum code space by measuring only a small number of qubits. Concretely we use the quantum locally testable (qLTC) codes based on cubical complexes from [13]. Such codes have good but not quite near-optimal parameters, i.e. for code length M they have linear rate, distance $\Omega(M/\mathrm{poly}\log M)$, soundness $\Omega(1/\mathrm{poly}\log M)$ and constant-weight parity checks.[10]

Secondly, we of course need to certify (and obtain) measurements on the entire witness, not only a small block of qubits. This is necessary because by the same principle of local indistinguishability, no information about the witness could be extracted by measuring only a small number of qubits. To address this we introduce an interactive procedure in which measurements are performed by the prover and only read and certified by the verifier when needed. We describe both aspects next.

1.2.5 Interactive measurement extraction using locally testable codes

We sketch the procedure by which the verifier is able to iteratively extract trusted measurement outcomes in basis W from the prover. The precise procedure and analysis are given in Section 8. The quantum witness sent by the prover to the verifier in the initial message is

[10] Some tradeoffs are possible between these parameters [34]; for clarity we do not explore these and stay with the "basic" construction.

divided into $r = M/m$ blocks of qubits, where $m \ll d_q$. The protocol then proceeds in r rounds (note that eventually we will have $r = \text{poly}\log(M)$). At the ith round, the m qubits in the ith block are sent to the prover. The prover is asked to measure them in basis W to obtain a string $x \in \{0,1\}^m$. In addition, the prover is asked to report an encoding $e = \mathtt{E}(x)$ of x according to an $[N, m, \Delta]$ linear error-correcting code C. It also provides PCPP proofs $b_1, \ldots, b_m$ where b_j is a proof for the claim that e is a valid codeword such that $\mathtt{D}(e)_j = x_j$, with $\mathtt{D}$ the decoding map of C (which is required to be efficient, up to some fraction of the distance).

In a randomly chosen round, the verifier samples a random basis-W stabilizer generator $g_W \in \{I, W\}^{\otimes M}$ for the quantum code C and verifies it as follows. If a qubit in the support of g_W has never been sent back to the prover, then the verifier measures it directly. If however the qubit has already been sent back to the prover, and thus measured in basis W and used in the preparation of PCPP proofs, then the verifier checks the associated PCPP and uses the measured value x_j returned by the prover instead.

To analyze this procedure we need to take into account that in later rounds, most of the qubits of the codeword have previously been acted on by the prover, and thus the local indistinguishability principle no longer applies. Nevertheless we are able to leverage the LTC property, which holds separately for both classical codes that underlie the CSS quantum LTC code, together with soundness of the PCPP to argue that any deviation from the prover in the complementary basis W' (i.e. X errors for an extraction in basis Z; these are the only errors that would affect a measurement in basis W) must have bounded weight. Using the linear distance property of the quantum code C, bounded weight errors do not affect logical operations. This guarantees that measurement outcomes are reported correctly, as desired.

To conclude it remains to bring together the two components, the constraint verification procedure described in Section 1.2.2 and the measurement extraction described here.

1.2.6 Protocol summary

We now summarize the structure of our protocol, from which it will be evident that the protocol has the parameters announced in Section 1.1. We describe the protocol in the case of a YES instance.

Let $H = \mathbb{E}_{W \in \{X,Z\}} \mathbb{E}_j \Pi_j^W$ be an instance of the amplified XZ Hamiltonian problem described in Section 1.2.1, let $|\Gamma\rangle$ be a ground state of H and $|\varphi\rangle_{\mathsf{W}} = E_q(|\Gamma\rangle\,|0\rangle_{\mathsf{S}})$, where E_q is the encoding unitary for a CSS code $C = (H_X, H_Z)$ with distance d_q that is locally testable. Let M be the code length. Divide M into r chunks of m qubits where $m \ll d_q$.

The protocol has two phases. The first phase is called the *measurement extraction* phase, and the second phase is the *constraint verification* phase. We first describe the measurement extraction phase. It consists of $1 + 2r$ messages, as follows:

1. The prover sends an M-qubit register W to the verifier. This register is supposed to contain $|\varphi\rangle_W$.
2. The verifier samples a basis $W \in \{X, Z\}$ uniformly at random.
3. For $i = 1, \ldots, r = M/m$ do the following:
   a. The verifier sends qubits in the ith block of W to the prover. If $i = 1$, the verifier also sends the basis W to the prover.
   b. The prover measures this block in basis W to obtain outcomes $x \in \{0,1\}^m$. He encodes x using a classical code C_c to obtain e. The prover returns the ith block of W together with x, e, and PCPP proofs b_j, for $j \in \{1, \ldots, m\}$, that e is a valid encoding and decodes to x_j.

c. In one of the rounds, chosen at random, the verifier executes the qLTC local tester for basis W. Whenever a query lands on a qubit that has previously been measured by the prover, the verifier uses x_j as the measurement outcome, checking the corresponding PCPP proof for correctness. Whenever the qubit has never been sent to the prover, the verifier measures it directly.

At the end of this phase, the verifier has in his possession a string x that is supposed to contain measurement outcomes corresponding to measuring the encoded witness $|\varphi\rangle_{\mathsf{W}}$ in basis W. Our analysis (see Lemma 8.11) shows that, indeed, any (efficiently computable) "logical" function $\widetilde{f}$ of x computed at the end of the phase is consistent with the evaluation of the "honest" observable associated with $\widetilde{f}$ on $|\varphi\rangle_{\mathsf{W}}$ at the beginning of the phase.

In the second and last phase, the *constraint verification* phase, the verifier interacts with the prover one more time to evaluate a non-local Hamiltonian constraint Π_j^W on x. This phase consists of only 2 messages, as follows:

1. The verifier selects a random constraint Π_j^W from the Hamiltonian H. Let $f_j : \{0,1\}^{B_q} \to \{0,1\}$ be the associated function, as in Section 1.2.2. The verifier sends j to the prover.
2. The prover sends the value $b = f(KE_W\mathtt{D}_W(x))$ together with a PCPP proof that $e_1, \ldots, e_r$ are valid codewords such that $f(KE_W\mathtt{D}_W(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_r))) = b$. Here, $\mathtt{D}_W$ is the W-basis decoder for the qLTC, E_W is the encoder for the code $C_W = \ker H_W$, and K is a matrix of logical operators (in rows) for basis W. $\mathtt{D}$ is the decoder for the classical code. (See Section 1.2.2 for why this representation is used.)
3. The verifier accepts if and only if the PCPP proof is accepted and $b = 0$.

The analysis of this phase, given the results of the previous phase, is fairly straightforward and follows the sketch given in Section 1.2.2.

1.3 Related work

Previous works have proved or conjectured characterizations of QMA in terms of efficient, "local" verification in other settings. The *quantum games PCP* conjecture [28] considers the case of a classical verifier who exchanges $O(\log n)$-length messages with two provers sharing entanglement, where n is the instance size. While this conjecture remains open, in contrast to the quantum PCP conjecture mentioned earlier there has been notable progress on it in the past years and its resolution appears close (at least to the present authors). In particular, the development of so-called multi-qubit tests as e.g. the Pauli Braiding test of [29, 30, 21], which emphasize the role of tensors product observables involving separate X and Z observables only, influenced this work.

In a different direction, previous works have considered the case of a classical verifier running in polylogarithmic time (given the right input access model) and shown that languages in QMA have interactive verification protocols of this sort, under computational assumptions [35, 4, 31, 27]. We only consider the information-theoretic setting (but see the open problems subsection below).

Our work is arguably more related to early information-theoretic protocols for quantum verification with a small-size quantum verifier [2, 16, 15]. The main difference with our work is that, in all these results, the total quantum effort of the verifier, aggregated across rounds, remains polynomial. In our paper, the total quantum effort of the verifier is polylogarithmic. The difference is substantial; and indeed, while the cited works do not rely on further structural results than the quantum circuit model and Kitaev's reduction showing that the local Hamiltonian problem is QMA-hard, our results require a tailored, amplified QMA-hard problem to start with; classical probabilistic proof verification technique (in the form of PCPPs); as well as recent advances in quantum codes and in particular the design of quantum locally testable codes.

1.4 Open problems

In the classical literature on delegated computation, interactive oracle proofs are a step towards more "practical" primitives. The next step is achieved by removing interaction and/or achieving succinctness by replacing the long, mostly unused messages by efficient commitment schemes such as a Merkle tree. It is thus natural that the next step in our work would be to seek methods for "compiling" our quantum interactive PCP into a succinct and/or non-interactive scheme, based e.g. on the quantum random oracle model. A few features of the protocol make this step, in particular the non-interactive part, non-trivial. Notably, our protocol is private-coin (for example, qLTC stabilizer checks are done privately), and moreover involves sending quantum messages from the verifier to the prover; such messages are not easily removed by the standard Fiat-Shamir heuristic. Some approaches exist to Merkle tree-like constructions in the quantum setting [11, 19] and it would be natural to try to use those first in order to achieve succinctness. We leave such attempts for future work.

Another potential direction would aim to connect our work to "compiled" protocols as in [27], which are interactive protocols with a classical succinct verifier, obtained at the cost of relatively heavy computational assumptions such as quantum fully homomorphic encryption. It is possible that our protocol could be more directly compiled into one with a classical verifier, ideally while using weaker computational assumptions.

More immediate directions aim to reduce the complexity of our interactive protocol along different axes. It is possible that the protocol can be made public-coin, as the most important random choices of the verifier, the basis in which to extract measurement outcomes and the term of the Hamiltonian that is estimated in the energy check, are already done publicly. Other random choices, such as which round to perform a test in or which code stabilizer is measured, are made privately in our formulation; yet it appears that it would be straightforward to make them publicly. A more difficult direction is to remove quantum communication from the verifier to the prover, which could aid compilation. One could also focus on the total communication length. At the moment the most important components, such as the qLTC-encoded witness, are quasi-linear, but other components, such as the PCPPs, remain super-linear, to the best of the authors' knowledge.

Finally, while our work focuses on establishing a lower bound on the power of qIOPs, i.e. on designing a concrete protocol, it is of course interesting to study potential limitations on the model. As one puts restrictions on the communication, the locality, etc. when does one cross the boundary between QMA and NP (or, say, QCMA)?

2 Preliminaries

We collect mostly standard material on classical and quantum error correcting codes (ECCs), locally testable codes (LTCs), and probabilistically checkable proofs of proximity (PCPPs).

2.1 Error correction codes

We begin by recalling the existence of good families of error-correcting codes.

▶ **Theorem 2.1** (Error correction codes). *There exist families of linear error-correcting codes $(C_k)_{k\geq 1}$ that encode k bits in $n = n(k)$ bits and have polynomial rate, constant relative distance, and deterministic polynomial-time encoding and decoding algorithms.*

▶ **Definition 2.2.** *Fix a family of codes $(C_k)_{k\geq 1}$ as in Theorem 2.1. Here, C_k encodes k bits into $n(k)$ bits with distance $d(k)$, where $n(k) = \mathrm{poly}(k)$ and $d(k) = \Omega(n)$. Let $\mathtt{E}$ and $\mathtt{D}$ be the associated encoding and decoding algorithms respectively (which implicitly depend on k). Further, let $\mathtt{T}$ be the associated code tester, which evaluates to 1 on the range of $\mathtt{E}$ and 0 elsewhere.*[11]

Next, we introduce a quantum operator corresponding to the classical encoding process.

▶ **Definition 2.3** (Encoding operator)**.** *Let $\mathtt{E}$ be the encoding algorithm from Definition 2.2. For $k \in \mathbb{Z}_+$, $a \in \mathbb{F}_2^{n(k)}$, and $w \in \mathbb{F}_2^k$, define the encoding operator*

$$E(|a\rangle\,|w\rangle) = |a \oplus \mathtt{E}(w)\rangle\,|w\rangle\,.$$

We now formally define classical locally testable codes, which guarantee that proximity to the codespace can be efficiently checked using a small number of queries.

▶ **Definition 2.4** (Classical locally testable codes)**.** *Let $C = \ker(H)$ be a classical code, where $H \in \mathbb{F}_2^{m\times n}$ is a parity-check matrix. We say that C is* locally testable *with soundness ρ if for all $x \in \mathbb{F}_2^n$, it holds that*

$$\frac{1}{m}|Hx| \geq \rho\frac{d_H(x,C)}{n}\,,$$

where $|x|$ denotes the Hamming weight (number of nonzero coordinates) of x and $d_H(x,C)$ denotes the Hamming distance between x and the code C.

We define the local tester associated with a classical locally testable code. This tester checks whether a given string is a valid codeword by examining a small number of randomly selected parity constraints. Note that the tester depends on the choice of a parity check matrix for the code.

▶ **Definition 2.5** (Local tester for a classical LTC)**.** *Let $c \in \mathbb{Z}_+$. Let $C = \ker(H)$ be a classical locally testable code, where $H \in \mathbb{F}_2^{m\times n}$ is a parity-check matrix. Define the local tester $\mathtt{T}$ for the code C as the following procedure:*

1. *Given input $w \in \mathbb{F}_2^n$,*
2. *Randomly sample $i_1, \ldots, i_c \in_R [m]$,*
3. *Compute*

$$r = \begin{bmatrix} H_{i_1} \\ \vdots \\ H_{i_c} \end{bmatrix} \cdot w \in \mathbb{F}_2^c,$$

4. *Reject if $\bigvee_{j\in[c]} r_j = 1$ (i.e., any parity constraint is violated).*

We now present a method to amplify the tester of any locally testable code, for ease of application:

▶ **Lemma 2.6.** *Let $C = \ker(H)$ be a locally testable code with soundness ρ, where each row of H has weight at most R. Let $\kappa > \rho$.*

[11] `T` can be implemented by e.g. executing `D` and then `E` and returning 1 if the result is unchanged, 0 otherwise.

Then, there exists a code tester for C with query complexity Q such that for all $x \in \mathbb{F}_2^n$,

$$\Pr[\textit{verifier rejects } x] \geq 1 - \left(1 - \rho \cdot \frac{d_H(x,C)}{n}\right)^{Q/R} .$$

Furthermore, there exists a code tester for C with query complexity

$$Q = R \cdot \frac{\kappa}{\rho} \cdot \frac{\log(1/\varepsilon)}{1-\varepsilon}$$

such that for all $x \in \mathbb{F}_2^n$, it holds that

$$\Pr[\textit{verifier rejects } x] \geq \min\left\{\kappa \cdot \frac{d_H(x,C)}{n}, 1-\varepsilon\right\} .$$

Proof. By independently sampling a row of H a total of $N = Q/R$ times and checking whether the codeword satisfies the corresponding parity check, we obtain a code tester that rejects a near-codeword with the following probability:

$$\begin{aligned}\Pr[\text{verifier rejects}] &= 1 - \left(1 - \frac{1}{m}|Hx|\right)^N \\ &\geq 1 - \left(1 - \rho\frac{d_H(x,C)}{n}\right)^N .\end{aligned}$$

By taking $N = \frac{\kappa}{\rho} \cdot \frac{\log(1/\varepsilon)}{1-\varepsilon}$, we have

$$1 - \left(1 - \rho\frac{d_H(x,C)}{n}\right)^N \geq 1 - \exp\left(-\rho\frac{d_H(x,C)}{n}N\right) .$$

If $\rho\frac{d_H(x,C)}{n}N \geq \log(1/\varepsilon)$, then

$$1 - \exp\left(-\rho\frac{d_H(x,C)}{n}N\right) \geq 1 - \varepsilon .$$

If $\rho\frac{d_H(x,C)}{n}N \leq \log(1/\varepsilon)$, note that for $x \in [0,a]$,

$$\exp(-x) \leq 1 - \frac{1-\exp(-a)}{a}x.$$

Then,

$$\begin{aligned}1 - \exp\left(-\rho\frac{d_H(x,C)}{n}N\right) &\geq 1 - \left(1 - \frac{1-\varepsilon}{\log(1/\varepsilon)} \cdot \rho\frac{d_H(x,C)}{n}N\right) \\ &= \kappa\frac{d_H(x,C)}{n}.\end{aligned}$$

Therefore, in all cases,

$$\Pr[\text{verifier rejects}] \geq \min\left\{\kappa\frac{d_H(x,C)}{n}, 1-\varepsilon\right\}. \qquad \blacktriangleleft$$

We introduce a quantum analogue of the local tester. This operator allows the verifier to coherently verify proximity of a string to the code space using a classical local test.

▶ **Definition 2.7** (Local tester operator). *Let T be the local tester algorithm for a classical LTC, as defined in Definition 2.5. Let l_r denote the number of random bits used by T, and let q denote the number of bits of q that are queried by the tester. For $r \in \mathbb{F}_2^{l_r}$, let $\mathsf{T}(w;r)$ denote the output of T when the randomness is r and the query results corresponding to that randomness are given by w.*

Then, for $a \in \mathbb{F}_2$, $w \in \mathbb{F}_2^q$, and $r \in \mathbb{F}_2^{l_r}$, define the local tester operator as

$$L_r(|a\rangle |w\rangle) = |a \oplus \mathsf{T}(w;r)\rangle |w\rangle \ .$$

2.2 Quantum CSS codes

We first introduce standard notation for Pauli operators.

▶ **Definition 2.8** (Pauli Operators). *The Pauli operators are defined as*

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

We introduce a convenient notation for applying a single-qubit operator selectively across multiple qubits, according to a binary string.

▶ **Definition 2.9** (Tensor product of single-qubit operators). *Let $e \in \mathbb{F}_2^n$ and let W be a single-qubit operator. Define*

$$W(e) = W^{e_1} \otimes W^{e_2} \otimes \cdots \otimes W^{e_n},$$

that is, $W(e)$ applies W at each position i where $e_i = 1$ and applies the identity operator otherwise.

Next, we define CSS codes.

▶ **Definition 2.10** (CSS Code). *An $[n, k, d]$ quantum CSS code on n qubits is a subspace $C \subseteq \mathcal{H} = (\mathbb{C}^2)^{\otimes n}$. It is defined by a pair of parity-check matrices H_X, H_Z such that $\mathrm{Im}(H_X^\top) \perp \mathrm{Im}(H_Z^\top)$. We denote the code as $C = CSS(H_X, H_Z)$. Explicitly, the code subspace is given by*

$$CSS(H_X, H_Z) = \left\{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : X(x)|\psi\rangle = |\psi\rangle \ \forall x \in H_X,\ Z(z)|\psi\rangle = |\psi\rangle \ \forall z \in H_Z \right\},$$

which can also be written as

$$CSS(H_X, H_Z) = Span\left\{ \frac{1}{\sqrt{|H_X|}} \sum_{x \in H_X} |z + x\rangle : z \in \ker(H_Z) \right\}.$$

The code encodes $k = \log_2 \left(|\ker(H_X)/\mathrm{Im}(H_Z^T)| \right)$ logical qubits and has distance

$$d = \min \left\{ |w| : w \in (\ker(H_X) \setminus \mathrm{Im}(H_Z^T)) \cup (\ker(H_Z) \setminus \mathrm{Im}(H_X^T)) \right\}.$$

Finally, we define the distance of the W-code modulo the image of the parity-check matrix of the complementary code.

▶ **Definition 2.11.** *Let C be a quantum CSS code and H_X, H_Z the associated parity check matrices, as in Definition 2.10. Define*

$$d_{q,X} = \min \left\{ |w| : w \in \ker(H_X) \setminus \mathrm{Im}(H_Z^T) \right\},$$

$$d_{q,Z} = \min \left\{ |w| : w \in \ker(H_Z) \setminus \mathrm{Im}(H_X^T) \right\}.$$

We now state a lemma that guarantees the existence of a natural basis of logical Pauli operators in CSS codes.

▶ **Lemma 2.12.** *Let C be an $[M, B_q, d_q]$ CSS code. Then there exist $F, G \in \mathbb{F}_2^{B_q \times M}$ such that*

$$X(F_1), \ldots, X(F_{B_q})$$

are B_q linearly independent logical operators composed only of X and identity operators, and

$$Z(G_1), \ldots, Z(G_{B_q})$$

are B_q linearly independent logical operators composed only of Z and identity operators. Moreover, for each $i, j \in \{1, \ldots, B_q\}$, these operators satisfy

$$X(F_i)Z(G_j) = (-1)^{1(i=j)} Z(G_j)X(F_i).$$

Finally, it holds that

$$FH_Z^T = GH_X^T = 0\,. \tag{1}$$

Proof. Eq. (1) follows since logical operators commute with the stabilizers. ◀

Finally we define appropriate unitary encoding and decoding maps. Regarding encoding, we provide a unitary encoding function based on the logical operators defined in Lemma 2.12.

▶ **Definition 2.13** (Encoding of a CSS code)**.** *Let C be an $[M, B_q, d_q]$ CSS code. Let H_X, H_Z be defined as in Definition 2.10. Let G be defined as in Lemma 2.12.*

For each $a \in \mathbb{F}_2^{B_q}$, let

$$|\varphi_a\rangle = \prod_{j \in [k]} \frac{I + (-1)^{a_j} Z(G_j)}{2} \cdot \prod_j \frac{I + X((H_X)_j)}{2} \cdot \prod_j \frac{I + Z((H_Z)_j)}{2} |+\rangle^{\otimes M}\,.$$

Let the following unitary E_q be the encoding unitary of C:

$$E_q(|a\rangle\,|0\rangle^{\otimes(M-B_q)}) = \frac{|\varphi_a\rangle}{\| \, |\varphi_a\rangle \, \|}\,.$$

For convenience, we also write $E_q(|\varphi\rangle)$ to denote $E_q(|\varphi\rangle\,|0\rangle^{\otimes(M-B_q)})$.

▶ Remark 2.14. Note that the $|\varphi_a\rangle$ are orthogonal for different a, and $\| \, |\varphi_a\rangle \, \| > 0$. Thus E_q is a well-defined unitary.

The following decomposition of arbitrary states as linear combinations of codewords with Pauli errors will be convenient for our analysis.

▶ **Lemma 2.15** (Stabilizer decomposition, [17])**.** *Let $C = CSS(H_X, H_Z)$ be a stabilizer code on n qubits. Let*

$$\mathcal{S} = \Big\{ (a,b) \,\Big|\, a, b \in \mathbb{F}_2^n,\ \exists r, s \text{ s.t. } |a \vee b| \text{ is minimized, and } (a,b) \text{ is lex smallest among all } (a',b') \text{ with } H_X b' = r \text{ and } H_Z a' = s \Big\}\,.$$

Then any n-qubit state $|\psi\rangle$ has a decomposition of the form

$$|\psi\rangle = \sum_{(a,b) \in \mathcal{S}} \alpha_{a,b} X(a)Z(b)\,|\eta_{a,b}\rangle\ ,$$

where $\| \, |\eta_{a,b}\rangle \, \| = 1$, $|\eta_{a,b}\rangle \in C$, the terms $X(a)Z(b)\,|\eta_{a,b}\rangle$ are pairwise othogonal, and $\sum_{a,b} |\alpha_{a,b}|^2 = 1$.

Proof. Any n-qubit state $|\psi\rangle$ decomposes as

$$|\psi\rangle = \sum_{r,s} \alpha_{r,s} |\psi_{r,s}\rangle \ ,$$

where $|\psi_{r,s}\rangle$ is a joint eigenvector of all $X(x)$, $x \in H_X$ and $Z(z)$, $z \in H_Z$ whose associated eigenvalues are given by the vectors r, s. This is because these eigenspaces form a direct sum decomposition of the entire space. For any such r, s, if we let (a, b) be the pair associated to it as in the definition of $\mathcal{S}$, then $|\psi_{r,s}\rangle = X(a)Z(b)|\eta_{a,b}\rangle$ for some codeword $|\eta_{a,b}\rangle$. ◀

We then define the weight of a Pauli error.

▶ **Definition 2.16** (Weight of an error in stabilizer codes)**.** *Let $C = CSS(H_X, H_Z)$ be a stabilizer code on n qubits. For $a, b \in \mathbb{F}_2^n$, let $E = X(a)Z(b)$. Define the weight of E modulo the centralizer of the code generating set as:*

$$wt_{cent}(E) = \min_{\substack{a',b' \in \mathbb{F}_2^n, \\ H_X b' = 0, H_Z a' = 0}} \left\{ \ wt(X(a')Z(b')E) \ \right\} .$$

▶ Remark 2.17. Note that, for $(a, b) \in \mathcal{S}$, $wt_{\text{cent}}(X(a)Z(b)) = |a \vee b|$.

We end with the decoding operator for CSS codes, which will be used to map an encoded quantum state back to its logical components and syndrome information. Note that this decoding map, as defined, is not necessarily efficient; and we will only use it for purposes of analysis. In Section 2.4, we discuss efficient maps associated with more restricted forms of decoding.

▶ **Definition 2.18** (Decoding qLTC code)**.** *Let H_X, H_Z be defined as in Definition 2.20. Let $\mathcal{S}$ be defined as in Lemma 2.15. Any n-qubit quantum state $|\psi\rangle$ decomposes as*

$$|\psi\rangle = \sum_{(a,b)\in\mathcal{S}} \alpha_{a,b} X(a)Z(b)|\eta_{a,b}\rangle \, ,$$

where $\| \, |\eta_{a,b}\rangle \, \| = 1$ and $|\eta_{a,b}\rangle \in C$. Let D_q be the following unitary:

$$D_q(|\psi\rangle) = \sum_{(a,b)\in\mathcal{S}} \alpha_{a,b} \left|\eta'_{a,b}\right\rangle |H_Z a\rangle \, |H_X b\rangle \, ,$$

where $\eta_{a,b} = E(\eta'_{a,b})$ and E is the qLTC encoding of $\eta'_{a,b}$.

▶ Remark 2.19. By Lemma 2.15, D_q is well-defined and is indeed a unitary. Note that D_q does not necessarily need to be efficient.

2.3 Quantum locally testable codes

We then introduce the standard definition of quantum locally testable codes.

▶ **Definition 2.20** (Local testability for quantum CSS codes, [13])**.** *Let $C = \text{CSS}(H_X, H_Z)$ be a quantum CSS code. We say that C is* locally testable *with soundness ρ if both $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$ are classical locally testable codes with soundness at least ρ.*

Next, we present a known construction of good quantum locally testable codes.

▶ **Theorem 2.21** (Quantum locally testable codes, [13])**.** *There exists a polynomial-time algorithm that, given an integer N (in unary), returns explicit descriptions of parity-check matrices H_X and H_Z on $\Theta(N)$ qubits such that the following properties hold:*

1. *The weight of each row of H_X and H_Z is $\Theta(1)$;*
2. *The code $C = \text{CSS}(H_X, H_Z)$ has dimension $\Theta(N)$ and distance $\Theta(\frac{N}{(\log N)^3})$;*
3. *C is a quantum locally testable code (qLTC) with soundness parameter $\rho = \Omega(\frac{1}{(\log N)^3})$.*

2.4 Decoding the X- and Z-codes of a CSS code

As stated in Theorem 2.21, we use the quantum LTC code based on cubical complexes from [13]. This code has an efficient decoder that generalizes the small-set bit-flip decoder, as adapted to quantum LDPC code construction in [12] and extended to the qLTC construction from [13] in [32]. As stated in [32, Theorem 3.3] the decoder runs in linear time and correctly decodes errors up to weight $\Theta(N/\mathrm{poly}\log(N))$, where N is the blocklength of the code.

For this work we need an additional property of the decoder, which is that if $e \in \{0,1\}^M$ is any "error" such that the associated syndrome in basis W, for $W \in \{X, Z\}$, is small, i.e. $\sigma = H_W e$ has small Hamming weight, then the decoder on input σ (deterministically) returns e' such that

$$|e + e'| \leq \mathrm{poly}\log(M) \cdot |\sigma| \quad \text{and} \quad H_W e' = \sigma \,. \tag{2}$$

Note that in general this property is not automatic, as an arbitrary decoder only needs to guarantee that e' has the correct syndrome σ; but e' may differ from e by an arbitrary element of $\mathrm{Im}(H_{W'}^T)$, where W' is the complementary basis to W. Such elements may a priori have arbitrary weight.

However, the decoder from [32] is a local decoder that progressively constructs an error e' such that $H_W e' = \sigma$ by finding $e_1, \ldots$, such that $|\sigma + H_W(e_1)| < \sigma$, etc. until no progress is found and one sets $e' = e_1 + \cdots$. Therefore, the number of iterations of the decoder is at most $|\sigma|$ and furthermore at each step only a "local" error is added, i.e. $|e_i| = O(\mathrm{poly}\log M)$ for each i. This is evident from the decoder description in Algorithm 3 and Algorithm 4 in [32] and establishes the desired property (2). We record it in the following fact:

▶ **Fact 2.22.** *Let C_q be the qLTC family from Theorem 2.21. Then C_q admits an efficient (in fact, linear time) decoder $\mathtt{D}_q$ for errors of weight up to $\eta_q d_q$, where d_q is the quantum distance of the code and η_q is a constant. Furthermore, let $\mathtt{D}_W$ be the decoder in basis $W \in \{X, Z\}$. We take this to be the map $\mathtt{D}_W : \mathbb{F}_2^M \to \mathbb{F}_2^{B_W}$ that to a noisy vector returns the "closest" element of the code H_W.* [12]

Then $\mathtt{D}_W$ is local in the sense that there exists $\eta_W = \Omega(1/\mathrm{poly}\log M)$ such that for any $e \in \{0,1\}^M$ such that $|e| < \eta_W d_{q,W}$ and any $z \in \{0,1\}^{B_W}$, then $|\mathtt{E}_W z + \mathtt{E}_W \mathtt{D}_W(E_W z + e)| \leq (1/\eta_W)|e|$.

For later use we define a partial composite decoding operator, which combines decoding results from different stages of the encoding process.

▶ **Definition 2.23** (Partial composite decoding operator)**.** *Let $M, m \in \mathbb{Z}_+$ such that m divides M. Let $\mathtt{D}$ denote the decoding algorithm from Theorem 2.1, and let $n(\cdot)$ be the length of the encoded message, as defined in Theorem 2.1. Let $W \in \{X, Z\}$. Let C_W be the linear code with parity-check matrix H_W from Theorem 2.21, with codeword length M, dimension B, and distance d. Let $\mathtt{D}_W$ denote its decoding algorithm.*

Let $e = (e_i)_{i \in [M/m]}$ be a tuple of encoded blocks, and let $x = (x_i)_{i \in [M/m]}$ be a tuple of plaintexts. Define the partial composite decoding function *as:*

$$\mathtt{D}_{p,W,i}(e, x) = \mathtt{D}_W(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_i), x_{i+1}, \ldots, x_{M/m}).$$

[12] Sometimes, the decoder is taken to be a map $\mathtt{D}'_W : \mathbb{F}_2^{c_W} \to \mathbb{F}_2^M$ that to a syndrome returns an associated error vector, where c_W would be the number of parity checks. With our notation, $\mathtt{D}_W(x) = (E_W)^{-1}(x + \mathtt{D}'_W(H_W x))$ where (provided the decoder succeeds) $x + \mathtt{D}'_W(H_W x) \in \ker H_W$ so it is indeed possible to efficiently find a unique preimage of it under the generator matrix E_W.

For $i \in \{0, \ldots, M/m\}$, $w \in \mathbb{F}_2^{M/m \cdot n(m)}$, $x \in \mathbb{F}_2^M$, and $r \in \mathbb{F}_2^B$, define the partial composite decoding operator as:

$$D_{p,W,i}(|r\rangle |w\rangle |x\rangle) = |r \oplus \mathsf{D}_{p,W,i}(w,x)\rangle |w\rangle |x\rangle .$$

Moreover, for $x \in \mathbb{F}_2^M$ and $r \in \mathbb{F}_2^B$, define the one-sided decoding operator:

$$D_W(|r\rangle |x\rangle) = |r \oplus \mathsf{D}_{p,W,0}(0,x)\rangle |x\rangle .$$

For $w \in \mathbb{F}_2^{M/m \cdot n(m)}$ and $r \in \mathbb{F}_2^B$, define the composite decoding operator:

$$D_{c,W}(|r\rangle |w\rangle) = \left|r \oplus \mathsf{D}_{p,W,M/m}(w,0)\right\rangle |w\rangle .$$

2.5 PCP of Proximity

In this section, we recall the notion of probabilistically checkable proofs of proximity (PCPPs), introduced in [5], and their application to the canonical $\mathcal{P}$-complete problem Circuit Value. These notions play an important role in our protocols.

▶ **Definition 2.24** ([5]). *We define the $\mathcal{P}$-complete language* Circuit Value *as* CktVal $= \{(C,w) : C(w) = 1\}$, *where C is a Boolean circuit and w is an input string. For a pair language $L \subseteq \{0,1\}^* \times \{0,1\}^*$, we define $L(x) = \{y : (x,y) \in L\}$.*

The following definition formalizes a proof system in which the verifier can efficiently check whether a given input is close to being valid, without reading the entire input.

▶ **Definition 2.25** (PCPs of Proximity (PCPPs), [5]). *Let $s, \delta : \mathbb{Z}^+ \to [0,1]$ be functions. A verifier V is said to be a probabilistically checkable proof of proximity (PCPP) system for a pair language L with proximity parameter δ and soundness error s if, for every pair of strings (x,y), the following conditions hold:*

- Completeness: *If $(x,y) \in L$, then there exists a proof string π such that $V(x)$ accepts the oracle $y \circ \pi$ with probability 1.*
- Soundness: *If y is $\delta(|x|)$-far from $L(x)$, then for every π, the verifier $V(x)$ accepts the oracle $y \circ \pi$ with probability strictly less than $s(|x|)$.*

If s and δ are not explicitly specified, they are assumed to be constants in the interval $(0,1)$.

The following theorem, due to [5], establishes that the Circuit Value problem admits a PCPP.

▶ **Theorem 2.26** ([5]). *There exists a constant $\delta_0 \in (0,1)$ such that for all $n \in \mathbb{Z}^+$ and all $\delta < \delta_0$, the language* Circuit Value *has a PCP of proximity (for circuits of size n) with the following parameters:*

- *randomness complexity $O(\log n)$,*
- *query complexity $O(1)$,*
- *perfect completeness, and*
- *soundness error $1 - \Omega(\delta)$ for proximity parameter δ.*

We now present an amplified version of the above theorem for ease of application:

▶ **Lemma 2.27.** *There exists a constant $\delta_0 \in (0,1)$ such that for all $n \in \mathbb{Z}^+$ and all $\delta < \delta_0$, the language* Circuit Value *has a PCP of proximity (for circuits of size n) with the following parameters:*

- *randomness complexity* $O(\log n)$,
- *query complexity* $O(\log(s)/\log(1-\Omega(\delta)))$,
- *perfect completeness, and*
- *soundness error* s *for proximity parameter* δ.

Proof. The lemma follows by repeatedly applying the PCPP verifier from Theorem 2.26 independently at least $N = \log(s)/\log(1-\Omega(\delta))$ times, and accepting if and only if all verifiers accept.

Note that for a no-instance, the acceptance probability satisfies

$$\Pr[\text{verifier accepts}] \leq (1-\Omega(\delta))^N \leq s.$$ ◀

We now define the quantum operator associated with generating a proof in a probabilistically checkable proof of proximity (PCPP).

▶ **Definition 2.28** (PCPP generating operator)**.** *Let* $s, l_x, l_y \in \mathbb{Z}_+$. *Let* $f : \mathbb{F}_2^{l_x} \times \mathbb{F}_2^{l_y} \to \mathbb{F}_2$ *be classical circuits of size* s. *For each* $x \in \mathbb{F}_2^{l_x}, y \in \mathbb{F}_2^{l_y}$, *let* $f_x(y) = f(x,y)$. *Let* $\mathtt{N}$ *be the classical algorithm that generates the proof for inputs to* CktVal *from Lemma 2.27. Suppose the output length of* $\mathtt{N}$ *on input of size* $\Theta(s)$ *is bounded by* $l_w(s)$. *For* $a \in \mathbb{F}_2^{l_w(s)}$, $x \in \mathbb{F}_2^{l_x}$, $y \in \mathbb{F}_2^{l_y}$, *define the operator*

$$N_f(|a\rangle|x\rangle|y\rangle) = |a \oplus \mathtt{N}(f_x, y)\rangle|x\rangle|y\rangle.$$

▶ Remark 2.29. In our protocol, x typically denotes the output of a circuit C on input y. We define f_x as the circuit $1(C(\cdot) = x)$. Thus, $(f_x, y) \in$ CktVal if and only if $C(y) = x$.

Finally, we define the quantum operator corresponding to verifying the PCPP proof.

▶ **Definition 2.30** (PCPP checking operator)**.** *Use the same notation as Definition 2.28. Let* $\mathtt{T}$ *be the verifier's algorithm from Lemma 2.27. Let the randomness complexity of* $\mathtt{T}$ *on instance size of* $\Theta(n)$ *be bounded by* $l_r(n)$.

For $a \in \mathbb{F}_2$, $x \in \mathbb{F}_2^{l_x}$, $y \in \mathbb{F}_2^{l_y}$, $w \in \mathbb{F}_2^{l_w(n)}$, *and* $r \in \mathbb{F}_2^{l_r(n)}$, *define*

$$T_f(|a\rangle|x\rangle|y\rangle|w\rangle|r\rangle) = |a \oplus \mathtt{T}(f_x, y, w; r)\rangle|x\rangle|y\rangle|w\rangle|r\rangle.$$

▶ Remark 2.31. In our protocol, r will be provided as classical randomness and thus does not consume quantum resources. However, for the sake of analysis, we prefer all quantum states to be represented as pure states.

3 Notational Conventions

In this section we summarize, for ease of reference, notation that is commonly used across the paper. It is advisable to skip the section at first read but refer to it whenever a notation used in a statement or proof is unclear.

3.1 General notation

For ℓ an integer we write $[\ell]$ for the set $[\ell] = \{1, \ldots, \ell\}$. For S a finite set, we write $s \in_R S$ for a uniformly random element s of S.

3.2 Codes

Our protocols rely on two different codes, a classical error-correcting code C_C and a quantum locally testable code C. In addition, we frequently consider the two codes C_X and C_Z that constitute C as a CSS code. We summarize notation used to discuss each code.

▶ **Notation 3.1** (Codes).

Conventions for classical code.

- *Let C_C be the error-correcting code defined in Theorem 2.1. Then C_C has codewords of length $n(k)$ on messages of length k, and it has relative distance Δ.*
- *Let $\mathtt{T}$ be the code tester of C_C.*
- *Let $\mathtt{D}$ be the decoding algorithm of C_C, which can decode any error of weight less than $\eta_C\Delta$; that is, for any $e \in \mathbb{F}_2^{n(k)}$ and $w \in C_C$ satisfying $|e| < \eta_C\Delta$, we have $\mathtt{D}(w+e) = \mathtt{D}(w)$.*

Convention for qLTC code.

- *C is the quantum LTC code from Theorem 2.21, instantiated such that the code has codeword length M. Let the code dimension be B_q, the distance d_q, and the locality q. C is locally testable with soundness ρ.*
- *Let E_q be the encoding unitary for C as defined in Definition 2.13. Let D_q be the decoding unitary as defined in Definition 2.18.*
- *Let $\mathcal{S}$ be defined as in Lemma 2.15.*

Convention for X/Z codes associated with the qLTC.

- *$W \in \{X, Z\}$ denotes a single-qubit observable. We often denote by W' the complementary observable, i.e. $W' = X$ if $W = Z$ and vice-versa.*
- *If $W = X$, let $S_W = H^{\otimes M}$. If $W = Z$, let $S_W = I$. When there is no ambiguity, we also use S to denote S_W.*
- *C_W is the classical linear code corresponding to the X-check matrix H_X when $W = X$, or the Z-check matrix H_Z when $W = Z$, as in Theorem 2.21. The code has word length M, dimension B_W. Let $d_{q,X}$ and $d_{q,Z}$ be as defined in Definition 2.11.*
- *Let $E_W \in \mathbb{F}_2^{M \times B_W}$ denote the generator matrix of the code C_W.*
- *Let $\mathtt{L}_W$ be the LTC code tester for C_W from Lemma 2.6, where the parameter ε is set to s. The tester has query complexity Q_W and soundness parameter κ_W. Define $Q = \max(Q_X, Q_Z)$ and $\kappa = \min(\kappa_X, \kappa_Z)$.*
- *Let $\mathtt{D}_W$ be the decoding algorithm for C_W. Note that $\mathtt{D}_W$ is efficient and a local decoder with locality parameter η_W in the sense of Fact 2.22. Let $\eta_q = \min(\eta_X, \eta_Z)$.*
- *Let D_W be defined as in Definition 2.23.*
- *Let F, G be defined as in Definition 2.12. Let $K_W = F$ if $W = X$, $K_W = G$ if $W = Z$. When there is no ambiguity, we also use K to denote K_W.*

3.3 PCPP

Regarding probabilistically chechable proofs of proximity, we use the following.

▶ **Notation 3.2** (PCPP).

- *$\mathtt{N}$ is the classical algorithm that generates the PCPP proof for inputs to* CktVal*, and N_f denotes its coherent application as in Definition 2.28 (on input a classical circuit $f : \mathbb{F}_2^{l_x} \times \mathbb{F}_2^{l_y} \to \mathbb{F}_2$).*
- *$\mathtt{T}$ is the PCPP verification algorithm from Lemma 2.27 and T_f denotes its coherent application as in Definition 2.30.*
- *Let δ and s denote the proximity parameter and soundness error from Lemma 2.27, respectively.*

3.4 Registers

Since the protocols we design can involve relatively large numbers of quantum registers, we strive to label them consistently using sans serif font as e.g. M, V, etc. In the protocols, the following registers will be commonly used.

▶ **Notation 3.3** (Registers).

- *$M, m \in \mathbb{Z}_+$ such that m divides M. The parameter m represents the size of a local block, and the witness is partitioned into $r = M/m$ blocks.*
- *P is a register of arbitrary dimension, initially held by the prover; it is the prover's private register.*
- *Let $\mathsf{W} = (\mathsf{W}_{i,j})_{i\in[M/m],\, j\in[m]}$ denote registers initially held by the verifier, containing the witness at the beginning of the protocol. Each $\mathsf{W}_{i,j}$ is a single-qubit register.*
- *Let $\mathsf{B} = (\mathsf{B}_{i,j})_{i\in[M/m],\, j\in[m]}$, where each $\mathsf{B}_{i,j}$ is an $l_w(\mathrm{poly}(m))$-qubit PCPP proof register. These registers hold proofs that the content of register E_i correctly decodes to the value stored in register X_i; each bit of X_i has a separate proof.*
- *Let $\mathsf{E} = (\mathsf{E}_i)_{i\in[M/m]}$ and $\mathsf{E}' = (\mathsf{E}'_i)_{i\in[M/m]}$, where each E_i, resp. E'_i, is an $n(m)$-qubit register intended to contain a codeword. Specifically, these registers store the classical ECC encoding of the content in W_i. The prover holds E'_i and returns E_i to the verifier.*
- *Let $\mathsf{X} = (\mathsf{X}_{i,j})_{i\in[M/m],\, j\in[m]}$ and $\mathsf{X}' = (\mathsf{X}'_{i,j})_{i\in[M/m],\, j\in[m]}$, where each $\mathsf{X}_{i,j}$, resp. $\mathsf{X}'_{i,j}$, is a single-qubit register. The registers X_i and X'_i are used to store copies (in the W basis) of the content in W_i during the global measurement extraction phase. The prover holds X'_i and returns X_i to the verifier.*

We often write W_i for the m-qubit register $(\mathsf{W}_{i,j})_{j\in[m]}$, and similarly for X_i and X'_i.

3.5 Operators

The following notation specifies unitary operators that play an important role in the proofs.

▶ **Notation 3.4** (Operators). *Fix a Boolean function $f : \{0,1\}^{B_q} \to \{0,1\}$. For $W \in \{X, Z\}$ let $W_f = \sum_x (-1)^{f(x)} S_W \left|x\right\rangle\!\left\langle x\right| S_W$.*

- *We label the register on which D_q acts as $\mathsf{W} = (\mathsf{L}, \mathsf{S})$, where L denotes the register holding the decoded quantum state, and S denotes the register holding the syndrome. Let*

$$W'_{f,W} = D^\dagger_{q,\mathsf{W}} (W_f)_{\mathsf{L}} D_{q,\mathsf{W}}.$$

When there is no ambiguity, we also use W'_f to denote $W'_{f,W}$.

- *Let T be a quantum register of B_W qubits. Let $\widetilde{f} : \{0,1\}^{B_W} \to \{0,1\}$ be defined by $\widetilde{f}(x) = f(KE_W x)$. Let*

$$W''_{f,W} = S^\dagger_{\mathsf{W}} D^\dagger_{W,\mathsf{TW}} (Z_{\widetilde{f}})_{\mathsf{T}} D_{W,\mathsf{TW}} S_{\mathsf{W}}.$$

When there is no ambiguity, we also use W''_f to denote $W''_{f,W}$.

- *Let T be a quantum register of B_W qubits. Let*

$$Q_{i,W} = S^\dagger_{\mathsf{W}} D^\dagger_{p,W,i,\mathsf{TEW}} (Z_{\widetilde{f}})_{\mathsf{T}} D_{p,W,i,\mathsf{TEW}} S_{\mathsf{W}}\ . \tag{3}$$

When there is no ambiguity, we also use Q_i to denote $Q_{i,W}$.

4 Formulation

We proceed with a formal definition of our model, which is a restricted variant of the qIOPs introduced in [33].

In the prior work, a general framework for quantum interactive oracle proofs (qIOPs) was proposed. Our result also fits within that framework, but our protocol exhibits a more structured and constrained form. We therefore define a restricted model, which we call a *Quantum Interactive Probabilistically Checkable Proof* (qIPCP) system. This name reflects the key characteristics of the model: the verifier begins the protocol holding a quantum witness and verifies it through limited interaction with a prover, making only a small number of queries to the prover's messages. Hence, the proof is checked in an interactive and probabilistic way.

More specifically: For YES instances, there exists a witness that is initially unentangled with the prover's private register, such that the verifier accepts with probability at least $2/3$. For NO instances, no matter what witness is used, even if it is initially entangled with the prover's private register, the verifier accepts with probability at most $1/3$.

Note that this model is equivalent to the one described in Section 1.1. In our formulation, for the malicious prover, the witness initially held by the verifier may be an arbitrary quantum state, potentially entangled with the prover's private register. Thus, this is equivalent to the prover sending a witness to the verifier at the start of the protocol.

We now describe the model more precisely. In each round, the verifier may send the prover a classical message and a subset of the quantum witness register. Importantly, once the verifier sends part of the quantum witness to the prover, he can no longer access it. This is because a malicious prover is not required to return the quantum state at a later stage (even though an honest prover would do so). Furthermore, we explicitly forbid the verifier from returning any of the prover's previous messages. Therefore, the only quantum messages the verifier can send are parts of the initial quantum witness.

Despite these strong restrictions, we demonstrate that the verifier can still verify QMA-complete languages within this model.

▶ **Definition 4.1** (Quantum interactive probabilistically checkable proof system)**.** *Let $n \in \mathbb{Z}_+$ denote the instance size, and let $m : \mathbb{Z}_+ \to \mathbb{Z}_+$ be a non-decreasing, even-valued function representing the total number of messages as a function of the instance size. Define $h(n) := m(n)/2$.*

An m-message quantum interactive probabilistically checkable proof system *(qIPCP) is specified by a tuple of quantum algorithms $\mathsf{V} = (\mathsf{V}_1, \mathsf{V}_2, \ldots, \mathsf{V}_{h+1})$.*

For each $i \in [h+1]$, the verifier's algorithm V_i takes x as input and acts on registers as follows:

$$\big((\mathsf{R}_{i-1}), (\mathsf{R}_1, \ldots, \mathsf{R}_{i-2}, \mathsf{W}_i, \ldots, \mathsf{W}_h)\big) \longrightarrow \big((\mathsf{W}_i, \mathsf{J}_i), (\mathsf{R}_1, \ldots, \mathsf{R}_{i-1}, \mathsf{W}_{i+1}, \ldots, \mathsf{W}_h)\big),$$

where

- R_i *is the quantum prover's message register, for each* $i \in \{-1, 0, \ldots, h\}$,
- W_i *is the quantum witness register, for each* $i \in \{1, \ldots, h+1\}$,
- J_i *is the verifier's classical message register, for each* $i \in \{1, \ldots, h+1\}$.
- R_{-1}, R_0, *and* W_{h+1} *are all empty registers.*
- *The register* J *is initialized to the all-zero classical string.*
- *The register* $\mathsf{W} = (\mathsf{W}_i)_{i \in [h]}$ *is initialized to a quantum witness for the input* x.

Each verifier algorithm V_i must be implementable in polynomial time.

Let $I = \{1, \ldots, h\}$. A prover strategy *is defined by a collection of quantum channels $\{\mathbb{P}_i\}_{i\in I}$ (possibly depending on x) and a private quantum register P initialized with an quantum state, where each $\mathbb{P}_i$ acts as:*

$$((\mathsf{W}_i, \mathsf{J}_i), \mathsf{P}) \longrightarrow (\mathsf{R}_i, \mathsf{P}).$$

The first (classical) bit of the register J_{h+1} in the output of V_{h+1} represents the verifier's final decision to accept or reject. Let $\omega(\mathsf{V}, |\varphi\rangle)$ denote the value of the interactive game defined by V_x when the initial state of the entire system is $|\varphi\rangle$. That is, $\omega(\mathsf{V}, |\varphi\rangle)$ is the maximum acceptance probability of V over all prover strategies $\{\mathbb{P}_i\}_{i\in I}$, where $|\varphi\rangle$ is defined on the joint quantum register WP.

Resource bounds. *We define the resource complexity of the system as follows:*

- ***Communication complexity:*** *The verifier V has communication complexity $l : \mathbb{Z}_+ \to \mathbb{Z}_+$ if*

$$l(n) \geq \sum_{i\in[h(n)]} l_i(n),$$

where $l_i(n)$ is the total size (in qubits) of the registers $\mathsf{R}_i, \mathsf{J}_i, \mathsf{W}_i$.

- ***Query complexity:*** *The verifier V has query complexity $q : \mathbb{Z}_+ \to \mathbb{Z}_+$ if*

$$q(n) \geq \sum_{i\in[h(n)+1]} q_i(n),$$

where $q_i(n)$ is the number of qubits accessed by V_i in the registers

$$(\mathsf{R}_1, \ldots, \mathsf{R}_{i-1}, \mathsf{W}_i, \ldots, \mathsf{W}_h).$$

We now formalize the notions of completeness and soundness for a QIPCP system verifying a promise problem.

▶ **Definition 4.2** (Quantum interactive probabilistically checkable proof systems for a promise problem)**.** *A quantum interactive probabilistically checkable proof system for a promise problem (S_{yes}, S_{no}) satisfies the following:*

Let $\omega(\mathsf{V}, |\varphi\rangle_{\mathsf{WP}})$ be defined as in Definition 4.1. The system must satisfy:

- Completeness: *There exists a constant $c \in (0, 1]$ such that for all $x \in S_{yes}$, the maximum acceptance probability satisfies $\max_{|\psi\rangle} \omega(\mathsf{V}, |\psi\rangle_{\mathsf{W}} |0\rangle_{\mathsf{P}}) \geq c$. That is, there exists a quantum witness $|\psi\rangle$ and an honest prover strategy under which the verifier accepts with probability at least c.*
- Soundness: *There exists a constant $s \in [0, c)$ such that for all $x \in S_{no}$, the maximum acceptance probability satisfies $\max_{|\varphi\rangle} \omega(\mathsf{V}, |\varphi\rangle_{\mathsf{WP}}) \leq s$. That is, no combination of prover strategy, prover's initial state, and jointly entangled quantum witness can cause the verifier to accept a no-instance with probability greater than s.*

We define the main complexity class studied in this paper.

▶ **Definition 4.3** (Quantum interactive probabilistically checkable proof)**.** *Let $r, l, q : \mathbb{Z}_+ \to \mathbb{Z}_+$. We define $\mathcal{QIPCP}(r, l, q)$ as the set of promise problems (S_{yes}, S_{no}) that have an r-message quantum interactive probabilistically checkable proof system with communication complexity l and query complexity q.*

5 QMA-Complete Problems

We introduce a variant of the local Hamiltonian problem where each term is diagonal in the X or Z basis respectively, and is furthermore not necessarily local but efficiently sampleable and computable. The QMA-completeness of this variant essentially follows from the results of [24] and [6].

We first introduce the Hamiltonian problem and then state its QMA-completeness. Before that, we define a basic term used in the Hamiltonian.

▶ **Definition 5.1** (X/Z projection). *For $W \in \{X, Z\}$ we say that a Hamiltonian (hermitian matrix) H is a W projection if and only if H acts non-trivially on a quantum register S of k qubits as $I + W_f/2$, where $f : \mathbb{F}_2^k \to \mathbb{F}_2$ is a boolean function.*[13] *More specifically,*

$$H = \left(\frac{I + W_f}{2}\right)_{\mathsf{S}} .$$

Next, we present the formal formulation of the Hamiltonian and define the corresponding promise problem associated with it.

▶ **Definition 5.2** (Dual-basis projector Hamiltonian). *A Hamiltonian $H \in \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}$ is a dual-basis projector Hamiltonian if it is the uniform average of a polynomial number $m = \mathrm{poly}(n)$ of X projection or Z projection terms H_i. That is,*

$$H = \frac{1}{m} \sum_{i \in [m]} H_i .$$

Moreover, H is k-local if each H_i acts non-trivially on at most k qubits.

▶ **Definition 5.3** (Local dual-basis projector Hamiltonian problem). *Let $k \in \mathbb{Z}^+$ and let $a, b : \mathbb{Z}^+ \to \mathbb{R}^+$ be non-decreasing functions such that $\forall n \in \mathbb{Z}^+$, $a(n) < b(n)$. The local dual-basis projector Hamiltonian problem $\mathcal{DLH}(k, a, b)$ is the promise problem (S_{yes}, S_{no}) where*

- *$S_{yes}(n)$ is the set of k-local dual-basis projector Hamiltonians $H \in \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}$ such that $\lambda_{\min}(H) \leq a(n)$,*
- *$S_{no}(n)$ is the set of k-local dual-basis projector Hamiltonians $H \in \mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}$ such that $\lambda_{\min}(H) \geq b(n)$.*

Finally, we state the QMA-completeness of the Hamiltonian problem. Note that this refers to the local Hamiltonian problem before amplification, where the promise gap is inverse-polynomial. The following follows immediately from [24].

▶ **Theorem 5.4.** *There exist polynomials p and q such that the problem*

$$\mathcal{DLH}(5, 2^{-p(n)}, 1/q(n))$$

is QMA-complete.

Proof. In [24], a somewhat incomparable result is shown, where $2^{-p(n)}$ above is replaced by 0, and the class characterized is QMA_1 (the variant of QMA with perfect completeness, meaning the verifier always accepts a yes-instance with probability 1). The present result follows using the same proof, noting that QMA is well-known to have verifiers with completeness exponentially close to 1 [26]. ◀

[13] Recall the notation $W_f = \sum_x (-1)^{f(x)} S_W |x\rangle\langle x| S_W$, with S_W a tensor product of I ($W = Z$) or H ($W = X$).

Then, we introduce the Hamiltonian amplification method from [6]. We first present their construction, followed by a statement of their result.

Essentially, the idea is to divide the Hamiltonian into commuting layers. In our case, there are two: the Z-layer, where each term is diagonal in the computational basis, and the X-layer, where each term is diagonal in the Hadamard basis. Each layer is then amplified separately by performing all tests corresponding to that layer simultaneously, and repeating the layer test sequentially t times.

▶ **Definition 5.5** (Amplified Hamiltonian, [6])**.** *Let H be a k-local Hamiltonian on n qubits expressed as an expectation over m projection terms,*

$$H = \frac{1}{m} \sum_{i \in [m]} \Pi_i.$$

where the local terms of H can be efficiently and equitably partitioned into $g = O(1)$ commuting layers, indexed by $\chi \in [g]$. We denote

$$H_\chi = \mathbb{E}_{i \in [m_\chi]} [\Pi_i^\chi], \quad w_\chi = \frac{m_\chi}{m} = \Omega(g^{-1}),$$

where w_χ represents the relative weight of clauses assigned to layer χ, $H = \mathbb{E}_\chi H_\chi$.

For any integer parameter $t \geq 1$, we define the amplified Hamiltonian *as*

$$H^{(t)} = I - \mathbb{E}_\chi \bigotimes_{j \in [t]} \Big(\prod_{i \in [m_\chi]} (I - \Pi_i^\chi) \Big), \tag{4}$$

where, $\mathbb{E}_\chi$ indicates sampling χ from the distribution $(w_1, w_2, \ldots, w_g)$ over $[g]$.

By applying this procedure, the minimum energy of the no-instance can be amplified until it exceeds a certain threshold.

▶ **Theorem 5.6** (Amplification from the DL, [6])**.** *Assume that the minimum eigenvalue of H satisfies $\lambda_{\min}(H) \geq \gamma$. Then, the minimum eigenvalue of the amplified Hamiltonian $H^{(t)}$ satisfies*

$$\lambda_{\min}\big(H^{(t)}\big) \geq \min \Big\{ \Theta(g^{-2}),\ \Omega\Big(\frac{t \cdot m \cdot \gamma}{g^4} \Big) \Big\}.$$

Finally, we apply the gap amplification theorem from [6] to the dual-basis projector Hamiltonian problem.

▶ **Corollary 5.7.** *There exists a polynomial p and a constant $c_{\mathrm{LH}} > 0$ such that the problem*

$$\mathcal{DLH}(\mathrm{poly}(n), 2^{-p(n)}, c_{\mathrm{LH}})$$

is QMA-complete.

Proof. The completeness follows from the fact that, within each layer, all projectors commute. The maximum eigenvalue $\lambda_{\max}(\prod_{i \in [m_\chi]} (I - \Pi_i^\chi))$ is lower bounded by $1 - \min_{|\varphi\rangle} \sum_i \langle\varphi| \Pi_i^\chi |\varphi\rangle$. In the yes-instance, there exists a state $|\varphi\rangle$ such that each $\langle\varphi| \Pi_i^\chi |\varphi\rangle \leq 1/\exp(n)$, since there are at most $\mathrm{poly}(n)$ such terms. Moreover, because each layer contains at most $\mathrm{poly}(n)$ terms, the completeness follows.

The soundness follows from Theorem 5.4 and Theorem 5.6. ◀

6 Evaluating Observables on Corrupted Codewords of a CSS Code

This section establishes some utility lemma on the structure of corrupted codewords of a quantum CSS code. The main lemma of the section is Lemma 6.6, which compares two different methods of evaluating a "logical" operation on a corrupted codeword.

We first formalize the notion of local testing for a quantum locally testable code (qLTC).

▶ **Definition 6.1** (Local testing of a qLTC code). *Let $C = CSS(H_X, H_Z)$ be a stabilizer code on n qubits, and let $|\varphi\rangle$ be a quantum state. The verifier performs the following steps c times:*

- *The verifier randomly samples $W \in \{X, Z\}$.*
- *The verifier randomly samples a row r_W from H_W.*
- *The verifier measures the observable $W(r_W)$ on the state $|\varphi\rangle$ and rejects if the measurement outcome is -1.*

If none of the tests reject, the verifier accepts.

▶ **Lemma 6.2.** *Let $C = CSS(H_X, H_Z)$ be a stabilizer code on n qubits such that $H_X \in \mathbb{F}_2^{c_X \times M}$, $H_Z \in \mathbb{F}_2^{c_Z \times M}$, are parity check matrices associated with classical locally testable codes with soundness ρ. Let $\mathcal{S}$ be defined as in Lemma 2.15. Let*

$$|\psi\rangle = \sum_{(a,b)\in\mathcal{S}} \alpha_{a,b} X(a)Z(b) |\eta_{a,b}\rangle$$

be a quantum state, where $\{|\eta_{a,b}\rangle\}$ are a family of codewords. Then the verifier in Definition 6.1 accepts $|\psi\rangle$ with probability at most

$$\Pr[\textit{verifier accepts}] \leq \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(1 - \frac{\rho}{2M} \cdot wt_{cent}(X(a)Z(b))\right)^c .$$

▶ Remark 6.3. A similar statement was first proposed in [3]. The result was further strengthened in [14], although no proof was provided. We strengthen the result even further.

Proof. Let the verifier in Definition 6.1 be denoted V. Suppose the sequence of observables measured by V is $(W_i(r_i))_{i\in[c]}$. Note that these observables pairwise commute. Thus

$$\begin{aligned}
\Pr\left[V \text{ accepts } \mid (W_i(r_i))_{i\in[c]}\right] &= \left\| \prod_{i\in[c]} \frac{I + W_i(r_i)}{2} |\varphi\rangle \right\|^2 \\
&= \langle\varphi| \prod_{i\in[c]} \frac{I + W_i(r_i)}{2} |\varphi\rangle \\
&= \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \langle\eta_{a,b}| Z(b)X(a) \\
&\qquad \cdot \Big(\prod_{i\in[c]} \frac{I + W_i(r_i)}{2} \Big) X(a)Z(b) |\eta_{a,b}\rangle \ .
\end{aligned}$$

Here, the second step follows from $(\frac{I+W_i(r_i)}{2})^2 = \frac{I+W_i(r_i)}{2}$, and the third step follows from the fact that the $X(a)Z(b)|\eta_{a,b}\rangle$ are orthonormal eigenvectors of the $W_i(r_i)$. Continuing, since the $|\eta_{a,b}\rangle$ are codewords,

$$\langle\eta_{a,b}| Z(b)X(a) \Big(\prod_{i\in[c]} \frac{I + W_i(r_i)}{2} \Big) X(a)Z(b) |\eta_{a,b}\rangle = \prod_{i\in[c]} 1_{[X(a)Z(b), W_i(r_i)]=0} \ .$$

Averaging over the verifier's choices of W and r_i,

$$\Pr[V \text{ accepts}] = \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(\begin{array}{l} \frac{1}{2} \cdot \mathop{\mathbb{E}}_{i\in_R[c_X]} 1_{[X(a)Z(b),X((H_X)_i)]=0} \\ + \frac{1}{2} \cdot \mathop{\mathbb{E}}_{i\in_R[c_Z]} 1_{[X(a)Z(b),Z((H_Z)_i)]=0} \end{array} \right)^c$$

$$= \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(\frac{c_X - wt(H_X b)}{2c_X} + \frac{c_Z - wt(H_Z a)}{2c_Z} \right)^c . \tag{5}$$

Let C_X be the locally testable code corresponding to the X-check matrix H_X. Let C_Z be the locally testable code corresponding to the Z-check matrix H_Z. By Definition 2.4,

$$wt(H_Z a) \geq \rho \cdot \frac{c_Z}{M} \cdot d_H(a, C_Z) ,$$

$$wt(H_X b) \geq \rho \cdot \frac{c_X}{M} \cdot d_H(b, C_X) .$$

Moreover, for any $W \in \{X, Z\}$ and $s \in \mathbb{F}_2^n$, let W' be the complementary basis of W, then

$$\begin{aligned} d_H(s, C_W) &= \min_{r\in\mathbb{F}_2^M, H_W\cdot r=0} \{wt(r+s)\} \\ &\geq \min_{a',b'\in\mathbb{F}_2^n, H_X b'=0, H_Z a'=0} \{wt(X(a')Z(b')(W')(s))\} \\ &= wt_{\text{cent}}(W'(s)) , \end{aligned}$$

and

$$wt_{\text{cent}}(X(a)) + wt_{\text{cent}}(Z(b)) \geq wt_{\text{cent}}(X(a)Z(b)) .$$

Plugging back into (5),

$$\begin{aligned} \Pr[V \text{ accepts}] &\leq \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(1 - \frac{\rho}{2M} \cdot d_H(b, C_X) - \frac{\rho}{2M} d_H(a, C_Z)\right)^c \\ &\leq \sum_{(a,b)\in\mathcal{S}} |\alpha_{a,b}|^2 \cdot \left(1 - \frac{\rho}{2M} \cdot wt_{\text{cent}}(X(a)Z(b))\right)^c . \end{aligned}$$ ◀

The next two lemmas, albeit simple, are used repeatedly throughout the proofs. In particular, they appear in the proof of the lemma that follows them.

▶ **Lemma 6.4.** *Recall the notation introduced in Section 3. For any $u, v \in C_W$, if $|u - v| < d_{q,W}$, then*

$$Ku = Kv.$$

Proof. Let W' be the complementary basis of W.

Since $u, v \in C_W$, it follows that $u - v \in C_W$. Moreover,

$$|u - v| < d_{q,W} = \min\{\, |w| : w \in \ker(H_W) \setminus \mathrm{Im}(H_{W'}^T) \,\},$$

implies that $u - v \in \mathrm{Im}(H_{W'}^T)$.

Because $KH_{W'}^T = 0$ by (1), we conclude that $K(u - v) = 0$, and hence $Ku = Kv$. ◀

▶ **Lemma 6.5.** *Recall the notation introduced in Section 3. Then, for any $p \in \mathbb{F}_2^M$ such that $|p| < \eta_W d_{q,W}$, then for any $v \in \mathbb{F}_2^{B_W}$, it holds that*

$$KE_W \, \mathtt{D}_W(p + E_W v) = KE_W v.$$

Proof. By Fact 2.22,

$$\left| \mathtt{E}_W v + \mathtt{E}_W \mathtt{D}_W (p + E_W v) \right| \leq \frac{1}{\eta_W} \cdot |p| < d_{q,W}.$$

Then, by Lemma 6.4,

$$K E_W \, \mathtt{D}_W (p + E_W v) = K E_W v. \qquad \blacktriangleleft$$

The following important lemma relates measurements obtained on the decoded quantum word (the observable W'), to measurements obtained directly on the encoded word but subsequently decoded according to the classical code (the observable W''). The lemma will be a key workhorse of soundness arguments in the following sections.

▶ **Lemma 6.6.** *Recall the notations introduced in Section 3. Let*

$$|\beta\rangle = \sum_{(a,b)\in\mathcal{S}} 1(wt_{cent}(X(a)Z(b)) < \eta_q d_q) \cdot \alpha_{a,b} X(a)Z(b) \, |\eta_{a,b}\rangle$$

be a quantum state, where $\{|\eta_{a,b}\rangle\}$ are codewords. Then, for any Boolean function $f : \{0,1\}^{B_q} \to \{0,1\}$,

$$W''_{\widetilde{f},W} \, |\beta\rangle = W'_{f,W} \, |\beta\rangle \, .$$

▶ Remark 6.7. Note that, for $v \in \mathbb{F}_2^{B_W}$, $\mathtt{D}_W(E_W v) = v$.

Proof.

Part I: analysis of the behavior of W'_f. Let $|\theta_{a,b}\rangle$ be such that $|\eta_{a,b}\rangle = E_q(|\theta_{a,b}\rangle)$. For $x \in \mathbb{F}_2^{B_q}$ let $|\varphi_x\rangle = E_q(|x\rangle)$ be as in Definition 2.13. Let $|\psi_x\rangle = \sum_{i\in\mathbb{F}_2^{B_q}} (-1)^{i\cdot x} |\varphi_i\rangle$ Let $f'_W : \{0,1\}^M \to \{0,1\}$ be $f'_W(x) = f(K_W x)$.

Note that in the definition of $|\varphi_x\rangle$ from Definition 2.13, the projection

$$\Pi_x = \prod_{j\in[k]} \frac{I + (-1)^{x_j} Z(G_j)}{2}$$

is diagonal in the computational basis. For any $y \in \mathbb{F}_2^M$ in the support of Π_x, we have $G_j y = x_j$. Therefore, $f'(y) = f(Gy) = f(x)$, and

$$\begin{aligned} &Z_{f'} E_q(|x\rangle) \\ &= (-1)^{f(x)} \cdot E_q(|x\rangle), \end{aligned}$$

and similarly,

$$\begin{aligned} &X_{f'} E_q(H\,|x\rangle) \\ &= (-1)^{f(x)} \cdot E_q(H\,|x\rangle). \end{aligned}$$

Using the definition of F and G,

$$Z_{f'} \, |\varphi_x\rangle = (-1)^{f(x)} \cdot E_q(|x\rangle) = E_q(Z_f \, |x\rangle) \, ,$$

and

$$X_{f'} \, |\psi_x\rangle = (-1)^{f(x)} \cdot E_q(H \, |x\rangle) = E_q(X_f H \, |x\rangle) \, ,$$

Expanding $|\eta_{a,b}\rangle$ in the codeword basis given by the $|\varphi_x\rangle$ or $|\psi_x\rangle$ it follows by linearity that

$$W_{f'}|\eta_{a,b}\rangle = E_q(W_f|\theta_{a,b}\rangle)\,.$$

By the definitions of d_q, W'_f, and D_q in Notation 3.1, we have, for $(a,b)\in\mathcal{S}$,

$$\begin{aligned} W'_f X(a)Z(b)|\eta_{a,b}\rangle &= D_q^\dagger W_f D_q X(a)Z(b)|\eta_{a,b}\rangle \\ &= D_q^\dagger W_f|\theta_{a,b}\rangle|H_Z a\rangle|H_X b\rangle \\ &= D_q^\dagger D_q(W_{f'}|\eta_{a,b}\rangle)|H_Z a\rangle|H_X b\rangle \\ &= X(a)Z(b)W_{f'}|\eta_{a,b}\rangle\,. \end{aligned} \tag{6}$$

Part II: analysis of the behavior of $W''_{w^T K E_W}$. Let $v\in\mathbb{F}_2^B$,

$$W''_{w^T K E_W} S|E_W v\rangle = (-1)^{w^T K E_W v} S|E_W v\rangle = W(w^T K)S|E_W v\rangle\,.$$

Therefore,

$$W''_{w^T K E_W}|\eta_{a,b}\rangle = W(w^T K)|\eta_{a,b}\rangle\,.$$

Moreover, we have that

$$\begin{aligned} &D^\dagger_{Z,\mathsf{TW}} Z_{\widetilde{f},\mathsf{T}} D_{Z,\mathsf{TW}} X(p)Z(q)|E_Z v\rangle \\ &= (-1)^{p^T q} D^\dagger_{Z,\mathsf{TW}} Z_{\widetilde{f},\mathsf{T}} D_{Z,\mathsf{TW}} Z(q)X(p)|E_Z v\rangle \\ &= (-1)^{p^T q} D^\dagger_{Z,\mathsf{TW}} Z_{\widetilde{f},\mathsf{T}} D_{Z,\mathsf{TW}} Z(q)|p+E_Z v\rangle \\ &= (-1)^{p^T q} Z(q) D^\dagger_{Z,\mathsf{TW}} Z_{\widetilde{f},\mathsf{T}} D_{Z,\mathsf{TW}}|p+E_Z v\rangle \\ &= (-1)^{p^T q + f(KE_Z \mathtt{D}_Z(p+E_Z v))} Z(q)|p+E_Z v\rangle \\ &= (-1)^{p^T q + f(KE_Z \mathtt{D}_Z(p+E_Z v))} Z(q)X(p)|E_Z v\rangle \\ &= (-1)^{f(KE_Z \mathtt{D}_Z(p+E_Z v))} X(p)Z(q)|E_Z v\rangle\,. \end{aligned}$$

Similarly, we have that

$$D^\dagger_{X,\mathsf{TW}} X_{\widetilde{f},\mathsf{T}} D_{X,\mathsf{TW}} X(p)Z(q)H|E_X v\rangle = (-1)^{f(KE_X \mathtt{D}_X(q+E_X v))} X(p)Z(q)H|E_X v\rangle\,.$$

Moreover,

$$\begin{aligned} &X(p)Z(q)Z_{f'}|E_Z v\rangle \\ &= (-1)^{f(KE_Z v)} X(p)Z(q)|E_Z v\rangle\,, \end{aligned}$$

and

$$\begin{aligned} &X(p)Z(q)X_{f'}H|E_X v\rangle \\ &= (-1)^{f(KE_X v)} X(p)Z(q)H|E_X v\rangle\,. \end{aligned}$$

Since $wt_{\mathrm{cent}}(X(a)Z(b)) < \eta_q d_q$, we have that $|a|,|b| < \eta_q d_q$. By Lemma 6.5,

$$GE_Z \mathtt{D}_Z(a+E_Z v) = GE_Z v\,, \qquad FE_X \mathtt{D}_X(b+E_X v) = FE_X v\,.$$

Therefore,

$$W''_{\widetilde{f}} X(a)Z(b)|\eta_{a,b}\rangle = X(a)Z(b)W_{f'}|\eta_{a,b}\rangle\,. \tag{7}$$

Part III: Putting things together. Combining (6) and (7),

$$W'_f X(a)Z(b)\,|\eta_{a,b}\rangle = W''_{\widetilde{f}} X(a)Z(b)\,|\eta_{a,b}\rangle\,,$$

and

$$W'_f\,|\beta\rangle = W''_{\widetilde{f}}\,|\beta\rangle\,. \qquad \blacktriangleleft$$

▶ **Lemma 6.8.** *Recall the notations introduced in Section 3. Let*

$$|\psi\rangle = \sum_{(a,b)\in\mathcal{S}} \alpha_{a,b} X(a)Z(b)\,|\eta_{a,b}\rangle$$

be a quantum state, where $\{|\eta_{a,b}\rangle\}$ *are a family of codewords. Then, for any boolean function* $f : \mathbb{F}_2^{B_q} \to \mathbb{F}_2$,

$$\|W''_{\widetilde{f}}\,|\psi\rangle - W'_f\,|\psi\rangle\,\|^2 \leq \Theta(1)\cdot \sum_{(a,b)\in\mathcal{S}} 1(wt_{cent}(X(a)Z(b)) \geq \eta_q d_q)\cdot |\alpha_{a,b}|^2\,.$$

Proof. This follows directly from Lemma 6.6. ◀

7 PCP of Proximity Applied to Quantum States

In this section we introduce and analyze a simple interaction between prover and verifier in which the prover is asked to compute a PCPP for the validity of a certain computation (abstracted through the function f below) when given an encoded input to the computation and a target output. This will be used as a building block later on.

7.1 Actions of the honest prover

In the following definition the prover is given registers U, B, and E. The register U represents the (target) output of a computation. The register B represents the witness for the verification of the computation, which the prover is to compute: initially, the register B is all $|0\rangle$. Finally, E is prepared as the encoded, according to a classical error correcting code, input of the computation.

Note that, to maintain the flexibility and generality of the results in this section, we do not adopt a specific classical code (e.g., the one defined in Notation 3.1); instead, we state the results for any classical error-correcting code C_P with appropriate parameters.

▶ Remark 7.1. In this section as in subsequent ones we describe the actions of a honest prover in a coherent form, i.e. replacing measurements by copying information, etc. This is to preserve the protocol's flexibility, e.g., the ability to generalize it to handle more complex observables. Any prover can be written in such a purified form, even though a dishonest prover may of course apply any unitaries they like.

▶ **Definition 7.2** (PCP of proximity for verifying a computation, honest prover)**.** *Recall the notations introduced in Section 3.*

Let C_P *be a classical error-correcting code with dimension* k*, codeword length* $n_P(k)$*, decoder* D_P*, code tester* T_P*, relative distance* Δ_P*, and* $\eta_P\Delta_P$ *the weight of errors that* D_P *can correct. That is, for any* $e \in \mathbb{F}_2^{n_P(k)}$ *and* $w \in C_P$ *satisfying* $|e| < \eta_P\Delta_P$*, we have* $D_P(w+e) = D_P(w)$*.*

Let U *be an* l_x*-qubit register denoting the output,* E *an* l_y*-qubit register denoting the encoded input, let* $f : \mathbb{F}_2^{n_P^{-1}(l_y)} \to \mathbb{F}_2^{l_x}$ *represent the verification of the computation. Let* $g : \mathbb{F}_2^{l_x} \times \mathbb{F}_2^{l_y} \to \mathbb{F}_2$ *be defined as*

$$g(x, y) = 1(f(D_P(y)) = x) \wedge T_P(y) .$$

Let s *be the size of the circuit* g*. Let* B *be a quantum register of* $l_w(s)$ *qubits. The honest prover performs the following steps:*

- *The prover computes an* $l_w(s)$*-bit long PCPP proof for a statement regarding the content of register* E*. This proof is stored in register* B*. Formally, the prover applies the following unitary:*

  $$N_{g,\mathsf{BUE}} .$$

- *The prover sends the verifier the quantum registers* B*.*

▶ Remark 7.3. Throughout the paper, C denotes either C_C itself or the code $C_C^{\oplus n}$, where each block corresponds to an independent codeword from the base code C_C.

7.2 Actions of the verifier

Before running the following protocol, the honest prover has already sent U, the register storing the computation output, to the verifier.

▶ **Definition 7.4** (PCP of proximity for verifying a computation, verifier)**.** *Use the notation in Definition 7.2. The verifier performs the following steps:*

- *The verifier executes the PCPP verifier on input* (g_x, y) *and stores the verification result in a single qubit register* C*, where* x *is the contents of register* U*,* y *is the contents of register* E *and the proof is in register* B*. The randomness is stored in register* R*. Formally, the verifier initialize* C *to* $|0\rangle$ *and* R *to* $l_r(s)$ *classical randomness bits, then the verifier applies*

  $$T_{g,\mathsf{CUEBR}}.$$

- *The verifier rejects if the PCPP verifier rejects. Formally, the verifier measures* C *and rejects if the outcome is* 0*.*

7.3 Main results

We now state the main lemma in this section. Informally, the lemma states that as long as the decoder of the classical ECC can decode errors smaller than the proximity parameter of the PCPP, the verifier's acceptance probability is at most the sum of the PCPP soundness parameter and the probability that a measurement of the whole quantum system yields a classical string that is a good instance. Note that we can obtain a PCPP verifier with improved soundness (while preserving the proximity parameter) by repeatedly applying the PCPP verifier, at the cost of increased query complexity.

▶ **Lemma 7.5** (Implication of the PCPP test)**.** *Let* s, δ *be defined as in Notation 3.2. Use the notation introduced in Definition 7.4.*

Let O *be a quantum register representing all registers in the system except for* U*,* E*, and* B*. Let* $|\varphi\rangle_{\mathsf{OUEB}}$ *be the quantum state of the system before the prover returns the quantum registers* B*. Let the state be written as*

$$|\varphi\rangle = \sum_{o,u,e,b} \alpha_{o,u,e,b} \, |o\rangle_{\mathsf{O}} \, |u\rangle_{\mathsf{U}} \, |e\rangle_{\mathsf{E}} \, |b\rangle_{\mathsf{B}} .$$

If $\eta_P \Delta_P \geq \delta$, then

- *The acceptance probability of the verifier is upper bounded by*

$$\Pr[\textit{PCPP test accepts}] \leq \max\left\{s, \sum |\alpha_{o,u,e,b}|^2 \cdot 1\Big(f(\mathtt{D}_P(e)) = u\Big)\right\}.$$

- *The rejection probability of the verifier is lower bounded by*

$$\Pr[\textit{PCPP test rejects}] \geq -s + \sum |\alpha_{o,u,e,b}|^2 \cdot 1\Big(f(\mathtt{D}_P(e)) \neq u\Big).$$

Proof. For $u \in \mathbb{F}_2^{l_x}$, define

$$S_u = \{e \in \mathbb{F}_2^{l_y} \mid 1(f(\mathtt{D}_P(e)) = u) \wedge \mathtt{T}_P(e) = 1\}.$$

We analyze the acceptance probability:

$$\Pr[\text{verifier accepts}] = \sum |\alpha_{o,u,e,b}|^2 \cdot \Pr[\text{verifier accepts } |o,u,e,b\rangle].$$

- If $\operatorname{dist}(e, S_u) > \delta$, then

$$\Pr[\text{PCPP accepts } u,e,b] \leq s.$$

- If $\operatorname{dist}(e, S_u) \leq \delta$, then

$$\Pr[\text{PCPP accepts } u,e,b] \leq 1 = 1\Big(f(\mathtt{D}_P(e)) = u\Big),$$

where the last step follows from

$$\delta < \eta_P \Delta_P.$$

As a result,

$$\begin{aligned}\Pr[\text{verifier accepts}] &\leq \sum |\alpha_{o,u,e,b}|^2 \cdot \max\left\{s, 1\Big(f(\mathtt{D}_P(e)) = u\Big)\right\} \\ &\leq \max\left\{s, \sum |\alpha_{o,u,e,b}|^2 \cdot 1\Big(f(\mathtt{D}_P(e)) = u\Big)\right\},\end{aligned}$$

and

$$\Pr[\text{verifier rejects}] \geq -s + \sum |\alpha_{o,u,e,b}|^2 \cdot 1\Big(f(\mathtt{D}_P(e)) \neq u\Big).$$ ◀

8 Global Measurement Extraction

In our protocol, the quantum proof is given as a qLTC encoding of a witness. The verifier first checks the correctness of the qLTC encoding. Ultimately, the witness used will be an amplified version of a witness for the QMA language instance provided as input. In the following, we present the general procedure for verification of the encoding, when applied to an arbitrary witness.

In the end, our goal is to measure observables of the form X_f or Z_f on the physical qubits. Since the procedures are very similar, we use the case of measuring Z_f as an example. To perform this measurement, the verifier must extract measurement outcomes in the Z basis. This overall process is referred to as *Global Measurement Extraction*. The Global Measurement Extraction procedure consists of repeatedly applying the *Single-Round Measurement Extraction* subroutine to successive portions of the qLTC-encoded state until all qubits are measured.

The Single-Round Measurement Extraction process works as follows: the verifier sends a small portion of the encoded witness, consisting of m qubits, to the prover. The only requirement is that this portion must be much smaller than the recoverable distance of the code, i.e. an error of weight at most m can be successfully corrected by the code decoder.

For concreteness, consider the case where the verifier sends the first m qubits. The prover is then asked to measure these m qubits in the Z basis, and to reply with:

- the measurement results themselves,
- a classical ECC encoding of the purified measurement results, and
- m different PCPPs, each proving that the corresponding bit of the ECC decoding matches the claimed measurement result.

The prover also retains a copy of the measurement outcomes.

Later in the protocol, the ECC encoding will be used as a proxy for the measurement outcomes on this portion of the qubits (as opposed to the measurement results themselves, also reported). Ideally, whenever the verifier wishes to access information about this portion, he would decode the ECC and compute based on the decoded information. However, decoding the ECC is not local. Fortunately, when trying to evaluate Z_f the verifier only needs access to information related to Z basis measurements. Therefore, instead of decoding the ECC, the verifier leverages classical PCPPs: he asks the prover to prove, via PCPPs, that any computation on the decoded results would yield the claimed outcome.

After receiving the prover's message, the verifier checks the Z part of the qLTC code. Specifically, the verifier randomly samples a Z-type stabilizer. Since stabilizers are local, they act on only a small number of qubits. For qubits that the verifier holds, he can measure them directly. For qubits that the prover already measured in the Z-basis, the verifier reads the reported bits and checks their correctness using the corresponding PCPP proofs. In particular, the prover must prove that the ECC decoding yields the correct value at the qubits involved in the stabilizer.

Note that the prover could introduce additional errors into the ECC-encoded portion during the Single-Round Measurement Extraction. Because Z (phase) errors do not affect Z basis measurement outcomes, we only need to worry about X (bit-flip) errors. The Z-LTC test performed after receiving the prover's message ensures that all X errors remain bounded to a small level.

The verifier then applies the Single-Round Measurement Extraction process to the next portion of the remaining qLTC-encoded witness. This process is repeated until all physical qubits have been measured in the Z-basis.

8.1 Actions of the honest prover

We now describe formally the basic procedure through which the prover extracts measurement results from a small portion of the quantum state. This process, which we refer to as *Single Round Measurement Extraction*, enables the prover to verifiably encode measurement outcomes for a small block of qubits. Even though in the protocol itself the prover is required to send back classical information, for purposes of modeling and analysis we will consider the actions of a purified prover, i.e. such that measurement outcomes are stored coherently in an outcome register. This register is later to be measured, as it is sent from prover to verifier.

Before we start, we introduce some notation that will be useful for presenting the prover's protocol.

▶ **Definition 8.1** (Controlled unitary). *Let* P *be a register and* U_{P} *a unitary acting on it. Let* R *be a one-qubit register. The controlled unitary* $C(U_{\mathsf{P}}, \mathsf{R})$ *applies* U_{P} *if the qubit in* R *is* $|1\rangle$*:*

$$C(U_{\mathsf{P}}, \mathsf{R}) = \begin{pmatrix} I_{\mathsf{P}} & 0 \\ 0 & U_{\mathsf{P}} \end{pmatrix} .$$

▶ **Definition 8.2** (Purified measurement). *Let* P, R *be registers and* O_{P} *an observable decomposed as* $O = \sum_{i=1}^{r} a_i P_i$*, where each* P_i *is an orthogonal projection and* $I = \sum_i P_i$*. The unitary that performs the purified measurement and stores the outcome in* R *is:*

$$P(O_{\mathsf{P}}, \mathsf{R}) = \sum_{i=1}^{r} P_{i,\mathsf{P}} \cdot (X(a_i))_{\mathsf{R}} \ .$$

The honest prover's actions in this procedure are defined below.

▶ **Definition 8.3** (Single round measurement extraction, honest prover). *Let* W *be a single-qubit observable. The honest prover performs the following steps:*

- *The verifier sends the prover a quantum register* W *consisting of* m *qubits.*
- *The prover measures the observable* W *m times, on each of the qubits of* W*, and stores two copies of each result, in single-qubit registers* X_i *and* X'_i *respectively.*
  Formally, the prover applies the following unitary:

$$\prod_{i\in[m]} P(W_{\mathsf{W}}(e_i), \mathsf{X}_i) P(W_{\mathsf{W}}(e_i), \mathsf{X}'_i) \ .$$

- *The prover encodes the* m*-qubit registers* X *and* X' *into* $n(m)$*-qubit registers* E *and* E' *respectively using the classical code from Theorem 2.1.*
  Formally, the prover applies the unitaries E_{EX}*,* $E_{\mathsf{E'X}}$ *where* E *is the encoding operator from Definition 2.3.*
- *For each* $i \in [m]$ *the prover computes a PCPP proof for the statement that the content of register* E *is a valid codeword such that the* i*-th bit of the encoded message matches the contents of register* X_i*. This proof is stored in register* B_i*.*
  Formally, for each $i \in [m]$*, the prover runs the procedure in Definition 7.2 with* $C = C_C, f(x) = (x)_i, \mathsf{U} = \mathsf{X}_i, \mathsf{E} = \mathsf{E}, \mathsf{B} = \mathsf{B}_i$*.*
- *Finally, the prover sends the verifier the quantum registers* WXE*.*

▶ Remark 8.4. Note that the register B is sent to the verifier during the procedure in Definition 7.2.

We extend the single round measurement extraction procedure to a global measurement extraction procedure. The global measurement extraction process involves repeating the single round measurement extraction across multiple disjoint portions of the qLTC encoded state. We formally define the honest prover's actions for this procedure below.

▶ **Definition 8.5** (Global measurement extraction, honest prover). *Recall the notations introduced in Section 3. The honest prover performs the following steps, for* $i = 1, \ldots, M/m$*:*

- *The prover executes the actions described in Definition 8.3, using* $W = W, \mathsf{B} = \mathsf{B}_i, \mathsf{E} = \mathsf{E}_i, \mathsf{E}' = \mathsf{E}'_i, \mathsf{X} = \mathsf{X}_i, \mathsf{X}' = \mathsf{X}'_i$*.*

8.2 Actions of the verifier

We now describe the verifier's actions in the single round measurement extraction process, corresponding to the honest prover's strategy previously defined.

▶ **Definition 8.6** (Single round measurement extraction, verifier). *Let* W *be a single-qubit observable. The verifier performs the following steps:*

- *The verifier sends the prover a quantum register* W *consisting of* m *qubits.*
- *The prover sends back the quantum registers* WXEB*, where* $\mathsf{B} = (\mathsf{B}_1, \ldots, \mathsf{B}_m)$*.*

We now describe the subroutine used by the verifier to check the validity of a partially extracted quantum state against a qLTC code. This procedure ensures that the measurement results encoded in the prover's messages are consistent with the structure of the code.

▶ **Definition 8.7** (Validity check, verifier). *Recall the notations introduced in Section 3. In addition, let $I \subseteq [M]$ such that, for every $i \in [M/m]$, either $((i-1)\cdot m, i\cdot m] \subseteq I$ or $((i-1)\cdot m, i\cdot m] \subseteq [M]\setminus I$ holds.*

The verifier performs the following steps:

- *The verifier samples queries $(j_1, \ldots, j_Q) \in [M]^Q$ for the code tester L_W, where Q is the query complexity.*
- *The verifier initializes a single-qubit register T' and a Q-qubit register T in state $|0\rangle$.*
- *For each $k \in [Q]$, the verifier performs the following.*
  If $j_k \in I$, the verifier directly copies register W_{j_k} in register T_k. Formally, the verifier applies $C(\mathsf{T}_k, \mathsf{W}_{j_k})$.
  If $j_k \in [M]\setminus I$, the verifier "extracts" W_{j_k} from registers provided by the prover, as follows:
  - *Let (p, q) be such that j_k is the q-th element of the p-th block of m qubits among $\mathsf{W}_1, \ldots, \mathsf{W}_M$.*
  - *The verifier runs the procedure in Definition 7.4 with $C = C_C, f(x) = (x)_q, \mathsf{U} = \mathsf{X}_{p,q}, \mathsf{E} = \mathsf{E}_p, \mathsf{B} = \mathsf{B}_{p,q}$. The verifier rejects if this procedure rejects.*
  - *The verifier copies register $\mathsf{X}_{p,q}$ in register T_k. Formally, the verifier applies $C(\mathsf{T}_k, \mathsf{X}_{p,q})$.*
- *The verifier rejects if the bits $(\mathsf{T}_1, \ldots, \mathsf{T}_Q)$ are rejected by the code tester L. Otherwise, it accepts.*
  Formally, the verifier applies the unitary $L_{\mathsf{T}'\mathsf{T},r}$, where L is the local tester operator as in Definition 2.7, measures T' and rejects if the outcome is 0.

Before defining the full global measurement extraction procedure for the verifier, we recall that this process allows the verifier to sequentially extract measurement outcomes from different parts of the qLTC-encoded state, while enforcing local consistency conditions verified through PCPP checks. We now describe the detailed steps of the verifier's actions.

▶ **Definition 8.8** (Global measurement extraction, verifier). *Recall the notations introduced in Section 3. Let $s \in \mathbb{F}_2^{n+1}$.*

- *If $s_0 = 1$, the verifier first runs the local tester of the qLTC on the quantum state in register W and rejects if the test fails. More specifically, the verifier runs the procedure from Definition 8.7 with parameters $I = [M]$, $W = W$, $\mathsf{W} = \mathsf{W}$, $\mathsf{B} = \mathsf{B}$, $\mathsf{E} = \mathsf{E}$, and $\mathsf{X} = \mathsf{X}$, and rejects if this procedure rejects.*
- *The verifier performs the following steps, repeating the actions described in Definition 8.6 a total of $r = M/m$ times. In the i-th iteration:*
  - *The verifier applies the procedure from Definition 8.6 using $W = W, \mathsf{W} = \mathsf{W}_i$, $\mathsf{B} = \mathsf{B}_i$, $\mathsf{E} = \mathsf{E}_i$, and $\mathsf{X} = \mathsf{X}_i$.*
  - *If $s_i = 1$, the verifier runs the procedure from Definition 8.7 with parameters $I = (i\cdot m, M]$, $W = W$, $\mathsf{W} = \mathsf{W}$, $\mathsf{B} = \mathsf{B}$, $\mathsf{E} = \mathsf{E}$, and $\mathsf{X} = \mathsf{X}$, and rejects if this procedure rejects.*

8.3 Main results

▶ **Lemma 8.9** (Implication of the validity test). *Fix $W \in \{X, W\}$. Recall the notations introduced in Section 3. Fix notation as in Definition 8.7. Let*

$$|\varphi\rangle_{\mathsf{WBEXP}} = \sum_{p,x,e,b,w} \alpha_{p,x,e,b,w} \, |p\rangle_{\mathsf{P}} \, |x\rangle_{\mathsf{X}} \, |e\rangle_{\mathsf{E}} \, |b\rangle_{\mathsf{B}} \, (S \, |w\rangle_{\mathsf{W}})$$

be the quantum state of the system before the test defined in Definition 8.7.

For $x = (x_i)_{i\in[M/m]}$, $e = (e_i)_{i\in[M/m]}$, $b = (b_i)_{i\in[M/m]}$, and $w = (w_i)_{i\in[M/m]}$, define for each $i \in [M/m]$:

- *If $((i-1)\cdot m, i\cdot m] \subseteq I$, let $r_i = w_i$;*
- *If $((i-1)\cdot m, i\cdot m] \subseteq [M]\setminus I$, let $r_i = \mathsf{D}(e_i)$.*

Suppose that $\delta < \eta_C\Delta$, then the rejection probability of the verifier from Definition 8.7 satisfies

$$\Pr[\textit{verifier rejects in the validity test}] \geq \kappa \cdot \sum |\alpha_{p,x,e,b,w}|^2 \cdot \frac{dist(r, C_W)}{M} - \Theta(s) \ .$$

Proof. Define for each $i \in [M/m]$:

- If $((i-1)\cdot m, i\cdot m] \subseteq I$, set $o_i = w_i$;
- If $((i-1)\cdot m, i\cdot m] \subseteq [M]\setminus I$, set $o_i = x_i$.

Fix a query $j_k \in [M]\setminus I$. Let $p = \lceil j_k/m\rceil$ and $q = j_k - m(p-1)$. By applying Lemma 7.5 with $C_P = C_C$ and $\delta < \eta_C\Delta$, we have that

$$\Pr[\text{PCPP accepts } x_{p,q}, e_p, b_{p,q}] \leq \max\Big\{1\big((r_p)_q = (o_p)_q\big),\ s\Big\}.$$

Then,

$$\begin{aligned}\Pr[\text{each PCPP accepts} \mid j_1,\dots,j_Q] &\leq \prod_{k=1}^{Q} \max\Big\{1(r_{j_k} = o_{j_k}),\ s\Big\} \\ &\leq \prod_{k=1}^{Q} 1(r_{j_k} = o_{j_k}) + s \ .\end{aligned}$$

Writing $j = (j_1,\dots,j_Q)$ for succinctness,

$$\begin{aligned}&\Pr[\text{verifier accepts } |p,x,e,b\rangle\, S\, |w\rangle] \\ &= \sum_j \Pr[j]\cdot\Pr[\text{local tester of } C_W \text{ accepts } o \mid j]\cdot\Pr[\text{each PCPP accepts} \mid j] \\ &\leq \sum_j \Pr[j]\cdot\Pr[\text{local tester of } C_W \text{ accepts } o \mid j]\cdot\Big(\prod_k 1(o_{j_k} = r_{j_k}) + s\Big) \\ &\leq \sum_j \Pr[j]\cdot\Pr[\text{local tester of } C_W \text{ accepts } r \mid j] + s \\ &\leq \Pr[\text{local tester of } C_W \text{ accepts } r] + s \\ &\leq 1 - \kappa\cdot\mathrm{dist}(r, C_W) + \Theta(s) \ .\end{aligned}$$

Therefore,

$$\begin{aligned}\Pr[\text{verifier rejects}] &= \sum |\alpha_{p,x,e,b,w}|^2 \cdot \Pr[\text{verifier rejects } |p,x,e,b\rangle\, S\, |w\rangle] \\ &\geq \sum |\alpha_{p,x,e,b,w}|^2 \cdot \Big(\kappa\cdot\frac{\mathrm{dist}(r,C_W)}{M} - \Theta(s)\Big) \\ &\geq \kappa\cdot\sum |\alpha_{p,x,e,b,w}|^2\cdot\frac{\mathrm{dist}(r,C_W)}{M} - \Theta(s) \ .\end{aligned}$$

◀

▶ **Lemma 8.10** (Implication of the single round measurement extraction)**.** *Recall the notations introduced in Section 3. Fix notation as in Definition 8.8. Let*

$$|\varphi\rangle = \sum_{p,x,e,b,w} \alpha_{p,x,e,b,w}\, |p\rangle_{\mathsf{P}}\, |x\rangle_{\mathsf{X}}\, |e\rangle_{\mathsf{E}}\, |b\rangle_{\mathsf{B}}\, S\, |w\rangle_{\mathsf{W}} \ ,$$

$$|\varphi'\rangle = \sum_{p,x,e,b,w} \alpha'_{p,x,e,b,w}\, |p\rangle_{\mathsf{P}}\, |x\rangle_{\mathsf{X}}\, |e\rangle_{\mathsf{E}}\, |b\rangle_{\mathsf{B}}\, S\, |w\rangle_{\mathsf{W}}$$

be the quantum state of the system respectively before and after the ith iteration of the single round measurement extraction in the global measurement extraction process introduced in Definition 8.8.

Let U_i be the global unitary that corresponds to the verifier and the prover's joint actions during the ith round of the interaction. For $x = (x_i)_{i\in[M/m]}$, $e = (e_i)_{i\in[M/m]}$, $b = (b_i)_{i\in[M/m]}$, and $w = (w_i)_{i\in[M/m]}$, define for each $j \in [M/m]$:

- *If $j \geq i$, let $r_j = w_j$;*
- *If $j < i$, let $r_j = D(e_j)$.*

Then,

$$\begin{aligned}&\|U_i Q_{i-1}(|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_i U_i(|0\rangle_{\mathsf{T}} |\varphi\rangle)\|^2 \\ &\leq \Theta(1) \cdot \Big(\sum_{dist(r,C_W)\geq \eta_q d_{q,W} - m} |\alpha_{p,x,e,b,w}|^2 + \sum_{dist(r,C_W)\geq \eta_q d_{q,W} - m} |\alpha'_{p,x,e,b,w}|^2 \Big) .\end{aligned}$$

Proof. Let

$$|\beta\rangle = \sum_{\mathrm{dist}(r,C_W)<\eta_q d_{q,W} - m} \alpha_{p,x,e,b,w} |p\rangle_{\mathsf{P}} |x\rangle_{\mathsf{X}} |e\rangle_{\mathsf{E}} |b\rangle_{\mathsf{B}} (S|w\rangle_{\mathsf{W}})$$

and

$$|\beta'\rangle = \sum_{\mathrm{dist}(r,C_W)<\eta_q d_{q,W} - m} \alpha'_{p,x,e,b,w} |p\rangle_{\mathsf{P}} |x\rangle_{\mathsf{X}} |e\rangle_{\mathsf{E}} |b\rangle_{\mathsf{B}} (S|w\rangle_{\mathsf{W}}) .$$

Let $\mathsf{E} = (\mathsf{E}_i)_{i\in[M/m]}$ and $\mathsf{W} = (\mathsf{W}_i)_{i\in[M/m]}$. Let W_0 be a quantum register of m qubits. Let E_0 be a quantum register of $n(m)$ qubits. Let $\mathsf{E}' = (\mathsf{E}_1, \ldots, \mathsf{E}_{i-1}, \mathsf{E}_0, \mathsf{E}_{i+1}, \ldots, \mathsf{E}_{M/m})$, $\mathsf{W}' = (\mathsf{W}_1, \ldots, \mathsf{W}_{i-1}, \mathsf{W}_0, \mathsf{W}_{i+1}, \ldots, \mathsf{W}_{M/m})$. Recall the partial composite decoding operator $D_{p,W,i}$ from Definition 2.23, and let

$$Q'_i = S^\dagger_{\mathsf{W}'} D^\dagger_{p,W,i,\mathsf{TE}'\mathsf{W}'} Z_{\widetilde{f},\mathsf{T}} D_{p,W,i,\mathsf{TE}'\mathsf{W}'} S_{\mathsf{W}'} ,$$

and

$$Q'_{i-1} = S^\dagger_{\mathsf{W}'} D^\dagger_{p,W,i-1,\mathsf{TE}'\mathsf{W}'} Z_{\widetilde{f},\mathsf{T}} D_{p,W,i-1,\mathsf{TE}'\mathsf{W}'} S_{\mathsf{W}'} .$$

By definition, for any quantum state $|\psi\rangle$ on registers $\mathsf{E}'_{-i}\mathsf{W}'_{-i}$:

$$Q'_{i-1}(|0\rangle_{\mathsf{TE}_0\mathsf{W}_0} |\psi\rangle) = Q'_i(|0\rangle_{\mathsf{TE}_0\mathsf{W}_0} |\psi\rangle) .$$

This is because $D_{p,W,i-1,\mathsf{TE}'\mathsf{W}'}$ in Q'_{i-1} uses the decoding of the first $i-1$ blocks of E' and the last $M/m - i + 1$ blocks of W', while $D_{p,W,i,\mathsf{TE}'\mathsf{W}'}$ in Q'_i uses the decoding of the first i blocks of E' and the last $M/m - i$ blocks of W'. However, both the ith block of E' and the ith block of W' are set to $|0\rangle$, so the two operators Q'_{i-1} and Q'_i act identically.

Note that

$$\begin{aligned}&Q_{i-1}(|0\rangle_{\mathsf{TE}_0\mathsf{W}_0} |\beta\rangle) \\ &= Q'_{i-1}(|0\rangle_{\mathsf{TE}_0\mathsf{W}_0} |\beta\rangle).\end{aligned}$$

This follows since $|\beta\rangle$ is supported on r such that $\mathrm{dist}(r, C_W) < \eta_q d_{q,W} - m$ and Lemma 6.5. More precisely, the difference between Q_{i-1} and Q'_{i-1} is that the former uses the ith block of W, while the latter replaces that block with $|0\rangle$. Decompose the state in the computational basis; then, applying the decoding to the first $i-1$ blocks of E (respectively E') and combining

the last $M/m - i + 1$ blocks of W (respectively W') gives us a string r (respectively r'). For an honest prover, r is a codeword of C_W, and r' is obtained from r by setting the ith block to 0. As long as the two decoded strings from r and r' are equal, or their respective differences from the same codeword both lie within the tolerance required by Lemma 6.5, the actions of Q_{i-1} and Q'_{i-1} will be the same (since we only consider $\widetilde{f}(x)$ of the form $f(KE_W x)$). Indeed, $\mathrm{dist}(r, C_W) < \eta_q d_{q,W} - m$ implies the existence of $s \in C_W$ such that $\mathrm{dist}(r, s) < \eta_q d_{q,W} - m \leq \eta_q d_{q,W}$, and hence $\mathrm{dist}(r', s) < \eta_q d_{q,W}$.

Therefore,

$$\begin{aligned}
&U_i Q_{i-1}(|0\rangle_{\mathsf{TE_0W_0}} |\beta\rangle) \\
&= U_i Q'_{i-1}(|0\rangle_{\mathsf{TE_0W_0}} |\beta\rangle) \\
&= U_i Q'_i(|0\rangle_{\mathsf{TE_0W_0}} |\beta\rangle) \\
&= Q'_i U_i(|0\rangle_{\mathsf{TE_0W_0}} |\beta\rangle) \ ,
\end{aligned}$$

Furthermore,

$$Q'_i(|0\rangle_{\mathsf{TE_0W_0}} |\beta'\rangle) = Q_i(|0\rangle_{\mathsf{TE_0W_0}} |\beta'\rangle) \ ,$$

Moreover,

$$\| \, |\beta\rangle - |\varphi\rangle \, \|^2 \leq \sum_{\mathrm{dist}(r, C_W) \geq \eta_q d_{q,W} - m} |\alpha_{p,x,e,b,w}|^2 \ ,$$

and

$$\| \, |\beta'\rangle - |\varphi'\rangle \, \|^2 \leq \sum_{\mathrm{dist}(r, C_W) \geq \eta_q d_{q,W} - m} |\alpha'_{p,x,e,b,w}|^2 \ .$$

Note that

$$\begin{aligned}
&\|U_i \, |\beta\rangle - |\beta'\rangle \, \|^2 \\
&\leq \Theta(1) \cdot (\|U_i \, |\beta\rangle - U_i \, |\varphi\rangle \, \|^2 + \| \, |\varphi'\rangle - |\beta'\rangle \, \|^2) \\
&= \Theta(1) \cdot (\| \, |\beta\rangle - |\varphi\rangle \, \|^2 + \| \, |\varphi'\rangle - |\beta'\rangle \, \|^2) \ .
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\|U_i Q_{i-1}(|0\rangle_{\mathsf{TE_0W_0}} |\varphi\rangle) - Q_i U_i(|0\rangle_{\mathsf{TE_0W_0}} |\varphi\rangle)\|^2 \\
&\leq \Theta(1) \cdot (\| \, |\beta\rangle - |\varphi\rangle \, \|^2 + \| \, |\varphi'\rangle - |\beta'\rangle \, \|^2) \\
&\leq \Theta(1) \cdot \Big(\sum_{\mathrm{dist}(r, C_W) \geq \eta_q d_{q,W} - m} |\alpha_{p,x,e,b,w}|^2 + \sum_{\mathrm{dist}(r, C_W) \geq \eta_q d_{q,W} - m} |\alpha'_{p,x,e,b,w}|^2 \Big) \ .
\end{aligned}$$

◀

▶ **Lemma 8.11** (Implication of the global measurement extraction)**.** *Use the same notation as in Lemma 8.10, except that now*

$$|\varphi\rangle_{\mathsf{WBEXP}} = \sum_{p,x,e,b,w} \alpha_{p,x,e,b,w} \, |p\rangle_{\mathsf{P}} \, |x\rangle_{\mathsf{X}} \, |e\rangle_{\mathsf{E}} \, |b\rangle_{\mathsf{B}} \, (S \, |w\rangle_{\mathsf{W}})$$

is the quantum state of the system before the global measurement extraction process. In particular, let Q_i, for $i \in \{0, \ldots, n\}$, be as defined in (3), for some $W \in \{X, Z\}$. Suppose that for each i,

$$\Pr[\textit{verifier rejects in the validity test in the } i\textit{th iteration}] \leq \varepsilon_i \ .$$

Assume that $\delta < \eta_C \Delta$. Suppose that during the interaction, the verifier and prover together apply the unitary U_b to the entire system. Then:

$$\|U_b Q_0(|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_n U_b(|0\rangle_{\mathsf{T}} |\varphi\rangle)\|^2 \leq \Theta\Big(\frac{r^2 \cdot (s + \mathbb{E}_i\, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)}\Big)\,,$$

where $r = M/m$.

Proof. Let $|\varphi_k\rangle_{\mathsf{WBEXP}}$ be the quantum state of the system after the kth iteration of the single round measurement extraction in the global measurement extraction process. Let the state be written as

$$|\varphi_k\rangle = \sum_{p,x,e,b,w} \alpha_{k,p,x,e,b,w} |p\rangle_{\mathsf{P}} |x\rangle_{\mathsf{X}} |e\rangle_{\mathsf{E}} |b\rangle_{\mathsf{B}} (S|w\rangle_{\mathsf{W}})\,.$$

For $x = (x_i)_{i\in[M/m]}$, $e = (e_i)_{i\in[M/m]}$, $b = (b_i)_{i\in[M/m]}$, and $w = (w_i)_{i\in[M/m]}$, define for each $j \in [M/m]$:

- If $j \geq i$, let $r_j = w_j$;
- If $j < i$, let $r_j = \mathtt{D}(e_j)$.

By Lemma 8.9,

$$\begin{aligned}
&\Pr[\text{verifier rejects in the validity test in the } k\text{th iteration}] \\
&\geq \kappa \cdot \sum |\alpha_{k-1,p,x,e,b,w}|^2 \cdot \frac{\mathrm{dist}(r, C_W)}{M} - \Theta(s) \\
&\geq \kappa \cdot \sum_{\mathrm{dist}(r,C_W) \geq \eta_q d_{q,W} - m} |\alpha_{k-1,p,x,e,b,w}|^2 \cdot \frac{\eta_q d_{q,W} - m}{M} - \Theta(s).
\end{aligned}$$

Therefore,

$$\sum_{\mathrm{dist}(r,C_W) \geq \eta_q d_{q,W} - m} |\alpha_{k-1,p,x,e,b,w}|^2 \leq \Theta\Big(\frac{(s + \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)}\Big)\,. \tag{8}$$

By Lemma 8.10,

$$\begin{aligned}
&\|U_k Q_{k-1}(|0\rangle_{\mathsf{T}} |\varphi_{k-1}\rangle) - Q_k U_k(|0\rangle_{\mathsf{T}} |\varphi_{k-1}\rangle)\|^2 \\
&\leq \Theta(1) \cdot \Big(\sum_{\mathrm{dist}(r,C_W) \geq \eta_q d_{q,W} - m} |\alpha_{k-1,p,x,e,b,w}|^2 + \sum_{\mathrm{dist}(r_k,C_W) \geq \eta_q d_{q,W} - m} |\alpha_{k,p,x,e,b,w}|^2 \Big)\,.
\end{aligned}$$

Using the triangle inequality and plugging in (8),

$$\begin{aligned}
&\|U_b Q_0(|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_n U_b(|0\rangle_{\mathsf{T}} |\varphi\rangle)\| \\
&\leq \sum_{k\in[M/m]} \|U_k Q_{k-1}(|0\rangle_{\mathsf{T}} |\varphi_{k-1}\rangle) - Q_k U_k(|0\rangle_{\mathsf{T}} |\varphi_{k-1}\rangle)\| \\
&\leq r \cdot \sqrt{\Theta\Big(\frac{(s + \mathbb{E}_i\, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)}\Big)}.
\end{aligned}$$

Squaring gives the desired result:

$$\|U_b Q_0(|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_n U_b(|0\rangle_{\mathsf{T}} |\varphi\rangle)\|^2 \leq \Theta(\frac{r^2 \cdot (s + \mathbb{E}_i\, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)}).$$ ◀

9 Measurement in the Computational or Hadamard Basis

To measure observables of the form X_f or Z_f on the physical qubits, the procedures are very similar. We continue with the Z_f case as an example. After completing the Global Measurement Extraction process described in the previous section, the verifier has all the necessary information to evaluate any measurement in the Z basis. To measure Z_f, the verifier sends the description of f to the prover. The prover then uses the stored measurement outcomes from each portion and provides a PCPP proof demonstrating that the ECC-decoded outcomes, when evaluated by the function f, yield the claimed value.

9.1 Actions of the honest prover

We introduce a basic quantum operator that computes a function on the string stored in a quantum register.

▶ **Definition 9.1** (Function evaluation operator). *Let $M \in \mathbb{Z}_+$ and $f : \{0,1\}^M \to \{0,1\}$. For $a \in \mathbb{F}_2$ and $w \in \mathbb{F}_2^M$, the evaluation operator R_f acts as:*

$$R_f(|a\rangle |w\rangle) = |a \oplus f(w)\rangle |w\rangle ,$$

Before proceeding, we introduce a notation for a blockwise extension of the classical code C_C. This construction will be useful when describing the encoding structure of the prover's registers and the corresponding decoding procedures.

▶ **Definition 9.2.** *Using the notation from Notation 3.1, define code $C_{C,r} = \bigoplus_{i\in[r]} C_C$.*

We now describe how the honest prover performs measurements in either the computational basis (Z basis) or the Hadamard basis (X basis), depending on the verifier's instruction. This procedure builds on the previously defined global measurement extraction process.

▶ **Definition 9.3** (Measurement in the computational or Hadamard basis, honest prover). *Recall the notations introduced in Section 3.*

The honest prover performs the following steps:

- *The verifier sends the prover the description of a classical function $h : \{0,1\}^M \to \{0,1\}$.*
- *The prover applies the unitary $R_{h,\mathsf{IX}'}$.*
- *The prover runs the procedure in Definition 7.2 with $C = C_{C,r}, f = h, \mathsf{U} = \mathsf{I}, \mathsf{E} = \mathsf{E}', \mathsf{B} = \mathsf{A}$, where $C_{C,r}$ is defined as in Definition 9.2.*
- *The prover sends the verifier the quantum register I.*

▶ Remark 9.4. The function h will be specified as $f(KE_W \mathtt{D}_W(\cdot))$ in a later section, but we present a general protocol here for clarity.

9.2 Actions of the verifier

In the following, we describe the verifier's procedure for performing a global measurement in either the computational (Z) basis or the Hadamard (X) basis. Since the measurement outcomes have already been extracted, the protocol remains the same for both bases. However, the evaluation function h differs between the two cases, even for the same function f. This process can be viewed as a verification of function evaluation using a PCPP.

▶ **Definition 9.5** (Measurement in the computational or Hadamard basis, verifier). *Recall the notations introduced in Section 3 and Definition 9.3. Let $s \in \mathbb{F}_2$.*

The verifier performs the following steps:

- *The verifier sends the prover the description of a classical function* $h : \{0,1\}^M \to \{0,1\}$.
- *The prover sends back to the verifier the quantum registers* IA.
- *If* $s = 1$*, the verifier runs the procedure in Definition 7.4 with* $C = C_{C,n}, f = h, \mathsf{U} = \mathsf{I}, \mathsf{E} = \mathsf{E}', \mathsf{B} = \mathsf{A}$*, where* $C_{C,n}$ *is defined as in Definition 9.2. The verifier accepts if and only if this procedure accepts.*

9.3 Main results

▶ **Lemma 9.6** (Implication of the PCPP check in the measurement procedure)**.** *Recall the notations introduced in Section 3 and Definition 9.3. Let* $\mathsf{O} = (\mathsf{P}, \mathsf{W}, \mathsf{B}, \mathsf{X})$ *be a quantum register. Let* $|\varphi\rangle_{\mathsf{OIEA}}$ *be the quantum state of the system after the measurement in the computational or Hadamard basis process. Let the state be written as*

$$|\varphi\rangle = \sum_{o,i,e,a} \alpha_{o,i,e,a} |o\rangle_{\mathsf{O}} |i\rangle_{\mathsf{I}} |e\rangle_{\mathsf{E}} |a\rangle_{\mathsf{A}} .$$

Assume that $r\delta < \eta_C \Delta$*. Then the rejection probability of the verifier is lower bounded by*

$$\Pr[\textit{verifier rejects}] \geq -s + \sum |\alpha_{o,i,e,a}|^2 \cdot 1\Big(h(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_n)) \neq i\Big).$$

Proof. Let $C_{C,n}$ be defined as in Definition 9.2. Let $\mathtt{D}'_C$ denote the decoder of $C_{C,n}$, $\mathtt{T}'_C$ the code tester, Δ'_C its relative distance, and $\eta'_C \Delta'_C$ the weight of errors that $\mathtt{D}'_C$ can correct. Note that $\mathtt{D}'_C$ can be implemented by applying $\mathtt{D}$ independently to each block of the code, $\mathtt{T}'_C$ can be implemented by applying $\mathtt{T}$ independently to each block of the code. In this case, $\Delta'_C = \Delta/n$ and $\eta'_C = \eta_C$.

Note that the assumption $n\delta < \eta_C \Delta$ implies that $\delta < \eta'_C \Delta'_C$.

The proof follows from Lemma 7.5 with $\eta_P = \eta'_C, \Delta_P = \Delta'_C, \mathtt{D}_P = \mathtt{D}'_C, \mathtt{T}_P = \mathtt{T}'_C$ and $\delta < \eta'_C \Delta'_C$. ◀

▶ **Lemma 9.7** (Implication of the measurement in the computational or Hadamard basis)**.** *Recall the notations introduced in Section 3 and Definition 9.3. Let* $\mathsf{O} = (\mathsf{P}, \mathsf{W}, \mathsf{B}, \mathsf{X})$ *be a quantum register. Let* $|\varphi\rangle_{\mathsf{OIEA}}$ *be the quantum state of the system after the measurement in the computational or Hadamard basis process. Let the state be written as*

$$|\varphi\rangle = \sum_{o,i,e,a} \alpha_{o,i,e,a} |o\rangle_{\mathsf{O}} |i\rangle_{\mathsf{I}} |e\rangle_{\mathsf{E}} |a\rangle_{\mathsf{A}} .$$

If $h(x) = f(KE_W \mathcal{D}_W(x))$*, then:*

$$\| Z_{\mathsf{I}} |\varphi\rangle - Q_r |\varphi\rangle \|^2 \leq \Theta(1) \cdot \sum |\alpha_{o,i,e,a}|^2 \cdot 1\Big(h(\mathcal{D}(e_1), \ldots, \mathcal{D}(e_r)) \neq i\Big).$$

Proof. Let

$$|\beta\rangle = \sum_{o,i,e,a} 1\Big(h(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_r)) = i\Big) \cdot \alpha_{o,i,e,a} |o\rangle_{\mathsf{O}} |i\rangle_{\mathsf{I}} |e\rangle_{\mathsf{E}} |a\rangle_{\mathsf{A}} .$$

Note that

$$Z_{\mathsf{I}} |\beta\rangle = Q_r |\beta\rangle .$$

Therefore,

$$\begin{aligned} &\| Z_{\mathsf{I}} |\varphi\rangle - Q_r |\varphi\rangle \|^2 \\ &\leq \Theta(1) \cdot (\| |\varphi\rangle - |\beta\rangle \|^2 + \| Z_{\mathsf{I}} |\beta\rangle - Q_r |\beta\rangle \|^2) \\ &\leq \Theta(1) \cdot \sum |\alpha_{o,i,e,a}|^2 \cdot 1\Big(h(\mathtt{D}(e_1), \ldots, \mathtt{D}(e_r)) \neq i\Big). \end{aligned}$$

◀

10 QIPCP protocol for QMA

In this section, we describe our protocol for estimating the energy of a dual-basis projector Hamiltonian, and thereby distinguishing between the yes and no instances of the corresponding QMA-complete problem. This will establish our main result, restated as Theorem 10.3 below.

To estimate the energy, we sample a term according to its coefficient in the Hamiltonian (each term being a projector) and then project the witness state according to that projector by implementing the corresponding measurement. The measurement procedures required for this have already been developed in Sections 8 and 9. Here we combine the results from these two sections and apply them to the specific case where the function f arises from a projector in the support of a dual-basis projector Hamiltonian (Definition 5.2).

10.1 Setup

We summarize the main notation, parameters and assumptions that are made throughout this section.

Firstly, $H = \mathbb{E}_i H_i$ s a dual-basis projector Hamiltonian, as introduced in Definition 5.2. According to Theorem 5.4, it is QMA-hard to distinguish between the case where H has smallest eigenvalue exponentially close to 0, or bounded away from 0 by a constant. We let B_q be the number of qubits on which H acts.

Secondly, two codes are chosen. A quantum LTC code C that encodes B_q qubits into M qubits, has distance d_q and locality q. We will use that C_q admits an efficient decoder $\mathtt{D}_q$ for errors of weight up to $\eta_q d_q$, where d_q is the quantum distance of the code. These parameters satisfy the conditions of Theorem 2.21 and Fact 2.22.

The M qubits are divided into r groups of m qubits, where r is chosen sufficiently large that the condition (10) on m holds; this can be satisfied with $r = \operatorname{poly}\log(B_q)$.

A classical code C_C encodes m bits into $n(m)$ bits. We let Δ be the distance of C_C and $\eta_C\Delta$ the (relative) decoding radius of a given efficient decoder. We can take $\Delta = \Theta(n(m))$ and $\eta_C = \Theta(1)$ by Theorem 2.1.

Further notation and parameters associated with the codes is summarized in Section 3.2. The quantum LTC code is used to encode the witness; while the classical code is used to encode measurement outcomes obtained from each of the r blocks of m qubits.

Thirdly, we fix a PCPP verifier and corresponding proof generating algorithm for the CktVal problem. The PCPP has proximity parameter δ and soundness error s. (Further notation associated with the PCPP is recalled in Section 3.3.)

The main assumption relating these parameters are the following.

- The PCPP proximity parameter δ is chosen in such a way that

$$r\delta < \eta_C\Delta\,. \tag{9}$$

  Intuitively, this relation is needed for the PCPP verifier to interact well with the classical code; it first arises from the use of Lemma 7.5. Given that Δ and η_C are both $\Theta(1)$ and $r = M/m = \Theta(\operatorname{poly}\log B_q)$, this assumption can be satisfied by choosing a sufficiently small δ such that $\delta^{-1} = \Theta(\operatorname{poly}\log B_q)$.
- The number m of qubits of the qLTC-encoded witness exchanged in each round is chosen such that

$$\eta_q d_q \geq 2m\,. \tag{10}$$

This can be achieved by choosing r to be sufficiently large, $r = \text{polylog}(B_q)$ will suffice given the quantum LTC code parameters (which are all a multiplicative polylog away from optimal). Intuitively this condition ensures that the prover manipulates too few qubits at a time to perform an undetected attack (in the basis currently being verified).

- The soundness error s of the PCPP is a small enough constant. This is because this soundness error appears as an additive term in the final soundness error.

10.2 Actions of the honest prover

We define the behavior of an honest prover during the energy test, which consists of two phases: the global measurement extraction phase and the measurement phase.

▶ **Definition 10.1** (Energy test, honest prover). *The honest prover performs the following steps:*

- *The verifier sends the basis $W \in \{X, Z\}$ to the prover.*
- *The honest prover performs the actions described in Definition 8.5.*
- *The honest prover performs the actions described in Definition 9.3.*

10.3 Actions of the verifier

We define the verifier's strategy in the energy test, which is designed to verify that the witness state has low energy. In what follows, the case $s_c = 3$ corresponds to the energy test, whose success probability is proportional to the energy of the Hamiltonian. The case $s_c = 1$ is intended to enforce honest behavior from the prover during the global measurement extraction phase, while the case $s_c = 2$ ensures honest behavior during the measurement phase.

▶ **Definition 10.2** (Energy test, verifier). *Recall the notations introduced in Section 3. Let $H = \mathbb{E}_i H_i$ be as in Definition 5.2, where $H_i = I + W_{i,f_i}/2$, W_i is either X or Z, $f_i : \mathbb{F}_2^{B_q} \to \mathbb{F}_2$ is a boolean function.*

The verifier performs the following steps:

- *The verifier samples the phase of the protocol to test by choosing $s_c \in_R [3]$, and selects a round to test in the global measurement extraction phase by choosing $s'_g \in_R [n]$ with probability $1/2$, and setting $s'_g = 0$ with probability $1/2$. If the verifier tests the global measurement extraction phase ($s_c = 1$), it sets $s_g = e_{s'_g}$, where $e_{s'_g} \in \{0,1\}^{n+1}$ is the bit string with all entries 0 except the s'_g-th bit, which is 1. If the verifier does not test during the global measurement extraction phase ($s_c \neq 1$), it sets $s_g = 0^{n+1}$.*
- *The verifier samples a term H_i by selecting the index i uniformly at random. The verifier sends the basis W_i to the prover.*
- *The verifier performs the actions described in Definition 8.8 with $W = W_i$, $s = s_g$.*
- *If $s_c = 1$, the verifier accepts if and only if the preceding procedure accepts.*
- *If $s_c = 2$, the verifier performs the actions described in Definition 9.5 with $h(x) = f_i(KE_W D_W(x))$ and $s = 1$. The verifier accepts if and only if the procedure accept.*
- *If $s_c = 3$, the verifier performs the actions described in Definition 9.5 with $h(x) = f_i(KE_W D_W(x))$ and $s = 0$. The verifier measures the register I and accept if and only if the measurement result is 0.*

10.4 Main results

▶ **Theorem 10.3.** $\mathrm{QMA} \subseteq \mathrm{QIPCP}(O(\mathrm{poly}), O(\mathrm{poly}\log), O(\mathrm{poly}\log))$.

Proof. Let $\mathtt{V}$ be the verifier in Definition 10.2. Assume that the different codes used for the protocol have been chosen in such a way that $r\delta < \eta_C \Delta$, where $r = M/m$ is the number of rounds, δ the proximity error of the PCPP and $\eta_C \Delta$ the (relative) decoding radius of the classical code C_C, as in Notation 3.1. This can be achieved by choosing good enough PCPP and classical code.

We will show that $\mathtt{V}$ is a poly log-message quantum interactive probabilistically checkable proof system for $\mathcal{DLH}(\mathrm{poly}, 2^{-p(n)}, c_{\mathrm{LH}})$ (see Definition 5.3), where $c_{\mathrm{LH}} > 0$ is the constant from Corollary 5.7. It follows that

$$\mathcal{DLH}(\mathrm{poly}, 2^{-p(n)}, c_{\mathrm{LH}}) \in \mathrm{QIPCP}(O(\mathrm{poly}), O(\mathrm{poly}\log), O(\mathrm{poly}\log)) \ .$$

By Corollary 5.7, $\mathcal{DLH}(\mathrm{poly}, 2^{-p(n)}, c_{\mathrm{LH}})$ is QMA-complete. Therefore,

$$\mathrm{QMA} \subseteq \mathrm{QIPCP}(O(\mathrm{poly}), O(\mathrm{poly}\log), O(\mathrm{poly}\log)) \ .$$

We now show the required properties of $\mathtt{V}$. Let n be the instance size, i.e., the description length of a $\mathcal{DLH}$ instance H. In particular, the number of qubits B_q on which H acts satisfies $B_q \leq n$. For the construction of the QIPCP, we use parameters as introduced in Section 10.1.

First, by inspecting Definitions 10.2 and 4.1, it is clear that $\mathtt{V}$ has polynomial-time complexity.

Rounds. By Definitions 10.2 and 4.1, the verifier and the prover exchange a total of $2M/m + 2$ messages.

Recall that M is the length of the qLTC codeword, and m is chosen to be smaller than $\eta_q d_q$ by at least a constant factor. By Theorem 2.21, $d_q = M/\mathrm{poly}\log(M)$, and by Fact 2.22, $\eta_q = 1/\mathrm{poly}\log(M)$. Moreover, the qLTC code has a linear code length $M = O(n)$. Therefore,

$$2M/m + 2 = \mathrm{poly}\log(n) \ .$$

Communication complexity. We compute the communication complexity according to Definitions 4.1 and 10.2. In each of the first M/m rounds, the verifier sends a quantum message of length m (plus 1 classical bit for the basis choice). The prover's message consists of two parts:

- an $n(m)$-bit classical ECC encoding of (measurements from) the verifier's message of length m;
- a PCPP proof of length $l_w(\mathrm{poly}(n(m)))$ for each bit measured from the verifier's message, where the $\mathrm{poly}(n(m))$ term arises from the size of the classical ECC decoding circuit.

The communication in the last round is $l_w(\mathrm{poly}(M/m \cdot n(m)))$. Here, the outer l_w comes from the PCPP proof, while $M/m \cdot n(m)$ corresponds to the size of the implicit input (i.e., the input to the circuit being verified). This circuit evaluates the logical Boolean function corresponding to the sampled projector after decoding the classical ECC code. Since decoding the classical ECC, transforming the Boolean function into its logical form, and evaluating the Boolean function itself are all polynomial-time operations, the PCPP also has an explicit input (the circuit being verified) of size $\mathrm{poly}(M/m \cdot n(m))$. Therefore, the total communication complexity is $\mathrm{poly}(n)$.

Query complexity. By Definitions 4.1 and 10.2, the query complexity in each of the first M/m rounds arises from the local tester for the W-code (in the qLTC) and the PCPP used to verify the queries of the local tester.

We first analyze the query complexity of a single invocation of the local tester for the W-code. Let $\varepsilon = \Theta(1)$ be a small constant that will be specified later, and let $s = \Theta(\varepsilon)$ be the soundness error of the PCPP. Note that the soundness parameter of the LTC is $\rho = \Theta(1/\operatorname{poly}\log(n))$. Define

$$\kappa = \Theta\left(\frac{r^2(s+\varepsilon)M}{\eta_q d_{q,W} - m}\right) = \operatorname{poly}\log(n)\ , \tag{11}$$

since $\eta_q d_{q,W} \geq \eta_q d_q \geq 2m$ by assumption (10) and $M/m = r = \operatorname{poly}\log(n)$. By setting $R = \Theta(1)$ and $\varepsilon = s$ in Lemma 2.6, we obtain the query complexity of the local tester for the W codes of the qLTC as

$$Q = \Theta(1) \cdot \frac{\kappa}{\rho} \cdot \frac{\log(1/s)}{1-s} = \operatorname{poly}\log(n)\ .$$

Next we analyze the query complexity of each invocation of the PCPP verifier. By applying Lemma 2.27, we obtain that the query complexity of the PCPP verifier is

$$O\left(\frac{\log(s)}{\log(1-\Omega(\delta))}\right) = \operatorname{poly}\log(n)\ .$$

Therefore, in each of the first $r = M/m$ rounds, the total query complexity is upper bounded by the product of the query complexity of the local tester and that of the PCPP verifier:

$$\operatorname{poly}\log(n) \cdot \operatorname{poly}\log(n)\ .$$

The query complexity in the final round is due to the PCPP verifier and is thus at most

$$O\left(\frac{\log(s)}{\log(1-\Omega(\delta))}\right) = \operatorname{poly}\log(n)\ ,$$

which again follows from the application of Lemma 2.27. Thus, the overall query complexity of the verifier is $\operatorname{poly}\log(n)$.

Completeness. By Definition 10.1, note that the honest prover uses a qLTC encoding with no error, so the local tester always accepts. Moreover, the PCPP statements in both the global measurement extraction and the measurement phases verify valid computations; hence, the corresponding instances are always yes-instances. By the completeness of the PCPP verifier, the PCPP verifier during these phases succeeds with probability 1. Therefore, all tests except the energy test pass with probability 1.

Let $|\psi\rangle_{\mathsf{OIEA}}$ denote the quantum state of the system after the measurement in the computational or Hadamard basis, as defined in Lemma 9.7, and let $|\varphi\rangle_{\mathsf{WBEXP}}$ be the quantum state of the system before the global measurement extraction, as defined in Lemma 8.11.

If $s_c = 3$, then by Definition 10.2, the energy test passes if and only if the measurement result of the register I is 0 with the corresponding probability:

$$\begin{aligned}
\mathop{\mathbb{E}}_i \left\| \frac{I + Z_{\mathsf{I}}}{2} |\psi\rangle \right\|^2 &= \mathop{\mathbb{E}}_i \left\| \frac{I + Q_r}{2} |\psi\rangle \right\|^2 \\
&= \mathop{\mathbb{E}}_i \left\| \frac{I + Q_0}{2} |\varphi\rangle \right\|^2 \\
&= \mathop{\mathbb{E}}_i \left\| (I - W_{i,f_i}) |\varphi\rangle \right\|^2 \\
&= \mathop{\mathbb{E}}_i \left\| (I - H_i) |\varphi\rangle \right\|^2 \\
&= \langle\varphi| (I - H) |\varphi\rangle\ ,
\end{aligned}$$

where the first step follows from Lemma 9.6, Lemma 9.7, and the fact that the verifier never rejects in the measurement phase; the second step follows from Lemma 8.11 and the verifier never rejecting in the global measurement extraction phase; and the third step follows from Lemma 6.8 and the fact that the quantum witness is a qLTC encoding. As a result,

$$\begin{aligned}\Pr[\mathtt{V}\text{ accepts}] =\ & \Pr[\mathtt{V}\text{ accepts} \mid s_c = 1]\cdot\Pr[s_c=1] + \Pr[\mathtt{V}\text{ accepts} \mid s_c = 2]\cdot\Pr[s_c=2] \\ & + \Pr[\mathtt{V}\text{ accepts} \mid s_c = 3]\cdot\Pr[s_c=3] \\ =\ & \frac{1}{3}+\frac{1}{3}+\frac{1}{3}\langle\varphi|(I-H)|\varphi\rangle \\ \geq\ & 1-\Theta(2^{-p(n)})\,.\end{aligned}$$

Soundness. Let $|\psi\rangle_{\mathsf{OIEA}}$ denote the quantum state of the system after the measurement in the computational or Hadamard basis, as defined in Lemma 9.7. This corresponds to the state of the quantum system after the verifier performs the procedure described in Definition 9.5 within Definition 10.2, in both cases $s_c = 2$ and $s_c = 3$. Let $|\varphi\rangle_{\mathsf{WBEXP}}$ denote the quantum state of the system before the global measurement extraction procedure, as defined in Lemma 8.11, i.e., the state of the quantum system before the verifier executes the procedure from Definition 8.8 in Definition 10.2. Then,

$$\Pr[\text{energy test accepts} \mid s_c = 3] = \mathop{\mathbb{E}}_i \left\| \frac{I+Z_{\mathsf{I}}}{2}|\psi\rangle \right\|^2 .$$

Note that for any vectors a, b and $k > 0$,

$$\|a+b\|^2 \leq \frac{1+k^2}{k^2}\|a\|^2 + (1+k^2)\|b\|^2\,.$$

Using this with $k = 10$,

$$\left\|\frac{I+Z_{\mathsf{I}}}{2}|\psi\rangle\right\|^2 \leq 1.01\left\|\frac{I+Q_r}{2}|\psi\rangle\right\|^2 + 101\left\|\frac{Z_{\mathsf{I}}|\psi\rangle - Q_r|\psi\rangle}{2}\right\|^2 .$$

Using the same reasoning,

$$\left\|\frac{I+Q_r}{2}|\psi\rangle\right\|^2 \leq 1.01\left\|U_b\frac{I+Q_0}{2}(|0\rangle_{\mathsf{T}}|\varphi\rangle)\right\|^2 + 101\left\|\frac{(U_bQ_0 - Q_rU_b)(|0\rangle_{\mathsf{T}}|\varphi\rangle)}{2}\right\|^2 .$$

Finally,

$$\begin{aligned}\left\|\frac{I+Q_0}{2}(|0\rangle_{\mathsf{T}}|\varphi\rangle)\right\|^2 \leq\ & 1.01\|(I-W_{f_i})(|0\rangle_{\mathsf{T}}|\varphi\rangle)\|^2 \\ & + 101\left\|\frac{(Q_0-(I-2W_{f_i}))(|0\rangle_{\mathsf{T}}|\varphi\rangle)}{2}\right\|^2 .\end{aligned}$$

Summarizing the preceding equations,

$$\begin{aligned}&\Pr[\text{energy test accepts} \mid s_c = 3] \\ &\quad\leq 101\mathop{\mathbb{E}}_i\left\|\frac{Z_{\mathsf{I}}|\psi\rangle - Q_r|\psi\rangle}{2}\right\|^2 \\ &\qquad + 1.01\cdot 101\left\|\frac{(U_bQ_0-Q_rU_b)(|0\rangle_{\mathsf{T}}|\varphi\rangle)}{2}\right\|^2 \\ &\qquad + 1.01^2\Big(1.01\|(I-W_{f_i})(|0\rangle_{\mathsf{T}}|\varphi\rangle)\|^2 \\ &\qquad\quad + 101\left\|\frac{(Q_0-(I-2W_{f_i}))(|0\rangle_{\mathsf{T}}|\varphi\rangle)}{2}\right\|^2\Big)\,. \qquad (12)\end{aligned}$$

Assume that

$$\begin{aligned}\Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g = 0] \geq 1 - \varepsilon \ , \\ \Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g \neq 0] \geq 1 - \varepsilon \ , \\ \Pr[\text{measurement phase accepts} \mid s_c = 2] \geq 1 - \varepsilon \ ,\end{aligned} \tag{13}$$

where ε is the previously introduced small constant that will be specified later. By Lemma 9.6 and Lemma 9.7,

$$\| Z_{\mathsf{I}} |\psi\rangle - Q_r |\psi\rangle \|^2 \leq \Theta(s + \varepsilon) \ . \tag{14}$$

Moreover, suppose $\Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g = i] = 1 - \varepsilon_i$ for some ε_i, then by Lemma 8.11,

$$\| U_b Q_0 (|0\rangle_{\mathsf{T}} |\varphi\rangle) - Q_r U_b (|0\rangle_{\mathsf{T}} |\varphi\rangle) \|^2 \leq \Theta\Big(\frac{r^2 \cdot (s + \mathbb{E}_i \, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)} \Big) \ . \tag{15}$$

Let

$$|\varphi\rangle = \sum_{(a,b) \in \mathcal{S}} \alpha_{a,b} X(a) Z(b) |\eta_{a,b}\rangle \ .$$

By Lemma 6.8,

$$\begin{aligned}\| Q_0 (|0\rangle_{\mathsf{T}} |\varphi\rangle) - (I - 2W_{f_i})(|0\rangle_{\mathsf{T}} |\varphi\rangle) \|^2 \leq \Theta(1) \cdot \sum_{(a,b) \in \mathcal{S}} 1(wt_{\text{cent}}(X(a)Z(b)) \geq \eta_q d_q) \\ \cdot |\alpha_{a,b}|^2 \ .\end{aligned}$$

Let c be the number of repetitions of the local testing verifier from Definition 6.1, which will be chosen below in (18). By Lemma 6.2,

$$\begin{aligned}1 - \varepsilon &\leq \sum_{(a,b) \in \mathcal{S}} |\alpha_{a,b}|^2 \cdot \Big(1 - \frac{\rho}{2M} \cdot wt_{\text{cent}}(X(a)Z(b)) \Big)^c \\ &\leq \sum_{(a,b) \in \mathcal{S}} 1(wt_{\text{cent}}(X(a)Z(b)) \geq \eta_q d_q) \cdot |\alpha_{a,b}|^2 \cdot \Big(1 - \frac{\rho}{2M} \cdot \eta_q d_q \Big)^c \\ &\quad + \sum_{(a,b) \in \mathcal{S}} 1(wt_{\text{cent}}(X(a)Z(b)) < \eta_q d_q) \cdot |\alpha_{a,b}|^2 \ .\end{aligned}$$

Therefore,

$$\| Q_0 (|0\rangle_{\mathsf{T}} |\varphi\rangle) - (I - 2W_{f_i})(|0\rangle_{\mathsf{T}} |\varphi\rangle) \|^2 \leq \Theta\Big(\varepsilon \Big(1 - \frac{\rho}{2M} \cdot \eta_q d_q \Big)^c \Big)^{-1} \Big) \ . \tag{16}$$

Plugging the bounds (14), (15) and (16) in (12) and using the assumptions (13) we have obtained that

$$\begin{aligned}\Pr[\text{energy test accepts}] \leq 1.1 \cdot \langle \varphi | \, (I - H) \, |\varphi\rangle + \Theta(s + \varepsilon) + \Theta\Big(\frac{r^2 \cdot (s + \mathbb{E}_i \, \varepsilon_i) \cdot M}{\kappa \cdot (\eta_q d_{q,W} - m)} \Big) \\ + \Theta\Big(\varepsilon \Big(1 - \Big(1 - \frac{\rho}{2M} \cdot \eta_q d_q \Big)^c \Big)^{-1} \Big) \ .\end{aligned} \tag{17}$$

Note that

$$\Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g \neq 0] = 1 - \mathop{\mathbb{E}}_i \varepsilon_i \ .$$

Choose

$$c := \frac{\Theta(1)}{-\log(1-\frac{\rho}{2M}\cdot\eta_q d_q)} = \operatorname{poly}\log(n)\ . \tag{18}$$

Plugging back into (17) and using also the choice of κ in (11),

$$\begin{aligned}\Pr[\text{energy test accepts}] &\leq 1.1\cdot\langle\varphi|\,(I-H)\,|\varphi\rangle + \Theta(\varepsilon)\\ &\leq 1.1 - 1.1c_{\mathrm{LH}} + \Theta(\varepsilon)\ .\end{aligned}$$

If any of the following fails:

$$\begin{aligned}\Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g = 0] &\geq 1-\varepsilon\ ,\\ \Pr[\text{global measurement extraction accepts} \mid s_c = 1, s'_g \neq 0] &\geq 1-\varepsilon\ ,\\ \Pr[\text{measurement phase accepts} \mid s_c = 2] &\geq 1-\varepsilon\ ,\end{aligned}$$

then

$$\Pr[\mathtt{V}\text{ accepts}] \leq 1-\Theta(\varepsilon)\ .$$

Note that we can take c_{LH} in Corollary 5.7 to be sufficiently large, and in particular strictly larger than 0.1. As a result, in all cases, we get that $\Pr[\mathtt{V}\text{ accepts}]$ is smaller than some constant less than 1, as desired.

By applying sequential repetition a constant number of times (since $c_{\mathrm{LH}} = \Theta(1), \varepsilon = \Theta(1)$), we obtain a protocol in which the verifier accepts yes-instances with probability greater than $2/3$ and accepts no-instances with probability less than $1/3$. ◀

References

1. Dorit Aharonov, Itai Arad, Zeph Landau, and Umesh Vazirani. The detectability lemma and quantum gap amplification. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 417–426, 2009. doi:10.1145/1536414.1536472.
2. Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. *arXiv preprint*, 2017. arXiv:1704.04487.
3. Dorit Aharonov and Lior Eldar. Quantum locally testable codes. *SIAM J. Comput.*, 44(5):1230–1262, 2015. doi:10.1137/140975498.
4. James Bartusek, Yael Tauman Kalai, Alex Lombardi, Fermi Ma, Giulio Malavolta, Vinod Vaikuntanathan, Thomas Vidick, and Lisa Yang. Succinct classical verification of quantum computation. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 195–211. Springer, 2022. doi:10.1007/978-3-031-15979-4_7.
5. Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs and applications to coding. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 1–10. ACM, 2004. doi:10.1145/1007352.1007361.
6. Thiago Bergamaschi, Tony Metger, Thomas Vidick, and Tina Zhang. Derandomised tensor product gap amplification for quantum Hamiltonians. *arXiv preprint arXiv:2510.01333*, 2025. doi:10.48550/arXiv.2510.01333.
7. Jacob D Biamonte and Peter J Love. Realizable hamiltonians for universal adiabatic quantum computers. *Physical Review A—Atomic, Molecular, and Optical Physics*, 78(1):012352, 2008.

8 Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 31–40. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.13.

9 Harry Buhrman, Jonas Helsen, and Jordi Weggemans. Quantum PCPs: on adaptivity, multiple provers and reductions to local hamiltonians. *Quantum*, 9:1791, 2025. doi:10.22331/Q-2025-07-11-1791.

10 Michael Chapman, Thomas Vidick, and Henry Yuen. Efficiently stable presentations from error-correcting codes. *arXiv preprint*, 2023. arXiv:2311.04681.

11 Lijie Chen and Ramis Movassagh. Quantum Merkle trees. *Quantum*, 8:1380, 2024. doi:10.22331/Q-2024-06-18-1380.

12 Irit Dinur, Min-Hsiu Hsieh, Ting-Chun Lin, and Thomas Vidick. Good quantum LDPC codes with linear time decoders. In *Proceedings of the 55th annual ACM symposium on theory of computing*, pages 905–918, 2023. doi:10.1145/3564246.3585101.

13 Irit Dinur, Ting-Chun Lin, and Thomas Vidick. Expansion of high-dimensional cubical complexes: with application to quantum locally testable codes. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 379–385. IEEE, 2024. doi:10.1109/FOCS61266.2024.00031.

14 Lior Eldar and Aram W. Harrow. Local hamiltonians whose ground states are hard to approximate. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 427–438. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.46.

15 Joseph F Fitzsimons, Michal Hajdušek, and Tomoyuki Morimae. Post hoc verification of quantum computation. *Physical review letters*, 120(4):040501, 2018.

16 Joseph F Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Physical Review A*, 96(1):012303, 2017.

17 Daniel Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, Thesis, eprint: quant-ph/9705052, 1997.

18 Alex Bredariol Grilo. A simple protocol for verifiable delegation of quantum computation in one round. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019*, volume 132, pages 28–1, 2019.

19 Sam Gunn, Nathan Ju, Fermi Ma, and Mark Zhandry. Commitments to quantum states. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1579–1588, 2023. doi:10.1145/3564246.3585198.

20 Sam Gunn, Yael Tauman Kalai, Anand Natarajan, and Agi Villanyi. Classical commitments to quantum states. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 234–244, 2025.

21 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*=RE. *Communications of the ACM*, 64(11):131–138, 2021.

22 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Quantum soundness of testing tensor codes. *Discrete Analysis*, December 6 2022. doi:10.19086/da.55554.

23 Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47 in Graduate Studies in Mathematics. American Mathematical Soc., 2002.

24 Henry Ma and Anand Natarajan. Two bases suffice for QMA_1-completeness. *arXiv preprint arXiv:2509.24390*, 2025. doi:10.48550/arXiv.2509.24390.

25 Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018. doi:10.1109/FOCS.2018.00033.

26 Chris Marriott and John Watrous. Quantum Arthur–Merlin games. *computational complexity*, 14(2):122–152, 2005. doi:10.1007/S00037-005-0194-X.

27 Tony Metger, Anand Natarajan, and Tina Zhang. Succinct arguments for QMA from standard assumptions via compiled nonlocal games. *CoRR*, abs/2404.19754, 2024. doi:10.48550/arXiv.2404.19754.

28 Anand Natarajan and Chinmay Nirkhe. The status of the quantum PCP conjecture (games version). *CoRR*, abs/2403.13084, 2024. doi:10.48550/arXiv.2403.13084.

29 Anand Natarajan and Thomas Vidick. Robust self-testing of many-qubit states. *CoRR*, abs/1610.03574, 2016. arXiv:1610.03574.

30 Anand Natarajan and John Wright. NEEXP is contained in MIP*. In *2019 ieee 60th annual symposium on foundations of computer science (focs)*, pages 510–518. IEEE, 2019.

31 Anand Natarajan and Tina Zhang. Bounding the quantum value of compiled nonlocal games: From CHSH to BQP verification. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 1342–1348. IEEE, 2023. doi:10.1109/FOCS57990.2023.00081.

32 Quynh T Nguyen and Christopher A Pattison. Quantum fault tolerance with constant-space and logarithmic-time overheads. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 730–737, 2025. doi:10.1145/3717823.3718318.

33 Baocheng Sun and Thomas Vidick. Quantum interactive oracle proofs. In *Theory of Cryptography Conference (TCC)*. Springer-Verlag, 2025.

34 Adam Wills, Ting-Chun Lin, and Min-Hsiu Hsieh. Tradeoff constructions for quantum locally testable codes. *IEEE Transactions on Information Theory*, 2024.

35 Jiayu Zhang. Succinct blind quantum computation using a random oracle. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1370–1383, 2021. doi:10.1145/3406325.3451082.

Condensing and Extracting Against Online Adversaries

Eshan Chattopadhyay
Cornell University, Ithaca, NY, USA

Mohit Gurumukhani
Cornell University, Ithaca, NY, USA

Noam Ringach
Cornell University, Ithaca, NY, USA

Rocco A. Servedio
Columbia University, New York, NY, USA

Abstract

We investigate the tasks of deterministically condensing and extracting randomness from Online Non-Oblivious Symbol Fixing (oNOSF) sources, a natural model of defective random sources for which it is known that extraction is impossible in many parameter regimes [AORSV, EUROCRYPT'20]. A (g, ℓ)-oNOSF source is a sequence of ℓ blocks $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_\ell) \sim (\{0,1\}^n)^\ell$, where at least g of the blocks are *good* (are independent and have some min-entropy), and the remaining *bad* blocks are controlled by an *online adversary* where each bad block can be arbitrarily correlated with any block that appears before it.

The existence of condensers (in regimes where extraction is impossible) was recently studied in [CGR, FOCS'24]. They proved condensing impossibility results for various values of g and ℓ, and they showed the existence of condensers matching the impossibility results in the special case when n is exponential in ℓ (i.e., the setting of few blocks of large length).

In this work, not only do we construct the first explicit condensers matching the existential results of [CGR, FOCS'24], but we make a doubly exponential improvement by handling the case when n is only polylogarithmic in ℓ. We also obtain a much improved explicit construction for transforming low-entropy oNOSF sources (where the good blocks only have min-entropy, as opposed to being uniform) into uniform oNOSF sources.

As our next result, we essentially resolve the question of the existence of condensers for oNOSF sources by showing the existence of condensers in almost all parameter regimes, even when n is a large enough constant and ℓ is growing.

We find interesting connections and applications of our results on condensers to collective coin flipping and collective sampling, problems that are well-studied in fault-tolerant distributed computing. We use our condensers to provide very simple protocols for these problems.

Next, we turn to understanding the possibility of extraction from oNOSF sources. For proving lower bounds, we introduce and initiate a systematic study of a new, natural notion of the influence of functions, which we call *online influence*, and establish tight bounds on the total online influence of functions, which imply extraction lower bounds. Lastly, we give explicit extractor constructions for oNOSF sources using novel connections to leader election protocols, and we further construct the required leader election protocols. These extractor constructions achieve parameters that go beyond the standard resilient functions of [AL, Combinatorica'93].

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization; Theory of computation → Expander graphs and randomness extractors; Computer systems organization → Dependable and fault-tolerant systems and networks

Keywords and phrases collective coin flipping, leader election, Boolean function analysis, fault tolerant distributed computing, full information model, resilient function, pseudorandomness, condensers, adversarial sources, non-oblivious symbol fixing sources, Chor-Goldreich sources

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.5



© Eshan Chattopadhyay, Mohit Gurumukhani, Noam Ringach, and Rocco A. Servedio;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 5; pp. 5:1–5:22



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version *Full Version*: https://arxiv.org/abs/2411.04115

Funding *Eshan Chattopadhyay*: Supported by a Sloan Research Fellowship, NSF CAREER Award 2045576, and NSF Award CCF-2514586.
Mohit Gurumukhani: Supported by a Sloan Research Fellowship, NSF CAREER Award 2045576, and NSF Award CCF-2514586.
Noam Ringach: Supported by NSF GRFP grant DGE – 2139899, NSF CAREER Award 2045576, a Sloan Research Fellowship, and NSF Award CCF-2514586.
Rocco A. Servedio: Supported by NSF Award CCF-2106429 and NSF Award CCF-2211238.

Acknowledgements We thank Madhur Tulsiani for asking a question that motivated us to consider the model of local oNOSF sources. We thank Miranda Christ for helpful feedback on an earlier version of the manuscript. We thank the organizers of the Dagstuhl Seminar on Algebraic and Analytic Methods in Computational Complexity and Schloss Dagstuhl for providing a stimulating research environment, where discussions between R.S. and E.C. contributed to this collaboration.

1 Introduction

Randomness is extremely useful in computation with wide-ranging applications in algorithm design, cryptography, distributed computing protocols, machine learning, error-correcting codes, and much more [34, 39]. Most of these applications require access to high quality randomness. However in a lot of settings, especially arising in practice, algorithms only have access to low quality source of randomness. This motivates the notion of *condensers*: functions that transform weak random sources into strong random sources that are of *better quality*.

The standard way of measuring the amount of randomness is using min-entropy. Formally, for a source (distribution) $\mathbf{X}$ with support Ω, define its min-entropy as $H_\infty(\mathbf{X}) = \min_{x\in\Omega} \log_2(1/\Pr[\mathbf{X} = x])$. If $\Omega = \{0,1\}^n$ and $H_\infty(\mathbf{X}) \geq k$, we say that $\mathbf{X}$ is an (n,k)-source. We will also need the notion of smooth min-entropy, which measures how close a distribution is to having high entropy. Formally, for a source $\mathbf{X}$, its smooth min-entropy with parameter ε is defined as $H^\varepsilon_\infty(\mathbf{X}) = \max_{\mathbf{Y}:|\mathbf{X}-\mathbf{Y}|\leq\varepsilon}\{H_\infty(\mathbf{Y})\}$, where $|\cdot|$ denotes the statistical distance.

With this, we are ready to formally define *deterministic condensers*:

▶ **Definition 1.** *A function* $\mathsf{Cond} : \{0,1\}^n \to \{0,1\}^m$ *is a* $(k_{in}, k_{out}, \varepsilon)$-condenser for a family of distributions $\mathcal{X}$ *if for all* $\mathbf{X} \in \mathcal{X}$ *with* $\mathbf{X} \sim \{0,1\}^n$ *and* $H_\infty(\mathbf{X}) \geq k_{in}$*, we have that* $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq k_{out}$.

We say $\frac{k_{in}}{n}$ *is the* input entropy rate*,* $\frac{k_{out}}{m}$ *is the* output entropy rate*, and* $m - k_{out}$ *is the* entropy gap *of* Cond.

The task of the condenser is to make the output entropy rate as high as possible compared to the input entropy rate; i.e., to make the output distribution more "condensed". Related to this, it is also desirable to have as small entropy gap as possible. Condensers with entropy gap 0 are known as *randomness extractors* and have been extensively studied in theoretical computer science.

When $\mathcal{X}$ is the family of all distributions, it is folklore that no non-trivial condensing is possible.[1] So, we additionally assume that $\mathcal{X}$ is a structured family of sources.[2] Since extractors are the highest quality condensers, a significant amount of work has focused on constructing extractors for many interesting families of sources [38, 11, 17, 30]. However, for many natural family of sources, one can provably show that no extractor can exist.

In this work, we focus on one natural family of sources where it is known that extraction is impossible (for many interesting parameter regimes): online non-oblivious symbol fixing sources (oNOSF sources).[3] Formally:

▶ **Definition 2.** *A (g, ℓ, n, k)-oNOSF source $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_\ell)$ with ℓ blocks is such that each block $\mathbf{X}_i$ is over $\{0,1\}^n$, g of the blocks are independent sources with min-entropy k ("good blocks"), and each "bad block" is an arbitrary function of the blocks with an index smaller than it. When $k = n$, we will call such sources uniform (g, ℓ, n)-oNOSF sources, and when n is clear from context, we will simply call them uniform (g, ℓ)-oNOSF sources.*

Our results at a glance

The previous work of [10] gave a condenser impossibility result for oNOSF sources and showed the existence of a condenser matching that result as long as the block length n was exponential in the number of blocks ℓ. We construct explicit condensers for oNOSF sources that match the results of [10]; in fact, we only require n to be polylogarithmic in ℓ, providing a doubly exponential improvement over the (existential) result of [10]. Next, we essentially resolve the existence question for oNOSF source condensers by showing that good condensers exist even when n is a (large) constant and ℓ grows. To go with these results, we obtain an improved construction for transforming low-entropy oNOSF sources into uniform oNOSF sources. Moreover, we find new applications of our results on condensers to collective coin flipping and collective sampling, and use these connections to provide simple protocols for these problems. In the complementary direction, we construct explicit extractors for uniform oNOSF sources by explicitly constructing the required leader election protocols; the results of which are summarized in Tables 1 and 2. Also in the context of extractors for oNOSF sources, we introduce the new, natural notion of *online influence* for Boolean functions and show extraction lower bounds for oNOSF sources by establishing tight bounds on the total online influence of functions.

On the Utility of Condensing for oNOSF sources

We note that condensers (and sources with high min-entropy rate) are very useful: the condensed distribution can be used to efficiently simulate randomized algorithms with small overhead, perform one-shot simulations for randomized protocols, cryptography, interactive proofs, and much more. For details on these applications and more, see [1, 15, 10, 14].

[1] Assuming $m \leq n$ (wlog this holds since $|\mathsf{Cond}(\{0,1\}^n)| \leq 2^n$), $m - k_{out} \geq (n - k_{in}) - \log(1/(1-\varepsilon))$ and hence the output entropy rate cannot be more than the input entropy rate without incurring extremely large error (> 0.999).

[2] A different route, that has been widely studied, is to assume access to a short independent seed. In this work, we will limit ourselves to the *deterministic or seedless setting*.

[3] These sources are in contrast to non-oblivious symbol fixing (NOSF) sources where bad blocks can be arbitrary functions of all the good blocks and oblivious symbol fixing sources (OSF) where bad blocks do not depend on any good blocks. NOSF sources were introduced in [30] and OSF sources were introduced in [12] with applications in leakage-resilient cryptography, and have been well-studied.

Practical applications to blockchains and cryptography

oNOSF sources are inspired by real-time randomness generation settings such as in blockchains where the adversary has some probability of corrupting a block. Moreover, it is known that non-corrupted blocks have some amount of min-entropy [7]. In fact, several works have attempted to use Bitcoin or Ethereum as a source of randomness in cryptographic protocols [7, 6, 35, 9]. However, the authors of [6] showed that even when the adversary has a small, constant probability of corrupting a block, randomness extraction is impossible from Bitcoin.[4] Our results show that in this setting, it is still possible to get a condensed source with a high min-entropy rate. It is known that such sources are still useful for cryptographic protocols, such as hedged public-key enryption [4]. Further, there are natural cryptographic settings, such as creating a Common Reference String, that are widely used in various cryptographic protocols where oNOSF sources naturally arise [1].

Practical applications to fault-tolerant distributed computing

One common scenario in distributed computing is that of many agents (e.g., servers in a network) attempting to collectively take a decision using several rounds of communication over a common broadcast channel in the presence of computationally unbounded adversarial agents, which render cryptographic primitives ineffective. Protocols for collective coin flipping, leader election, and collective sampling are prime examples of this scenario that have been intensively studied ([5, 19, 13, 3, 18] and many more), and already connected to the extractor literature [31, 32]. In Section 3, we explain how condensing or extracting from oNOSF sources can be viewed as a variant of these protocols. As a consequence, our new results on condensers provide a new protocol for collective sampling and impossibility results for these protocols can be translated into lower bounds against extractors and condensers for oNOSF sources.

Organization

The remainder of our introduction is structured as follows. We give an overview of previous work in Section 1.1 before presenting our main existential and explicit condenser results in Section 1.2. In Section 1.3, we present our results on the limits of extraction from oNOSF sources. Later on, in Section 3, we explain how our results on condensers have implications for collective coin flipping and sampling protocols.

1.1 Previous Work

Extractors

The study of extractors for oNOSF sources was initiated by [1].[5] Their results include the following:

- It is impossible to extract from uniform oNOSF sources when the fraction of good blocks is 0.99.
- An explicit transformation from $(g, \ell, n, 0.9n)$-oNOSF source into a source over $(\{0,1\}^{O(n)})^{\ell-1}$ where $g-1$ of the blocks are uniform and independent.
- An explicit transformation from $(g, \ell, n, 0.1n)$-oNOSF source into a source over $(\{0,1\}^{O(n)})^{100\ell}$ where $g-1$ of the blocks are uniform and independent.

[4] This mirrors our extraction impossibility result for oNOSF sources.
[5] In [1], these sources were called SHELA (Somewhere Honest Entropic Look Ahead) sources.

Even though the output entropy rate is only slightly more than the input-entropy rate in the second result and smaller in the third result, the fact that a lot of the blocks are truly uniform is very useful, and they find interesting cryptographic applications of these "somewhere-extractors".

Before our work, the best known extractors for oNOSF sources could be obtained by using resilient functions or equivalently, extractors for NOSF sources (non-online version of oNOSF sources) constructed by [2, 11, 33, 26, 27] ; these require $g \geq \ell - \frac{\ell}{(\log \ell)^2}$.

Condensers

oNOSF sources were further studied by [10], where they obtained the following results regarding condensers:

- When $n \geq k \geq \ell$, there exist functions that can transform a (g, ℓ, n, k)-oNOSF source into a uniform $(g-1, \ell-1, O(k/\ell))$-oNOSF source (this function can be made explicit with slightly worse dependence on output length).
- When $n \geq 2^{\omega(\ell)}$ and $g > 0.5\ell$, there exists condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^{m=O(n\cdot\ell/g)}$ such that for any uniform (g, ℓ, n)-oNOSF source $\mathbf{X}$, $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - O(\log(n/\varepsilon))$.[6] Their result is not explicit.
- It is impossible to condense from uniform $(0.5\ell, \ell, n)$-oNOSF sources with output entropy rate more than 0.5.[7]

We also mention a related family of sources, namely adversarial Chor-Goldreich sources. Uniform oNOSF sources can be seen as a special case of adversarial Chor-Goldreich sources where the good blocks are uniform. Constructing condensers where the output entropy rate is g/ℓ for adversarial Chor-Goldreich sources is already a challenging task, although such condensers in various parameter regimes have been recently constructed [15, 22]. The paper of [16] recently constructed condensers for a related, more general model.

1.2 This Work: New Condenser Constructions

Previous works only showed the existence of condensers for oNOSF sources when $n \geq 2^{\omega(\ell)}$. We vastly improve on this result in two ways. First, we construct explicit condensers that work even when $n \geq \text{polylog}(\ell)$ and provide an explicit transformation from low-entropy oNOSF sources to uniform oNOSF sources that works even when the min-entropy of a block k is only $\text{polylog}(n)$. Second, we show that condensers for oNOSF sources exist when n is just a large constant, only leaving open the question of the existence of such condensers for when n is a very small constant (e.g., $n = 1$). We also discover surprising connections between condensers for oNOSF sources and protocols for natural problems in distributed computing, such as collective coin flipping and collective sampling. Lastly, we initiate the study of *online influence* of Boolean functions, a natural generalization of influence that captures the one-sided nature of our online adversary to help us analyze the setting of $n = 1$. We now discuss our results in detail below.

1.2.1 Explicit Condensers

We construct the first explicit condensers for oNOSF sources. Our ultimate result is founded on a baseline construction that itself is an explicit condenser for oNOSF sources that matches the existential results of [10] and works for any block length $n = 2^{\Omega(\ell)} \log(1/\varepsilon)$ as long as at least 51% of blocks are good.

[6] They get a tradeoff for $g \leq 0.5\ell$ as well
[7] They get impossibility for other smaller g as well

▶ **Theorem 3** (Theorem 16 restated). *For all $\varepsilon > 0$ and $n, \ell \in \mathbb{N}$ where $n \geq 2^{\Omega(\ell)} \log(1/\varepsilon)$, there exists an explicit condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any $(g = 0.51\ell, \ell, n)$-oNOSF source $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - 2^{O(\ell)} \log(1/\varepsilon)$ where $m = 0.0001\ell n$.*

Surprisingly, we are able to improve upon this baseline to obtain explicit condensers that work for oNOSF sources where the block length n is only at least $\mathrm{poly}(\log(\ell)/\varepsilon)$.

▶ **Theorem 4** (Informal version of Theorem 13). *For all $\varepsilon > 0$ and $n, \ell \in \mathbb{N}$ where $n \geq \mathrm{poly}(\log(\ell)/\varepsilon)$, there exists an explicit condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any $(g = 0.51\ell, \ell, n)$-oNOSF source $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - \mathrm{poly}(\log(\ell)/\varepsilon) \cdot \log(n)$ where $m = 0.001\ell n - O(\ell \log(\ell) \log(1/\varepsilon))$.*

Since condensing when $g = 0.5\ell$ is impossible, both results are tight. We note that neither result completely subsumes the other. Our baseline construction in Theorem 3 has an exponential dependence of n on ℓ instead of the polylogarithmic dependence achieved in Theorem 4; however, the latter result requires the dependence $n \geq \mathrm{poly}(1/\varepsilon)$ compared to $n \geq \log(1/\varepsilon)$ for the baseline construction.

Using our new results regarding transforming oNOSF sources to uniform oNOSF sources, we also obtain explicit condensers for $(0.51\ell, \ell, n, k)$-oNOSF sources for the same parameter regime:

▶ **Corollary 5.** *For all $n, \ell, k \in \mathbb{N}$ where $n \geq \mathrm{poly}(\ell)$ and $k \geq \mathrm{polylog}(n)$, there exists an explicit condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any $(g = 0.51\ell, \ell, n, k)$-oNOSF source $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - \mathrm{polylog}(\ell) \cdot \log(n)$ where $m = 0.001\ell n - O(\ell \log(\ell) \log\log(\ell))$ and $\varepsilon = \mathrm{poly}(1/\log(\ell))$.*

We can also extend our result to explicitly condense from uniform (g, ℓ, n)-oNOSF sources in the same parameter regime so that the output entropy rate is $1/\lfloor \ell/g \rfloor - o(1)$, which is tight according to the impossibility result of [10].

Previously, [10] showed how to existentially condense from uniform (g, ℓ, n)-oNOSF sources when $n = 2^{\Omega(\ell)}$. However, they relied on the existence of a very strong pseudorandom object: "output-light" low-error two-source extractors. Such extractors, even without the output-lightness requirement, are extremely hard to construct and it is a major open problem to obtain such extractors. We are able to construct explicit condensers by creating new tools that allow us to use an oNOSF source to sample indices within an oNOSF source, and stitching them together so that the base pseudorandom object we rely on are seeded extractors that we know how to explicitly construct with near optimal parameters.

1.2.2 Transforming Low-Entropy oNOSF sources to uniform oNOSF sources

We show how to existentially, as well as explicitly, with a slight loss in parameters, transform (g, ℓ, n, k)-oNOSF sources into uniform $(0.99g, \ell - 1, n)$-oNOSF sources. Formally, we show:

▶ **Theorem 6.** *For all ℓ, n, k, ε where $n = \mathrm{poly}(\log(\ell)), k = O(\log(\ell/\varepsilon))$, there exists a function f such that f transforms $(0.51\ell, \ell, n, k)$-oNOSF sources into uniform $(0.509\ell, \ell, m)$-oNOSF sources with error ε where $m = \Omega(k)$.*

Our construction can also be made explicit with slightly worse dependence on m and ε. See the full version of our paper for the full tradeoff.

Previously, [10] provided such a transformation only for $n \geq k \geq \Omega(\ell)$. Hence, our transformation makes a major improvement on their parameters. Such an improvement allows us to obtain better condensers for low-entropy oNOSF sources in the regime $n = \text{poly}(\log(\ell/\varepsilon))$ (see Theorem 8).

1.2.3 Existential Condensers

We show how to condense from uniform (g, ℓ, n)-oNOSF sources for almost all settings of ℓ and n when $g \geq 0.51\ell$. In particular, we show:

▶ **Theorem 7.** *For all ℓ, ε where $\ell \geq O(\log(1/\varepsilon))$, and $n = 10^4$, there exists a condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any uniform $(0.51\ell, \ell, n)$-oNOSF source $\mathbf{X}$, we have $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq 0.99m$ where $m = \Omega(\ell + \log(1/\varepsilon))$. Furthermore, when $n = \omega(1)$, the output entropy rate becomes $1 - o(1)$.*

This is tight since [10] showed it is impossible to condense uniform $(0.5\ell, \ell, n)$-oNOSF sources beyond output entropy rate 0.5.

Using our new results regarding transforming oNOSF sources to uniform oNOSF sources, we also obtain condensers for $(0.51\ell, \ell, n, k)$-oNOSF sources when $n \geq \text{poly}(\log(\ell))$,

▶ **Theorem 8.** *For all ℓ, n, ε where $n = \text{poly}(\log(\ell/\varepsilon)), k = \Omega(\log(\ell/\varepsilon))$, there exists a condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any $(0.51\ell, \ell, n, k)$-oNOSF source $\mathbf{X}$, we have $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - O(m/\log(m)) - O(\log(1/\varepsilon))$ where $m = \Omega(k)$.*

We sketch the proof of both of these theorems in Section 2.1 We can also extend our result to condense from uniform (g, ℓ, n)-oNOSF sources for all g, ℓ and constant n where the output entropy rate is $1/\lfloor \ell/g \rfloor - 0.001$. This is tight since [10] showed it is impossible to condense such sources beyond output entropy rate $1/\lfloor \ell/g \rfloor$.

Previously, [10] showed how to existentially condense from uniform (g, ℓ, n)-oNOSF sources when $g \geq 0.51\ell$, provided $n \geq 2^{\omega(\ell)}$. As n gets smaller, condensing becomes harder since a uniform (g, ℓ, n)-oNOSF source is also a uniform $(g \cdot n/1000, \ell \cdot n/1000, 1000)$-oNOSF source. Hence, we greatly improve the parameters while using different and much simpler techniques.

1.3 Extraction from oNOSF Sources

Next we discuss our positive and negative results on the limits of extraction from oNOSF sources. Our upper bound results (explicit extractors) are based on a novel connection to leader election and coin-flipping protocols; to instantiate this connection and give explicit extractors, we construct novel protocols for these distributed problems. Our lower bounds are based on a new notion of influence of functions, namely *online influence*, that we introduce and analyze.

Extraction Lower bounds via Online Influence

For simplicity, we focus on the case of $n = 1$, which leads to interesting new questions about Boolean functions. We refer to such uniform $(g, \ell, 1)$-oNOSF sources as (g, ℓ)-oNOBF sources; oNOBF stands for online non-oblivious bit-fixing sources. We ask what is the exact tradeoff between g, ℓ, and ε for extracting from oNOBF sources. Towards this, we introduce the notion of online influence.

▶ **Definition 9** (Online influence). *For a function $f : \{0,1\}^\ell \to \{0,1\}$, the* online influence *of the i-th bit is*

$$\mathbf{oI}_i[f] = \mathop{\mathbb{E}}_{x\sim\mathbf{U}_{i-1}}\left[\left|\mathop{\mathbb{E}}_{y\sim\mathbf{U}_{\ell-i}}[f(x,1,y)] - \mathop{\mathbb{E}}_{y\sim\mathbf{U}_{\ell-i}}[f(x,0,y)]\right|\right]$$

and the total online influence *is* $\mathbf{oI}[f] = \sum_{i=1}^{\ell} \mathbf{oI}_i[f]$.

We establish new structural results on online influence, including a Poincaré-style inequality and use them to obtain the following extraction lower bound.

▶ **Theorem 10.** *For $\varepsilon < 0.01$, there do not exist extractors for $(0.97\ell, \ell)$-oNOBF sources with error at most ε.*

A similar extraction lower bound was shown in [1] using different techniques.

Explicit Extractors via Leader Election Protocols

Here we present our explicit constructions of extractors for oNOBF and oNOSF sources. The following are our main results.

▶ **Theorem 11.** *There exists an explicit function $\mathsf{Ext} : \{0,1\}^\ell \to \{0,1\}$ such that for any (g,ℓ)-oNOBF source $\mathbf{X}$ where $g \geq \ell - \ell/(C\log(\ell))$, we have $\mathsf{Ext}(\mathbf{X}) \approx_{\varepsilon=1/100} \mathbf{U}_1$, where C is a large constant.*

▶ **Theorem 12.** *There exists an explicit function $\mathsf{Ext} : (\{0,1\}^n)^\ell \to \{0,1\}^n$ such that for any (g,ℓ,n)-oNOSF source $\mathbf{X}$ where $g \geq \ell - \ell/(C\log^*(\ell))$ and $n \geq \log(\ell)$, we have $\mathsf{Ext}(\mathbf{X}) \approx_{\varepsilon=1/100} \mathbf{U}_n$, where C is a large constant.*

It is instructive to contrast our results with the non-online setting (where adversarial bits may depend on any good bit), called NOSF sources and NOBF sources. For both these sources, the current best extractors require $g \geq \ell - \frac{\ell}{(\log \ell)^2}$, which is much more than what Theorem 11 and Theorem 12 require.

We contrast the results for both settings in Tables 1 and 2. In these tables, we are providing known upper and lower bounds on the value of $b(\ell)$, defined as the maximum number of bad symbols for which extraction is still possible with a small constant error – so lower bounds correspond to best known constructions of such functions and upper bounds refers to the best known limitation of such functions. We write "$O(\ell)$" to mean "$c\ell$ for some small universal constant $c < 1$".

To interpret our results in terms of (online) influence of coalitions, it will be useful to extend the definition of online influence to subsets of coordinates. We formally do so in the full version of the paper. Intuitively, we're measuring the influence of the exact same adversary as in an oNOSF source. Using this connection, we observe that online-resilient functions are equivalent to extractors for uniform oNOSF source sources (with one bit output). Thus, our explicit extractor results immediately imply explicit online-resilient functions. We present this connection in the full version of the paper.

Our main technique for Theorems 11 and 12 is a new generic way to transform leader election and coin flipping protocols into extractors for oNOBF and oNOSF sources. The general idea of constructing an extractor is to simulate an appropriate leader election

[8] Recall that this lower bound is for (g,ℓ,n)-oNOSF sources with $n \geq \log(\ell)$.

Table 1 $b(\ell)$ bounds in the non-online setting.

Source	Lower bound	Upper bound
NOBF	$\Omega\left(\frac{\ell}{\log^2 \ell}\right)$, [2]	$O\left(\frac{\ell}{\log \ell}\right)$, [28]
NOSF	$\Omega\left(\frac{\ell}{\log^2 \ell}\right)$, [2]	$O(\ell)$, [8]

Table 2 $b(\ell)$ bounds in the online setting.

Source	Lower bound	Upper bound
oNOBF	$\Omega\left(\frac{\ell}{\log \ell}\right)$, [Theorem 11]	$O(\ell)$, Theorem 10 or [1]
oNOSF	$\Omega\left(\frac{\ell}{\log^* \ell}\right)$, [Theorem 12] [8]	$O(\ell)$, [1]

protocol with the source at hand (oNOBF or oNOSF), and output according to the chosen leader. To instantiate this transformation, we revisit previous leader election protocols. Our leader election protocols provide a slightly stronger than usual guarantee: a good player is elected as the leader with probability close to 1. This contrasts with the usual guarantee in leader election protocols, where a good leader is chosen with only a non-trivial (constant) probability. We give more connections to distributed computing in Section 3 where we delineate applications of our results to collection coin flipping and collective sampling. For further details, see the full version of our paper.

2 Proof Overview

Our proof overview begins by outlining our new explicit result for condensers in Section 2.1 that is able to handle polylogarithmic block length. Next, we present our transformation of low-entropy oNOSF sources to uniform oNOSF sources in Section 2.2 before discussing our existential results for condensers that can handle constant block length in Section 2.3. We present the main ideas behind our results regarding online influence and extractor lower bounds in Section 2.4. In Section 2.5, we overview our extractor constructions for oNOBF and oNOSF sources that are based on a general transformation from leader election protocols. We refer the reader to the full version of this paper for more details and for formal proofs of all our results.

2.1 Explicit Condensers for Uniform oNOSF Sources

We sketch here our constructions (as well as proof ideas) of explicit condensers for uniform $(0.51\ell, \ell, n)$-oNOSF sources.[9] The goal will be to construct explicit condensers that work with as few good sources as possible while minimizing the block length n. In particular, we will show:

▶ **Theorem 13.** *For all $\varepsilon > 0$ and $n, \ell \in \mathbb{N}$ where $n \geq \left(\frac{\log(\ell)}{\varepsilon}\right)^{\Omega(1)}$, there exists an explicit condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any $(g = 0.51\ell, \ell, n)$-oNOSF source $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - \left(\frac{\log(\ell)}{\varepsilon}\right)^{O(1)} \cdot \log(n)$ where $m = 0.001 \cdot \ell n - O\left(\ell \log(\ell) \log(1/\varepsilon)\right)$.*

[9] Since all sources are uniform here, we will not explicitly mention this again.

We will require two main tools to show this. The first one uses ideas from the leader election literature and allows us to sample an $O(\log(\ell))$ sized committee starting from ℓ players while essentially maintaining the fraction of bad players. Formally:

▶ **Lemma 14.** *For all $\varepsilon_s > 0$, $n, \ell \in \mathbb{N}$, and constant $\varepsilon_a > 0$ where $n \geq \Omega(\log(\ell)\log(1/\varepsilon_s))$, there exists an explicit function* $\mathsf{oNOSFSamp} : (\{0,1\}^n)^\ell \to [\ell]^D$ *where $D \leq O(\log(\ell/\varepsilon_s))$ with the following property.*[10] *For all $S \subset [\ell]$ and (g, ℓ, n)-oNOSF sources $\mathbf{X}$, we have that*

$$\Pr_{x \sim \mathbf{X}}\left[\left|\frac{|\mathsf{oNOSFSamp}(x) \cap S|}{D} - \frac{|S|}{\ell}\right| \geq \varepsilon_a\right] \leq \varepsilon_s$$

At a high level, to construct $\mathsf{oNOSFSamp}$, we slightly modify the committee selection procedure from [36] and instantiate it with a seeded extractor with near optimal dependence on the seed [40].

Our second tool is a seeded condenser for general min-entropy sources $\mathbf{X}$ that uses an oNOSF source $\mathbf{Y}$ as the seed, where the bad bits in $\mathbf{Y}$ can depend on $\mathbf{X}$.

▶ **Lemma 15.** *There exists a constant C_{2Cond} such that for all $n_x, k, n_y, t \in \mathbb{N}$ with $\varepsilon > 0$ and $n_y \geq (C_{\mathsf{2Cond}})^t \log(tn_x/\varepsilon)$, there exists an explicit condenser* $\mathsf{2Cond} : \{0,1\}^{n_x} \times (\{0,1\}^{n_y})^t \to \{0,1\}^m$ *where $m = \frac{1}{3}(k - (C_{\mathsf{2Cond}})^t \log(tn_x/\varepsilon))$ so that the following holds: For all (n_x, k)-sources $\mathbf{X}$ and $(g = 1, \ell = t)$-oNOSF sources $\mathbf{Y} \sim (\{0,1\}^{n_y})^t$ such that the good blocks in $\mathbf{Y}$ are independent of $\mathbf{X}$ and the bad blocks in $\mathbf{Y}$ can depend on $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{2Cond}(\mathbf{X}, \mathbf{Y})) \geq m - (C_{\mathsf{2Cond}})^t \log(tn_x/\varepsilon)$.*

We will sketch how to construct this in Section 2.1.6. Using these tools, we are ready to present our explicit condenser. In fact, we will provide sketches of five different constructions of explicit condensers with increasingly better parameter dependence; the fifth (and final) one is the construction given by Theorem 13. The parameters they will vary in are the fraction of good blocks (i.e., g/ℓ) and the block length n. As we will see, each construction will build on ideas from the previous construction.

2.1.1 Construction 1: 51% good and $n \geq 2^{\Omega(\ell)}$

We here construct the following condenser:

▶ **Theorem 16.** *For all $\varepsilon > 0$ and $n, \ell \in \mathbb{N}$ where $n \geq 2^{\Omega(\ell)} \log(1/\varepsilon)$, there exists an explicit condenser* $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ *such that for any $(g = 0.51\ell, \ell, n)$-oNOSF source $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - 2^{O(\ell)} \log(1/\varepsilon)$ where $m = 0.0001 \cdot n\ell$.*

Proof sketch. Let $\gamma = 0.01$ and $\ell' = \ell/2$. We decompose the input $\mathbf{X}$ into two equal sized parts so that $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$ where both $\mathbf{X}_1, \mathbf{X}_2 \sim (\{0,1\}^n)^{\ell/2} \equiv (\{0,1\}^n)^{\ell'}$. Since $\mathbf{X}$ has $(0.5 + \gamma)\ell$ good players, we infer that each of $\mathbf{X}_1, \mathbf{X}_2$ has at least $\gamma\ell = (2\gamma) \cdot \ell'$ good players. In particular, we use the fact that $H_\infty(\mathbf{X}_1) \geq (2\gamma) \cdot (\ell n/2)$ and that $\mathbf{X}_2$ is a $(g = (2\gamma)\ell/2, \ell/2, n)$-oNOSF source. With this, we let $\mathsf{2Cond}$ be the condenser from our second tool Lemma 15 and let our final output be $\mathsf{2Cond}(\mathbf{X}_1, \mathbf{X}_2)$. Note that here $t = \ell$ and so, this requires $n \geq (C_{\mathsf{2Cond}})^\ell \log(\ell n/\varepsilon)$, an inequality that we indeed satisfy. The guarantees from Lemma 15 provide us with the desired claim. ◀

[10] Even though the output domain of $\mathsf{oNOSFSamp}$ is a vector, we will abuse notation and often treat it as a set.

Each of the subsequent constructions will use a similar construction idea as above. However, they will try to decrease t as much as possible, where t is the number of players in the oNOSF source when applying 2Cond from Lemma 15. Note that any reduction results in a decrease in the block length requirement n.

2.1.2 Construction 2: 67% good and $n \geq \text{poly}(\ell)$

Our next construction requires a slightly larger fraction of good blocks. However, the block length required is exponentially improved.

▶ **Theorem 17.** *For all $\varepsilon > 0$ and $n, \ell \in \mathbb{N}$ where $n \geq \left(\frac{\ell}{\varepsilon}\right)^{\Omega(1)}$, there exists an explicit condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any $(g = 0.67\ell, \ell, n)$-oNOSF source $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - \left(\frac{\ell}{\varepsilon}\right)^{O(1)} \log(n)$ where $m = 0.001 \cdot \ell n$.*

Proof sketch. Let $\gamma = 0.67 - (2/3)$. Let $\ell' = \ell/3$. We decompose the input $\mathbf{X}$ into three equal sized parts so that $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$ where all $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \sim (\{0,1\}^n)^{\ell'}$. Since $\mathbf{X}$ has $((2/3) + \gamma)\ell$ good players, we easily see that each $\mathbf{X}_i$ is a $(g = 3\gamma\ell', \ell', n)$-oNOSF source.

We use $\mathbf{X}_1$ to sample a $O(\log(\ell))$ sized committee from $\mathbf{X}_3$. To do so, we use $\mathsf{oNOSFSamp}_{1\to 3} : (\{0,1\}^n)^{\ell'} \to (\{0,1\}^{\ell'})^D$ from Lemma 14 with $S \subset [\ell']$ being the set of good players from $\mathbf{X}_3$, the approximation factor $\varepsilon_a = \gamma$ and sampling error $\varepsilon_s = \varepsilon/3$. Let $\mathcal{C}_3 \subset [\ell'], |\mathcal{C}_3| \leq D_3 = O(\log(\ell'/\varepsilon)) = O(\log(\ell/\varepsilon))$ be the committee of players thus obtained. The approximation property of the sampler guarantees us that out of $\geq 3\gamma\ell'$ good players in $\mathbf{X}_3$, at least $2\gamma\,|\mathcal{C}_3|$ many good players will be in $\mathcal{C}_3$ with probability $1 - \varepsilon/3$. Let $\mathbf{Y}_3$ be the $(2\gamma D_3, D_3, n)$-oNOSF source obtained by restricting the players in $\mathbf{X}_3$ to the committee $\mathcal{C}_3$. We finally use 2Cond from Lemma 15 and output $\mathsf{2Cond}(\mathbf{X}_2, \mathbf{Y}_3)$. Here, the parameter t in Lemma 15 will be set to $D \leq O(\log(\ell/\varepsilon))$, and hence, Lemma 15 would only require that $n \geq (C_{\mathsf{2Cond}})^t \log(\ell n/\varepsilon) = \left(\frac{\ell}{\varepsilon}\right)^{O(1)}$, a condition that we do meet. We carefully compute the remaining parameters to infer the claim. ◀

2.1.3 Construction 3: 76% good and $n \geq \text{poly}(\log(\ell))$

We build on our previous construction and show how to condense when the block length requirement is again exponentially decreased. This comes at a cost of slightly larger fraction of good blocks.

▶ **Theorem 18.** *For all $\varepsilon > 0$ and $n, \ell \in \mathbb{N}$ where $n \geq \left(\frac{\log(\ell)}{\varepsilon}\right)^{\Omega(1)}$, there exists an explicit condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any $(g = 0.76\ell, \ell, n)$-oNOSF source $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - \left(\frac{\log(\ell)}{\varepsilon}\right)^{O(1)} \log(n)$ where $m = 0.001 \cdot \ell n$.*

Proof sketch. Let $\gamma = 0.01$. Let $\ell' = \ell/4$. We decompose the input $\mathbf{X}$ into four equal sized parts such that $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4)$ and conclude that each $\mathbf{X}_i$ is a $(g = 4\gamma\ell', \ell', n)$-oNOSF source. However here, we claim something stronger. Call $i \in [\ell']$ a totally good index if it corresponds to a good player across each of the four blocks. Since $\mathbf{X}$ has $(3/4 + \gamma) \cdot (4\ell')$ good players, there must be $\geq 4\gamma\ell'$ totally good indices.

We first use $\mathbf{X}_1$ to sample a $D_2 = O(\log(\ell/\varepsilon))$ sized committee $\mathcal{C}_2 \subset [\ell']$ using the sampler from Lemma 14 such that $\mathcal{C}_2$ will have at least 3γ fraction of totally good indices. We let $\mathbf{Y}_2$ be the $(g = 3\gamma D_2, D_2, n)$-oNOSF source obtained by restricting $\mathbf{X}_2$ to indices from $\mathcal{C}_2$.

Second, we use $\mathbf{Y}_2$ to sample a $D_4 = O(\log(\log(\ell)/\varepsilon))$ sized committee $\mathcal{C}_4 \subset \mathcal{C}_2$ such that $\mathcal{C}_4$ has at least 2γ fraction of totally good indices. We let $\mathbf{Y}_4$ be the $(g = 2\gamma D_4, D_4, n)$-oNOSF source obtained by restricting $\mathbf{X}_4$ to indices from $\mathcal{C}_4$.

Third and last, we use $\mathsf{2Cond}$ from Lemma 15 and output $\mathsf{2Cond}(\mathbf{X}_3, \mathbf{Y}_4)$. Here, the parameter t in Lemma 15 will be set to $D_4 \leq O(\log(\log(\ell)/\varepsilon))$, and hence, Lemma 15 would only require that $n \geq (C_{\mathsf{2Cond}})^t \log(\ell n/\varepsilon) = \left(\frac{\log(\ell)}{\varepsilon}\right)^{O(1)}$, a condition that we do meet. We carefully compute the remaining parameters to infer the claim. ◀

2.1.4 Construction 4: 67% good and $n \geq \mathbf{poly}(\log(\ell))$

Our next construction maintains a similar guarantee as before on the block length and decreases the requirement on the fraction of good blocks.

▶ **Theorem 19.** *For all $\varepsilon > 0$ and $n, \ell \in \mathbb{N}$ where $n \geq \left(\frac{\log(\ell)}{\varepsilon}\right)^{\Omega(1)}$, there exists an explicit condenser $\mathsf{Cond} : (\{0,1\}^n)^\ell \to \{0,1\}^m$ such that for any $(g = 0.67\ell, \ell, n)$-oNOSF source $\mathbf{X}$, we have that $H^\varepsilon_\infty(\mathsf{Cond}(\mathbf{X})) \geq m - \left(\frac{\log(\ell)}{\varepsilon}\right)^{O(1)} \log(n)$ where $m = 0.001 \cdot \ell n$.*

Proof sketch. Let $\gamma = 0.67 - (2/3)$. Let $\ell' = \ell/3$. We decompose $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$ so that each $\mathbf{X}_i$ is a $(g = 3\gamma\ell', \ell', n)$-oNOSF source with $3\gamma\ell'$ totally good indices.

For the first step, we again use $\mathbf{X}_1$ along with a sampler from Lemma 14 to obtain $\mathcal{C}_2 \subset [\ell']$ with $D_2 = |\mathcal{C}_2| \leq O(\log(\ell/\varepsilon)$ and a subsource of $\mathbf{X}_2$ restricted to $\mathcal{C}_2$ - Y_2 that is a $(g = 2\gamma D_2, D_2, n)$-oNOSF source. For the second step, we again use $\mathbf{Y}_2$ to sample a $D_3 = O(\log(\log(\ell)/\varepsilon))$ sized committee $\mathcal{C}_3 \subset \mathcal{C}_2$ such that $\mathcal{C}_3$ has $\geq \gamma$ fraction of totally good indices. We let $\mathbf{Y}_3$ be the $(g = \gamma D_3, D_3, n)$-oNOSF source obtained by restricting $\mathbf{X}_3$ to indices from $\mathcal{C}_3$. Third and last, we use $\mathsf{2Cond}$ from Lemma 15 and output $\mathsf{2Cond}(\mathbf{X}_2, \mathbf{Y}_3)$. Again the parameter t in Lemma 15 will be set to $D_3 \leq O(\log(\log(\ell)/\varepsilon))$ and would only require that $n \geq \left(\frac{\log(\ell)}{\varepsilon}\right)^{O(1)}$.

Analyzing this construction requires more care since we use $\mathbf{X}_2$ to both sample from $\mathbf{X}_3$ and as a source for $\mathsf{2Cond}$. We use the chain rule for min-entropy to argue that most fixings of $\mathbf{Y}_2 = y_2$ will leave $\mathbf{X}_2$ with lot of entropy (since sampler requires few random bits) and observe that such a fixing still leaves $\mathbf{X}_3$ as oNOSF source with the same parameters. Also since for most fixings of $\mathbf{Y}_2 = y_2$, the committee $\mathcal{C}_3$ has γ fraction of good players, we obtain our claim. ◀

2.1.5 Construction 5: 51% good and $n \geq \mathbf{poly}(\log(\ell))$

We lastly construct the condenser promised in our main result (Theorem 13).

Proof sketch of Theorem 13. Let $\gamma = 0.01$. Let $\ell' = \ell/2$. We decompose $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)$ so that each $\mathbf{X}_i$ is a $(g = 2\gamma\ell', \ell', n)$-oNOSF source with $2\gamma\ell'$ totally good indices.

For the first step, we let $\mathbf{X}'_1$ be the $(2\gamma\ell', \ell', n'_1)$-oNOSF source obtained by taking a prefix of length n'_1 from each source where we set $n'_1 \ll n$ but also n'_1 is long enough to be used by the sampler from Lemma 14. We use $\mathbf{X}'_1$ with such a sampler to obtain a committee $\mathcal{C}_{1\to2} \subset [\ell']$ with $D_{1\to2} = |\mathcal{C}_{1\to2}| \leq O(\log(\ell/\varepsilon)$ and a subsource of $\mathbf{X}_2$ restricted to $\mathcal{C}_{1\to2}$, which we call $\mathbf{Y}_{1\to2}$, that is a $(g = (3/2)\gamma D_{1\to2}, D_{1\to2}, n)$-oNOSF source.

We argue that: 1) most fixings of $\mathbf{X}'_1$ are such that they leave $\mathbf{X}_1$ with high entropy, and 2) that the committee $\mathcal{C}_{1\to2}$ obtained will have $3\gamma/2$ fraction of good players. We obtain this using the chain rule for min-entropy and by guarantees of the sampler. We condition on such a fixing from here on.

For the second step, we use $\mathbf{Y}_{1\to2}$ to sample a $D_{2\to2} = O(\log(\log(\ell)/\varepsilon))$ sized committee $\mathcal{C}_{2\to2} \subset \mathcal{C}_{1\to2}$ such that $\mathcal{C}_{2\to2}$ has $\geq \gamma$ fraction of totally good indices. We let $\mathbf{Y}_{2\to2}$ be the $(g = \gamma D_{2\to2}, D_{2\to2}, n)$-oNOSF source obtained by restricting $\mathbf{X}_2$ to indices from $\mathcal{C}_{2\to2}$.

Lastly, we use 2Cond from Lemma 15 and output $\mathsf{2Cond}(\mathbf{X}_1, \mathbf{Y}_{2\to 2})$. The parameter t in Lemma 15 will be set so that it would only require that $n \geq \left(\frac{\log(\ell)}{\varepsilon}\right)^{O(1)}$. We must be a bit careful about the error parameter $\varepsilon_{\mathsf{2Cond}}$ for 2Cond and we will choose it to be extremely small.

Analyzing this construction requires care since we use $\mathbf{X}_2$ both to sample from within $\mathbf{X}_2$ itself and also as a seed for 2Cond. We cannot use the chain rule since a fixing of $\mathbf{Y}_{1\to 2}$ will destroy the structure of the source $\mathbf{X}_2$. We first see that the sampler guarantees that no matter how the adversary behaves, with probability $1 - \varepsilon/4$, the sampler will succeed in selecting a committee $\mathcal{C}_{2\to 2}$ with γ fraction of good players. We pay $\varepsilon/4$ in error and now assume that the sampler always succeeds in doing so. We then compare two scenarios: one scenario Opt where the bits in $\mathbf{Y}_{1\to 2}$ are all uniform and independent of all other bits and another scenario Adv where the bits in $\mathbf{Y}_{1\to 2}$ are all controlled by an adversary (guarantees on this latter scenario suffice for our claim). Let the total number of bits in $\mathbf{Y}_{1\to 2}$ be equal to $b_{1\to 2}$. Under scenario Opt, we easily see that we succeed and with error $\varepsilon_{\mathsf{2Cond}}$ will have high entropy - say k. We compare this to scenario Adv and use the fact that the adversary can only control few bits, to conclude that in scenario Adv, with error $\varepsilon_{\mathsf{2Cond}} \cdot 2^{b_{1\to 2}}$, the output will have entropy $k - b_{1\to 2}$. Since we carefully chose $\varepsilon_{\mathsf{2Cond}}$ to be small enough and $b_{1\to 2}$ is small since we only use $\mathbf{Y}_{1\to 2}$ as source for the sampler, the output will still have small error and will have high-entropy as desired. ◀

2.1.6 Construction of 2Cond

We now sketch how to construct our desired 2Cond.

Proof sketch of Lemma 15. Let $\mathbf{X}$ and $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_t)$ be the two sources. For $i \in [t]$, let $n_i \approx C^{t-i+1} \log(tn_x/\varepsilon)$ where C is a large constant. For $i \in [t]$, let $\mathbf{Z}_i$ be the length n_i prefix of the block $\mathbf{Y}_i$. Our final construction will be the parity of the outputs of seeded extractors applied with source $\mathbf{X}$ and seeds $\mathbf{Z}_i$. More formally, we output

$$\bigoplus_{i=1}^{t} \mathsf{sExt}_i(\mathbf{X}, \mathbf{Z}_i),$$

where sExt_i is any explicit near optimal seeded extractor such a from [25].

We proceed to sketch the analysis. We are guaranteed that there exists at least one $j \in [t]$ from $\mathbf{Y}$ that is good. We first condition on fixing blocks $\mathbf{Z}_1, \ldots, \mathbf{Z}_{j-1}$. Since these blocks can depend on $\mathbf{X}$, we apply the chain rule for min-entropy and conclude $\mathbf{X}$ will only lose some small amounts of entropy (the amount will be very small since these blocks are tiny compared to the amount entropy in $\mathbf{X}$). Moreover, since the adversary is online, $\mathbf{Z}_j$ remains uniform even after this fixing. We now view our construction as

$$g(\mathbf{X}) \oplus \bigoplus_{i=j}^{t} \mathsf{sExt}_i(\mathbf{X}, \mathbf{Z}_i)$$

where g is the fixed function obtained by fixing $\mathbf{Y}_1, \ldots, \mathbf{Y}_{j-1}$.

We now compare two scenarios: (1) Where all of $\mathbf{Z}_j, \ldots, \mathbf{Z}_t$ are uniform (2) Only $\mathbf{Z}_j$ is uniform and $\mathbf{Z}_{j+1}, \ldots, \mathbf{Z}_t$ are arbitrarily controlled by an adversary and can even depend on $\mathbf{X}$:

In the first scenario, we further condition on fixing $\mathbf{Z}_{j+1}, \ldots, \mathbf{Z}_t$. Since in this scenario $\mathbf{Z}_i$ are independent and random, $\mathbf{X}$ retains the same entropy and $\mathbf{Y}_j$ remains uniform. So our overall output is of the form $h(\mathbf{X}) \oplus \mathsf{sExt}_j(\mathbf{X}, \mathbf{Z}_j)$ for some fixed function h. We condition

on fixing output $h(\mathbf{X})$. Since the number of output bits $m \ll H_\infty(\mathbf{X})$, we apply the chain rule to infer that $\mathbf{X}$ still has lots of entropy when we do this fixing. Now the output is just $z \oplus \mathsf{sExt}_j(\mathbf{X}, \mathbf{Z}_j)$ where z is a fixed string, and hence is uniform.

The second scenario is more realistic and, in the worst case, this is what can actually happen. We then use the result that if an adversary controls few bits in the input distribution, then they cannot make the output of the condenser too bad. With this, since we carefully chose geometrically decreasing lengths of $\mathbf{Z}_i$ to help control the error, we indeed obtain that the output will be condensed. ◀

2.2 Converting Low-Entropy oNOSF Sources to Uniform oNOSF Sources

They key part of our proof for condensing from low-entropy oNOSF sources is a transformation from low-entropy oNOSF sources to uniform oNOSF sources. Here, we sketch the proof for our transformation in Theorem 6 and compare it to that of [10]. Both these transformations rely on two-source extractors as a basic primitive.

Given a (g, ℓ, n, k)-oNOSF source $\mathbf{X} = \mathbf{X}_1, \ldots, \mathbf{X}_\ell$, [10] uses excellent existential two-source extractors to define output blocks $\mathbf{O}_i = 2\mathsf{Ext}(\mathbf{X}_1 \circ \cdots \circ \mathbf{X}_{i-1}, \mathbf{X}_i)$ for $i \in \{2, \ldots, \ell\}$ and define their transformation as $f(\mathbf{X}) = \mathbf{O}_2, \ldots, \mathbf{O}_\ell$. They show that $\mathbf{O}_i$ is a good block if: (1) $\mathbf{X}_i$ is a good block and (2) at least one block amongst $\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}$ is a good block. They showed that such a good block will be uniform and independent of the blocks $\mathbf{O}_2, \ldots, \mathbf{O}_{i-1}$ and argued there will be $g - 1$ such good output blocks. This indeed shows their output is a uniform $(g - 1, \ell - 1, m)$-oNOSF source. However, each of their output blocks has length $m = O\left(\frac{k}{\ell}\right) \leq O\left(\frac{n}{\ell}\right)$, and so they were not able to handle the case of $n = o(\ell)$. We improve on their construction by using a "sliding window" based technique to obtain a much better transformation that can even handle $n = \mathrm{poly}(\log(\ell))$.

▶ **Theorem 20.** *Let $d, g, g_{out}, \ell, n, m, k, \varepsilon$ be such that $g_{out} \leq g - \frac{\ell - g + 2}{d}, n \geq k \geq \log(nd - k) + md + 2\log(2g_{out}/\varepsilon)$. Then, there exists a function $f : (\{0,1\}^n)^\ell \to (\{0,1\}^m)^{\ell-1}$ such that for any (g, ℓ, n, k)-oNOSF source $\mathbf{X}$, there exists uniform $(g_{out}, \ell - 1, m)$-oNOSF source $\mathbf{Y}$ for which $|f(\mathbf{X}) - \mathbf{Y}| \leq \varepsilon$.*

The parameter d in our theorem statement above is the width of our sliding window. When we set $d = \ell$ we recover the analysis of [10]. The true advantage of our transformation emerges when d is very small compared to ℓ. For instance, when $g = 0.51\ell, n = \mathrm{poly}(\log(\ell))$ and $k = \mathrm{poly}(\log(\ell))$, we set d to be a large constant and conclude that the output distribution is a uniform $(0.509\ell, \ell, \mathrm{poly}(\log(\ell))$-NOSF source.

Proof sketch of Theorem 20. Define $\mathbf{O}_i = 2\mathsf{Ext}(\mathbf{X}_{i-d} \circ \cdots \circ \mathbf{X}_{i-1}, \mathbf{X}_i)$. We call $\mathbf{O}_i$ a good output block when $\mathbf{X}_i$ is good and there is at least one good block amongst $\{\mathbf{X}_{i-d}, \ldots, \mathbf{X}_{i-1}\}$.

We first compute the number of good output blocks g_{out}. Let $j_1, \ldots, j_g$ be the indices of the good input blocks in $\mathbf{X}$ and $d_i = j_{i+1} - j_i$ be the gap between the i-th good block and the next $(i+1)$-th good block. If the gap d_i is at most d, then $\mathbf{O}_{i+1}$ must be a good output block. So, g_{out} is the number of i such that $d_i \leq d$. Since $g \geq 0.51\ell$, such large gaps cannot appear too often and we compute that $g_{out} \geq g - \frac{\ell - g + 2}{d}$ as desired.

Next, we show that the good output blocks are indeed uniform conditioned on all previous output blocks. With this, we will obtain that the output distribution will be uniform $(g_{out}, \ell - 1, m)$-oNOSF source as desired. Let i be the index of a good output block. We want to show that $\mathbf{O}_i$ is uniform conditioned on $\mathbf{O}_1, \ldots, \mathbf{O}_{i-1}$. To do this, we first observe that any input block contributes to at most $d + 1$ good output blocks. This means

that$(\mathbf{X}_{i-d} \circ \cdots \circ \mathbf{X}_{i-1})$, which has min-entropy at least k, loses at most $d \cdot m$ min-entropy conditioned on fixing $\mathbf{O}_1, \ldots, \mathbf{O}_{i-1}$. Moreover, $\mathbf{X}_i$ still remains uniform and independent of $(\mathbf{X}_{i-d} \circ \cdots \circ \mathbf{X}_{i-1})$ when fixing these previous output blocks. Hence, the output of the two-source extractor will indeed be uniform as desired. ◀

We can make Theorem 20 explicit by using the explicit two-source extractors at a slight cost of dependence on m and ε.

2.3 Existence of oNOSF Condensers for All ℓ and n

Here we sketch the proof of Theorem 7. This result states that when $g = 0.51\ell$ and $n = 1000$, there exists a condenser Cond for uniform (g, ℓ, n)-oNOSF sources so that the output entropy rate is 0.99, the number of output bits is $m = O(\ell + \log(1/\varepsilon))$, and the error of the condenser is ε where $\varepsilon \leq 2^{-\Omega(\ell)}$ is arbitrary.

Our construction uses amazing seeded condensers with $1 \cdot \log(1/\varepsilon)$ dependence on seed length. We slightly modify our source and then apply such seeded condenser. Here is a proof sketch:

Proof sketch for Theorem 7. Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_\ell)$ be such a source. Let $\mathbf{Y}_1 \sim (\{0,1\}^n)^{0.5\ell}$ be the source obtained by concatenating the first 0.5ℓ blocks of $\mathbf{X}$. Since 0.51ℓ blocks are good, there exist at least 0.01ℓ uniform blocks in $\mathbf{Y}_1$. We treat $\mathbf{Y}_1$ as a single distribution over $n\ell$ bits with min-entropy $\geq 0.01\ell n$. Let $\mathbf{Y}_2 \sim \{0,1\}^{0.5\ell}$ be the source obtained by concatenating 1 bit from each of the last 0.5ℓ blocks of $\mathbf{X}$. Once again, since 0.51ℓ blocks are good, there exist at least 0.01ℓ uniform bits in $\mathbf{Y}_2$. We will use the following seeded condenser:

▶ **Theorem 21.** *For all d, ε such that $d \geq \log(\ell n/\varepsilon) + O(1)$, there exists a seeded condenser $\mathsf{sCond} : \{0,1\}^{0.5\ell n} \times \{0,1\}^d \to \{0,1\}^m$ such that for all $\mathbf{X} \sim \{0,1\}^{0.5\ell n}$ with $H_\infty(\mathbf{X}) \geq 0.01\ell n$, we have $H^\varepsilon_\infty(\mathsf{sCond}(\mathbf{X}, \mathbf{U}_d)) \geq 0.01\ell n + d$ where $m = 0.01\ell n + d + \log(1/\varepsilon) + O(1)$.*

Our condenser Cond will output $\mathsf{sCond}(\mathbf{Y}_1, \mathbf{Y}_2)$. Observe that not only is $\mathbf{Y}_2$ not uniform, there could be as many as 0.49ℓ "bad bits" in $\mathbf{Y}_2$ that can depend on $\mathbf{Y}_1$. To remedy this, we use the well known fact that the behavior of such adversarial $\mathbf{Y}_2$ cannot be far worse than the behavior if $\mathbf{Y}_2$ were uniform. Concretely, suppose if $\mathbf{Y}_2$ were uniform and the output entropy and error were k and ε. Then for the actual $\mathbf{Y}_2$, the output entropy will be $k - 0.49\ell$ and error will be $\varepsilon \cdot 2^{0.49\ell}$.

For us, it means the following: let $\varepsilon_{\mathsf{sCond}}, k_{\mathsf{sCond}}$ be such that $H^{\varepsilon_{\mathsf{sCond}}}_\infty(\mathsf{sCond}(\mathbf{Y}_1, \mathbf{U}_{0.5\ell})) \geq k_{\mathsf{sCond}}$. Then, it must be that $H^{2^{0.49\ell} \cdot \varepsilon_{\mathsf{sCond}}}_\infty(\mathsf{sCond}(\mathbf{Y}_1, \mathbf{Y}_2)) \geq k_{\mathsf{sCond}} - 0.49\ell$. So, for our final error to be some ε, we need to have $\varepsilon_{\mathsf{sCond}} = \varepsilon \cdot 2^{-0.49\ell}$. For seeded condensers to exist, we need $0.5\ell \geq \log(\ell n/\varepsilon_{\mathsf{sCond}}) + O(1)$ and we check that such an inequality can indeed be satisfied if $\varepsilon \geq 2^{-0.01\ell}$.

Hence, we finally obtain that our seeded condenser will output $0.01\ell n + O(\ell)$ bits and will have output entropy $m - \Delta$ where $\Delta = O(\ell)$. Hence, if n is a large enough constant, our output entropy rate, $\frac{m-\Delta}{m}$, will be ≥ 0.99 as desired. ◀

▶ Remark 22. Here (in the inequality $0.5\ell \geq 1 \cdot \log(\ell n/\varepsilon_{\mathsf{sCond}})$) we crucially used the fact that there exist seeded condensers with seed length dependence $1 \cdot \log(1/\varepsilon)$. Currently, we do not have explicit constructions with this dependence. We also could not have used a seeded extractor since for them, the seed length dependence is $2 \cdot \log(1/\varepsilon)$. For that to work, we would need to assume $g \geq 0.76\ell$.

2.4 Online Influence and Extractor Lower Bounds

In this subsection, we provide a brief overview of our results regarding online influence and sketch how they imply extractor lower bounds against oNOBF sources. We also contrast this with the established notion of influence for Boolean functions. For any function $f : \{0,1\}^n \to \{0,1\}$, define the function $e(f)(x) = (-1)^{f(x)}$.

A Poincaré inequality and extractor lower bounds

One fundamental inequality about regular influence is the Poincaré inequality which states that $\mathrm{Var}(f) \leq \mathbf{I}[f]$. We prove a similar result for online influence.

▶ **Theorem 23.** *For any $f : \{0,1\}^\ell \to \{0,1\}$, we have $\mathrm{Var}(e(f)) \leq \mathbf{oI}[f] \leq \sqrt{\ell \,\mathrm{Var}(e(f))}$.*

It is not hard to derive extractor lower bounds for oNOBF sources from the above result. The high level idea is to collect bits with high online influence, which is guaranteed by the first inequality in the above theorem (using an averaging argument) to form a *coalition of coordinates* that has enough online influence to bias the claimed extractor.

The proof of Theorem 23 is based on techniques from the Fourier analysis of Boolean functions. The following key result implies Theorem 23.

▶ **Lemma 24.** *For any $f : \{0,1\}^\ell \to \{0,1\}$ and $i \in [\ell]$,*

$$\mathbf{oI}_i(f)^2 \leq \sum_{\substack{S \subseteq [i] \\ S \ni i}} \widehat{f}(S)^2 \leq \mathbf{oI}_i(f).$$

Influence vs Online Influence

It is not hard to see that $\mathbf{oI}_i[f] \leq \mathbf{I}_i[f]$ for all $i \in [\ell]$, with equality always holding for $i = \ell$ as an adversarial online bit in the last index can see every good bit. Moreover, we observe that for monotone functions, the notion of online influence is equivalent to regular influence, so any separation between the two notions must come from non-monotone functions.

We exactly exhibit such a separation via the address function $\mathrm{Addr}_\ell : \{0,1\}^{\log \ell + \ell} \to \{0,1\}$ which considers its first $\log \ell$ bits as an index in $\{1, \ldots, \ell\}$ and then outputs the value of the chosen index. It is easy to show that the first $\log \ell$ bits of Addr_ℓ have no online influence, while the remaining bits have online influence of $O\left(\frac{1}{\ell}\right)$. This is in contrast to the well known result of [28] showing that, for a balanced function such as Addr_ℓ, there must exist a bit with influence at least $\Omega\left(\frac{\log \ell}{\ell}\right)$.

2.5 Extractors via Leader Election Protocols

We sketch our main idea for constructing an extractor for oNOBF sources. Similar ideas work more generally for extracting from oNOSF sources. As mentioned above, we use a novel connection to leader election protocols to construct extractors.

Suppose π is an $(r-1)$-round leader election protocol over ℓ players where in each round, each player sends 1 bit and with the guarantee that if there are at most $\delta\ell$ bad players, then a good player is chosen as leader with probability $1 - \epsilon$. Suppose $\mathbf{X}$ is an $(g, \ell r)$-oNOBF source, where $g \geq lr - \delta\ell$. We simply partition the bits of $\mathbf{X}$ into chunks $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_r$, where each X_i is on ℓ bits, and simulate the protocol π by using the j'th bit of $\mathbf{X}_i$ as the message of the j'th player in round i, for all $1 \leq j \leq \ell$ and $1 \leq i \leq r-1$. At the end of this simulation suppose $j^* \in [\ell]$ is the chosen leader. Then we output the j^*'th bit of $\mathbf{X}_r$ as the output of the extractor.

Briefly, the reason that the above is a valid simulation of π is the fact that the value of any bad bit in this online setting just depends on bits that appear before it, which is allowed in the leader election protocol (where in round i, the message of a bad player can be any function of the messages in the same round or previous rounds). The correctness of the extractor now follows from the fact that since the number of bad players (i.e., bad bits in $\mathbf{X}$) is at most $\delta\ell$, the guarantee of the protocol ensures that the chosen leader $j^* \in [\ell]$ is a good player with probability at least $1 - \epsilon$, and in this case the j^*'th bit of $\mathbf{X}_r$ must be uniform.

We note here that in the usual definition of leader election protocols, the requirement is to select a good leader with constant probability, which is a weaker guarantee than what we need to instantiate the above plan. It turns out that we can combine leader election protocols from prior works, in particular from [18] and [3], to construct protocols with the stronger guarantee we require.

3 Application to Collective Coin Flipping and Collective Sampling

We now discuss applications of our results on condensers for oNOSF sources to fault-tolerant distributed computing. Condensing from oNOSF sources can be viewed as a special case of coin flipping and collective sampling protocols in the full information model that arise in fault-tolerant distributed computing.

3.1 Background

Say there are ℓ players who have a common broadcast channel and want to jointly perform a task such as collectively flipping a coin. Some b players out of them are "bad" and want to deter the task. We assume the bad players are computationally unbounded so cryptographic primitives are of no use. We further assume that each player has private access to uniform randomness. [5] initiated the study of this model and aptly termed this task as "collective coin flipping."

The simplest way to collectively flip a coin would be for all the players to initially agree on a function $f : \{0,1\}^\ell \to \{0,1\}$, then synchronously broadcast one random bit r_i, and to finally agree on the output being $f(r_1, \ldots, r_\ell)$. However, synchronizing broadcasts is hard, and it could be that the bad players set their output as function of the bits of the good players. [28] showed that no function f can handle more than $O\left(\frac{\ell}{\log \ell}\right)$ corruptions.

One way to allow for more corruptions (almost linear) among players is to consider "protocols" that allow more rounds of communication. In particular, a protocol can be thought of as a tree where each vertex represents a "round" where in every round the following happens: all good players sends their bits, then all bad players send their bits as a function of the bits of the good players, and they jointly compute a function of these bits. Depending on the outcome of the function, everyone branches on one branch in this tree. Furthermore, every leaf is labeled with final outcomes (say 0 or 1) and, once a leaf is reached, that is the outcome that everybody agrees on. [20] initiated the study of protocols where the outcomes are from a larger range and where the bad players are trying to minimize the largest probability of any outcome. They called this problem "collective sampling."

3.1.1 Known Results

[5] showed that for protocols with outcomes $\{0,1\}$, b bad players can always ensure that some outcome occurs with probability at least $\frac{1}{2} + \frac{b}{2\ell}$. [3] first constructed a protocol that can handle a linear number of corruptions. Follow-up works tried to reduce the number of rounds in this protocol where, in some settings, players were allowed to send more than one bit per round [36, 18].

[20] showed that for all collective sampling protocols and all outcomes, there exists a way for b bad players to coordinate and ensure that an outcome that happens without corruption with probability p, now happens with probability $p^{1-(b/n)} \geq p\left(1 + \frac{b}{n}\log(1/p)\right)$. Nearly matching collective sampling protocols were constructed by [20, 37, 24]. For an overview of further results and bounds, see [13].

3.2 Connection to oNOSF Sources

The problem of extracting or condensing from oNOSF sources can be seen as special cases or variants of collective coin flipping and collective sampling that provide very simple protocols. For instance, suppose one has an extractor or condenser f for uniform (g, ℓ, n)-oNOSF sources. Then, consider a protocol where all ℓ players take turns and output n random bits. The agreed final outcome is f applied on these ℓn bits. This leads to protocols that are structurally much simpler since players don't have to carefully compute whose turn it is to go in various rounds and can obliviously prepare for their turn.

The above protocol can also be viewed as a relaxed version of a 1-round protocol where instead of everyone providing their output asynchronously, they take turns and provide outputs one after another in a simple sequential manner.

3.2.1 Previous Results Interpreted in oNOSF source context

Previous impossibility results can be interpreted in the context of extracting / condensing from uniform oNOSF sources. For instance, collective coin flipping impossibility results of [5] imply extraction impossibility results for uniform (g, ℓ, n)-oNOSF sources when $n = 1$. They imply:

▶ **Corollary 25.** *There does not exist an $\frac{b}{2\ell}$-extractor for uniform $(g, \ell, 1)$-oNOSF sources.*

Similarly, we observe that the notion of collective sampling is equivalent to 0-error condensing. Hence, lower bounds of [20] imply 0-error condensing lower bounds for uniform (g, ℓ, n)-oNOSF sources when $n = 1$. Formally:

▶ **Corollary 26.** *There does not exist a condenser $\mathsf{Cond} : \{0,1\}^\ell \to \{0,1\}^m$ for uniform $(g, \ell, 1)$-oNOSF sources that can guarantee output smooth min-entropy (with parameter $\varepsilon = 0$) more than $k = \frac{g}{\ell} \cdot m$.*

3.2.2 ε-Collective Sampling

Since collective sampling lower bounds show that for any protocol, 0-error condensing beyond rate g/ℓ is impossible, one can naturally ask whether condensing with small error ε is possible. We call this problem *ε-collective sampling*, where the goal is to output a distribution which is ε-close to a distribution where every output has small probability.

Interpreted this way, this is exactly what protocols arising out of our condensers for uniform oNOSF sources provide: Using Theorem 7, when each player has access to 10^4 random bits, there exists a simple protocol that can handle 0.49ℓ corrupt players such that the players can collectively sample a distribution over $m = O(\ell)$ bits which is $2^{-\Omega(\ell)}$-close to having entropy $0.99m$. As far as we are aware, such a protocol is not implied by any other previous protocol. Most previous protocols are obtained through *leader election* protocols, which do not seem useful here since the leader has access to only constant number of bits.

We similarly obtain explicit protocols using Theorem 4 for the case when each player has access to $n \geq \text{poly}(\log(\ell)/\varepsilon)$ many bits.

3.2.3 Collective Coin Flipping and Sampling with Weak Random Sources

A natural extension to collective coin flipping and sampling in the full information model is when all players only have access to weak source of randomness (that are independent from each other) instead of true uniform randomness. This question was first studied by [21]. [29] used network extractor protocol to transform weak random sources of each player into independent private random sources. This way, after using the network extraction protocol, players can follow the usual collective coin flipping / sampling protocol. [23] improved the network extraction protocol using two-source non-malleable extractors.

Using our (g, ℓ, n, k)-oNOSF source condensers, we obtain alternative, simple ε-collective sampling protocols in the setting where players have access to weak sources of randomness. We obtain such an existential protocol using Theorem 8, and explicit protocol using Corollary 5.

4 Open Problems

We list here some interesting open problems left by our work:

- While we obtain explicit condensers for almost all parameter regimes, it remains open to construct them when the bock length is constant, matching the parameters of our existential results. As we show, one way of achieving this would be to explicitly construct a seeded condenser with dependence on seed length being $1 \cdot \log(1/\varepsilon)$.
- All our condensers have entropy gap much larger than a constant. It will be interesting to show there exist condensers with constant entropy gap (for any values of n, ℓ) for uniform oNOSF sources. A slightly weaker but equally interesting question is to construct seeded extractors for uniform oNOSF sources with constant seed length.
- Show that there exist non-trivial condensers for oNOBF sources or show no such condenser exists. We conjecture that no condenser exists with output entropy rate larger than the input entropy rate for such sources.
- Construct ε-collective sampling protocols with fewer rounds than the ones obtained using uniform oNOSF source condensers. It will also be interesting to explicitly construct such protocols when the number of players are very large compared to the number of bits each player has access to. Further, proving lower bounds for ε-collective sampling protocols is a natural direction to explore.
- Determine the exact threshold for extracting from oNOBF sources and oNOSF sources. Our lower bounds show extraction is impossible when $g \leq 0.99\ell$ while our constructions using leader election protocols require $g \geq \ell - \Omega\left(\frac{\ell}{\log \ell}\right)$ for oNOBF sources and $g \geq \ell - \Omega\left(\frac{\ell}{\log^*(\ell)}\right)$ for (g, ℓ, n)-oNOSF sources where $n \geq \log(\ell)$. Using the connection between extractors and leader election protocols, lower bounds for extraction imply lower bounds for leader election protocols. In particular, matching lower bounds for extraction would imply all current leader election protocols are tight, a long standing open problem.

References

1 Divesh Aggarwal, Maciej Obremski, João Ribeiro, Luisa Siniscalchi, and Ivan Visconti. How to Extract Useful Randomness from Unreliable Sources. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, Lecture Notes in Computer Science, pages 343–372, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-45721-1_13.

2 Miklós Ajtai and Nathan Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993. doi:10.1007/BF01303199.

3 Noga Alon and Moni Naor. Coin-flipping games immune against linear-sized coalitions. *SIAM J. Comput.*, 22(2):403–417, 1993. doi:10.1137/0222030.

4 Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged Public-Key Encryption: How to Protect against Bad Randomness. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, pages 232–249, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-10366-7_14.

5 Michael Ben-Or and Nathan Linial. Collective Coin Flipping. *Advances In Computing Research*, 5:91–115, 1989. URL: https://www.cs.huji.ac.il/~nati/PAPERS/coll_coin_fl.pdf.

6 Iddo Bentov, Ariel Gabizon, and David Zuckerman. Bitcoin Beacon, 2016. arXiv:1605.04559 [cs]. doi:10.48550/arXiv.1605.04559.

7 Joseph Bonneau, Jeremy Clark, and Steven Goldfeder. On Bitcoin as a public randomness source, 2015. Publication info: Preprint. MINOR revision. URL: https://eprint.iacr.org/2015/1015.

8 Jean Bourgain, Jeff Kahn, Gil Kalai, Yitzhak Katznelson, and Nathan Linial. The influence of variables in product spaces. *Israel Journal of Mathematics*, 77(1):55–64, 1992. doi:10.1007/BF02808010.

9 Benedikt Bünz, Steven Goldfeder, and Joseph Bonneau. Proofs-of-delay and randomness beacons in ethereum. *IEEE Security and Privacy on the blockchain (IEEE S&B)*, 2017.

10 Eshan Chattopadhyay, Mohit Gurumukhani, and Noam Ringach. On the existence of seedless condensers: Exploring the terrain. In *Proceedings of the 65th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2024. To appear.

11 Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. *Annals of Mathematics*, 189(3):653–705, 2019. doi:10.4007/annals.2019.189.3.1.

12 Benny Chor, Oded Goldreich, Johan Hasted, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*, SFCS '85, pages 396–407, USA, 1985. IEEE Computer Society. doi:10.1109/SFCS.1985.55.

13 Yevgeniy Dodis. Fault-tolerant leader election and collective coin-flipping in the full information model, 2006.

14 Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. Key derivation without entropy waste. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 93–110. Springer, 2014. doi:10.1007/978-3-642-55220-5_6.

15 Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Almost Chor-Goldreich Sources and Adversarial Random Walks. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1–9, New York, NY, USA, June 2023. Association for Computing Machinery. doi:10.1145/3564246.3585134.

16 Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Online Condensing of Unpredictable Sources via Random Walks. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference (CCC 2025)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:17, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2025.30.

17 Zeev Dvir, Ariel Gabizon, and Avi Wigderson. Extractors and rank extractors for polynomial sources. *Comput. Complex.*, 18(1):1–58, 2009. doi:10.1007/S00037-009-0258-4.

18 Uriel Feige. Noncryptographic selection protocols. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 142–153. IEEE Computer Society, 1999. doi:10.1109/SFFCS.1999.814586.

19 O. Goldreich, S. Goldwasser, and N. Linial. Fault-tolerant computation in the full information model. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 447–457, 1991. doi:10.1109/SFCS.1991.185405.

20 Oded Goldreich, Shafi Goldwasser, and Nathan Linial. Fault-tolerant computation in the full information model. *SIAM J. Comput.*, 27(2):506–544, 1998. doi:10.1137/S0097539793246689.

21 Shafi Goldwasser, Madhu Sudan, and Vinod Vaikuntanathan. Distributed computing with imperfect randomness. In Pierre Fraigniaud, editor, *Distributed Computing, 19th International Conference, DISC 2005, Cracow, Poland, September 26-29, 2005, Proceedings*, volume 3724 of *Lecture Notes in Computer Science*, pages 288–302. Springer, 2005. doi:10.1007/11561927_22.

22 Jesse Goodman, Xin Li, and David Zuckerman. Improved condensers for chor-goldreich sources. In *Proceedings of the 65th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2024. To appear.

23 Vipul Goyal, Akshayaram Srinivasan, and Chenzhi Zhu. Multi-source non-malleable extractors and applications. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 468–497. Springer, 2021. doi:10.1007/978-3-030-77886-6_16.

24 Ronen Gradwohl, Salil P. Vadhan, and David Zuckerman. Random selection with an adversarial majority. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006. doi:10.1007/11818175_25.

25 Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM*, 56(4):20:1–20:34, 2009. doi:10.1145/1538902.1538904.

26 Peter Ivanov, Raghu Meka, and Emanuele Viola. Efficient resilient functions. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2867–2874. SIAM, 2023. doi:10.1137/1.9781611977554.CH108.

27 Peter Ivanov and Emanuele Viola. Resilient functions: Optimized, simplified, and generalized. *CoRR*, abs/2406.19467, 2024. doi:10.48550/arXiv.2406.19467.

28 J. Kahn, G. Kalai, and N. Linial. The influence of variables on Boolean functions. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 68–80, 1988. doi:10.1109/SFCS.1988.21923.

29 Yael Tauman Kalai, Xin Li, Anup Rao, and David Zuckerman. Network extractor protocols. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 654–663. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.73.

30 Jesse Kamp and David Zuckerman. Deterministic Extractors for Bit-Fixing Sources and Exposure-Resilient Cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, January 2007. doi:10.1137/S0097539705446846.

31 Xin Li. Extractors for a Constant Number of Independent Sources with Polylogarithmic Min-Entropy. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 100–109, USA, 2013. IEEE Computer Society. doi:10.1109/FOCS.2013.19.

32 Xin Li. New independent source extractors with exponential improvement. In *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, STOC '13, pages 783–792, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2488608.2488708.

33 Raghu Meka. Explicit Resilient Functions Matching Ajtai-Linial. In *Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 1132–1148. Society for Industrial and Applied Mathematics, 2017. doi:10.1137/1.9781611974782.73.

34 Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, 1995. doi:10.1017/CBO9780511814075.

35 Cécile Pierrot and Benjamin Wesolowski. Malleability of the blockchain's entropy. *Cryptography Commun.*, 10(1):211–233, 2018. doi:10.1007/s12095-017-0264-3.

36 Alexander Russell and David Zuckerman. Perfect information leader election in log* n+o (1) rounds. *J. Comput. Syst. Sci.*, 63(4):612–626, 2001. doi:10.1006/JCSS.2001.1776.

37 Saurabh Sanghvi and Salil P. Vadhan. The round complexity of two-party random selection. *SIAM J. Comput.*, 38(2):523–550, 2008. doi:10.1137/050641715.

38 Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 32–42. IEEE Computer Society, 2000. doi:10.1109/SFCS.2000.892063.

39 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012. doi:10.1561/0400000010.

40 David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3:103–128, August 2007. Number: 6. doi:10.4086/toc.2007.v003a006.

Improved Parallel Repetition for GHZ-Supported Games via Spreadness

Yang P. Liu
Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Shachar Lovett
Department of Computer Science and Engineering, University of California, San Diego, CA, USA

Kunal Mittal
Department of Computer Science, Courant Institute of Mathematical Sciences,
New York University, NY, USA

Abstract

We prove that for any 3-player game $\mathcal{G}$, whose query distribution has the same support as the GHZ game (i.e., all $x, y, z \in \{0,1\}$ satisfying $x + y + z = 0 \pmod 2$), the value of the n-fold parallel repetition of $\mathcal{G}$ decays exponentially fast:

$$\mathrm{val}(\mathcal{G}^{\otimes n}) \leq \exp(-n^c)$$

for all sufficiently large n, where $c > 0$ is an absolute constant.

We also prove a concentration bound for the parallel repetition of the GHZ game: For any constant $\epsilon > 0$, the probability that the players win at least a $\left(\frac{3}{4} + \epsilon\right)$ fraction of the n coordinates is at most $\exp(-n^c)$, where $c = c(\epsilon) > 0$ is a constant.

In both settings, our work exponentially improves upon the previous best known bounds which were only polynomially small, i.e., of the order $n^{-\Omega(1)}$. Our key technical tool is the notion of *algebraic spreadness* adapted from the breakthrough work of Kelley and Meka (FOCS '23) on sets free of 3-term progressions.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Parallel Repetition, GHZ Game, Algebraic Spreadness

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.6

Related Version *Full Version*: https://arxiv.org/abs/2602.09290 [46]

Funding *Shachar Lovett*: Supported by Simons Investigator Award #929894 and NSF award CCF-2425349.
Kunal Mittal: Research supported a Simons Investigator Award.

1 Introduction

In a k-player game $\mathcal{G}$, a verifier samples a tuple of questions $(x^{(1)}, \ldots, x^{(k)})$ from a distribution Q. Then, for each $j \in \{1, \ldots, k\}$, the verifier gives the question $x^{(j)}$ to player j, and the player gives back an answer $a^{(j)}$, which depends only on $x^{(j)}$. The verifier now declares whether the players win or lose based on the evaluation of a predicate $V(x, a) \in \{0, 1\}$ depending on the questions $x = (x^{(1)}, \ldots, x^{(k)})$ and answers $a = (a^{(1)}, \ldots, a^{(k)})$. We define the game value, denoted $\mathrm{val}(\mathcal{G})$, as the maximum winning probability (with respect to the distribution Q) over all possible player strategies; see Definitions 14, 15 for formal definitions.

A natural question that arises is: How does the value of the game behave under parallel repetition [29]? The n-fold parallel repetition, denoted $\mathcal{G}^{\otimes n}$, is a game where the players play, and try to win, n independent copies of the game in parallel. More precisely, the verifier samples questions $(x_i^{(1)}, \ldots, x_i^{(k)}) \sim Q$ independently for $i = 1, \ldots, n$, and for each

© Yang P. Liu, Shachar Lovett, and Kunal Mittal;
licensed under Creative Commons License CC-BY 4.0

$j \in \{1, \ldots, k\}$, sends questions $(x_1^{(j)}, \ldots, x_n^{(j)})$ to player j, to which they give back answers $(a_1^{(j)}, \ldots, a_n^{(j)})$. The verifier says the players win if $V((x_i^{(1)}, \ldots, x_i^{(k)}), (a_i^{(1)}, \ldots, a_i^{(k)})) = 1$ for each $i \in \{1, \ldots, n\}$; see Definition 17 for a formal definition.

Note that for any game $\mathcal{G}$, it holds that $\mathrm{val}(\mathcal{G}^{\otimes n}) \geq \mathrm{val}(\mathcal{G})^n$, since the players can achieve this value by repeating an optimal strategy in each of the n coordinates. Although one might expect the naïve bound $\mathrm{val}(\mathcal{G}^{\otimes n}) \leq \mathrm{val}(\mathcal{G})^n$ to hold as well, this turns out to be false [28, 22, 26, 55]. Roughly speaking, this failure occurs because the players do not have to treat the n copies of the game independently, and can instead correlate their answers among different copies. Remarkably, it turns out that this failure is intimately connected to the geometry of high-dimensional Euclidean tilings [25, 45, 1, 14].

Parallel repetition of 2-player games is well-understood. Raz [53] showed that for any game with value less than 1, the value of the n-fold parallel repetition decays exponentially in n. Subsequent works have simplified this proof and strengthened the quantitative bounds [38, 6, 52, 56, 21, 12]. These and related works have led to several applications in various domains, including the theory of interactive proofs [8], PCPs and hardness of approximation [24, 2, 3, 4, 7, 23, 36, 43, 44, 35, 18, 20], quantum information [17, 16], and communication complexity [50, 5, 15]. The reader is referred to the survey [54] for more details.

Parallel repetition of multiplayer games is much less understood. The only general bound says that for any game $\mathcal{G}$ with value less than 1, it holds that $\mathrm{val}(\mathcal{G}^{\otimes n}) \leq 1/\alpha(n)$, where $\alpha(n)$ is a very slowly growing inverse-Ackermann function [58, 30, 51]. Recent work has made substantial progress in understanding special cases of multiplayer games [19, 39, 31, 32, 33, 13, 10, 9, 11]; however, the general question remains wide open, even for 3-player games.

Proving improved parallel repetition for multiplayer games has several potential applications. It is known that a strong parallel repetition theorem for a certain class of multiplayer games implies super-linear lower bounds for non-uniform Turing machines, which is a long-standing open problem in complexity theory [48]. Additionally, parallel repetition in the large answer alphabet regime is equivalent to many problems in high-dimensional extremal combinatorics, such as the density Hales-Jewett problem, and the problem of square-free sets in finite fields [26, 37, 47]. Also, as stated in [19], it is believable that improved understanding of parallel repetition can lead to a better understanding of communication complexity in the number-on-forehead (NOF) model, which is intimately connected to circuit lower bounds.

The focus of this paper is the 3-player GHZ game [34], which proceeds as follows: The verifier samples questions $(x, y, z) \in \{0, 1\}^3$ uniformly at random such that $x + y + z = 0 \pmod 2$, and the players' goal is to give answers $a, b, c \in \{0, 1\}$ respectively satisfying $a + b + c \pmod 2 = x \vee y \vee z$. The GHZ game has played a foundational role in quantum information theory, due in part to the fact that quantum strategies can win this game with probability 1, whereas any classical strategy wins with probability at most 3/4. Moreover, the GHZ game satisfies a self-testing property, which says that all quantum strategies achieving value 1 are essentially the same; this has led to applications like entanglement testing and device-independent cryptography [57].

The problem of parallel repetition of the GHZ game has been discussed in several recent works. Dinur, Harsha, Venkat and Yuen [19] extend the 2-player information-theoretic techniques of Raz [53] and prove parallel repetition for a specific class of multiplayer games satisfying a certain connectivity property; they identify the GHZ game as a multiplayer game which is in some sense maximally far from this class of games; regarding the difficulty of the problem, they write

> "We believe that the strong correlations present in the GHZ question distribution represent the "hardest instance" of the multiplayer parallel repetition problem."

Subsequent research established polynomial decay bounds for the n-fold repetition of the GHZ game, showing $\mathrm{val}(\mathrm{GHZ}^{\otimes n}) \leq n^{-\Omega(1)}$ [39, 31]. Notably, these results apply to any 3-player game $\mathcal{G}$ whose query distribution Q has the same support as the GHZ game (i.e., $\{(x,y,z) \in \{0,1\}^3 : x+y+z = 0 \pmod 2\}$).

More recently, an exponential decay bound was proven for the GHZ game [13], and then extended to all 3-player XOR games[1] satisfying a certain distributional assumption [9]. However, these works rely heavily on the XOR structure of the game predicate, exploiting it via sophisticated tools from Fourier analysis and additive combinatorics. For instance, in the case of the GHZ game, they observe that the predicate corresponds to a linear equation over the group $\mathbb{Z}/4\mathbb{Z}$. Consequently, these techniques do not extend to general game predicates lacking such XOR structure.

In this work, we improve upon the existing polynomial decay bounds [39, 31] by establishing a stretched exponential bound for all games sharing the query support of the GHZ game. Crucially, unlike the aforementioned works on XOR games [13, 9], our result is agnostic to the answer sets and the game predicate – it solely depends on the support of the query distribution. Our technical approach is distinct as well: we utilize spreadness-based arguments from recent works [42, 41, 40], whose application to the context of parallel repetition is novel.

Our main result is the following:

▶ **Theorem 1** (Parallel Repetition for the GHZ query support). *Let $\mathcal{G}$ be any 3-player game with value $\mathrm{val}(\mathcal{G}) < 1$, whose query distribution has support*

$$\left\{(x,y,z) \in \{0,1\}^3 : x+y+z = 0 \pmod 2\right\}.$$

Then, for all sufficiently large n,[2] it holds that

$$\mathrm{val}(\mathcal{G}^{\otimes n}) \leq \exp\left(-n^c\right),$$

where $c > 0$ is an absolute constant.

We complement this result with a concentration bound for games with the GHZ query distribution. To the best of our knowledge, the exponential decay bounds established in prior works [13, 10] do not imply such concentration. We prove the following:

▶ **Theorem 2** (A Concentration Bound; restated and proved as Theorem 42). *Let $\mathcal{G}$ be any 3-player game with value $\mathrm{val}(\mathcal{G}) < 1$, whose query distribution is uniform over the set*

$$\left\{(x,y,z) \in \{0,1\}^3 : x+y+z = 0 \pmod 2\right\}.$$

Then, for every constant $\epsilon > 0$, there exists $c = c(\epsilon) > 0$, such that for all sufficiently large n,[3] the probability that the players can win at least $\mathrm{val}(\mathcal{G}) + \epsilon$ fraction of the n coordinates in the game $\mathcal{G}^{\otimes n}$ is at most $\exp\left(-n^c\right)$.

As a direct consequence, we obtain a concentration bound for the standard GHZ game.

▶ **Corollary 3.** *For every constant $\epsilon > 0$, there exists $c = c(\epsilon) > 0$, such that the probability of the players winning at least $\left(\frac{3}{4} + \epsilon\right) n$ coordinates in the game $\mathrm{GHZ}^{\otimes n}$ is at most $\exp\left(-n^c\right)$.*

[1] In a 3-player XOR game, the answers a, b, c of the three players lie in some finite Abelian group H, and the predicate is of the form $a + b + c = \varphi(x,y,z)$, for some function φ mapping questions (x,y,z) of the players to elements of H.

[2] i.e., $n \geq N$, where N is a constant depending on the game $\mathcal{G}$.

[3] i.e., $n \geq N$, where N is a constant depending on the game $\mathcal{G}$ and the parameter ϵ.

It remains an interesting open problem to establish analogous concentration bounds (as in Theorem 2) for games over the GHZ support where the underlying distribution is not uniform.[4]

Organization

In Section 2, we give an overview of our proofs. In Section 3, we establish some preliminaries. In Section 4, we define a notion of pseudorandomness, called *algebraic spreadness*, that will be useful throughout this paper; we also show how to decompose arbitrary sets into components satisfying this spreadness condition. In Section 5, we show that any *diagonal-product* set composed of algebraically spread sets is *uniformly covered* by *squares*. Finally, in Section 6, we use the results of the previous sections and prove parallel repetition for games with the GHZ query support (Theorem 1). In Section 7, we prove our concentration bound (Theorem 2).

2 Overview

In this section, we outline the proof of our main result (Theorem 1).

2.1 General Inductive Framework and High-Level Approach

We begin by introducing a general inductive framework for parallel repetition. Our proof follows the basic setup established by Raz for parallel repetition of 2-player games [53].

Let $\mathcal{G}$ be a 3-player game, whose query distribution Q has support[5]

$$\mathrm{supp}(Q) = \left\{(x,y,z) \in \mathbb{F}_2^3 : x+y+z=0\right\}.$$

Here, we identified $\{0,1\}$ with the finite field $\mathbb{F}_2$. For the purpose of parallel repetition, we may assume without loss of generality that Q is the uniform distribution over its support (see Lemma 18).[6] Now, consider the game $\mathcal{G}^{\otimes n}$, and consider any strategy for the 3 players in this game. For each $i \in [n]$, let Win_i be the event that the players win the i^{th} coordinate. By the chain rule, for any permutation $i_1, i_2, \ldots, i_n$ of $[n]$, we can write the winning probability as

$$\Pr[\mathrm{Win}_1 \wedge \mathrm{Win}_2 \wedge \cdots \wedge \mathrm{Win}_n] = \prod_{k=1}^{n} \Pr[\mathrm{Win}_{i_k} \mid \mathrm{Win}_{i_1} \wedge \cdots \wedge \mathrm{Win}_{i_{k-1}}].$$

To bound this, we proceed inductively. Assuming the players have won a set of coordinates $i_1, \ldots, i_k$, we aim to identify a *hard coordinate* $i \in [n]$ whose conditional winning probability is at most $1 - \Omega(1)$. Formally, we wish to prove the following condition:[7]

---

[4] We remark that Theorem 2, combined with a reduction similar to Lemma 18 (also see Footnote 6), already implies the following: the probability that the players win at least $1-\beta+\epsilon$ fraction of the n coordinates is at most $\exp(-n^{\Omega_\epsilon(1)})$, where $\beta = \beta(\mathcal{G}) > 0$ is a constant.

[5] The framework applies to all k-player games, but we restrict to 3 player games with the GHZ queries.

[6] Roughly speaking, this is because any distribution Q *contains* a small copy of the uniform distribution over $\mathrm{supp}(Q)$, and hence winning n copies under the distribution Q is at least as hard as winning $\Omega(n)$ copies under the the uniform distribution.

[7] Conditioning on the event $\mathrm{Win}_{i_1} \wedge \cdots \wedge \mathrm{Win}_{i_k}$ can introduce complex correlations among the players' inputs. However, this event depends deterministically on the questions and answers of the players in coordinates $i_1, \ldots, i_k$, and hence instead of conditioning on the event $\mathrm{Win}_{i_1} \wedge \cdots \wedge \mathrm{Win}_{i_k}$, we can condition on typical questions and answers to the players in coordinates $i_1, \ldots, i_k$. This induces a product event on the players' input.

Inductive Step: For every product event $\mathcal{E} = E \times F \times G \subseteq \mathbb{F}_2^n \times \mathbb{F}_2^n \times \mathbb{F}_2^n$, with measure $\Pr[\mathcal{E}] \geq \alpha$, there exists a coordinate $i \in [n]$ such that $\Pr[\mathrm{Win}_i \mid \mathcal{E}] \leq 1 - \Omega(1)$.

Establishing this condition for $\alpha = \exp(-n^{\Omega(1)})$ implies the desired bound on $\mathrm{val}(\mathcal{G}^{\otimes n})$ via the inductive strategy above (see Lemma 19).

For the remainder of this overview, consider any product event $\mathcal{E} = E \times F \times G \subseteq (\mathbb{F}_2^n)^3$, with measure $\Pr[\mathcal{E}] \geq \alpha = \exp(-n^{\Omega(1)})$. Our goal is to identify a coordinate $i \in [n]$ that remains hard to win when the inputs are conditioned on $\mathcal{E}$. The proof proceeds in two main steps:

1. **Identify Hard Sets:** We define a class of sets called *squares*, such that if the input distribution is restricted to a square, many coordinates are hard to win.
2. **Distributional Approximation:** We show that the conditional distribution $Q^{\otimes n}|\mathcal{E}$ can be approximated (in ℓ_1 distance) by a convex combination of such square distributions.

Combining these steps establishes that $\Pr[\mathrm{Win}_i \mid \mathcal{E}] \leq 1 - \Omega(1)$ for randomly chosen $i \in [n]$.

2.2 Step 1: Squares are Hard

Since the third player's input in $\mathcal{G}^{\otimes n}$ is fully determined by the inputs to the first two players (inputs $(x, y, z) \in \mathrm{supp}(Q)^n \subseteq (\mathbb{F}_2^n)^3$ satisfy $z = x + y$), we can analyze the game primarily by only looking at inputs of the first two players. The restriction of a product event $\mathcal{E} = E \times F \times G$ to the first two players is captured by a *diagonal-product set*, defined as follows:

▶ **Definition 4** (Diagonal-Product Set). *Given sets $X, Y, Z \subseteq \mathbb{F}_2^n$, we define the corresponding diagonal-product set, denoted $S(X, Y, Z)$, as*

$$S(X, Y, Z) = \{(x, y) \in \mathbb{F}_2^n \times \mathbb{F}_2^n : x \in X, y \in Y, x + y \in Z\}.$$

Sampling inputs $(x, y, z) \sim Q^{\otimes n}|\mathcal{E}$ is the same as sampling $(x, y) \sim S(E, F, G)$ uniformly and setting $z = x + y$.

We now define our hard sets, that are *squares* in $\mathbb{F}_2^n \times \mathbb{F}_2^n$:

▶ **Definition 5** (Square). *A* square $s_{x,y,w} \subseteq \mathbb{F}_2^n \times \mathbb{F}_2^n$, *for $x, y, w \in \mathbb{F}_2^n$, is the set*

$$s_{x,y,w} = \{(x, y), (x + w, y), (x, y + w), (x + w, y + w)\}.$$

▶ Remark 6. Given a square $s \subseteq \mathbb{F}_2^n \times \mathbb{F}_2^n$, suppose we wish to represent it as $s = s_{x,y,w}$. Note that the width w is uniquely determined by the square s. If $w = 0$, the square contains a single point, in which case (x, y) is uniquely determined. If $w \neq 0$, the square $s_{x,y,w}$ has 4 different representations, given by $s_{x,y,w} = s_{x+w,y,w} = s_{x,y+w,w} = s_{x+w,y+w,w}$.

The crucial property of a square is its local hardness: Consider a square $s = s_{x_0,y_0,w}$, and a coordinate $i \in [n]$ such that $w_i \neq 0$ (a *non-trivial* coordinate).[8] Suppose the inputs (x, y, z) to the three players are sampled from the square s as follows: let $(x, y) \sim s$ be chosen uniformly at random, and let $z = x + y$. Under this distribution, no strategy of the players can win coordinate i with probability more than $\mathrm{val}(\mathcal{G})$. Roughly speaking, this is true since this distribution "looks exactly the same" as the base distribution Q, and hence the players can embed a single copy of the game $\mathcal{G}$ into coordinate i of this distribution; see Lemma 38 for formal details of this step.[9]

[8] A coordinate is non-trivial if and only if the points in s do not have a constant value in this coordinate.

[9] We remark that the set of inputs $\{(x, y, x + y) : (x, y) \in s\}$ was introduced in the work [31], where they call this set a *bow-tie*. We also note that squares correspond to the maximal *forbidden-subgraphs* inside the GHZ query distribution (in the sense that no player strategy can win on all four points of a square), and the largest size of square-free sets in $\mathbb{F}_2^n \times \mathbb{F}_2^n$ exactly captures the value of the n-fold parallel repetition of the GHZ query distribution in the large answer alphabet regime [47].

2.3 Step 2: Approximating the Distribution

The second, and more technically demanding, step of the proof is to show that the distribution $Q^{\otimes n}|\mathcal{E}$ is well-approximated by a mixture of square distributions as above. Equivalently, we wish to approximate the uniform distribution over the diagonal-product set $S = S(E, F, G)$, denoted U_S, by a convex combination of uniform distributions over squares. Specifically, we analyze the distribution μ generated as follows: sample a square $s \subseteq S$ uniformly at random, and output an element uniformly at random from s.

Note that a priori it is not even clear that the set S contains squares, and that the distribution μ is well-defined. However, we utilize the concept of *algebraic spreadness*, a pseudorandomness notion introduced in the breakthrough work of Kelley and Meka on bounds for sets without 3-term arithmetic progressions [42]. They show that algebraically spread subsets behave like random subsets in a certain sense (see Definition 20 and Theorem 21); in our setting, we show that if all E, F, G are algebraically spread, the set S contains the expected density of squares. Our approximation argument proceeds in two stages:

Uniformization

Given arbitrary sets E, F, G, we decompose the diagonal-product set $S(E, F, G)$ into disjoint components $S(E_1, F_1, G_1) \cup S(E_2, F_2, G_2) \cdots \cup S(E_T, F_T, G_T)$ (plus a negligible remainder), such that within each component, the sets E_i, F_i, G_i are all algebraically spread. This generalizes the recent decomposition technique of [40], which achieved algebraic spreadness for only two out of the three sets and used it to prove bounds on corner-free sets in finite fields. We extend this to all three sets via a careful recursive argument; see Section 4.2 for more details.

Counting and Approximation

Assuming that all of E, F, G are algebraically spread, we demonstrate that they behave like random sets of the same density, and that the square-sampling distribution μ approximates the uniform distribution U_S in ℓ_1 distance. We prove this by establishing tight bounds on the ℓ_2 norm (collision probability) of the distribution μ. A key technical tool here is a recent graph-counting result of [27], based on a pseudorandomness notion called combinatorial spreadness, which we show holds in our setting. See Lemma 33 and Proposition 34 for more details on this step.

Why Algebraic Spreadness?

A crucial difference between our work and previous polynomial decay bounds [31] lies in the choice of the pseudorandomness property for the sets E, F, G. Prior works use the concept of Fourier-uniformity (pseudorandomness against linear tests), and yield bounds when the density $\alpha = \Pr[\mathcal{E}]$ is at least polynomial, i.e., of the order $n^{-O(1)}$. The use of algebraic spreadness enables us to surpass this barrier and get bounds even when the density $\alpha = \exp(-n^{\Omega(1)})$ is exponentially smaller. However, this improvement incurs a significant technical cost. Both the decomposition and approximation steps are standard under Fourier uniformity, whereas establishing them under algebraic spreadness constitutes the main technical contribution of this work.

3 Preliminaries

Let $\mathbb{N} = \{1, 2, \ldots\}$ denote the set of natural numbers. For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

3.1 Probability Distributions

Let P be a distribution (over an underlying finite set Ω, which is usually clear from context). We use $\mathrm{supp}(P) = \{\omega \in \Omega : P[\omega] > 0\}$ to denote the support of the distribution P. For an event $E \subseteq \Omega$ with $P[E] > 0$, we use $P|E$ to denote the conditional probability distribution P conditioned on E.

For distributions P and Q over a set Ω, the ℓ_1-distance between them is defined as

$$\|P - Q\|_1 = \sum_{\omega \in \Omega} |P[\omega] - Q[\omega]| .$$

We state a useful tail bound on the sum of independent random variables:

▶ **Fact 7** (Chernoff Bounds, see [49] for reference). *Let $X_1, \ldots, X_n \in \{0, 1\}$ be independent random variables each with mean μ, and let $X = \sum_{i=1}^{n} X_i$. Then, for all $\delta \in (0, 1)$,*

$$\Pr[X \leq (1 - \delta)\mu n] \leq e^{-\frac{\delta^2 \mu n}{2}},$$

$$\Pr[X \geq (1 + \delta)\mu n] \leq e^{-\frac{\delta^2 \mu n}{3}}.$$

We also state another useful lemma:

▶ **Lemma 8.** *Let $t, n \in \mathbb{N}$, $t < n$. Let μ be the uniform distribution on $[n]$, and let ν be the uniform distribution on $[n - t]$. Then, $\|\mu - \nu\|_1 = 2t/n$.*

Proof. We have $\|\mu - \nu\|_1 = \sum_{i=1}^{n-t} \left(\frac{1}{n-t} - \frac{1}{n}\right) + \sum_{i=n-t+1}^{n} \frac{1}{n} = \frac{2t}{n}$. ◀

3.2 Vector Spaces over GF(2)

Let $\mathcal{V}$ be a finite dimensional vector space over $\mathbb{F}_2$, with the uniform measure; often we shall have $\mathcal{V} = \mathbb{F}_2^n$ for some $n \in \mathbb{N}$.

▶ **Definition 9** (Inner Product). *For functions $f, g : \mathcal{V} \to \mathbb{R}$, define their inner product as*

$$\langle f, g \rangle = \mathop{\mathbb{E}}_{x \sim \mathcal{V}}[f(x)g(x)],$$

where $x \sim \mathcal{V}$ denotes that x is uniformly chosen from $\mathcal{V}$.

▶ **Definition 10** (L^p norm). *For $f : \mathcal{V} \to \mathbb{R}$, and $p \geq 1$, we define*

$$\|f\|_p = \mathbb{E}[f^p]^{1/p}.$$

▶ **Definition 11** (Convolution). *For $f, g : \mathcal{V} \to \mathbb{R}$, we define their convolution as*

$$(f * g)(x) = \mathop{\mathbb{E}}_{y \sim \mathcal{V}}[f(y)g(x + y)].$$

We observe the following simple fact:

▶ **Fact 12.** *For functions* $f, g, h : \mathcal{V} \to \mathbb{R}$,

$$\langle f * g, h\rangle = \langle f * h, g\rangle = \langle g * h, f\rangle = \mathop{\mathbb{E}}_{x,y\sim\mathcal{V}}[f(x)g(y)h(x+y)].$$

▶ **Definition 13** (Density Function). *For a nonempty set* $A \subseteq \mathcal{V}$*, we define its density function as*

$$\varphi_A = \frac{\mathbb{1}_A}{\mathbb{E}\left[\mathbb{1}_A\right]} = \frac{\mathbb{1}_A}{|A|/|\mathcal{V}|}.$$

Note that $\mathbb{E}[\varphi_A] = 1$.

3.3 3-Player Games and Parallel Repetition

We overview some basic definitions regarding multiplayer games. We shall restrict our focus to 3-player games.

▶ **Definition 14** (3-Player Game). *A 3-player game* $\mathcal{G}$ *is a tuple* $\mathcal{G} = (\mathcal{X}\times\mathcal{Y}\times\mathcal{Z},\ \mathcal{A}\times\mathcal{B}\times\mathcal{C},\ Q, V_{pred})$*, where the question sets* $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ *and the answer sets* $\mathcal{A}, \mathcal{B}, \mathcal{C}$ *are finite sets,* Q *is a probability distribution over* $\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$*, and* $V_{pred} : (\mathcal{X}\times\mathcal{Y}\times\mathcal{Z})\times(\mathcal{A}\times\mathcal{B}\times\mathcal{C}) \to \{0,1\}$ *is a predicate.*

The game $\mathcal{G}$ proceeds as follows: A verifier samples questions $(X, Y, Z) \sim Q$; then, the verifier sends X to player 1, Y to player 2, and Z to player 3, to which the players respond back with answers $A \in \mathcal{A},\ B \in \mathcal{B},\ C \in \mathcal{C}$ respectively. Finally, the verifier declares that the players win if and only if $V_{pred}((X,Y,Z),(A,B,C)) = 1$.

▶ **Definition 15** (Game Value). *Let* $\mathcal{G} = (\mathcal{X}\times\mathcal{Y}\times\mathcal{Z},\ \mathcal{A}\times\mathcal{B}\times\mathcal{C},\ Q, V_{pred})$ *be a 3-player game. The value of the game* $\mathcal{G}$*, denoted* $\mathrm{val}(\mathcal{G})$*, is defined as*

$$\mathrm{val}(\mathcal{G}) = \max_{f,g,h} \Pr_{(X,Y,Z)\sim Q}\left[V_{pred}\left((X,Y,Z),(f(X),g(Y),h(Z))\right) = 1\right],$$

where the maximum is over player strategies $f : \mathcal{X}\to\mathcal{A},\ g : \mathcal{Y}\to\mathcal{B},\ h : \mathcal{Z}\to\mathcal{C}$.

▶ **Fact 16.** *The game value is unchanged even if the players are allowed to use public and private randomness, since there always exists some optimal fixed values of the random strings.*

Next, we define the parallel repetition of a 3-player game, which corresponds to playing n independent copies of the game in parallel.

▶ **Definition 17** (Parallel Repetition). *Let* $\mathcal{G} = (\mathcal{X}\times\mathcal{Y}\times\mathcal{Z},\ \mathcal{A}\times\mathcal{B}\times\mathcal{C},\ Q, V_{pred})$ *be a 3-player game. We define its* n*-fold repetition as* $\mathcal{G}^{\otimes n} = (\mathcal{X}^n\times\mathcal{Y}^n\times\mathcal{Z}^n,\ \mathcal{A}^n\times\mathcal{B}^n\times\mathcal{C}^n,\ Q^{\otimes n}, V_{pred}^{\otimes n})$*. The distribution* $Q^{\otimes n}$ *is the* n*-fold product of the distribution* Q *with itself, i.e.,* $Q^{\otimes n}[(x,y,z)] = \prod_{i=1}^n Q[(x_i,y_i,z_i)]$ *for each* $x\in\mathcal{X}^n, y\in\mathcal{Y}^n, z\in\mathcal{Z}^n$*. The predicate* $V_{pred}^{\otimes n}$ *is defined as* $V_{pred}^{\otimes n}((x,y,z),(a,b,c)) = \bigwedge_{i=1}^n V_{pred}((x_i,y_i,z_i),(a_i,b_i,c_i))$.

Some Basic Results on Parallel Repetition

We state a lemma from [26], which shows that it suffices to prove parallel repetition in the case when the game's distribution is uniform over its support:

▶ **Lemma 18** ([32, Lemma 3.14]). *Let* $\mathcal{G} = (\mathcal{X}\times\mathcal{Y}\times\mathcal{Z},\ \mathcal{A}\times\mathcal{B}\times\mathcal{C},\ Q, V_{pred})$ *be a 3-player game such that* $\mathrm{val}(\mathcal{G}) < 1$*. Let* $\tilde{\mathcal{G}} = (\mathcal{X}\times\mathcal{Y}\times\mathcal{Z},\ \mathcal{A}\times\mathcal{B}\times\mathcal{C},\ U, V_{pred})$*, where* U *is the uniform distribution over* $\mathrm{supp}(Q)$*. Then,* $\mathrm{val}(\tilde{\mathcal{G}}) < 1$*, and there exists a constant* $c = c(\mathcal{G}) > 0$*, such that for every sufficiently large* $n\in\mathbb{N}$*,*

$$\mathrm{val}(\mathcal{G}^{\otimes n}) \le 2\cdot\mathrm{val}(\tilde{\mathcal{G}}^{\otimes\lfloor cn\rfloor}).$$

We state an inductive parallel repetition criterion from [53]:

▶ **Lemma 19** ([11, Lemma B.1]). *Let $\mathcal{G} = (\mathcal{X} \times \mathcal{Y} \times \mathcal{Z},\ \mathcal{A} \times \mathcal{B} \times \mathcal{C},\ Q, V_{pred})$ be a 3-player game, and consider its n-fold repetition $\mathcal{G}^{\otimes n} = (\mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n,\ \mathcal{A}^n \times \mathcal{B}^n \times \mathcal{C}^n,\ Q^{\otimes n}, V_{pred}^{\otimes n})$ for some sufficiently large $n \in \mathbb{N}$. Fix optimal strategies for the 3 players in this game, and for each $i \in [n]$, let Win_i be the event that this strategy wins the i^{th} coordinate of the game.*

Let $\epsilon > 0$ be a constant, and $\alpha \in (0, 1],\ \alpha \geq 2^{-n}$ be such that the following condition holds: For every product event $E \times F \times G \subseteq \mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n$ with $\Pr_{Q^{\otimes n}}[E \times F \times G] \geq \alpha$, there exists a coordinate $i \in [n]$ such that $\Pr[\mathrm{Win}_i \mid E \times F \times G] \leq 1 - \epsilon$.

Then, for some constant $c = c(\mathcal{G})$, it holds that $\mathrm{val}(\mathcal{G}^{\otimes n}) \leq \alpha^c$.

4 Algebraic Spreadness and Uniformization

In this section, we overview a notion of pseudorandomness called algebraic spreadness. We also show how to decompose arbitrary sets into components that satisfy this spreadness definition.

4.1 Algebraic Spreadness

▶ **Definition 20** (Algebraic Spreadness). *Let $\mathcal{V} \subseteq \mathbb{F}_2^n$ be an affine subspace. We say that a subset $A \subseteq \mathcal{V}$ is (r, ϵ)-algebraically spread within $\mathcal{V}$ if for all affine subspaces $\mathcal{V}' \subseteq \mathcal{V}$ satisfying $\dim(\mathcal{V}') \geq \dim(\mathcal{V}) - r$, it holds that*

$$\frac{|A \cap \mathcal{V}'|}{|\mathcal{V}'|} \leq (1 + \epsilon) \cdot \frac{|A|}{|\mathcal{V}|}.$$

We state a useful result about spread subsets.

▶ **Theorem 21.** *Let $d \geq 1, \epsilon \in (0, 1/4)$. Then, there exists a sufficiently large integer $r = d^8 \epsilon^{-O(1)}$, and a sufficiently small $\delta = \Omega(\epsilon)$, such that the following holds: Suppose $A, B, C \subseteq \mathbb{F}_2^n$ are sets each of size at least $2^{-d} \cdot |\mathbb{F}_2^n|$. Then,*

- **(i)** *([42, Proposition 2.16]) If at least two of A, B, C are (r, δ)-algebraically spread within $\mathbb{F}_2^n$, then*[10]

$$|\langle \varphi_A * \varphi_B,\ \varphi_C \rangle - 1| \leq \epsilon.$$

- **(ii)** *([42, Proposition 4.10]) If at least one of A, B, C is (r, δ)-algebraically spread within $\mathbb{F}_2^n$, then*

$$\langle \varphi_A * \varphi_B,\ \varphi_C \rangle \leq 1 + \epsilon.$$

4.2 Uniformization

In this section, we describe a procedure that takes an arbitrary set $X \subseteq \mathbb{F}_2^n$ and approximately decomposes it into components that are algebraically spread. More generally, we find a simultaneous decomposition for three sets X, Y, Z, compatible with the diagonal-product:

[10] For example, if both A, B are algebraically spread within $\mathbb{F}_2^n$, we have $|\langle \varphi_A * \varphi_B,\ \varphi_C \rangle - 1| \leq 2 \|\varphi_A * \varphi_B - 1\|_d \leq \epsilon$, where the first inequality follows from Hölder's inequality, and the second inequality follows from [42, Proposition 2.16].

▶ **Proposition 22.** *Let $r \in \mathbb{N}$, $\epsilon, \eta \in (0, 1/10)$, and let $X, Y, Z \subseteq \mathcal{V}$ for a linear subspace $\mathcal{V} \subseteq \mathbb{F}_2^n$. Define $\alpha = |S(X,Y,Z)| / |\mathcal{V}|^2$. Then, there exists $C_\eta = (1/\eta)^{O(1)} \geq 1$ (that depends only on η), an integer $T \in \mathbb{N}$, and for each $i \in [T]$, a linear subspace $\mathcal{V}_i \subseteq \mathcal{V}$, points $x_i, y_i \in \mathcal{V}/\mathcal{V}_i$, and subsets $X_i \subseteq x_i + \mathcal{V}_i$, $Y_i \subseteq y_i + \mathcal{V}_i$, $Z_i \subseteq x_i + y_i + \mathcal{V}_i$, such that:*

1. *$\dim(\mathcal{V}_i) \geq \dim(\mathcal{V}) - r\epsilon^{-3} \log_2(4/\alpha)^{C_\eta}$ for all $i \in [T]$.*
2. *$S(X_1, Y_1, Z_1), \ldots, S(X_T, Y_T, Z_T)$ are disjoint subsets of $S(X, Y, Z)$ such that*

$$\left|S(X,Y,Z) \setminus \cup_{i=1}^{T} S(X_i, Y_i, Z_i)\right| \leq \eta \left|S(X,Y,Z)\right|.$$

   The diagonal-product sets are defined as in Definition 4.
3. *For all $i \in [T]$, we have $|X_i|, |Y_i|, |Z_i| \geq 2^{-(\log_2(4/\alpha))^{C_\eta}} \cdot |\mathcal{V}_i|$, and X_i, Y_i, Z_i are (r, ϵ)-algebraically spread within $x_i + \mathcal{V}_i, y_i + \mathcal{V}_i, x_i + y_i + \mathcal{V}_i$ respectively.*

The proof of the above proposition is based on [40, Section 5.3], with the exception that we work to make all the 3 sets spread (instead of just 2). As a result, we obtain a weaker dependence on the parameter η compared to their two set version; however, this is inconsequential in our setting, as we use this only for constant η. In the following subsections, we first show how to decompose one set, then two sets, and finally three sets.

4.2.1 Uniformization for 1 Set

We first show how to decompose a single set X into spread components. Before that, we show that any large set contains a spread set in it.

▶ **Lemma 23** (Existence of a Spread Subset)**.** *Let $r \in \mathbb{N}$, $\epsilon \in (0, 1)$, and let $X \subseteq \mathcal{V}$ for a linear subspace $\mathcal{V} \subseteq \mathbb{F}_2^n$. Define $\alpha_X = |X| / |\mathcal{V}|$. Then, there exists an affine subspace $\mathcal{V}' \subseteq \mathcal{V}$ such that:*

1. *$\dim(\mathcal{V}') \geq \dim(\mathcal{V}) - r\epsilon^{-1} \log_2(1/\alpha_X)$.*
2. *The set $X' = X \cap \mathcal{V}'$ satisfies $\frac{|X'|}{|\mathcal{V}'|} \geq \alpha_X$, and X' is (r, ϵ)-algebraically spread within $\mathcal{V}'$.*

Proof. The reader is referred to the full version of this paper [46]. ◀

Using the above lemma, we show how to decompose a single set into spread components.

▶ **Lemma 24** (Uniformization for 1 Set)**.** *Let $r \in \mathbb{N}$, $\epsilon, \eta \in (0, 1)$. Let $X \subseteq \mathcal{V}$ for a linear subspace $\mathcal{V} \subseteq \mathbb{F}_2^n$. Then, there exists $T \in \mathbb{N}$, and for each $i \in [T]$, an affine subspace $\mathcal{V}_i \subseteq \mathcal{V}$ and a subset $X_i \subseteq \mathcal{V}_i$, such that:*

1. *$\dim(\mathcal{V}_i) \geq \dim(\mathcal{V}) - r\epsilon^{-1} \log_2(1/\eta)$ for all $i \in [T]$.*
2. *$X_1, \ldots, X_T$ are disjoint subsets of X such that $\left|X \setminus \cup_{i \in [T]} X_i\right| \leq \eta |\mathcal{V}|$.*
3. *For all $i \in [T]$, $|X_i| \geq \eta |\mathcal{V}_i|$ and X_i is (r, ϵ)-algebraically spread within $\mathcal{V}_i$.*

Proof. The reader is referred to the full version of this paper [46]. ◀

4.2.2 Uniformization for 2 Sets

Next, we state a lemma about decomposing the product $X \times Y$ of two sets X, Y into spread components.

▶ **Lemma 25** (Uniformization for 2 Sets, [40, Lemma 5.9])**.** *Let $r \in \mathbb{N}$, $\epsilon, \eta \in (0, 1/10)$, and let $X, Y \subseteq \mathcal{V}$ for a linear subspace $\mathcal{V} \subseteq \mathbb{F}_2^n$. Define $\alpha_{XY} = |X|\,|Y| / |\mathcal{V}|^2$. Then, there exists $T \in \mathbb{N}$, and for each $i \in [T]$, a linear subspace $\mathcal{V}_i \subseteq \mathcal{V}$, points $x_i, y_i \in \mathcal{V}/\mathcal{V}_i$, and subsets $X_i \subseteq x_i + \mathcal{V}_i$, $Y_i \subseteq y_i + V_i$, such that:*

1. $\dim(\mathcal{V}_i) \geq \dim(\mathcal{V}) - O\left(r\epsilon^{-2}\log_2(1/\alpha_{XY})^2\log_2(1/\eta) + r\epsilon^{-2}\log_2(1/\eta)^5\right)$ *for all* $i \in [T]$.
2. $X_1 \times Y_1, \ldots, X_T \times Y_T$ *are disjoint subsets of* $X \times Y$ *such that*

$$\left|(X \times Y) \setminus \cup_{i\in[T]}(X_i \times Y_i)\right| \leq \eta\left|X \times Y\right|$$

3. *For all* $i \in [T]$, *we have* $|X_i \times Y_i| \geq \kappa \cdot |\mathcal{V}_i|^2$ *for* $\kappa \geq 2^{-O(\log_2(1/\eta)^2)} \cdot \alpha_{XY}$, *and* X_i, Y_i *are* (r,ϵ)*-algebraically spread within* $x_i + \mathcal{V}_i, y_i + \mathcal{V}_i$ *respectively.*

4.2.3 Uniformization for 3 Sets

We shall prove Proposition 22 and show how to decompose three sets into spread components, compatible with the diagonal-product. We start by stating a simple fact about the size of a diagonal-product $S(X,Y,Z)$, which along with Theorem 21 allows us to control this size when at least one or two of X, Y, Z are spread.

▶ **Fact 26.** *Let* $X, Y, Z \subseteq \mathcal{V}$ *for a linear subspace* $\mathcal{V} \subseteq \mathbb{F}_2^n$, *and let* $\alpha_X = |X|/|\mathcal{V}|$, $\alpha_Y = |Y|/|\mathcal{V}|$, $\alpha_Z = |Z|/|\mathcal{V}|$. *Then,* $|S(X,Y,Z)| = \alpha_X\alpha_Y\alpha_Z \cdot \langle \varphi_X * \varphi_Y,\ \varphi_Z\rangle$.

First, we show a one round partitioning result for $S(X,Y,Z)$:

▶ **Lemma 27.** *Let* $\eta \in (0, 1/50)$, *and let* $X, Y, Z \subseteq \mathcal{V}$ *for a linear subspace* $\mathcal{V} \subseteq \mathbb{F}_2^n$. *Define* $\alpha = |S(X,Y,Z)|/|\mathcal{V}|^2$. *Then, there exists an integer* $r' = O(\log_2(1/(\eta\alpha))^{16})$, *and constant* $\epsilon' = \Omega(1)$, *such that the following hold:*

Let $r \in \mathbb{N}$, $\epsilon \in (0, 1/10)$ *satisfy* $r \geq r'$, $\epsilon \leq \epsilon'$. *Then, there exists* $T \in \mathbb{N}$, *a set* $\mathcal{G} \subseteq [T]$, *and for each* $i \in [T]$, *a linear subspace* $\mathcal{V}_i \subseteq \mathcal{V}$, *points* $x_i, y_i \in \mathcal{V}/\mathcal{V}_i$, *and subsets* $X_i \subseteq x_i + \mathcal{V}_i, Y_i \subseteq y_i + \mathcal{V}_i, Z_i \subseteq x_i + y_i + \mathcal{V}_i$, *such that*

1. $\dim(\mathcal{V}_i) \geq \dim(\mathcal{V}) - O\left(r\epsilon^{-3}\log_2(1/(\eta\alpha))^6\right)$ *for all* $i \in [T]$.
2. $S(X_1,Y_1,Z_1), \ldots, S(X_T,Y_T,Z_T)$ *are disjoint subsets of* $S(X,Y,Z)$ *such that*

$$\left|S(X,Y,Z) \setminus \cup_{i=1}^{T} S(X_i,Y_i,Z_i)\right| \leq 4\eta\left|S(X,Y,Z)\right|.$$

3. $|S(X_i,Y_i,Z_i)| \geq 2^{-O(\log_2(1/(\eta\alpha))^2)} \cdot |\mathcal{V}_i|^2$ *for all* $i \in [T]$.
4. *For all* $i \in \mathcal{G}$, *the sets* X_i, Y_i, Z_i *are* (r,ϵ)*-algebraically spread within* $x_i + \mathcal{V}_i, y_i + \mathcal{V}_i, x_i + y_i + \mathcal{V}_i$ *respectively.*
5. $\sum_{i\in\mathcal{G}} |S(X_i,Y_i,Z_i)| \geq \frac{1}{10} \cdot |S(X,Y,Z)|$.

Proof. The reader is referred to the full version of this paper [46]. ◀

We write down a simpler version of the above lemma, with the assumption on r, ϵ removed:

▶ **Corollary 28.** *Let* $r \in \mathbb{N}$, $\epsilon, \eta \in (1/10)$, *and let* $X, Y, Z \subseteq \mathcal{V}$ *for a linear subspace* $\mathcal{V} \subseteq \mathbb{F}_2^n$. *Define* $\alpha = |S(X,Y,Z)|/|\mathcal{V}|^2$. *Then, there exists* $T \in \mathbb{N}$, *a set* $\mathcal{G} \subseteq [T]$, *and for each* $i \in [T]$, *a linear subspace* $\mathcal{V}_i \subseteq \mathcal{V}$, *points* $x_i, y_i \in \mathcal{V}/\mathcal{V}_i$, *and subsets* $X_i \subseteq x_i + \mathcal{V}_i, Y_i \subseteq y_i + \mathcal{V}_i, Z_i \subseteq x_i + y_i + \mathcal{V}_i$, *such that*

1. $\dim(\mathcal{V}_i) \geq \dim(\mathcal{V}) - O\left(r\epsilon^{-3}\log_2(1/(\eta\alpha))^6 + \epsilon^{-3}\log_2(1/(\eta\alpha))^{22}\right)$ *for all* $i \in [T]$.
2. $S(X_1,Y_1,Z_1), \ldots, S(X_T,Y_T,Z_T)$ *are disjoint subsets of* $S(X,Y,Z)$ *such that*

$$\left|S(X,Y,Z) \setminus \cup_{i=1}^{T} S(X_i,Y_i,Z_i)\right| \leq \eta\left|S(X,Y,Z)\right|.$$

3. $|S(X_i,Y_i,Z_i)| \geq 2^{-O(\log_2(1/(\eta\alpha))^2)} \cdot |\mathcal{V}_i|^2$ *for all* $i \in [T]$.
4. *For all* $i \in \mathcal{G}$, *the sets* X_i, Y_i, Z_i *are* (r,ϵ)*-algebraically spread within* $x_i + \mathcal{V}_i, y_i + \mathcal{V}_i, x_i + y_i + \mathcal{V}_i$ *respectively.*
5. $\sum_{i\in\mathcal{G}} |S(X_i,Y_i,Z_i)| \geq \frac{1}{10} \cdot |S(X,Y,Z)|$.

Proof. We plug in the parameter $\eta/5$ in Lemma 27 to get $r' = O(\log_2(1/(\eta\alpha))^{16})$, and $\epsilon' = \Omega(1)$. Then, we use the lemma with parameters $r_0 = r + r'$ and $\epsilon_0 = \epsilon \cdot \epsilon'$, noting that any set which is (r_0, ϵ_0)-algebraically spread is also (r, ϵ)-algebraically spread. ◀

Finally, we complete the main result of this section:

Proof of Proposition 22. The reader is referred to the full version of this paper [46]. ◀

5 Uniform Square Covers

In this section, we show that any *diagonal-product* set (see Definition 4) composed of algebraically spread sets (see Definition 20) is *uniformly covered* by *squares* (see Definition 5). In Section 5.1, we introduce a combinatorial notion of spreadness that will be useful for us; in Section 5.2, we show that algebraically spread sets satisfy this definition (in a specific way); finally, in Section 5.3, we prove Proposition 34, our square covering result.

5.1 Combinatorial Spreadness and Graph Counts

We state some useful definitions from [41]:

▶ **Definition 29** (Combinatorial Spreadness). *Let X, Y be finite sets, and let $f : X \times Y \to \{0,1\}$. We say that f is (r,ϵ)-combinatorially spread if for every $S \subseteq X$, $T \subseteq Y$ satisfying $|S \times T| \geq 2^{-r} |X \times Y|$, it holds that*

$$\mathop{\mathbb{E}}_{(x,y)\sim S\times T}[f(x,y)] \leq (1+\epsilon)\,\mathbb{E}[f],$$

where $\mathbb{E}[f] = \mathbb{E}_{(x,y)\sim X\times Y}[f(x,y)]$.

▶ **Definition 30** (Lower Bounded Left-Marginals). *Let X, Y be finite sets, and let $f : X \times Y \to \{0,1\}$. We say that f has (r,ϵ)-lower bounded left-marginals if*

$$\Pr_{x\sim X}\left[\mathop{\mathbb{E}}_{y\sim Y}[f(x,y)] \leq (1-\epsilon)\cdot\mathbb{E}[f]\right] \leq 2^{-r}.$$

We state a result about graph counts in spread sets:

▶ **Theorem 31** ([27, Theorem 2.1]). *For any $k \in \mathbb{N}$, $\epsilon \in (0,1)$, there exists sufficiently small $\gamma = \gamma(\epsilon,k) > 0$, and sufficiently large $C = C(\epsilon,k) \in \mathbb{N}$, such that the following holds:*

Let $H = ([k], E)$ be an oriented graph, where $(i,j) \in E$ implies $i < j$. Let $X_1, \ldots, X_k$ be finite sets; let $d \geq 1$, and for each $(i,j) \in E$, let $f_{ij} : X_i \times X_j \to \{0,1\}$ be a function satisfying:

1. $\mathbb{E}[f_{ij}] \geq 2^{-d}$,
2. *f_{ij} is (Cd^2, γ)-combinatorially spread, and*
3. *f_{ij} has (Cd, γ)-lower bounded left-marginals.*

Then, it holds that

$$\left|\mathop{\mathbb{E}}_{x_1\sim X_1,\ldots,x_k\sim X_k}\left[\prod_{(i,j)\in E} f_{ij}(x_i,x_j)\right] - \prod_{(i,j)\in E}\mathbb{E}[f_{ij}]\right| \leq \epsilon\cdot\prod_{(i,j)\in E}\mathbb{E}[f_{ij}].$$

5.2 Algebraic Spreadness implies Graph Counts

We show that a certain function corresponding to algebraically spread sets is both combinatorially spread and has lower bounded left-marginals.

▶ **Lemma 32.** *Let $d, r \geq 1, \epsilon \in (0, 1/4)$. Then, there exists a sufficiently large integer $s = (r^8 + d^8) \cdot \epsilon^{-O(1)}$, and a sufficiently small $\delta = \Omega(\epsilon)$, such that the following holds:*

Let $X, Y, Z \subseteq \mathbb{F}_2^n$ be subsets each of density at least 2^{-d}, and such that Y, Z are (s, δ)-algebraically spread. Then, the function $f : X \times Y \to \{0, 1\}$, given by $f(x, y) = \mathbb{1}[x + y \in Z]$, satisfies:

1. *$|\mathbb{E}[f] - \alpha_Z| \leq \epsilon \cdot \alpha_Z$, where $\alpha_Z = |Z| / 2^n$,*
2. *f is (r, ϵ)-combinatorially spread, and*
3. *f has (r, ϵ)-lower bounded left-marginals.*

Proof. The reader is referred to the full version of this paper [46]. ◀

5.3 Squares inside Spread Diagonal-Product Sets

In this subsection, we prove our square covering result. We start by proving the following counting lemma:

▶ **Lemma 33.** *Let $\epsilon \in (0, 1/4)$ be a constant, and let $d \geq 1$. Then, there exists a sufficiently large integer $r = O_\epsilon(d^{16})$ and sufficiently small $\delta = \delta(\epsilon) > 0$ such that the following holds:*

Let $X, Y, Z \subseteq \mathbb{F}_2^n$ be each of density at least 2^{-d}, and such that all of X, Y, Z are (r, δ)-algebraically spread. Let $S = S(X, Y, Z)$ be as in Definition 4, and let $\Gamma : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{R}$ be the function mapping (x, y) to the normalized number of squares (see Definition 5) in S containing (x, y), i.e.,

$$\Gamma(x, y) = \mathop{\mathbb{E}}_{w \sim \mathbb{F}_2^n} [\mathbb{1}[s_{x,y,w} \subseteq S]].$$

Define $\alpha_X = \mathbb{E}[\mathbb{1}_X], \alpha_Y = \mathbb{E}[\mathbb{1}_Y], \alpha_Z = \mathbb{E}[\mathbb{1}_Z]$. Then, it holds that

1. *$\left||S| \cdot 2^{-2n} - \alpha_X \alpha_Y \alpha_Z\right| \leq \epsilon \cdot \alpha_X \alpha_Y \alpha_Z$.*
2. *$\left|\|\Gamma\|_1 - \alpha_X^2 \alpha_Y^2 \alpha_Z^2\right| \leq \epsilon \cdot \alpha_X^2 \alpha_Y^2 \alpha_Z^2$.*
3. *$\left|\|\Gamma\|_2^2 - \alpha_X^3 \alpha_Y^3 \alpha_Z^3\right| \leq \epsilon \cdot \alpha_X^3 \alpha_Y^3 \alpha_Z^3$.*

Proof. The reader is referred to the full version of this paper [46]. ◀

With the above, we are ready to prove the main result of this section.

▶ **Proposition 34.** *Let $\epsilon \in (0, 1/4)$ be a constant, and let $d \geq 1$. Then, there exists a sufficiently large integer $r = O_\epsilon(d^{16})$ and sufficiently small $\delta = \delta(\epsilon) > 0$ such that the following holds:*

Let $X, Y, Z \subseteq \mathbb{F}_2^n$, be each of density at least 2^{-d}, and such that all of X, Y, Z are (r, δ)-algebraically spread. Let $S = S(X, Y, Z)$ be as in Definition 4, and let $\mathcal{T}$ denote the set of all squares in S (see Definition 5); formally,

$$\mathcal{T} = \left\{(x, y, w) \in (\mathbb{F}_2^n)^3 : s_{x,y,w} \subseteq S\right\}.$$

Define $\alpha_X = \mathbb{E}[\mathbb{1}_X], \alpha_Y = \mathbb{E}[\mathbb{1}_Y], \alpha_Z = \mathbb{E}[\mathbb{1}_Z]$. Then, it holds that

1. *$\left||S| \cdot 2^{-2n} - \alpha_X \alpha_Y \alpha_Z\right| \leq \epsilon \cdot \alpha_X \alpha_Y \alpha_Z$.*
2. *$\left||\mathcal{T}| \cdot 2^{-3n} - \alpha_X^2 \alpha_Y^2 \alpha_Z^2\right| \leq \epsilon \cdot \alpha_X^2 \alpha_Y^2 \alpha_Z^2$.*

3. *Let μ be the distribution on S obtained as follows: Pick a random square in S, by picking uniformly at random $(x, y, w) \sim \mathcal{T}$; then, output a uniformly random element from the set $s_{x,y,w}$. It holds that*

$$\|\mu - U_S\|_1 \leq \epsilon,$$

where U_S denotes the uniform distribution over S.

Proof. The reader is referred to the full version of this paper [46]. ◀

6 Parallel Repetition for Games with the GHZ Query Support

This section is devoted to the proof of Theorem 1. By Lemma 18, it suffices to consider the case where Q is the uniform distribution over $\mathrm{supp}(Q)$. We prove the following:

▶ **Theorem 35.** *Let $\mathcal{G} = (\mathcal{X} \times \mathcal{Y} \times \mathcal{Z},\ \mathcal{A} \times \mathcal{B} \times \mathcal{C},\ Q, V_{pred})$ be any 3-player game with $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \mathbb{F}_2$, and with Q the uniform distribution over*

$$\mathrm{supp}(Q) = \left\{(x, y, z) \in \mathbb{F}_2^3 : x + y + z = 0\right\},$$

and such that $\mathrm{val}(\mathcal{G}) < 1$. Then, for all sufficiently large n,[2] it holds that

$$\mathrm{val}(\mathcal{G}^{\otimes n}) \leq \exp\left(-n^c\right),$$

where $c > 0$ is an absolute constant.

The remainder of this section is devoted to the proof of the above theorem. Towards this, we fix a 3-player game $\mathcal{G}$ as in the statement of the theorem; in particular, we must have that $\mathrm{val}(\mathcal{G}) \leq 3/4$.

The main step of the proof is to prove that the game $\mathcal{G}^{\otimes n}$ has a hard coordinate when the inputs to the players are conditioned on a product event with sufficiently large measure. We carry this out in two parts: first, in Section 6.1, we prove this statement assuming that each of the three sets in the product event are well spread (as in Definition 20) inside some large subspace of $\mathbb{F}_2^n$; then, in Section 6.2, we prove the statement for general product events, using our uniformization strategy.

6.1 Hard Coordinate under Spread Product Events

We consider the game $\mathcal{G}^{\otimes n} = (\mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n,\ \mathcal{A}^n \times \mathcal{B}^n \times \mathcal{C}^n,\ Q^{\otimes n}, V_{pred}^{\otimes n})$ for some sufficiently large $n \in \mathbb{N}$. Fix any strategies for the 3 players in this game, and for each $i \in [n]$, let $\mathrm{Win}_i \subseteq \mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n$ be the event that this strategy wins the i^{th} coordinate of the game.

▶ **Lemma 36.** *For any $1 \leq d \leq o(n)$, and constant $\epsilon \in (0, 1/4)$, there exists a (sufficiently large) integer $r = O_\epsilon(d^{16})$, and a (sufficiently small) constant $0 < \delta = \delta(\epsilon) < 1$, such that the following holds:*

Let $\mathcal{V} \subseteq \mathbb{F}_2^n$ be a linear subspace of dimension $\dim(\mathcal{V}) \geq n - o(n)$, and let $E, F, G \subseteq \mathcal{V}$ be such that each of them is (r, δ)-algebraically spread within $\mathcal{V}$ (see Definition 20), and such that each of the densities $\frac{|E|}{|\mathcal{V}|}, \frac{|F|}{|\mathcal{V}|}, \frac{|G|}{|\mathcal{V}|} \geq 2^{-d}$. Then,

$$\mathop{\mathbb{E}}_{i \sim [n]} \left[\Pr_{Q^{\otimes n}} \left[\mathrm{Win}_i \mid E \times F \times G \right] \right] \leq \frac{1}{2} + \frac{\mathrm{val}(\mathcal{G})}{2} + \epsilon.$$

The remainder of this subsection is devoted to the proof of the Lemma 36. We fix some $1 \leq d \leq o(n)$, and constant $\epsilon \in (0, 1/4)$, and let $r = O_\epsilon(d^{16})$, $\delta = \delta(\epsilon) > 0$ be so that Proposition 34 holds with the parameters $d, \frac{\epsilon}{2}$. Let $\mathcal{V} \subseteq \mathbb{F}_2^n$ be a linear subspace of dimension $\dim(\mathcal{V}) \geq n - o(n)$, and let $E, F, G \subseteq \mathcal{V}$ be each (r, δ)-algebraically spread within $\mathcal{V}$, and such that each of the densities $\alpha_E = \frac{|E|}{|\mathcal{V}|}$, $\alpha_F = \frac{|F|}{|\mathcal{V}|}$, $\alpha_G = \frac{|G|}{|\mathcal{V}|}$ is at least 2^{-d}.

We start by proving a lemma that demonstrates the usefulness of squares. It says that the no strategy for the game $\mathcal{G}^{\otimes n}$ can win on non-trivial coordinates of squares.

▶ **Definition 37** (Non-trivial coordinates). *For a square $s = s_{x,y,w} \subseteq \mathbb{F}_2^n \times \mathbb{F}_2^n$, as in Definition 5, we define its set of* non-trivial coordinates *as*

$$\mathcal{I}_s = \{i \in [n] : w_i \neq 0\}.$$

This is well-defined since w is uniquely determined by the square s (see Remark 6).

▶ **Lemma 38.** *Let s be a square, and let $i \in \mathcal{I}_s$ be a non-trivial coordinate in s. Then,*

$$\Pr_{(x,y)\sim s}[(x, y, x + y) \in \mathrm{Win}_i] \leq \mathrm{val}(\mathcal{G}) < 1.$$

Proof. The reader is referred to the full version of this paper [46]. ◀

Next, we show that the spreadness of $E, F, G \subseteq \mathcal{V}$ guarantees that $S = S(E, F, G)$ (see Definition 4) is covered uniformly by squares. For this, let $\mathcal{T}$ denote the set of all squares contained in S; formally,

$$\mathcal{T} = \left\{(x, y, w) \in \mathcal{V}^3 : s_{x,y,w} \subseteq S\right\}.$$

Let U_S denote the uniform distribution over S. Also, let μ be the distribution on S obtained as follows: Pick a random square in S, by picking uniformly at random $(x, y, w) \sim \mathcal{T}$; then, output a uniformly random element from the set $s_{x,y,w}$.

▶ **Lemma 39.** *We have that*

1. $\left|\frac{|S|}{|\mathcal{V}|^2} - \alpha_E\alpha_F\alpha_G\right| \leq \frac{\epsilon}{2} \cdot \alpha_E\alpha_F\alpha_G$.
2. $\left|\frac{|\mathcal{T}|}{|\mathcal{V}|^3} - \alpha_E^2\alpha_F^2\alpha_G^2\right| \leq \frac{\epsilon}{2} \cdot \alpha_E^2\alpha_F^2\alpha_G^2$.
3. $\|\mu - U_S\|_1 \leq \frac{\epsilon}{2}$.

Proof. This follows directly by Proposition 34, applied with respect to $\mathcal{V} \cong \mathbb{F}_2^{\dim(\mathcal{V})}$. Note that the definition of squares is *coordinate-free*, and hence the proposition is applicable. ◀

Before completing the proof of Lemma 36, we show that a random square inside $S = S(E, F, G)$ contains many non-trivial coordinates.

▶ **Lemma 40.** *Let $(x, y, w) \sim \mathcal{T}$, and let $s = s_{x,y,w}$. Then,*

$$\mathbb{E}\,|\mathcal{I}_s| \geq \frac{n}{2} \cdot (1 - o(1)).$$

Proof. The reader is referred to the full version of this paper [46]. ◀

Finally, we complete the proof.

Proof of Lemma 36. For any $i \in [n]$, and $S = S(E, F, G)$, we can write

$$\begin{aligned}\Pr_{Q^{\otimes n}}[\mathrm{Win}_i \mid E \times F \times G] &= \Pr_{x,y\sim\mathbb{F}_2^n}[(x, y, x+y) \in \mathrm{Win}_i \mid (x, y, x+y) \in E \times F \times G] \\ &= \Pr_{x,y\sim\mathbb{F}_2^n}[(x, y, x+y) \in \mathrm{Win}_i \mid (x, y) \in S] \\ &= \Pr_{(x,y)\sim S}[(x, y, x+y) \in \mathrm{Win}_i]\,.\end{aligned}$$

By Lemma 39, this can be bounded as

$$\Pr_{Q^{\otimes n}}[\mathrm{Win}_i \mid E \times F \times G] \le \mathop{\mathbb{E}}_{(x_0,y_0,w)\sim\mathcal{T}} \mathop{\mathbb{E}}_{(x,y)\sim s_{x_0,y_0,w}} [\mathbb{1}[(x, y, x+y) \in \mathrm{Win}_i]] + \frac{\epsilon}{2}. \tag{1}$$

Observe that for a fixed square $(x_0, y_0, w) \in \mathcal{T},\ s = s_{x_0,y_0,w}$, by Lemma 38, we can bound

$$\mathop{\mathbb{E}}_{(x,y)\sim s_{x_0,y_0,w}} [\mathbb{1}[(x, y, x+y) \in \mathrm{Win}_i]] \le 1 - (1 - \mathrm{val}(\mathcal{G})) \cdot \mathbb{1}[i \in \mathcal{I}_s]\,.$$

Plugging this into the above, and taking expectation over $i \sim [n]$, we obtain

$$\begin{aligned}\mathop{\mathbb{E}}_{i\sim[n]}\left[\Pr_{Q^{\otimes n}}[\mathrm{Win}_i \mid E \times F \times G]\right] &\le \mathop{\mathbb{E}}_{i\sim[n]} \mathop{\mathbb{E}}_{(x_0,y_0,w)\sim\mathcal{T}} \left[1 - (1 - \mathrm{val}(\mathcal{G})) \cdot \mathbb{1}\left[i \in \mathcal{I}_{s_{x_0,y_0,w}}\right]\right] + \frac{\epsilon}{2} \\ &= 1 - \frac{(1 - \mathrm{val}(\mathcal{G}))}{n} \mathop{\mathbb{E}}_{(x_0,y_0,w)\sim\mathcal{T}} |\mathcal{I}_{s_{x_0,y_0,w}}| + \frac{\epsilon}{2}. \\ &\le 1 - (1 - \mathrm{val}(\mathcal{G})) \cdot \frac{1}{2} \cdot (1 - o(1)) + \frac{\epsilon}{2} \qquad \text{(by Lemma 40)} \\ &\le \frac{1}{2} + \frac{\mathrm{val}(\mathcal{G})}{2} + \frac{\epsilon}{2} + o(1) \le \frac{1}{2} + \frac{\mathrm{val}(\mathcal{G})}{2} + \epsilon.\end{aligned}$$

◀

6.2 Hard Coordinate under General Product Events

Now, we work with general product events. Consider the game $\mathcal{G}^{\otimes n} = (\mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n,\ \mathcal{A}^n \times \mathcal{B}^n \times \mathcal{C}^n,\ Q^{\otimes n}, V_{pred}^{\otimes n})$ for some sufficiently large $n \in \mathbb{N}$. Fix any strategies for the 3 players in this game, and for each $i \in [n]$, let $\mathrm{Win}_i \subseteq \mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n$ be the event that this strategy wins the i^{th} coordinate of the game. We prove that:

▶ **Lemma 41.** *For any constant $\epsilon \in (0, 1/4)$, there exists a constant $c = c(\epsilon) > 0$, such that the following holds: For any sets $E, F, G \subseteq \mathbb{F}_2^n$ with $\Pr_{Q^{\otimes n}}[E \times F \times G] \ge 2^{-n^c}$, it holds that*

$$\mathop{\mathbb{E}}_{i\sim[n]}\left[\Pr_{Q^{\otimes n}}[\mathrm{Win}_i \mid E \times F \times G]\right] \le \frac{1}{2} + \frac{\mathrm{val}(\mathcal{G})}{2} + \epsilon.$$

Proof. The reader is referred to the full version of this paper [46]. ◀

With the above, the proof of our main theorem is direct:

Proof of Theorem 35. The theorem follows by combining Lemma 19 with Lemma 41, applied with the parameter choice $\epsilon = 0.1$, recalling that $\mathrm{val}(\mathcal{G}) \le 3/4$. ◀

Proof of Theorem 1. The theorem follows by combining Theorem 35 and Lemma 18. ◀

7 A Concentration Bound

We also prove a concentration bound for the GHZ game:

▶ **Theorem 42.** *Let $\mathcal{G} = (\mathcal{X} \times \mathcal{Y} \times \mathcal{Z},\ \mathcal{A} \times \mathcal{B} \times \mathcal{C},\ Q, V_{pred})$ be any 3-player game with $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \mathbb{F}_2$, and with Q the uniform distribution over*

$$\mathrm{supp}(Q) = \left\{(x, y, z) \in \mathbb{F}_2^3 : x + y + z = 0\right\},$$

and such that $\mathrm{val}(\mathcal{G}) < 1$. Then, for every constant $\epsilon \in (0, 1/4)$, there exists a constant $c = c(\epsilon) > 0$, such that the following holds for all sufficiently large n:[3]

Consider the game $\mathcal{G}^{\otimes n}$, and consider any strategies for the 3 players in this game. Then, if Z denotes the number of coordinates that the players win, it holds that

$$\Pr\left[Z \geq (\mathrm{val}(\mathcal{G}) + \epsilon) \cdot n\right] \leq \exp\left(-n^c\right).$$

The remainder of this section is devoted to the proof of the above theorem. Towards this, we fix a game $\mathcal{G}$ as in the statement of the theorem. Consider the game $\mathcal{G}^{\otimes n} = (\mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n,\ \mathcal{A}^n \times \mathcal{B}^n \times \mathcal{C}^n,\ Q^{\otimes n}, V_{pred}^{\otimes n})$ for some sufficiently large $n \in \mathbb{N}$. Also, fix any strategies for the 3 players in this game, and for each $i \in [n]$, let $\mathrm{Win}_i \subseteq \mathcal{X}^n \times \mathcal{Y}^n \times \mathcal{Z}^n$ be the event that this strategy wins the i^{th} coordinate of the game. For each $S \subseteq [n]$, we use Win_S to denote the event $\wedge_{i \in S} \mathrm{Win}_i$.

7.1 An Improved Bound for Product Events

The first step in our proof is to prove an improved version of Lemma 41, where we don't lose a "factor of 2." We prove the following:

▶ **Lemma 43.** *For any constant $\epsilon \in (0, 1/4)$, there exists $c = c(\epsilon) > 0$, such that the following holds: For any sets $E, F, G \subseteq \mathbb{F}_2^n$ with $\Pr_{Q^{\otimes n}}[E \times F \times G] \geq 2^{-n^c}$, it holds that*

$$\mathop{\mathbb{E}}_{i \sim [n]}\left[\Pr_{Q^{\otimes n}}\left[\mathrm{Win}_i \mid E \times F \times G\right]\right] \leq \mathrm{val}(\mathcal{G}) + \epsilon.$$

The proof of the above lemma exactly follows the proof in Section 6, so we only mention how to improve it. We note that the only place where we lost a factor of 2 in the bound was in the proof of Lemma 36; the proof of Lemma 41, using uniformization, goes through as is if an improved version of Lemma 36 is proved. Now, in Lemma 36, we proceed in exactly the same manner upto Equation (1), which says (after taking expectation over $i \sim [n]$)

$$\mathop{\mathbb{E}}_{i \sim [n]}\left[\Pr_{Q^{\otimes n}}\left[\mathrm{Win}_i \mid E \times F \times G\right]\right] \leq \mathop{\mathbb{E}}_{i \sim [n]} \mathop{\mathbb{E}}_{(x_0, y_0, w) \sim \mathcal{T}} \mathop{\mathbb{E}}_{(x,y) \sim s_{x_0, y_0, w}} \left[\mathbb{1}\left[(x, y, x + y) \in \mathrm{Win}_i\right]\right] + \frac{\epsilon}{2}.$$

The factor 2 loss now occurred because when $(x_0, y_0, w) \sim \mathcal{T}$, the event $w_i = 1$ occurs with probability roughly $1/2$ (see Lemma 40), and we can only apply Lemma 38 when this holds. To improve upon this bound, we will replace the distribution with one where we are already conditioning on the event $w_i = 1$, and only then apply Lemma 38; this requires proving a stronger version of Lemma 40, which says that the distribution of (x, y) in the above statement remains unaffected even when the square is sampled conditioned on $w_i = 1$. Formally, we prove:

▶ **Lemma 44.** *Recall the distribution μ on S obtained as follows: choose $(x_0, y_0, w) \sim \mathcal{T}$, and let $(x, y) \sim s_{x_0, y_0, w}$. For any $i \in [n]$, define the distribution ν_i as follows: choose $(x_0, y_0, w) \sim \mathcal{T}$ conditioned on $w_i = 1$, and let $(x, y) \sim s_{x_0, y_0, w}$. Then, we have*

$$\mathop{\mathbb{E}}_{i \sim [n]} \|\mu - \nu_i\|_1 \leq o(1).$$

Proof. The reader is referred to the full version of this paper [46]. ◀

Finally, we complete the proof of the improved bound:

Proof of Lemma 43. As observed before, it suffices to improve Lemma 36, and we have

$$\mathop{\mathbb{E}}_{i\sim[n]}\left[\Pr_{Q^{\otimes n}}\left[\mathrm{Win}_i \mid E\times F\times G\right]\right] \le \mathop{\mathbb{E}}_{i\sim[n]}\mathop{\mathbb{E}}_{(x_0,y_0,w)\sim\mathcal{T}}\mathop{\mathbb{E}}_{(x,y)\sim s_{x_0,y_0,w}}\left[\mathbb{1}\left[(x,y,x+y)\in\mathrm{Win}_i\right]\right]+\frac{\epsilon}{2}.$$

By Lemma 44 and Lemma 38, the above is at most

$$\mathop{\mathbb{E}}_{i\sim[n]}\mathop{\mathbb{E}}_{(x_0,y_0,w)\sim\mathcal{T}|w_i=1}\mathop{\mathbb{E}}_{(x,y)\sim s_{x_0,y_0,w}}\left[\mathbb{1}\left[(x,y,x+y)\in\mathrm{Win}_i\right]\right]+\frac{\epsilon}{2}+o(1)\le \mathrm{val}(\mathcal{G})+\frac{\epsilon}{2}+o(1)$$
$$\le \mathrm{val}(\mathcal{G})+\epsilon.$$ ◀

7.2 Random Sets are Hard under Parallel Repetition

We prove a strengthening of Theorem 35, by using an argument similar to the proof of Lemma 19.

▶ **Lemma 45.** *For every constant $\epsilon\in(0,1/4)$, there exists $c=c(\epsilon)>0$, such that for every integer $1\le t\le n^c$, it holds that*

$$\mathop{\mathbb{E}}_{|S|=t}\left[\Pr_{Q^{\otimes n}}\left[\mathrm{Win}_S\right]\right]\le 2\cdot\left(\mathrm{val}(\mathcal{G})+\epsilon\right)^t,$$

where $\mathbb{E}_{|S|=t}$ denotes that the expectation is over a uniformly chosen subset $S\subseteq[n]$ of size t.

Proof. The reader is referred to the full version of this paper [46]. ◀

7.3 Proof of the Concentration Bound

Finally, we complete the proof of our concentration bound, using standard arguments [52]:

Proof of Theorem 42. The reader is referred to the full version of this paper [46]. ◀

References

1 Noga Alon and Bo'az Klartag. Economical toric spines via Cheeger's inequality. *J. Topol. Anal.*, 1(2):101–111, 2009.

2 Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. System Sci.*, 54(2, part 2):317–331, 1997. (also in FOCS 1993). doi:10.1006/JCSS.1997.1472.

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

4 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

5 Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. *SIAM J. Comput.*, 42(3):1327–1363, 2013. (also in STOC 2010). doi:10.1137/100811969.

6 Boaz Barak, Anup Rao, Ran Raz, Ricky Rosen, and Ronen Shaltiel. Strong parallel repetition theorem for free projection games. In *APPROX-RANDOM*, pages 352–365, 2009. doi:10.1007/978-3-642-03685-9_27.

7 Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998. (also in FOCS 1995). doi:10.1137/S0097539796302531.

8 Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *STOC*, pages 113–131, 1988. doi:10.1145/62212.62223.

9 Amey Bhangale, Mark Braverman, Subhash Khot, Yang Liu, and Dor Minzer. Parallel repetition for 3-player XOR games. In *STOC*, pages 104–110, 2025. doi:10.1145/3717823.3718190.

10 Amey Bhangale, Mark Braverman, Subhash Khot, Yang P. Liu, and Dor Minzer. Parallel repetition of k-player projection games. In *APPROX-RANDOM*, pages Art. No. 54, 16, 2024.

11 Amey Bhangale, Mark Braverman, Subhash Khot, Yang P. Liu, Dor Minzer, and Kunal Mittal. An analytical approach to parallel repetition via CSP inverse theorems. In *STOC*, 2026. (To appear).

12 Mark Braverman and Ankit Garg. Small value parallel repetition for general games. In *STOC*, pages 335–340, 2015. doi:10.1145/2746539.2746565.

13 Mark Braverman, Subhash Khot, and Dor Minzer. Parallel repetition for the GHZ game: exponential decay. In *FOCS*, pages 1337–1341, 2023. doi:10.1109/FOCS57990.2023.00080.

14 Mark Braverman and Dor Minzer. Optimal tiling of the Euclidean space using permutation-symmetric bodies. In *CCC*, pages Art. No. 5, 48, 2021.

15 Mark Braverman, Anup Rao, Omri Weinstein, and Amir Yehudayoff. Direct products in communication complexity. In *FOCS*, pages 746–755, 2013. doi:10.1109/FOCS.2013.85.

16 Jop Briët, Harry Buhrman, Troy Lee, and Thomas Vidick. Multipartite entanglement in XOR games. *Quantum Inf. Comput.*, 13(3-4):334–360, 2013. doi:10.26421/QIC13.3-4-11.

17 Richard Cleve, Peter Høyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *CCC*, pages 236–249, 2004. doi:10.1109/CCC.2004.1313847.

18 Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM J. Comput.*, 34(5):1129–1146, 2005. (also in STOC 2003). doi:10.1137/S0097539704443057.

19 Irit Dinur, Prahladh Harsha, Rakesh Venkat, and Henry Yuen. Multiplayer parallel repetition for expanding games. In *ITCS*, pages Art. No. 37, 16, 2017.

20 Irit Dinur, Oded Regev, and Clifford Smyth. The hardness of 3-uniform hypergraph coloring. *Combinatorica*, 25(5):519–535, 2005. doi:10.1007/S00493-005-0032-4.

21 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633, 2014. doi:10.1145/2591796.2591884.

22 Uriel Feige. On the success probability of the two provers in one-round proof systems. In *Structure in Complexity Theory Conference*, pages 116–123, 1991. doi:10.1109/SCT.1991.160251.

23 Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998. (also in STOC 1996). doi:10.1145/285055.285059.

24 Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996. doi:10.1145/226643.226652.

25 Uriel Feige, Guy Kindler, and Ryan O'Donnell. Understanding parallel repetition requires understanding foams. In *CCC*, pages 179–192, 2007. doi:10.1109/CCC.2007.39.

26 Uriel Feige and Oleg Verbitsky. Error reduction by parallel repetition – a negative result. *Combinatorica*, 22(4):461–478, 2002. doi:10.1007/S00493-002-0001-0.

27 Yuval Filmus, Hamed Hatami, Kaave Hosseini, and Esty Kelman. Sparse graph counting and Kelley-Meka bounds for binary systems. In *FOCS*, pages 1559–1578, 2024. doi:10.1109/FOCS61266.2024.00098.

28 Lance Fortnow. *Complexity theoretic aspects of interactive proof systems*. PhD thesis, MIT, 1989.

29 Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. *Theoret. Comput. Sci.*, 134(2):545–557, 1994. doi:10.1016/0304-3975(94)90251-8.

30 H. Furstenberg and Y. Katznelson. A density version of the Hales-Jewett theorem. *J. Anal. Math.*, 57:64–119, 1991.

31 Uma Girish, Justin Holmgren, Kunal Mittal, Ran Raz, and Wei Zhan. Parallel repetition for the GHZ game: A simpler proof. In *APPROX-RANDOM*, pages 62:1–62:19, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.62.

32 Uma Girish, Justin Holmgren, Kunal Mittal, Ran Raz, and Wei Zhan. Parallel repetition for all 3-player games over binary alphabet. In *STOC*, pages 998–1009, 2022. doi:10.1145/3519935.3520071.

33 Uma Girish, Kunal Mittal, Ran Raz, and Wei Zhan. Polynomial bounds on parallel repetition for all 3-player games with binary inputs. In *APPROX-RANDOM*, pages 6:1–6:17, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.6.

34 Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger. Going beyond bell's theorem. In *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, pages 69–72. Springer Netherlands, 1989.

35 Venkatesan Guruswami, Johan Håstad, and Madhu Sudan. Hardness of approximate hypergraph coloring. *SIAM J. Comput.*, 31(6):1663–1686, 2002. (also in FOCS 2000). doi:10.1137/S0097539700377165.

36 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. (also in STOC 1997). doi:10.1145/502090.502098.

37 Jan Hązła, Thomas Holenstein, and Anup Rao. Forbidden subgraph bounds for parallel repetition and the density hales-jewett theorem. *CoRR*, abs/1604.05757, 2016. Available at http://arxiv.org/abs/1604.05757. arXiv:1604.05757.

38 Thomas Holenstein. Parallel repetition: simplifications and the no-signaling case. *Theory Comput.*, 5:141–172, 2009. (also in STOC 2007). doi:10.4086/TOC.2009.V005A008.

39 Justin Holmgren and Ran Raz. A parallel repetition theorem for the GHZ game. *CoRR*, abs/2008.05059, 2020. Available at https://arxiv.org/abs/2008.05059. arXiv:2008.05059.

40 Michael Jaber, Yang P. Liu, Shachar Lovett, Anthony Ostuni, and Mehtaab Sawhney. Quasi-polynomial bounds for the corners theorem. In *FOCS*, 2025.

41 Zander Kelley, Shachar Lovett, and Raghu Meka. Explicit separations between randomized and deterministic Number-on-Forehead communication. In *STOC*, pages 1299–1310, 2024. doi:10.1145/3618260.3649721.

42 Zander Kelley and Raghu Meka. Strong bounds for 3-progressions. In *FOCS*, pages 933–973, 2023. doi:10.1109/FOCS57990.2023.00059.

43 Subhash Khot. Hardness results for approximate hypergraph coloring. In *STOC*, pages 351–359, 2002. doi:10.1145/509907.509962.

44 Subhash Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *FOCS*, pages 23–32, 2002. doi:10.1109/SFCS.2002.1181879.

45 Guy Kindler, Ryan O'Donnell, Anup Rao, and Avi Wigderson. Spherical cubes and rounding in high dimensions. In *FOCS*, pages 189–198, 2008. doi:10.1109/FOCS.2008.50.

46 Yang P. Liu, Shachar Lovett, and Kunal Mittal. Improved parallel repetition for GHZ-supported games via spreadness, 2026. Available at https://arxiv.org/abs/2602.09290. doi:10.48550/arXiv.2602.09290.

47 Kunal Mittal. Multiplayer parallel repetition is the same as high-dimensional extremal combinatorics, 2025. Available at https://arxiv.org/abs/2510.24910. doi:10.48550/arXiv.2510.24910.

48 Kunal Mittal and Ran Raz. Block rigidity: strong multiplayer parallel repetition implies super-linear lower bounds for Turing machines. In *ITCS*, pages Art. No. 71, 15, 2021.

49 Michael Mitzenmacher and Eli Upfal. *Probability and computing*. Cambridge University Press, Cambridge, 2005. Randomized algorithms and probabilistic analysis.

50 Itzhak Parnafes, Ran Raz, and Avi Wigderson. Direct product results and the GCD problem, in old and new communication models. In *STOC*, pages 363–372, 1997. doi:10.1145/258533.258620.

51 D. H. J. Polymath. A new proof of the density Hales-Jewett theorem. *Ann. of Math. (2)*, 175(3):1283–1327, 2012.

52 Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM J. Comput.*, 40(6):1871–1891, 2011. (also in STOC 2008). doi:10.1137/080734042.

53 Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. (also in STOC 1995). doi:10.1137/S0097539795280895.

54 Ran Raz. Parallel repetition of two prover games. In *CCC*, pages 3–6, 2010.

55 Ran Raz. A counterexample to strong parallel repetition. *SIAM J. Comput.*, 40(3):771–777, 2011. (also in FOCS 2008). doi:10.1137/090747270.

56 Ran Raz and Ricky Rosen. A strong parallel repetition theorem for projection games on expanders. In *CCC*, pages 247–257, 2012. doi:10.1109/CCC.2012.11.

57 Ivan Šupić and Joseph Bowles. Self-testing of quantum systems: a review. *Quantum*, 4:337, 2020. doi:10.22331/Q-2020-09-30-337.

58 Oleg Verbitsky. Towards the parallel repetition conjecture. *Theoret. Comput. Sci.*, 157(2):277–282, 1996. doi:10.1016/0304-3975(95)00165-4.

The Log-Rank Conjecture: New Equivalent Formulations

Lianna Hambardzumyan
The University of Copenhagen, Denmark

Shachar Lovett
University of California San Diego, CA, USA

Morgan Shirley
Lund University, Sweden

Abstract

The log-rank conjecture is a longstanding open problem with multiple equivalent formulations in complexity theory and mathematics. In its linear-algebraic form, it asserts that the rank and partitioning number of a Boolean matrix are quasi-polynomially related.

We propose a relaxed but still equivalent version of the conjecture based on a new matrix parameter, signed rectangle rank: the minimum number of all-1 rectangles needed to express the Boolean matrix as a ± 1-sum. Signed rectangle rank lies between rank and partition number, and our main result shows that it is in fact equivalent to rank up to a logarithmic factor. Additionally, we extend the main result to tensors. This reframes the log-rank conjecture as: can every signed decomposition of a Boolean matrix be made positive with only quasi-polynomial blowup?

As an application, we prove an equivalence between the log-rank conjecture and a conjecture of Lovett and Singer–Sudan on cross-intersecting set systems.

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases cross-intersecting set systems, Log-rank conjecture, monochromatic rectangle, partition number

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.7

Related Version *Preprint*: https://arxiv.org/abs/2510.02583

Funding Most of the work was done while the first and third authors were postdoctoral researchers at University of Victoria, Canada, and funded by NSERC.
Lianna Hambardzumyan: This work is funded by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA).
Shachar Lovett: Research supported by Simons Investigator Award #929894 and NSF award CCF-2425349.
Morgan Shirley: Supported by Knut and Alice Wallenberg grant KAW 2023.0116.

Acknowledgements We are grateful to Adi Shraibman and Amir Yehudayoff for enlightening discussions.

1 Introduction

The log-rank conjecture is one of the most well-known problems in complexity theory, and despite extensive work it remains unsolved. It asserts that for a Boolean matrix its communication complexity and the logarithm of its matrix rank over the reals are polynomially related. An equivalent linear-algebraic formulation of the conjecture is that for Boolean matrices, the matrix rank and partitioning number (sometimes called the binary rank) are quasi-polynomially related.



© Lianna Hambardzumyan, Shachar Lovett, and Morgan Shirley;
licensed under Creative Commons License CC-BY 4.0



More formally, for a Boolean matrix M let $\mathrm{r}(M)$ denote its rank over the reals, and let $\mathrm{p}(M)$ denote its partitioning number: the minimum number of all-1 submatrices that partition the 1-entries of M. Equivalently, the partitioning number is the minimum number p such that $M = \sum_{i=1}^{p} R_i$, where each R_i is a *primitive* matrix – an all-1 submatrix, possibly after adding all-zero rows and columns. If we relax the decomposition to allow general rank-1 matrices instead of primitive matrices, we recover the standard matrix rank. That is, $\mathrm{r}(M)$ is the minimum number r such that $M = \sum_{i=1}^{r} M_i$, with each M_i of rank-1. In this sense, matrix rank can be seen as a relaxation of the partitioning number, and trivially $\mathrm{r}(M) \leq \mathrm{p}(M)$. The log-rank conjecture asks how well this relaxation estimates the partitioning number:

▶ **Conjecture 1** (Log-rank conjecture [11]). *For any Boolean matrix M,*

$$\log \mathrm{p}(M) \leq (\log \mathrm{r}(M))^{O(1)}.$$

The log-rank conjecture was first posed by Lovász and Saks [11] in the context of communication complexity: is the deterministic communication complexity of a Boolean matrix polynomially related to the logarithm of its rank over the reals? A closely related question had appeared even earlier in graph theory. There, the log-rank conjecture is equivalent to asking whether the logarithm of a graph's chromatic number is quasi-polynomially related to the rank of its adjacency matrix [16, 3, 11].

Regarding the state of the art, the best known upper bound is due to Sudakov and Tomon [20], improving on previous work of Lovett [13], and shows that

$$\log \mathrm{p}(M) \leq O\left(\sqrt{\mathrm{r}(M)}\right).$$

The largest known separation follows from [1, Corollary 3], which yields a matrix M satisfying $\log \mathrm{p}(M) \geq \Omega(\log^2 \mathrm{r}(M))$. For a more detailed overview of the log-rank conjecture and its equivalent formulations, we refer the reader to the survey by Lee and Shraibman [8], as well as the textbooks by Jukna [6] and Rao and Yehudayoff [17].

In this paper, we consider a "gradual relaxation" from partitioning number to rank via an intermediate complexity measure: the *signed rectangle rank*. This notion allows decomposition of the matrix into *signed* primitive matrices. Formally, let the *signed rectangle rank* of M, denoted by $\mathrm{srr}(A)$, be the minimum number t such that $M = \sum_{i=1}^{t} \varepsilon_i R_i$, where $\varepsilon_i \in \{1, -1\}$ and each R_i is a primitive matrix. Trivially, we have

$$\mathrm{r}(M) \leq \mathrm{srr}(M) \leq \mathrm{p}(M).$$

A promising approach to the log-rank conjecture is to study how "close" signed rectangle rank is to rank and partitioning number. To resolve the conjecture, it would suffice to either prove both of the following statements or disprove one of them:

- $\mathrm{srr}(M)$ is quasi-polynomially related to $\mathrm{r}(M)$, and
- $\mathrm{srr}(M)$ is quasi-polynomially relate to $\mathrm{p}(M)$.

The main result of the paper is that rank and signed rectangle rank are tightly related.

▶ **Theorem 2** (Main Theorem). *Every Boolean matrix M of rank r can be written as a ± 1-linear-combination of at most $O(r \log r)$ primitive matrices, that is,*

$$\mathrm{srr}(M) \leq O(r \log r).$$

From this we immediately get an equivalent formulation of the log-rank conjecture:

▶ **Conjecture 3.** *For every Boolean matrix M,* $\log \mathrm{p}(M) \leq (\log \mathrm{srr}(M))^{O(1)}$.

▶ **Corollary 4** (from Theorem 2)**.** *The log-rank conjecture (Conjecture 1) is equivalent to Conjecture 3.*

A natural question left open here is whether the bound in Theorem 2 can be improved, or if it is already tight:

▶ **Question 5.** *Is it true that for every Boolean matrix M,* $\mathrm{srr}(M) \leq O(\mathrm{r}(M))$*?*

A result of Lindström [10] shows that the bound is tight for our proof technique (see the discussion at the end of Section 2). Therefore, any improvements on the bound require different ideas.

Power of cancellations vs. monotonicity

The log-rank conjecture compares two ways of measuring the complexity of a matrix: rank and partitioning number. These measures differ in two fundamental ways:

- **Combinatorial structure**: Rank expresses a matrix as a summation of rank-1 matrices with arbitrary real coefficients and no apparent combinatorial structure. In contrast, partitioning number only allows primitive matrices, which are purely *combinatorial* objects.
- **Cancellations**: Rank allows for *cancellations* in the summation. Partitioning, on the other hand, is a *monotone* model: all-1 submatrices (also known as *rectangles*) must be combined positively and disjointly.

The log-rank conjecture asks whether these two measures are nevertheless quasi-polynomially related.

The measure we study, *signed rectangle rank*, is an intermediate notion that allows cancellations, but only at the level of rectangles. In this way, it relaxes partitioning by permitting signed combinations of rectangles, while still restricting to simple combinatorial building blocks. Intuitively, our main theorem shows that allowing cancellation between rectangles already captures essentially all of the linear structure present in a Boolean matrix.

An analogous perspective arises from studying another measure of a matrix, the *non-negative rank*. If one forbids cancellations but allows non-negative real values, without imposing any combinatorial structure on the decomposition, the resulting measure is the non-negative rank. Formally, non-negative rank of a $n \times n$ matrix M, denoted by $\mathrm{nnr}(M)$, is the minimum number r such that $M = \sum_{i=1}^{r} u_i v_i^T$, where $u_i, v_i \in \mathbb{R}^n$ are non-negative column vectors. Trivially, we have

$$\mathrm{r}(M) \leq \mathrm{nnr}(M) \leq \mathrm{p}(M).$$

Analogous to our main theorem, it is known that for Boolean matrices this relaxation of partitioning number does not increase its power too much (this follows from a result of Lovász and Saks [12]; see e.g. Watson [22]):

$$\log \mathrm{p}(M) \leq O(\log^2 \mathrm{nnr}(M)).$$

Consequently, the log rank conjecture is equivalent to asking whether there is a quasi-polynomial relationship between rank and non-negative rank, namely, is it true that $\log \mathrm{nnr}(M) \leq (\log \mathrm{r}(M))^{O(1)}$? Comparing this with Corollary 4, our results imply that the log-rank conjecture also reduces to converting a signed rectangle decomposition into a fully monotone one, with at most a quasi-polynomial increase in the number of primitive matrices. See Figure 1 for an illustration of the relationships between these measures.

Together, these paint a picture of what is left to understand about the log-rank conjecture. The fact that rank allows arbitrary real coefficients and partitioning number is combinatorial is a red herring: the only relevant question is whether allowing cancellations is significantly stronger than monotonicity.

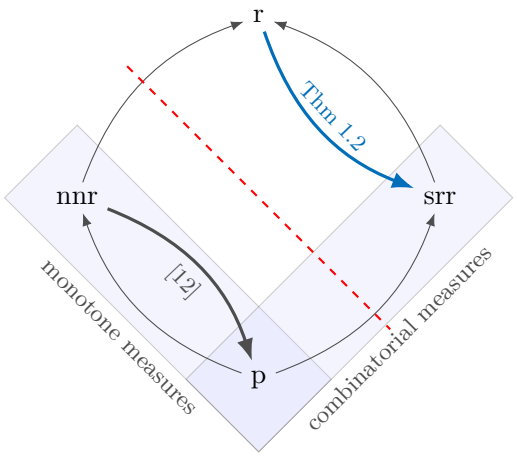


Figure 1 Relationships between the matrix measures discussed; an arrow $a \to b$ indicates that $\log b \leq \log^{O(1)} a$. The thin arrows indicate bounds which are trivial from the definitions. The red dashed line indicates that proving the log-rank conjecture is equivalent to showing that *any* measure below the line is quasi-polynomially related to *any* measure above the line.

Equivalent conjecture on cross-intersecting set systems

Let $\mathcal{S}, \mathcal{T} \subseteq 2^{[d]}$ be two set families. We say the pair $(\mathcal{S}, \mathcal{T})$ is *L-cross-intersecting* for some $L \subseteq \{0, \dots, d\}$ if for all $S \in \mathcal{S}$ and $T \in \mathcal{T}$, we have $|S \cap T| \in L$; that is, the size of every pairwise intersection belongs to L. Cross-intersecting set families have been widely studied in combinatorics, with much of the work focusing on their extremal properties [4, 18, 7, 5].

The following conjecture about $\{a, b\}$-cross-intersecting set systems was independently proposed by Lovett [14] and Singer–Sudan [19], who both observed that it is implied by the log-rank conjecture.

▶ **Conjecture 6.** *Let $\mathcal{S} = \{S_1, \dots, S_m\}$ and $\mathcal{T} = \{T_1, \dots, T_n\}$ be an $\{a, b\}$-cross-intersecting pair of families from $2^{[d]}$, where $a, b \in \{0, \dots, d\}$. Then there exist subfamilies $\mathcal{A} \subseteq \mathcal{S}$ and $\mathcal{B} \subseteq \mathcal{T}$ such that $(\mathcal{A}, \mathcal{B})$ is either $\{a\}$- or $\{b\}$-cross-intersecting, and*

$$|\mathcal{A}||\mathcal{B}| \geq 2^{-\operatorname{polylog}(d)} \cdot |\mathcal{S}||\mathcal{T}|.$$

As an application of Theorem 2, we show that the log-rank conjecture is equivalent to this conjecture.[1]

▶ **Theorem 7.** *The log-rank conjecture (Conjecture 1) is equivalent to Conjecture 6.*

[1] This equivalence was claimed in [19] as a parenthetical remark. However, this was a typo. The remark was meant to claim the implication from the log-rank conjecture [21].

Tensors

In addition to the matrix case we extend Theorem 2 to Boolean tensors, showing an analogous relation between the tensor rank and signed rectangle rank of a tensor. This has implications for the generalization of the log-rank conjecture to *number-in-hand* communication complexity (asked explicitly in [2]).

Organization

We give the proof of Theorem 2 in Section 2 and discuss the open question of how tight the bound is. In Section 3, we prove the equivalence of the log-rank conjecture to Conjecture 6. Finally, we prove the extension of the main theorem to tensors in Section 4.

2 Proof of the Main Theorem

Let M be an $m \times n$ Boolean matrix of rank r. For a subset S of the columns, define its *column-sum* to be the column vector c such that for $i \in [m]$, $c(i) = \sum_{j \in S} M(i,j)$. That is, c is the entrywise sum of all the columns of S along all the rows. Denote the column-sum of S as $\mathrm{sum}(S)$. Call a subset S of columns *independent* if all the subsets of S have distinct column-sums.

▷ Claim 8. Let S be an independent set of columns of M. Then $|S| \leq O(r \log r)$, where r is the rank of M.

Proof. Let A_S be the matrix whose columns are the column-sums of S, so A_S has $2^{|S|}$ columns. The entries of A_S are in $\{0, \ldots, |S|\}$ as they are sums of at most $|S|$ entries of M, which take value $\{0, 1\}$. The rank of A_S is at most r because its columns are in the span of the columns of M, which itself has rank r. Thus, there is a set of rows R with $|R| = r$ such that, for any column c of A_S, the values $(c(i))_{i \in R}$ are sufficient to determine every entry of c. Therefore, there are at most $(|S| + 1)^r$ unique columns of A_S, and since every column of A_S is unique, we have that the subsets of S have at most $(|S| + 1)^r$ column-sums. Therefore, $2^{|S|} \leq (|S| + 1)^r$, which after taking logarithms gives $|S| \leq O(r \log |S|)$, which implies $|S| \leq O(r \log r)$. ◁

▷ Claim 9. Let S be a maximal independent set of columns of M. Then every column of M can be expressed as a ± 1-linear combination of columns in S.

Proof. This trivially holds for the columns in S. Fix a column $c \notin S$. Since S is a maximal independent set, $S \cup \{c\}$ is not independent. This means that there are two subsets A and B of $S \cup \{c\}$ with the same column-sum. We can assume that these subsets are disjoint; if they are not, then removing the common columns still results in equal column-sums. Note A and B cannot both exclude c, as this would contradict S being independent. Assume $c \in B$, and let $B' = B \setminus \{c\}$. Combining this with $\mathrm{sum}(A) = \mathrm{sum}(B)$, we get $\mathrm{sum}(A) = \mathrm{sum}(B') + c$. Noting that $A \cap B' = \emptyset$ and $A, B' \subseteq S$ concludes that $c = \mathrm{sum}(A) - \mathrm{sum}(B')$ is the desired linear combination. ◁

Combining these two claims concludes the proof of Theorem 2 as follows. Let S be the largest independent set of columns of M. By Claim 9 every column y can be expressed as a ± 1-linear combination of columns in S. Thus, M can be written as

$$M(x,y) = \sum_{c \in S} \alpha_c(y) c(x),$$

where each coefficient $\alpha_c(y) \in \{-1, 0, 1\}$. Decompose $\alpha_c(y) = \alpha_c^+(y) - \alpha_c^-(y)$, where $\alpha_c^+(y), \alpha_c^-(y) \in \{0, 1\}$. For each column $c \in S$, define

$$R_c^+(x, y) = \alpha_c^+(y)c(x) \qquad \text{and} \qquad R_c^-(x, y) = \alpha_c^-(y)c(x)$$

Each of R_c^+ and R_c^- is either an all-zeroes matrix or a primitive matrix, since they are outer products of $\{0, 1\}$-valued vectors. Hence, $M = \sum_{c \in S}(R_c^+ - R_c^-)$, which expresses M as ± 1-sum of at most $2|S|$ primitive matrices. Finally, applying the bound on $|S|$ from Claim 8, we conclude that $\mathrm{srr}(M) \leq O(r \log r)$.

We remind the reader of the following open question:

▶ **Question 5.** *Is it true that for every Boolean matrix M, $\mathrm{srr}(M) \leq O(\mathrm{r}(M))$?*

An approach one might hope to take is to improve the statement of Claim 8, that is, to prove a better upper bound on the size of an independent set of columns in a low-rank Boolean matrix. However, a result of Lindström [10, Theorem 1] shows that Claim 8 is tight. Concretely, Lindström gives a construction of a set of $\Theta(r \log r)$ vectors in $\{0, 1\}^r$ which are independent (that is, all their column sums are distinct).

3 Equivalence to the cross-intersecting set systems conjecture

In order to prove Theorem 7, we will use the equivalent version of the log-rank conjecture by Nisan and Wigderson [15]. A submatrix of a matrix is called a *monochromatic rectangle* if all of its entries have the same value.

▶ **Conjecture 10** ([15])**.** *Any Boolean matrix M has a monochromatic rectangle of density $2^{-\operatorname{polylog}(\mathrm{r}(M))}$.*

Sgall [18] considered Conjecture 6 in the case of $\{a, a+1\}$-cross-intersecting set systems and noted that it would follow from the log-rank conjecture. Sgall noted that any $\{a, b\}$-cross-intersecting set pair of families $(\mathcal{S}, \mathcal{T})$ from $2^{[d]}$ can be represented as an $m \times n$ matrix $M_{\mathcal{S},\mathcal{T}}$ over $\{a, b\}$ with rank at most d, where $M_{\mathcal{S},\mathcal{T}}(i, j) := |S_i \cap T_j|$. One can write $M_{\mathcal{S},\mathcal{T}}$ as a sum of d primitive matrices R_k indexed by elements of $[d]$, where $R_k(i, j) = 1$ iff $k \in S_i \cap T_j$. Each R_k has rank 1, so the total rank of $M_{\mathcal{S},\mathcal{T}}$ is at most d. Conversely, any $\{a, b\}$-valued matrix that is a sum of d primitive matrices corresponds to an $\{a, b\}$-cross-intersecting set system over a universe of size d.

▶ **Theorem 7.** *The log-rank conjecture (Conjecture 1) is equivalent to Conjecture 6.*

Proof. As mentioned above, the fact that Conjecture 6 is implied by the log-rank conjecture was proved by Sgall [18]. We include the proof here for completeness. For an $\{a, b\}$-cross-intersecting family pair $(\mathcal{S}, \mathcal{T})$ with $a \leq b$, we can write $M_{\mathcal{S},\mathcal{T}} = (b - a)B + aJ$, where B is some Boolean matrix and J is the all-ones matrix. Then, $\mathrm{r}(B) - 1 \leq \mathrm{r}(M_{\mathcal{S},\mathcal{T}}) \leq \mathrm{r}(B) + 1$. By Conjecture 10, B has a monochromatic rectangle of density $2^{-\operatorname{polylog}(\mathrm{r}(B))}$, which yields a monochromatic rectangle in $M_{\mathcal{S},\mathcal{T}}$ of density $2^{-\operatorname{polylog}(\mathrm{r}(M))} \geq 2^{-\operatorname{polylog}(d)}$ from the discussion above.

For the reverse direction, assume Conjecture 6 holds. Let A be a Boolean $m \times n$ matrix with signed rectangle rank u. By Theorem 2 and Conjecture 10, it suffices to find a monochromatic rectangle in A of density at least $2^{-\operatorname{polylog}(u)}$. We reduce this to the problem of finding a large $\{a\}$- or $\{b\}$-cross-intersecting subfamily in an $\{a, b\}$-cross-intersecting set system. Let a, b be integers chosen later. Define a matrix $A' \in \{a, b\}^{m \times n}$ as follows:

$$A'(i, j) := \begin{cases} a & \text{if } A(i, j) = 1, \\ b & \text{if } A(i, j) = 0. \end{cases}$$

Our goal is to show that A' corresponds to an $\{a, b\}$-cross-intersecting set system over a universe of size $d = \Theta(u)$.

Let $A = \sum_{i=1}^{u} \varepsilon_i R_i$ be the signed rectangle rank decomposition for A. We now construct a set of $2u$ primitive matrices $\{R'_i\}$ such that $A' = \sum_{i=1}^{2u} R'_i$, by replacing each R_i with a pair of primitive matrices R'_{2i-1}, R'_{2i} depending on the sign of ε_i.

- If $\varepsilon_i = 1$, let $R'_{2i-1} = J$ (the all-ones matrix), and $R'_{2i} = R_i$. Then, $R'_{2i-1} + R'_{2i} = J + R_i$
- If $\varepsilon_i = -1$, let $A_i \subseteq [m]$, $B_i \subseteq [n]$ be such that $R_i = 1$ on $A_i \times B_i$ and zero elsewhere. Define R'_{2i-1} to be 1 on $([m] \setminus A_i) \times [n]$ and 0 elsewhere, and R'_{2i} to be 1 on $A_i \times ([n] \setminus B_i)$. Then $R'_{2i-1} + R'_{2i} = J - R_i$.

In either case, the pair (R'_{2i-1}, R'_{2i}) replaces $\varepsilon_i R_i$ by $J + \varepsilon_i R_i$. Thus, each entry of A' satisfies $A'(i, j) = A(i, j) + u$, and therefore A' corresponds to a $\{u, u + 1\}$-cross-intersecting set family over $[d]$ for $d = 2u$. Applying Conjecture 6, we obtain a large monochromatic rectangle in A', and hence in A, of density at least $2^{-\operatorname{polylog}(u)}$. ◀

▶ Remark 11. The above proof shows that the log-rank conjecture is equivalent to a special case of Conjecture 6 for $\{a, a + 1\}$-cross-intersecting set systems.

4 Generalizing the Main Theorem to tensors

The log-rank conjecture extends naturally to Boolean tensors. Motivated by this perspective, we show that Theorem 2 continues to hold for the tensor rank of constant-order Boolean tensors, giving an equivalent formulation of the log-rank conjecture for tensors.

Let T be an order-ℓ Boolean tensor: a multilinear map $[n_1] \times \ldots \times [n_\ell] \to \{0, 1\}$ (for some natural numbers $n_1, \ldots, n_\ell$). In other words, T can be expressed as an ℓ-dimensional array whose entries are all in $\{0, 1\}$.

We say T has tensor rank 1 if there are maps $v_i : [n_i] \to \mathbb{R}$ for $i \in [\ell]$ such that

$$T(x_1, \ldots, x_\ell) = v_1(x_1) \cdot \ldots \cdot v_\ell(x_\ell).$$

The *tensor rank* of a tensor T, denoted by $\operatorname{tr}(T)$ is the minimum number r such that T can be expressed as the sum of r rank-1 tensors.

A *primitive tensor* is the natural generalization of a primitive matrix: it is a rank-1 tensor where v_i's map to $\{0, 1\}$. That is, a primitive tensor can be expressed as a multidimensional array which is all-1 on some product set $Q_1 \times \ldots \times Q_\ell$ for $Q_i \subseteq [n_i]$ and is all-0 elsewhere. Having this, the *partitioning number* and *signed rectangle rank* of a tensor are defined analogously as the minimum integers p and r, respectively, such that $T = \sum_{i=1}^{p} S_i$ and $T = \sum_{i=1}^{r} \varepsilon_i S_i$, where S_i are primitive tensors and $\varepsilon_i \in \{\pm 1\}$.

The log-rank conjecture for tensors can be formulated as follows.

▶ **Conjecture 12** (Log-rank conjecture for tensors). *For any constant-order Boolean tensor T,*

$$\log \mathrm{p}(T) \leq (\log \operatorname{tr}(T))^{O(1)}.$$

In the communication complexity language, the log-rank conjecture for tensors states that the *multiparty number-in-hand communication complexity* of a tensor T is polynomially related to the logarithm of the tensor rank of T (see [2]). It is known that the log-rank conjecture does not hold for tensors of super-polynomial order [9].

We prove that Theorem 2 generalizes to tensors.

▶ **Theorem 13.** *Let $T : [n_1] \times \ldots \times [n_\ell] \to \{0,1\}$ be an order-ℓ Boolean tensor with $\ell \geq 2$. If the tensor rank of T is r, then T can be written as a ± 1-linear-combination of at most $(cr \log r)^{(\ell-1)}$ primitive tensors, where c is an absolute constant.*

A direct corollary of this theorem is an equivalent formulation of the log-rank conjecture for tensors, analogous to Conjecture 3 for matrices.

▶ **Conjecture 14.** *Every constant-order Boolean tensor T satisfies* $\log \mathrm{p}(T) \leq (\log \mathrm{srr}(T))^{O(1)}$.

Finally, let us set up the proof of Theorem 13. A *slice* of an order-ℓ tensor is the order-$(\ell-1)$ tensor obtained by taking a coordinate and setting it to a fixed value. For $\lambda \in [\ell]$, a *λ-slice* is a slice where we specify that λ is the coordinate to be fixed. Similar to the proof of Theorem 2, we define a slice-sum to be the entrywise sum of slices, and an independent set of slices is one where all of its subsets have distinct slice-sums.

▷ Claim 15. Let S be an independent set of λ-slices of T. Then $|S| \leq O(r \log r)$, where r is the tensor rank of T.

Proof. Let $m = \prod_{i \neq \lambda} n_i$. Consider the flattening of T to a matrix $M_T \in \{0,1\}^{m \times n_\lambda}$. Then M_T has rank at most r and, following the bijection between λ-slices of T and columns of M_T, there is an independent set of columns S' in M_T with $|S'| = |S|$. Apply Claim 8 to obtain $|S'| \leq O(r \log r)$. ◁

▷ Claim 16. Let S be a maximal independent set of λ-slices of T. Then every λ-slice of T can be expressed as a ± 1-linear combination of λ-slices in S.

We omit the proof of Claim 16 as it is analogous to Claim 9.

Proof of Theorem 13. The proof is by induction. The base case is $\ell = 2$, which is proven by Theorem 2. For $\ell > 2$, arbitrarily choose a coordinate λ and use Claim 16 to express T as a ± 1-linear combination of a maximal independent set of λ-slices S. Each λ-slice of S has tensor rank at most r, and so by the inductive hypothesis, each λ-slice in S can be expressed as a ± 1-linear combination of $(cr \log r)^{(\ell-2)}$ order-$(\ell-1)$ primitive tensors.

In a similar fashion to the proof of Theorem 2, we therefore can write T as a ± 1-sum of $|S| \cdot (cr \log r)^{(\ell-2)}$ order-ℓ primitive tensors. Conclude by substituting $|S| = O(r \log r)$ using Claim 15. ◀

▶ Remark 17. We note that the proof of Theorem 13 only uses the tensor rank as an upper bound for the *flattening rank* of a tensor, which is the maximal rank of any flattening of it as an $n_\lambda \times \left(\prod_{i \neq \lambda} n_i\right)$ matrix over $\lambda \in [\ell]$. If this flattening rank is r, then the conclusion of Theorem 13 still holds.

Moreover, in this formulation, the quantitative bound of $O((r \log r)^{\ell-1})$ primitive tensors is near optimal. Indeed, consider all tensors $T : [r]^\ell \to \{0,1\}$. Any such tensor has flattening rank $\leq r$. On the other hand, a simple counting argument shows that most such tensors need $\Omega(r^{\ell-1}/\ell)$ primitive tensors in their decomposition.

If we return however to the original formulation of Theorem 13 using tensor rank, it is unclear if the bound is close to tight. In fact, we suspect that it can be significantly improved.

▶ **Question 18.** *Can the bound in Theorem 13 be improved to $o(r^{\ell-1})$?*

References

1 Kaspars Balodis, Shalev Ben-David, Mika Göös, Siddhartha Jain, and Robin Kothari. Unambiguous DNFs and Alon–Saks–Seymour. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 116–124. IEEE Computer Society, 2021. doi:10.1109/FOCS52979.2021.00020.

2 Jan Draisma, Eyal Kushilevitz, and Enav Weinreb. Partition arguments in multiparty communication complexity. *Theoretical Computer Science*, 412(24):2611–2622, May 2011. doi:10.1016/j.tcs.2010.01.018.

3 Siemion Fajtlowicz. On conjectures of Graffiti. In *Proceedings of the First Japan Conference on Graph Theory and Applications (Hakone, 1986)*, volume 72, pages 113–118, 1988. doi:10.1016/0012-365X(88)90199-9.

4 Peter Frankl and Vojtěch Rödl. Forbidden intersections. *Trans. Amer. Math. Soc.*, 300(1):259–286, 1987. doi:10.2307/2000598.

5 Zach Hunter, Aleksa Milojević, Benny Sudakov, and István Tomon. Disjoint pairs in set systems and combinatorics of low rank matrices. *arXiv preprint*, 2024. arXiv:2411.13510.

6 Stasys Jukna. *Boolean function complexity: advances and frontiers*, volume 27. Springer, 2012. doi:10.1007/978-3-642-24508-4.

7 Peter Keevash and Benny Sudakov. Set systems with restricted cross-intersections and the minimum rank of inclusion matrices. *SIAM J. Discrete Math.*, 18(4):713–727, 2005. doi:10.1137/S0895480103434634.

8 Troy Lee and Adi Shraibman. Around the log-rank conjecture. *Israel Journal of Mathematics*, 256(2):441–477, 2023.

9 Yang D Li. Applications of monotone rank to complexity theory. *arXiv preprint*, 2011. arXiv:1102.2932.

10 Bernt Lindström. On a combinatorial problem in number theory. *Canadian Mathematical Bulletin*, 8(4):477–490, 1965. doi:10.4153/CMB-1965-034-2.

11 László Lovász and Michael Saks. Lattices, Möbius functions and communications complexity. In *Proceedings of 29th Annual Symposium on Foundations of Computer Science*, pages 81–90. IEEE Computer Society, 1988. doi:10.1109/SFCS.1988.21924.

12 László Lovász and Michael Saks. Communication complexity and combinatorial lattice theory. *Journal of Computer and System Sciences*, 47(2):322–349, October 1993. doi:10.1016/0022-0000(93)90035-U.

13 Shachar Lovett. Communication is bounded by root of rank. *J. ACM*, 63(1):Art. 1, 9, 2016. doi:10.1145/2724704.

14 Shachar Lovett. Unpublished note, 2021.

15 Noam Nisan and Avi Wigderson. On rank vs. communication complexity. *Combinatorica*, 15(4):557–565, 1995. doi:10.1007/BF01192527.

16 Cyriel van Nuffelen. Research problems: A bound for the chromatic number of a graph. *Amer. Math. Monthly*, 83(4):265–266, 1976. doi:10.2307/2318218.

17 Anup Rao and Amir Yehudayoff. *Communication complexity and applications*. Cambridge University Press, Cambridge, 2020.

18 Jiří Sgall. Bounds on pairs of families with restricted intersections. *Combinatorica*, 19(4):555–566, 1999. doi:10.1007/s004939970007.

19 Noah Singer and Madhu Sudan. Point-hyperplane incidence geometry and the log-rank conjecture. *ACM Trans. Comput. Theory*, 14(2):Art. 7, 16, 2022.

20 Benny Sudakov and István Tomon. Matrix discrepancy and the log-rank conjecture. *Math. Program.*, 212(1-2):567–579, 2025. doi:10.1007/s10107-024-02117-9.

21 Madhu Sudan and Noah Singer. Personal communication, 2025.

22 Thomas Watson. Nonnegative rank vs. binary rank. *Chicago Journal of Theoretical Computer Science*, 2016(2), 2016. doi:10.4086/cjtcs.2016.002.

Efficient Adversaries

Erfan Khaniki
University of Oxford, UK

Ján Pich
University of Oxford, UK

Dmitry Sokolov
Université de Montréal, Canada

Abstract

The size of Frege proofs can be characterized in terms of prover-adversary games of Pudlák and Buss. We consider a generalization of prover-adversary games to many standard proof systems and show that some of the major proof complexity lower bounds such as the constant-depth Frege lower bound for the pigeonhole principle based on the method of k-evaluations, the Resolution lower bound for the weak pigeonhole principle based on the method of pseudo-width and Razborov's $\mathsf{Res}(k)$ lower bound for Nisan-Wigderson generators based on expansion and a width lower bound (which is used to derive the $\mathsf{Res}(k)$-hardness of formulas expressing circuit lower bounds) are constructive in the sense that they yield efficient algorithms computing winning strategies of adversaries in the generalized games. This is in contrast with our second result saying that if (a) such a constructive lower bound exists for Extended Frege system EF for formulas expressing succinct circuit lower bounds for SAT and (b) EF is strong enough to prove efficiently the correctness of anticheckers for SAT, then it is easy to separate the canonical pair of EF.

2012 ACM Subject Classification Theory of computation → Proof complexity; Theory of computation → Circuit complexity; Theory of computation → Complexity theory and logic

Keywords and phrases proof complexity, circuit complexity, lower bounds, barriers, truth-table formula, pigeonhole principle

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.8

Funding *Erfan Khaniki*: This work was supported by the Royal Society University Research Fellowship URF\R1\211106 "Proof complexity and circuit complexity: a unified approach".
Ján Pich: This work was supported by the Royal Society University Research Fellowship URF\R1\211106 "Proof complexity and circuit complexity: a unified approach".
Dmitry Sokolov: This work was supported by the Swiss State Secretariat for Education, Research, and Innovation (SERI) under contract number MB22.00026.

Acknowledgements We are indebted to Moritz Müller for a simplification of the proof of Theorem 1, which allowed us to derive Theorem 2 for EF instead of G_1^*, and for many comments that clarified the presentation of the paper. We thank Neil Thapen for a correction of Lemma 19, Noel Arteche for pointing out to us that Item 2 in Theorem 1 implies $\mathsf{NP} \not\subseteq \mathsf{Circuit}[n^\ell]$, and Pavel Pudlák, Jan Krajíček and anonymous reviewers for suggestions and comments. This project was initiated at the Institute of Mathematics of the Czech Academy of Sciences. For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

1 Introduction

Proving circuit lower bounds for explicit Boolean functions is a central open problem in complexity theory, underlying questions such as the P vs NP problem.

The notorious difficulty of the problem has led to the development of a theory of formal barrier results. For example, already in the 80's Razborov [57] showed that the approximation method, used to derive monotone circuit lower bounds [55] and lower bounds against $\mathsf{AC}^0[p]$

© Erfan Khaniki, Ján Pich, and Dmitry Sokolov;
licensed under Creative Commons License CC-BY 4.0

[56, 69], cannot yield superquadratic lower bounds for non-monotone circuits. In the 90s, Razborov and Rudich [65] discovered the influential natural proofs barrier clarifying the limitations of major lower bound methods such as the method of random restrictions or the algebraic method of Razborov and Smolensky [56, 69]. More recently, the locality barrier explained why a successful implementation of hardness magnification, an approach to strong explicit circuit lower bounds sidestepping the natural proofs barrier, requires more substantial new ideas [13]. Such barrier results could potentially guide us towards the solution of the original problem, but it is also possible that they will eventually lead to the conclusion that strong circuit lower bounds are practically impossible to establish or to the discovery of new surprising algorithms, cf. [12].

The possibility that circuit lower bounds could lie beyond what can be achieved within feasible logical reasoning was investigated extensively by Razborov who used natural proofs to derive the unprovability of circuit lower bounds in certain theories of bounded arithmetic [59]. Such unprovability results can be typically interpreted as lower bounds on lengths of proofs of tautologies expressing circuit lower bounds in propositional proof systems. Indeed, after composing [59] Razborov switched to the more elegant and transparent framework of propositional logic and formulated his subsequent results in this setting; including a degree lower bound for deriving circuit lower bounds in Polynomial Calculus [60], a Resolution lower bound for formulas expressing circuit lower bounds based on the pseudo-width method [61, 62, 63] (which simplified and strengthened the original result of Raz [54]), and the hardness of circuit lower bounds for $\mathsf{Res}(k)$, an extension of Resolution operating with k-DNFs, for a small parameter k [64]. The last result remains the strongest propositional lower bound for tautologies encoding circuit lower bounds.[1]

Ultimately, in [64] Razborov formulated intriguing conjectures about a conditional hardness of Nisan-Wigderson (NW) generators for Frege and Extended Frege system EF: suitable NW-generators should form good proof complexity generators in the sense that tautologies postulating the existence of elements outside their range require super-polynomial-size proofs in Frege systems, assuming $\mathsf{P/poly}$ is hard on average for NC^1, and in EF, assuming $\mathsf{NP}\cap\mathsf{coNP}$ is hard on average for $\mathsf{P/poly}$. If true, these conjectures would present a dramatic barrier establishing that feasibly-constructive logical reasoning (which can be defined in terms of short EF proofs or as the provability in Cook's theory PV_1 [29, 16]) cannot prove explicit circuit lower bounds – $\mathsf{P/poly}$ lower bounds in the case of Frege and lower bounds for a specific kind of $(\mathsf{NP}\cap\mathsf{coNP})/\mathsf{poly}$ circuits in the case of EF.[2] A similar earlier conjecture of Krajíček [30, Conjecture 7.9], implying that, under the existence of strong pseudorandom generators, subexponential circuit lower bounds are hard for EF was motivated by the role of dual weak pigeonhole principle in bounded arithmetic. In [35], Krajíček formulated the conjecture in an unconditional form.

The conjectures of Razborov and Krajíček are interesting not only because of the potential meta-mathematical consequences but also because they suggest the existence of a strong connection between circuit complexity and proof complexity.

---

[1] There are several related results about the impossibility of deriving strong circuit lower bounds in bounded arithmetic [46, 49, 39, 26] which appear to be incomparable to the lower bounds for propositional systems, but it seems that strengthening them in a more substantial way requires new propositional lower bounds. In the propositional setting, Austrin and Risse [5] showed that Sum-of-Squares proof system, which proves efficiently the pigeonhole principle, cannot prove efficiently circuit lower bounds for certain functions f, but the lower bound does not hold for all functions f. An analogous meta-mathematical theory is currently under development also in the algebraic setting [42].

[2] A lot of contemporary complexity theory is formalizable within EF, e.g., the PCP theorem has short EF proofs [47], see also [45]. Some candidates for PV_1-unprovable theorems from computational complexity are given in [14] but they exploit witnessing theorems which (under standard hardness assumptions) cannot be used to get the PV_1-unprovability of deterministic circuit lower bounds for SAT [46, 44].

Proof complexity presents a logic-oriented approach to the separation of NP and coNP which proceeds by proving lower bounds on lengths of proofs in increasingly more powerful propositional proof systems. Unfortunately, Razborov's unprovability results as well as other existing proof complexity lower bounds work only for very weak proof systems. In fact, in some sense proving proof complexity lower bounds has turned out to be harder than proving circuit lower bounds: turning AC^0 lower bounds into constant-depth Frege lower bounds required developing a substantially more involved adaptation of the method of random restrictions in the form of k-evaluations [1, 38, 51] and it still remains an open problem to derive lower bounds for Frege systems operating with $\mathsf{AC}^0[2]$ circuits despite the existence of explicit $\mathsf{AC}^0[2]$ lower bounds [56, 69]. From the perspective of the attempts to derive propositional lower bounds for tautologies expressing circuit lower bounds this is somewhat ironic as it suggests that deriving such hardness results unconditionally might be harder than proving circuit lower bounds themselves.

Generally, the relation between circuit complexity and proof complexity is rather mysterious. Formal connections between the two areas exist if we restrict ourselves to weak computational models. For example, the method of feasible interpolation derives proof complexity lower bounds for many weak proof systems from circuit lower bounds for monotone circuits and related models [34]. In the opposite direction, lifting theorems yield lower bounds for various monotone computational models from lower bounds for weak propositional proof systems, see e.g. [66, 21]. In the algebraic setting, IPS lower bounds imply $\mathsf{VP} \neq \mathsf{VNP}$ [22]. Unfortunately, these connections break when moving to strong proof systems and the Boolean setting. Recently a conditional connection between lower bounds for strong proof systems and strong circuit lower bounds such as $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ has been established [50, 3], but it requires the assumption that the proof system proves efficiently some circuit lower bound such as E being hard on average for subexponential-size circuits.

As a consequence of the lack of formal links between circuit and proof complexity, the barrier results in circuit complexity do not seem to say much about the possibility of deriving proof complexity lower bounds for strong proof systems. The question of identifying a "natural proofs"-like barrier in proof complexity was raised already by Beame and Pitassi [6]. Given that some prominent proof complexity lower bounds are harder to derive than their counterparts in circuit complexity, it might seem even more ironic that we have not been able to identify such a barrier, see Section 1.2 for a discussion of earlier attempts. Is there an explanation for our inability to prove Frege lower bounds like those in Razborov's conjecture? In this paper we introduce a notion that has a potential to answer this question.

1.1 Our contribution

Following the paradigm of natural proofs, which turn constructive circuit lower bounds into efficient algorithms, we want to find a way to interpret (sufficiently constructive) proof complexity lower bounds as a construction of new surprising algorithms.

In order to get closer to such a conclusion we focus on prover-adversary games used by Pudlák and Buss [53] to characterize the lengths of proofs in Frege systems. We outline a slightly modified formulation of these games which is sufficiently general to capture uniformly the existing lower bounds for many different proof systems.

An s-size lower bound for a formula ϕ in a proof system P is often obtained by defining a strategy for an adversary who responds to queries of the prover in the following manner. Given a supposed s-size P-refutation π of $\neg\phi$, the prover sends to the adversary the last formula ψ of π and the adversary responds with 1 or 0. After that, the prover sends formulas

$\psi_1, \ldots, \psi_k$ from which ψ was derived in π by a single application of a derivation rule and the adversary responds with 1 or 0 to each of them. The prover then proceeds to a subproof of π which derives a formula $\psi_i \in \{\psi_1, \ldots, \psi_k\}$ assigned 0 by the adversary, if such a formula exists. If such a formula does not exist and the adversary assigned 0 to ψ, the adversary has been proven to be inconsistent (caught). The game continues in this way possibly until the players reach the axioms or $\neg\phi$. The adversary has to assign 0 to the contradiction (which is the last formula ψ, if π is a refutation) and 1 to the axioms and $\neg\phi$ (indicating that the adversary posses an assignment falsifying ϕ). If we can show that the adversary maintains consistency, we can conclude that π was not a correct refutation of $\neg\phi$. For a more elaborated discussion of the prover-adversary games see Section 2.

Intuitively, we define a constructive proof complexity lower bound by the existence of an "efficient adversary"; more precisely, by an efficient algorithm generating the responses of the adversary on given formulas such that for each short P-refutation (not necessarily of $\neg\phi$) the adversary (who might not be given the entire refutation) stays consistent in the game described above, cf. Definitions 8-9. The adversary always has to assign 1 to $\neg\phi$ and the axioms of P but not necessarily to other initial formulas of a given P-refutation. A randomized efficient adversary is allowed to use randomness so that for each short P-refutation the adversary can choose a random string and use it to determine its responses. To indicate that an adversary corresponds to a lower bound (for a formula ϕ) in a proof system P we speak of P-adversaries (claiming that $\neg\phi$ is satisfiable).

Note that in order to make a proof complexity lower bound for a formula ϕ constructive, we do not need to construct adversaries for short refutations of $\neg\phi$ – such refutations do not exist if the lower bound holds. In a sense, an efficient adversary rules out more than just the existence of a short refutation of $\neg\phi$, it could be said that it rules out the existence of a short "pseudoproof" of ϕ, cf. Proposition 10.

Our first theorem shows that constructive EF lower bounds for formulas $\mathsf{lb}_A(\mathsf{SAT}, n^2)$ (expressing that SAT is not computable by n^2-size circuits on a set of inputs $A \subseteq \{0,1\}^n$) given by randomized efficient EF-adversaries[3] claiming that $\neg\mathsf{lb}_A(\mathsf{SAT}, n^2)$ is satisfiable can be used to break the canonical pair of EF assuming that EF is capable of efficiently deriving a statement about antickeckers. Informally, the canonical pair of a proof system P consists of two disjoint NP sets: the set of all satisfiable formulas and the set of all unsatisfiable formulas whose unsatisfiability has a short P-proof. The canonical pair of P is hard to separate if there is no small circuit accepting all inputs from the first set and rejecting all inputs from the second set, cf. [58].

▶ **Theorem 1** (From adversaries to algorithms – Informal, cf. Theorem 11)**.**
Assume 1 - 2:

1. (The canonical pair of EF is hard.) *For each poly(n)-size circuit D with n inputs, there is z such that $D(z) = 0$ but z is a satisfiable formula or $D(z) = 1$ but there is a poly(n)-size EF-refutation of z.*
2. (Provably constructive antickeckers.) *There is $A \subseteq \{0,1\}^n$, $|A| = poly(n)$, and a poly(n)-size circuit F such that for each $z \in \{0,1\}^n$ there is a poly(n)-size EF-proof of the statement expressing that "if an n^2-size circuit B solves SAT on A, then a poly(n)-size circuit $F(B)$ solves SAT on z".*

Then for each randomized poly(n)-time EF-adversary claiming that $\neg\mathsf{lb}_A(\mathsf{SAT}, n^2)$ is satisfiable, there is a poly(n)-size EF-refutation (not necessarily of $\neg\mathsf{lb}_A(\mathsf{SAT}, n^2)$) catching the adversary with probability $\geq 1/2 - 1/n$ over the randomness of the adversary.

[3] The definition of EF-adversaries requires a careful treatment of the extension axioms, see Section 2.

Reading it contrapositively, Theorem 1 says that efficient adversaries defining constructive EF lower bounds for formulas $\mathsf{lb}_A(\mathsf{SAT}, n^2)$ yield efficient algorithms breaking the canonical pair of EF (a condition similar to the weak automatability of EF), assuming Condition 2 holds. Condition 2, the second assumption of the theorem, is less standard. It postulates the existence of a set of "anticheckers" A such that a small circuit which solves SAT on A can be turned into a circuit which solves SAT on any other input. Such A is known to exist unconditionally, cf. Section 2, but Condition 2 additionally says that this fact is efficiently provable in EF. It can be shown that Condition 2 follows if, e.g., EF proves efficiently that (a) E is hard on average for subexponential-size circuits and EF proves efficiently that (b) the existence of one-way permutation can be based on $\mathsf{NP} \not\subseteq \mathsf{P/poly}$. Both statements (a) and (b) are considered to be more approachable than non-uniform circuit lower bounds for SAT. The plausibility of Condition 2 can be further enhanced by replacing EF in Theorem 1 by stronger systems such as (a propositional equivalent of) ZFC, the theorem remains valid in this case as well. As pointed out by Noel Arteche (private communication), Condition 2 implies also that $\mathsf{NP} \not\subseteq \mathsf{Circuit}[n^\ell]$, for each ℓ. See Section 2 for a more detailed discussion.

The parameter $1/2 - 1/n$ in Theorem 1 is not essential for the purpose of a barrier result, any constant would work equally well.

The proof of Theorem 1 is straightforward: Since the adversary claims that it has a circuit solving SAT on A, using Condition 2 we can try to force it (with EF-queries) to claim that it has a circuit C which solves SAT everywhere and to simulate the computation of circuit C on any input z. This almost works except that we only get an efficient algorithm breaking the canonical pair of EF: If the adversary claims that C says that z is a satisfiable formula and z is unsatisfiable, we need the additional assumption that z has a short EF-refutation to reveal the inconsistency of the adversary (and force it to avoid such claims).

The proof works also for formulas $\neg\mathsf{tt}(\mathsf{SAT}, n^2) := \neg\mathsf{lb}_{\{0,1\}^n}(\mathsf{SAT}, n^2)$ instead of $\neg\mathsf{lb}_A$, but it crucially exploits the fact that the adversary is efficient w.r.t. the queries of the prover, not w.r.t. the formula for which the lower bound is being proved (which is of size $2^{O(n)}$ in this case). In fact, in the case of the tt-formulas we can avoid Condition 2 which can be replaced by a formalization of the search-to-decision reduction for SAT. Further, the switch from $poly(n)$-size to $2^{O(n)}$-size formulas allows us to replace the first assumption in Theorem 2 with $\mathsf{SAT} \notin \mathsf{P/poly}$. As a result we get

▶ **Theorem 2** (Informal, cf. Theorem 11)**.**
Assume that $\mathsf{SAT} \notin \mathsf{P/poly}$*. Then for each randomized p-time* EF*-adversary claiming that* $\neg\mathsf{tt}(\mathsf{SAT}, n^2)$ *is satisfiable, there is a* $2^{O(n)}$*-size* EF*-refutation catching the adversary with probability* $\geq 1/2 - 1/n$ *over the randomness of the adversary.*

Constructive proof complexity lower bounds. We complement Theorem 1 by showing that some of the major existing proof complexity lower bounds are indeed constructive.

As the first example we design efficient adversaries for the strongest known lower bound for lb-tautologies due to Razborov [64], who showed that there are no $2^{t^{\Omega(1)}}$-size $\mathsf{Res}(\Omega(\log t))$-refutations of $\neg\mathsf{lb}_W(f,t)$, for $n^2 \leq t \leq 2^n$ and any $W \subseteq \{0,1\}^n$. More precisely, we design efficient adversaries for "proof complexity generator" formulas $\tau_W(A,b)$ (expressing that a system of linear equations $Ax = b$ is solvable) instead of $\neg\mathsf{lb}_W(f,t)$. This is justified by the fact that Razborov's proof proceeds by reducing the lower bound for $\mathsf{lb}_W(f,t)$ to a lower bound for $\neg\tau_W(A,b)$ and Theorem 1 holds even if the formula $\neg\mathsf{lb}_W(\mathsf{SAT}, n^2)$ is replaced by any formula ϕ such that EF proves efficiently $\phi \to \neg\mathsf{lb}_W(\mathsf{SAT}, n^2)$, cf. Remark 13. The focus on the τ-formulas allows us to avoid the problem of encoding the reduction of lower bounds for $\mathsf{lb}_W(f,t)$ to lower bounds for $\neg\tau_W(A,b)$ in the strategy of the adversary, see Section 3.1 for more details.

▶ **Theorem 3** (Constructive Res(k) lower bound of Razborov – Informal, cf. Theorem 14)**.** *For each sufficiently small $\delta > 0$, there is $\epsilon > 0$ and a poly(t^{t^δ})-size randomized* Res(k)*-adversary C claiming that $\tau_W(A, b)$ is satisfiable, where $k = \epsilon \log t$ and $|W| = poly(t)$, such that for each $2^{t^{\Omega(\delta)}}$-size* Res(k)*-refutation π of any set of clauses, with probability $\geq 9/10$ over the randomness of C, π does not catch C.*

The complexity of the adversary in Theorem 3 is subexponential, but the formal version of Theorem 1 (Theorem 11) captures this setting of parameters. Our proof of Theorem 3 does not yield better than 2^n-size adversaries for $W = \{0,1\}^n$, which is the setting that would correspond to a constructive Res(k)-adversary for $\neg\mathsf{tt}(f, t)$, so we can match Theorem 3 with Theorem 1 but not with Theorem 2.

The proof of Theorem 3 requires a detailed analysis of Razborov's original proof combined with a different proof of the final width lower bound. We did not manage to exploit [64] in a more black-box fashion. Verifying Theorem 3 thus requires reading the entire proof of Razborov (with a more sparse restriction while keeping the original expansion parameters) except the switching lemma which can be treated mostly as a black-box. As a result of this our write-up becomes dense, with many references to the intricacies of the proof in [64]. We replace the width lower bound in Razborov's proof by an adaptation of a width lower bound presented by Krajíček [34, Lemma 13.4.5]. In the adaptation, instead of using Krajíček's expansion, we use Razborov's "lossless" expansion.

The second example we present is a constructive version of the method of pseudowidth, a sophisticated analogue of the popular method of width [7], which was used to derive Resolution (Res) lower bounds for the weak pigeonhole principle and for formulas encoding circuit (or even DNF) lower bounds [54, 61, 62, 63, 20]. Pseudowidth and width are currently incomparable: It is not known how to prove Res lower bounds for the weak (enough) pigeonhole principle or for formulas encoding DNF lower bounds using width and it is not known how to prove Res lower bounds for proof complexity generator formulas $\tau_W(A, b)$ considered in Theorem 3 using pseudowidth.

We present a constructive version of the lower bound for the weak pigeonhole principle PHP^m_n mapping m pigeons to n holes.

▶ **Theorem 4** (Constructive pseudowidth lower bound – Informal, cf. Theorem 20)**.** *There is a poly(m)-size randomized* Res*-adversary C claiming that $\neg\mathsf{PHP}^m_n$ with $m = 2^{n^{1/5}}$ is satisfiable such that for each $2^{\Omega(n^{1/4})}$-size* Res*-refutation π of any set of clauses, with probability $\geq 9/10$ over the randomness of C, π does not catch C.*

Our proof of Theorem 4 differs substantially from the proofs in [54, 61, 62, 63, 20] and might be of independent interest. The strategy of the adversary from Theorem 4 is guided by random partial multi-matchings. The notion of pseudowidth arises here naturally from the observation that the partial multi-matchings satisfy certain clauses with high probability – these clauses can be thus treated as pseudo-axioms.

Finally, we look at the crown jewel of proof complexity, the lower bound on the length of proofs of the pigeonhole principle PHP_n (mapping $n+1$ pigeons to n holes) in depth d Frege systems (AC^0_d-Frege), which is obtained via k-evaluations [1, 38, 51].

Unfortunately, it is not known how to prove a lower bound for tt-tautologies in AC^0-Frege. The closest result of this kind was obtained in [27], where it was shown that a particular kind of $\mathsf{NP} \cap \mathsf{coNP}$ lower bounds (similar to formulas $\neg\tau_W(A, b)$), formalized in a suitable way, does not admit short proofs in AC^0-Frege. The lower bound from [27] proceeds by a reduction to the PHP_n lower bound.

▶ **Theorem 5** (Constructive AC^0-Frege lower bound – Informal, cf. Theorem 21)**.** *Let $\epsilon < 1/9^d$. There is a poly(n^{n^ϵ})-size randomized AC^0_d-Frege-adversary C claiming that $\neg\mathsf{PHP}_n$ is satisfiable such that for each 2^{n^ϵ}-size AC^0_d-Frege-refutation π of any set of clauses, with probability $\geq 9/10$ over the randomness of C, π does not catch C.*

The proof of Theorem 5 follows directly by inspection of the original proof. On high level, the reason why the theorem holds is that a random partial matching "kills" short AC^0-Frege proofs of PHP_n with high probability.

Unification. Theorems 3-5 imply the corresponding proof complexity lower bounds for $\mathsf{Res}(k)$, Res and AC^0-Frege. Moreover, our presentation of lower bound methods of width, pseudowidth and k-evaluation in terms of adversaries provides a more unifying perspective on these methods by showing how they arise from natural choices of strategies of adversaries.

1.2 Related work

There are several results in proof complexity that could be interpreted as barriers for deriving proof complexity lower bounds.

Incompleteness. Already Gödel's second incompleteness theorem implies that, e.g., if ZFC is consistent, then ZFC cannot prove superlinear lower bounds for the propositional proof system obtained from ZFC in a natural way (otherwise ZFC would prove its own soundness). For similar reasons, a result of Krajíček [31, Corollary 3.2] implies that Buss's theory S^1_2 cannot prove superlinear lower bounds on the so called Implicit EF. These results, however, do not present an actual barrier for proving proof complexity lower bounds because in proof complexity we prove lower bounds under the assumption that the proof system in question is sound.

Cook's translation. A more substantial proof complexity barrier was obtained by Krajíček-Pudlák [36], who showed that Cook's theory PV_1 cannot prove that there exists a tautology which requires superpolynomial, say $n^{\log n}$-size, EF proofs (Buss [11] and Cook-Urquhart [18] derived the same unprovability result for the intuitionistic PV_1). The result crucially relies on the formalization of the statement of the lower bound, which is unfortunately not natural from the perspective of PV_1 in the sense that $n^{\log n}$ is not the length of a number. Consequently, PV_1 cannot use induction over the proofs of size $n^{\log n}$. Informally, the theory cannot see (and analyze) the whole $n^{\log n}$-size proof properly.

Nisan-Wigderson generators and the sum-check protocol. This drawback was eliminated in a proof complexity barrier obtained in [49] (building on [32, 46] and extended in [39, 26]), where it was shown that subexponential average-case lower bounds for nondeterministic circuits are not provable in PV_1 (and certain extensions of PV_1). In a related work using completely different methods, Chen, Rothblum and Tell [15] constructed a non-deterministic algorithm P such that under a derandomization assumption S^1_2 cannot prove that P is not a p-bounded propositional proof system.[4] Both [49] and [15] use a formalization which allows the theory to perform induction over objects of even exponential size $2^{O(n)}$. Unfortunately, the complexity of these formalizations is Σ^p_2 and while theories like PV_1 and S^1_2 can prove a

[4] [15] presents the unprovability result for APC_1 but it works for S^1_2 as well.

lot of complexity theory and are generally very powerful when reasoning with P-time objects and deriving coNP statements, they are not well-designed for reasoning with NP statements[5], which is why these results do not present a significant barrier in the practice of proving lower bounds.

This issue was partially addressed in [28]. We believe that [28] implicitly shows that there is a concrete MA proof system R such that, if it is hard to derive superpolynomial circuit lower bounds in a sufficiently strong propositional proof system P, then the theory T_P corresponding to P cannot prove that R does not admit p-size proofs of all tautologies of length n. Here, the lower bound for R assumes that $2^{O(n)}$ is the length of a number and it is formulated as a coNP statement, so the T_P-provability of the lower bound for R does not necessarily imply that P simulates R and T_P should be in principle very powerful when proving lower bounds for R. The result in [28] would therefore give us a remarkable proof complexity barrier if we could only prove the assumption or at least replace it by a standard complexity-theoretic conjecture.

Krajíček [33] used the methods of [32] to show that under standard hardness assumptions, there is no efficient algorithm which given an algorithm P (potentially defining a propositional proof system) outputs a tautology hard for P or a witness that P does not define a consistent proof system.

Natural properties. In an attempt to mimic the natural proofs barrier, [48] defined a version of natural properties against proof systems roughly as the efficient provability of EF lower bounds for formulas $\mathsf{tt}(f, n^{O(1)})$, for many functions f, and showed that if super-bits exist (an assumption about the existence of pseudorandom generators safe against nondeterministic circuits), then such natural properties against EF do not exist or EF does not prove efficiently any formula $\mathsf{tt}(f, n^{O(1)})$. If we knew how to avoid the second disjunct about the EF-hardness of $\mathsf{tt}(f, n^{O(1)})$, this would present a close analogue of the natural proofs barrier in proof complexity.

A combination of the results in [28] and [48] shows that under the assumption of the existence of super-bits, either there is no natural proof against P or P cannot prove proof complexity lower bounds for a concrete MA system. The conclusion is interesting, but it does not tell us much about the hardness of deriving lower bounds for systems like EF.

Self-provability. More recently it has been shown that EF lower bounds imply circuit lower bounds such as $\mathsf{SAT} \notin \mathsf{P/poly}$ [50, 3] under various assumptions such as that EF proves efficiently that E is hard on average for subexponential-size circuits. If we could avoid this assumption, the resulting strong connection between EF lower bounds and circuit lower bounds could explain a part of the difficulty of deriving EF lower bounds.

A problem with interpreting the results from [48] or [50, 3] as proof complexity barriers is that they require the assumption that EF proves efficiently a sufficiently strong circuit lower bound, which is the very statement we are trying to refute when proving proof complexity lower bounds (in Razborov's conjecture).

TFNP, automatability & algebraic complexity. Li, Li and Ren [40] initiated a systematic investigation of proof complexity of proof complexity lower bounds through the lens of TFNP, but they restrict their attention to Resolution lower bounds, see also [19]. Similarly, the

[5] For example, it follows from Buss's witnessing theorem [10] that S^1_2 cannot prove that a TFNP problem is total unless the problem is in P.

NP-hardness results for the automatability of various weak proof systems are based on the hardness of deriving proof complexity lower bounds in systems such as Resolution [4, 2]. Lu, Santhanam and Tzameret [42] showed that certain lower bounds for a constant-depth version of IPS are hard to prove in $\mathsf{AC}^0[2]$-Frege. The result builds on a work of Santhanam and Tzameret [67] where IPS lower bounds for formulas expressing IPS lower bounds for VP $\neq$ VNP were shown to follow from VP $\neq$ VNP. A drawback of [67] when interpreted as a barrier result is that the conclusion holds only for infinitely many input lengths, [67] does not rule out the possibility that IPS proves efficiently lower bounds for IPS for VP $\neq$ VNP on an infinite and dense set of input lengths. Nevertheless, the potential sensitivity to the input length is a valuable information about the possibility of deriving proof complexity lower bounds for strong proof systems.

Upper bounds. Ultimately, the prospect of proving proof complexity lower bounds is limited by the possible existence of surprising upper bounds. In this context, it is worth mentioning an MA protocol of Williams which breaks a version of SETH for MA [70]. A recent result of Mazor and Pass [43] and Hirahara, Ilango and Williams [24] constructs subexponentials-size (deterministic) circuits for more specific problems like MCSP, but so far all these upper bounds remain in the regime of SETH.

All in all, while there have been several previous results that can be seen as some kind of proof complexity barriers, they are significantly less impactful than natural proofs in circuit complexity. In our view, the concept of efficient adversaries introduced in the present paper comes the closest to the potential "natural proofs barrier" in proof complexity so far despite the fact that the second assumption in Theorem 1 is similar to the assumption that EF proves efficiently some circuit lower bound. Theorem 2 could turn out as the main barrier, if the existing lower bounds such as the $\mathsf{Res}(k)$ lower bound for $\mathsf{tt}(f,t)$ can be made more constructive by designing adversaries who are efficient w.r.t. the queries of the prover, not w.r.t. the formula for which the lower bound is proved.

1.3 Open problems

Constructivity of proof complexity lower bounds. The main problem we leave open is the question of constructing efficient adversaries for all the remaining major proof complexity lower bounds. As the proofs of Theorems 3-4 show, it is not always obvious how to constructivize an existing lower bound. For example, in Theorem 4 we only obtained efficient adversaries for the weak pigeonhole principle, but the question of constructing such adversaries for Res lower bounds for the *functional* weak pigeonhole principle [62, 63] remains open. Further, it is not clear how to get efficient adversaries from lower bounds exploiting "deterministic" restrictions such as the recent super-linear lower bound for constant-depth Frege systems by Gryaznov and Talebanfard [23]. In general, lower bounds obtained via the width-size relation of Ben-Sasson and Wigderson [7] use deterministic restrictions as well and we do not know how to make them constructive. That is, we do not have a general strategy for replacing deterministic restricitions whose existence is shown non-constructively by random restrictions which would work with high probability. A prominent example we leave as an open problem is the constructivization of lower bounds based on feasible interpolation such as Pudlák's lower bound for Cutting Planes CP [52] – notably, it is still an open problem to show that $\mathsf{tt}(f,s)$ tautologies are hard for CP. Nevertheless, a significant barrier for using feasible interpolation to derive lower bounds for strong proof systems has been established already in [37, 9, 8].

▶ **Problem 6.** *Does the* CP *lower bound of Pudlák [52] admit efficient adversaries?*

If there are lower bounds which cannot be made constructive, can we rule out the possibility of a more substantial proof complexity barrier by showing at least that under a standard complexity-theoretic assumption some proof complexity lower bound does not admit an efficient adversary?

If we cannot construct an efficient adversary for some lower bound, if for each efficient P-adversary claiming that $\neg\phi$ is satisfiable, there is a short P-refutation (not necessarily of $\neg\phi$) catching the adversary, we can say that the refutations constitute a short *P-pseudo-proof* of ϕ. Is there a proof system P with p-size pseudo-proofs of all tautologies? To formulate the question more formally we consider only proof systems EF_P obtained by extending EF with axioms postulating the soundness of a proof system P. EF_P simulates P [34]. An EF_P-adversary is an EF-adversary who is additionaly required to assign 1s to all instantiations of the axiom postulating the soundness of P.

▶ **Problem 7** (NP = coNP w.r.t. efficient adversaries)**.** *Is there a proof system P such that for each tautology ϕ, for each $(|\phi|+n)^3$-size EF_P-adversary claiming that $\neg\phi$ is satisfiable there is a poly$(|\phi|)$-size EF_P-refutation catching the adversary?*

Avoiding anticheckers. Is it possible to remove the assumption about the provably constructive anticheckers from Theorem 1? In order to do so, it might be helpful to interpret efficient adversaries as algorithms for the search problems of finding errors in supposed short refutations of hard formulas, cf. [40].

Efficient adversaries w.r.t. tt-tautologies and other formulas. Theorem 2 does not rule out the possibility of proving constructively EF lower bounds for $\mathsf{tt}(f,s)$ formulas with adversaries which are efficient (not w.r.t. the queries of the prover but) w.r.t. the size of the $\mathsf{tt}(f,s)$ formula, i.e. with adversaries of size $2^{O(n)}$. Is it possible to make Theorem 2 work in this case as well? Can we further extend the theorem to all formulas instead of just formulas expressing circuit lower bounds? The latter question is related to the problem of proving that MCSP is NP-complete. Another relevant problem in this direction is to make Theorem 3 work for $\mathsf{tt}(f,t)$ with better than $2^{O(n)}$-size adversaries, i.e. to construct $\mathsf{Res}(k)$-adversaries for $\neg\mathsf{tt}(f,t)$ which are efficient w.r.t. the queries of the prover.

Simulations preserving efficient adversaries. The notion of efficient adversaries appears to be sensitive to the specific choice of proof system in question. Do p-simulations preserve efficient adversaries? That is, is it possible to show that if there is an efficient P-adversary for Φ and the proof system P is p-equivalent to a proof system R, then there is an efficient R-adversary for Φ? Is there another natural notion of simulation between proof systems which preserves efficient adversaries?

Connection to automatability. Automatability of a proof system P should yield a close-to-optimal P-adversary because the optimal strategy for the adversary proceeds by deciding in each step if a given query has a short derivation from formulas that have been already assigned 1. Is it possible to use this connection to shed new light on the question of automatability of standard proof systems?

2 From efficient adversaries to algorithms

In this section we formally introduce efficient adversaries and prove Theorem 1.

Standard proof systems. We define adversaries for standard propositional proof systems from the set SYS consisting of:

- Extended Frege EF, Frege, AC^0_d-Frege and Circuit Frege CF systems;
- Resolution Res; Cutting Planes CP; k-DNF Resolution Res(k);
- tree-like versions of the systems listed above.[6]

The definitions of these systems can be found in [34], but we will use the definition of Res(k) from [64, Definition 2.6] and the definition of CF from [25]. Each instantiation of EF, Frege, AC^0_d-Frege and CF system is represented in SYS separately. This is because we do not know whether the existence of an efficient adversary for a system P guarantees the existence of an efficient adversary for a system P' if P' is p-equivalent to P. We consider Frege, EF and CF systems for all finite and complete languages for propositional logic. SYS^- will stand for SYS without EF systems.

We denote by $O(P)$ the set of "objects" a proof system $P \in \mathsf{SYS}$ operates with. EF and Frege systems operate with formulas built from the connectives in the respective language. AC^0_d-Frege operates with formulas of depth d defined as in [34, Chapter 15], CF systems operate with circuits in the respective language, Res operates with disjunctions of literals (clauses), CP operates with integer inequalities, Res(k) with k-DNFs.

In order to unify the presentation, we treat all systems $P \in \mathsf{SYS}$ as refutational systems, refer to the objects of $O(P)$ as formulas (in the case of CF this is an abuse of terminology but it should not cause any confusion) and denote by 0 the last formula of a refutation.

Pudlák-Buss game. Our definition of efficient adversaries is inspired by the Pudlák-Buss game [53], which can be used to characterize short Frege proofs. The game is played between Prover and Liar. In each round Prover sends a formula to Liar who responds by 0 or 1. Denote the response of Liar on formula ψ by $C(\psi)$, not displaying the history of the game for simplicity. Intuitively, we interpret $C(\psi) = 0/1$ as claims that Liar possesses a falsifying/satisfying assignment of ψ. In each round, Liar has to satisfy $C(0) = 0$ and $C(\phi) = 1$, if we want to estimate the size of the refutation of a formula ϕ.

Given an s-size Frege-refutation of ϕ, there is a strategy of Prover "catching" Liar in $K \log s$ rounds by querying formulas of size $\leq Ks^K$, where $K \in \mathbb{N}$ depends only on the Frege system: Using binary search Prover can find in $K \log s$ rounds formulas $\phi_1, \ldots, \phi_{k+1}$ such that $C(\phi_i) = 1$, for $i \in [k]$, $C(\phi_{k+1}) = 0$ and ϕ_{k+1} follows from $\phi_1, \ldots, \phi_k$ by a single application of a Frege-rule.

In the opposite direction, such a strategy of Prover that catches Liar in r rounds with queries of size $\leq 2^{O(r)}$ yields a $2^{O(r)}$-size tree-like Frege-refutation of ϕ, if the logical connectives in the language of the system include $0, 1, \neg, \vee, \wedge$, cf. [34, Lemma 2.2.3].

▶ **Definition 8** (Efficient adversary). *For $P \in \mathsf{SYS}^-$, an s-size P-adversary is a sequence $C := \{C_i\}_{i=1}^{\infty}$, where C_i computes the responses of the adversary in the i-th round as follows. Each C_i is a sequence of s-size circuits $C_i := \{C_{i,j}\}$ which for each $\langle a_1, \ldots, a_i \rangle \in \mathbb{N}^i$ contains a single circuit $C_{i,j}$ with i blocks of inputs partitioned so that the k-th block, $k \leq i-1$, consists of $a_k + 1$ inputs and the i-th block consists of a_i inputs. Given a history of the adversary's responses $h := \{\langle \phi_k, b_k \rangle\}_{k=1}^{i-1}$, where $\phi_k \in O(P)$ and $b_k \in \{0, 1\}$ is a response of the adversary to ϕ_k in the k-th round, the response of the adversary to $\phi_i \in O(P)$ in the i-th round is $C_i(h, \phi_i) := C_{i,j}(h, \phi_i)$, where $C_{i,j}$ is the circuit in C_i with inputs partitioned in accord with*

[6] Strictly speaking, it does not make much sense to consider tree-like EF as we want to allow the system to reuse extension axioms.

h, ϕ_i. The P-adversary is honest *if it responds consistently with its responses in previous rounds, i.e. for each $\phi \in O(P)$ and each history $h = \{\langle \phi_k, b_k \rangle\}_{k=1}^{i-1}$, $C_i(h, \phi) = b_\ell$ if $\phi = \phi_\ell$ for some $\langle \phi_\ell, b_\ell \rangle \in h$ with $\ell < i$.*

If the adversary responds to $\phi \in O(P)$ with a fixed $b \in \{0, 1\}$ in each round independently of the history, we write $C(\phi) = b$. If the P-adversary satisfies $C(0) = 0$ and $C(\phi) = 1$, for all $\phi \in \Phi \subseteq O(P)$, we say that the adversary is for Φ*. We will consider only finite sets Φ.*

A randomized s-size P-adversary *(*for Φ*) is an s-size P-adversary $\{C_i\}_{i=1}^{\infty}$ (*for Φ*) such that for each i, each $C_{i,j} \in C_i$ has access to $\leq s$ bits of a finite random string of bits shared among all circuits $C_{i,j}$, for all i, j. The random strings are distributed according to some arbitrary distribution.*

The definition of EF-adversaries requires a small twist which will become apparent after Definition 9. For $P \in \mathsf{SYS}^-$, a P-prover is a function mapping histories $\{\langle \phi_k, b_k \rangle\}_{k=1}^{i}$, where $\phi_k \in O(P)$, $b_k \in \{0, 1\}$, $i \in \mathbb{N}$, to new formulas (queries) in $O(P)$.

Switching quantifiers and the role of randomness. If for a given prover with $\leq s$-size queries, there is an adversary who stays consistent for r rounds, then there is a $poly(sr)$-size adversary who achieves that as well. To make the notion of efficient adversaries interesting we will thus switch quantifiers and consider s-size adversaries staying consistent for r rounds against all possible provers. A randomized adversary will be slightly more flexible: for each prover, the randomized adversary will choose a random string and then proceed as the deterministic adversary specified by the random string. The definition ensures that circuits in C_2 responding in the second round can access the random bits used by circuits C_1 responding in the first round.

Next we introduce a notion that allows us to generalize the characterization of proof-size from Pudlák-Buss games to all systems in SYS^-.

▶ **Definition 9** (Evaluation of adversary). *Given a refutation π of $\Phi \subseteq O(P)$ in a proof system $P \in \mathsf{SYS}^-$ and an (s-size) P-adversary C for some $\Phi' \subseteq O(P)$, define the* evaluation of C on π *by the following process. In round 0, let $h_0 := \emptyset$ and start at the last formula of π, i.e. 0. If in the i-th round, the current formula $\phi \notin \Phi$ is derived in π by applying a single derivation rule on (a possibly empty set of) formulas $\phi_j, j \in [k]$, proceed to the first $\phi_{j'}, j' \in [k]$, such that $C_{i+1}(h_i, \phi_{j'}) = 0$ and let $h_{i+1} := h_i \cup \langle \phi_{j'}, 0 \rangle$. If there is no such $\phi_{j'}, j' \in [k]$, output 0. If $\phi \in \Phi$ is reached, output 1. If π evaluated C to 0, we say that π* caught *C. Otherwise, C* stayed consistent on *π.*

Obviously, there is an s-size refutation of Φ in a proof system $P \in \mathsf{SYS}^-$ if and only if there is an s-size P-refutation of Φ which catches all P-adversaries for Φ.

EF adversaries. Definitions 8-9 can be straightforwardly adapted to the case of EF systems, but the extension axioms need to be treated carefully: It would be unfair to ask an EF adversary to assign a value to an extension variable e and only subsequently reveal that e is equivalent to 1 or 0 – the adversary would be caught immediately. The issue does not appear in the case of the p-equivalent CF systems. This also suggests the right way to define EF adversaries: We adapt Definitions 8-9 so that whenever an EF adversary evaluated on an EF refutation π receives a formula ϕ, they receive also all extension axioms in π defining the extension variables appearing in ϕ and the extension axioms of all the extension variables in the received extension axioms. In other words, the adversary gets to know the whole circuit defining the extension variables in ϕ.

To justify the definition, following the proof of the simulation of EF by CF [25, Lemma 2.4], it can be shown that for each EF system P and each CF system R such that P and R have the same language and Frege rules, if each $poly(s)$-size P-adversary is caught by some t-size P-refutation, then each s-size R-adversary is caught by some $poly(t)$-size R-refutation.

Proposition 10 points out that for Frege systems, an honest adversary for Φ stays consistent against all provers with $poly(n)$-size queries for $O(\log n)$ rounds if and only if it stays consistent on all $poly(n)$-size $O(\log n)$-height refutations (not necessarily of Φ). (Note that an s-size Frege proof can be transformed to a $poly(s)$-size $O(\log s)$-height Frege proof of the same formula [34].) A height of a Frege derivation is the maximal number of edges on a directed path in the derivation graph.

▶ **Proposition 10.** *Let P be a Frege system containing modus ponens rule (which derives ψ from ϕ and $\neg\phi \vee \psi$).*

1. *If an s-size r-height P-refutation (not necessarily of Φ) catches a P-adversary for Φ, then some P-prover with s-size queries catches the adversary in $O(r)$ rounds.*
2. *If a P-prover with s-size queries catches an honest P-adversary for Φ in r rounds, then some $poly(s, r)$-size $(r + 2)$-height P-refutation (not necessarily of Φ) catches the adversary.*

Proof. To prove Item 1, we define a strategy of the prover by querying formulas on the path in the P-refutation going from 0 to an axiom where the adversary assigns 0s.

Item 2 is obtained by treating each query as a resolvent ϕ in the modus ponens rule deriving ψ from ϕ and $\neg\phi \vee \psi$. The formula ψ is initially 0. If ϕ is assigned 0, we proceed with extending the refutation we have built so far by building a derivation of ϕ and leaving $\neg\phi \vee \psi$ as an additional axiom. Otherwise, we proceed with building a derivation of $\neg\phi \vee \psi$ and leaving ϕ is an axiom. The evaluation of the adversary on the resulting refutation π ends up in one of the following two cases. In Case 1, the last queried formula ϕ_r is assigned 0 but ϕ_r follows by a single application of a Frege rule from formulas assigned 1. In Case 2, the last queried formula is assigned 1 and again a "direct contradiction" is found in which some formula ϕ_r is assigned 0. In Case 1, we extend π by a single derivation step deriving ϕ_r from the formulas assigned 1. In Case 2, we extend π by a modus ponens rule resolving on ϕ_r and then adding the derivation rule deriving ϕ_r from the formulas assigned 1. Since the adversary has to stay consistent with its previous responses, in both cases, we obtain an $\leq (r + 2)$-height P-refutation of size $poly(s, r)$ which catches them. ◀

Proposition 10 holds also for AC^0_d-Frege systems with a minor adjustment of parameters. We do not know how to improve the conclusion in Proposition 10 (Item 2) into a P-refutation of Φ catching the adversary, but we will show that a typical proof complexity lower bound for a formula Φ yields a randomized efficient (possibly dishonest) adversary for Φ who stays consistent on all small refutations (not necessarily of Φ) with probability $\geq 9/10$ over the randomness of the adversary, if the adversary is not required to assign 1s to formulas outside Φ. This should be contrasted with the next theorem showing that under certain assumptions it is not possible to construct such EF-adversaries for formulas encoding circuit lower bounds for SAT.

In order to state the theorem we need to introduce some notation. Let $\neg\mathsf{lb}_A(\mathsf{SAT}, s)[B]$ be a propositional formula encoding the $poly(s, |A|)$-size predicate $R_1(B) :=$ "variables B encode an s-size circuit with n inputs computing SAT on inputs from $A \subseteq \{0, 1\}^n$". The formula can use other hidden auxiliary variables as well and is satisfiable iff $\exists B, R_1(B) = 1$. The values of SAT on A are "hardwired" bits. Sometimes we write $\mathsf{lb}_A(\mathsf{SAT}, s)$ without displaying $[B]$. We

define $\mathsf{tt}(\mathsf{SAT}, s)$ as $\mathsf{lb}_{\{0,1\}^n}(\mathsf{SAT}, s)$. The notation SAT should not be confused with $\mathsf{SAT}(x)$ which we use as a shortcut for saying that "x is a satisfiable formula" and with $\mathsf{SAT}(x, y)$ used to denote a propositional formula encoding the predicate $R_2(x, y) :=$ "x encodes a propositional formula satisfied by assignment y". For concreteness, we assume that $\mathsf{SAT}(x, y)$ uses extension variables encoding the gates of a $poly(|x|, |y|)$-size circuit computing $R_2(x, y)$ and is a tautology for an assignment a, b of x, y iff $R_2(a, b) = 1$. Similarly, given a circuit F with inputs y, the predicate $R_3(x, y) :=$ "x is a propositional formula satisfied by $F(y)$" can be encoded by a propositional formula $\mathsf{SAT}(x, F(y))$ using extension variables for the gates of a $poly(|F|, |x|)$-size circuit computing $R_3(x, y)$ and is a tautology for an assignment a, b of x, y iff $R_3(a, b) = 1$. We do not specify the encoding of these formulas further as the results in this section work for any natural encoding.

▶ **Theorem 11.**
For each EF *system* P *there is* $K \in \mathbb{N}$ *such that the following holds.*

Let $v : \mathbb{N} \mapsto \mathbb{N}$, $n, s, t, u, w \in \mathbb{N}$ *be such that* $n < s \leq t \leq u$ *and assume 1 - 2:*

1. (The canonical pair of P is hard.) *For each* $(n \cdot v(t^K))^K$*-size circuit* D *with* n *inputs, there is* z *such that* $\mathsf{SAT}(z) \wedge D(z) = 0$ *or* $D(z) = 1$ *while there is a* w*-size* P*-refutation of (the formula encoded by)* z*.*
2. (Provably constructive anticheckers.) *There is a* t*-size set* $A \subseteq \{0,1\}^n$ *and a* t*-size circuit* F *such that for each* $z \in \{0,1\}^n$ *there is a* u*-size* P*-proof of*

$$\neg\mathsf{lb}_A(\mathsf{SAT}, s)[B] \wedge \mathsf{SAT}(z, y) \to \mathsf{SAT}(z, F(z, B)).$$

Then,

a) *For each (possibly dishonest) randomized* v*-size* P*-adversary* $C = \{C_i\}_{i=1}^{\infty}$ *for* $\neg\mathsf{lb}_A(\mathsf{SAT}, s)$*, there is a* $(w + u)^K$*-size* P*-refutation catching* C *with probability* $\geq 1/2 - 1/n$ *over the randomness of* C*.*

b) *For each randomized* v*-size* P*-adversary* C *for* $\neg\mathsf{tt}(\mathsf{SAT}, s)$*, there is a* $(2^n + w + u)^K$*-size* P*-refutation catching* C *with probability* $\geq 1/2 - 1/n$ *over the randomness of* C*.*

Moreover, Item b) holds even without the assumption in Item 2, if $s \leq 2^n$*.*

Typical parameters of interest in Theorem 11 are $s, t, u = poly(n)$ and v being a polynomial function in n and the size of the input, where the size of the input might be $\ll n$ or $\gg n$. In Item a), we typically consider $w = poly(n)$ while in Item b) it is best to set $w = 2^{O(n)}$.

▶ Remark 12. For $w = 2^{O(n)}$, which is the setting suitable for the "moreover" part of Theorem 11, Item 1 in the assumptions can be replaced by $\mathsf{SAT} \notin \mathsf{Circuit}[(n \cdot v(t^K))^K]$ because each unsatisfiable z has a $2^{O(|z|)}$-size EF-refutation.

Proof of Theorem 11.

Item a). Assume that the modus ponens rule is available in P and let D^r be the n-input $v(poly(t))$-size randomized circuit

$$D^r(x) := C_{1,j}(\mathsf{SAT}(x, F(x, B))),$$

where $C_{1,j}$ is the randomized circuit in C_1 with the appropriate input size, F is the circuit from the second assumption and B are the variables from the displayed formula in the second assumption. In more detail, D^r maps $x \in \{0,1\}^n$ to $\mathsf{SAT}(x, F(x, B))$ and sends the formula to $C_{1,j}$ together with the extension axioms of the extension variables in $\mathsf{SAT}(x, F(x, B))$. Intuitively, D^r attempts to use C to solve SAT. However, there is an input z of D^r on which D^r fails to separate the canonical pair of P (for the given w) with probability

$\geq 1/2 - 1/n$. Otherwise, by an amplification employing Chernoff's bound and the subsequent derandomization, a circuit of size $poly(n) \cdot |D^r| + poly(n)$ would separate the canonical pair of P, contradicting Item 1.

The last step of the refutation catching C derives 0 from $\mathsf{SAT}(z, F(z, B))$ and its negation (using modus ponens).[7] By the choice of z, with probability $\geq 1/2 - 1/n$ over the randomness of C, one of the following two cases happens.

Case I: $D^r(z) = 1$ and there is a w-size P-refutation of z. A w-size P-refutation of z yields a $poly(w + t)$-size P-derivation π_0 of $\neg\mathsf{SAT}(z, F(z, B))$ from empty assumptions, cf. [34, Chapter 12.4]. Since extension variables cannot appear in the initial formulas of a P-refutation containing π_0 (they would not be allowed to be introduced in π_0), we also need to use a $poly(t)$-size P-derivation π_2 of $\mathsf{SAT}(z, F(z, B))$ from 0. The refutation catching C is then constructed by composing π_0, π_2 with the application of modus ponens on $\mathsf{SAT}(z, F(z, B))$ and its negation. It is a refutation of 0.

Case II: $D^r(z) = 0 \wedge \mathsf{SAT}(z)$. Since $\mathsf{SAT}(z)$ and we assume Item 2, there is a $poly(u)$-size P-derivation π_1 of $\mathsf{SAT}(z, F(z, B))$ from $\neg\mathsf{lb}_A(\mathsf{SAT}, s)$. Again, in order to deal with the extension axioms, we construct a $poly(t)$-size P-derivation π_2' of $\neg\mathsf{SAT}(z, F(z, B))$ from 0 and then the P-refutation catching C analogously as in Case I. It is a refutation of $\neg\mathsf{lb}_A(\mathsf{SAT}, s)$ and 0.

Since only one of the cases I and II can happen, this yields the desired refutation.

If modus ponens is not available in P, we proceed similarly. Given $x \in \{0,1\}^n$, D^r constructs a $poly(t)$-size P-derivation π_x of 0 from $\mathsf{SAT}(x, F(x, B))$ and its negation (expressed in the language of the system). Then, D^r simulates the evaluation of C on π_x (it suffices to store finitely many $v(poly(t))$-size circuits from C to simulate the evaluation for all $x \in \{0,1\}^n$). If the evaluation ends up in the formula $\mathsf{SAT}(x, F(x, B))$, D^r outputs 0. If the evaluation ends up in the formula $\neg\mathsf{SAT}(x, F(x, B))$, D^r outputs 1. Otherwise, D^r outputs an arbitrary bit, say, 0. The size of $D^r(x)$ is $poly(v(poly(t)))$. The existence of $z \in \{0,1\}^n$ such that Case I or Case II happens follows as before. The refutation catching C is constructed as before but the final application of modus ponens, i.e. the last step of the refutation, is replaced by π_z. The refutation catches C because in Case I, the evaluation of C on π_z cannot end up in $\mathsf{SAT}(z, F(z, B))$ – in such case D^r would output 0 on z – and in Case II the evaluation on π_z cannot end up in $\neg\mathsf{SAT}(z, F(z, B))$ – in such case D^r would output 1 on z. This completes the proof of Item a).

Item b). We proceed as in Item a) but instead of the assumption from Item 2 we use a formalization of the search-to-decision reduction for SAT. In more detail, let F be a $poly(s)$-size circuit which solves the search version of SAT given an s-size circuit B witnessing that $\neg\mathsf{tt}(\mathsf{SAT}, s)$. As before, the last step of the refutation resolves on $\mathsf{SAT}(z, F(z, B))$, for a suitable z (or this step is simulated). The only difference from Item a) is that in Case II we use a $2^{O(n)}$-size P-derivation of $\mathsf{SAT}(z, F(z, B))$ from $\neg\mathsf{tt}(\mathsf{SAT}, s)$. The existence of the P-derivation follows from the S^1_2-formalization of the search-to-decision reduction for SAT, see [46, Proposition 3.1] or [17], and the fact that Π^b_1-consequences of S^1_2 have p-size proofs in EF [29]. ◀

While the assumption in Item 1 in Theorem 11 is standard (roughly corresponding to the non-automatability of P), Item 2 deserves a discussion.

[7] If we formulated the proof in terms of prover-adversary games, $\mathsf{SAT}(z, F(z, B))$ would be the first formula sent by the prover.

Plausibility of provably constructive anticheckers. A theorem of Lipton and Young [41] about "anti-checkers" guarantees that if SAT is hard for circuits of size $poly(s)$, there is a set A of size $poly(s)$ such that $\mathsf{lb}_A(\mathsf{SAT}, s)$ is a tautology. In fact, a simple case analysis (considering the cases that $\mathsf{SAT} \in \mathsf{Circuit}[poly(s)]$ and $\mathsf{SAT} \notin \mathsf{Circuit}[poly(s)]$) yields a circuit F with $t = poly(s)$ such that the displayed formulas in Item 2 are tautologies, but we do not have the guarantee that the tautologies are efficiently provable. The existence of such F with the efficiently provable property from Item 2 follows, if we assume that EF proves efficiently that a concrete Boolean function is hard on average and that EF proves efficiently that a concrete efficient circuit given a circuit breaking a one-way permutation (OWP) produces a circuit for SAT, see [50, Lemma 2]. In other words, if EF can prove a circuit lower bound and reduce OWPs to $\mathsf{SAT} \notin \mathsf{P/poly}$, Item 2 holds. In fact, it holds with z being a free variable in the displayed formula.

Item 2 is similar to the assumption used in [50] to establish the implication that lower bounds on lengths of proofs in concrete propositional proof systems imply explicit circuit lower bounds. Indeed, as pointed out to us by Noel Arteche, Item 2 (with $s, t, u = poly(n)$) implies that $\mathsf{NP} \not\subseteq \mathsf{Circuit}[n^\ell]$, for each $\ell \in \mathbb{N}$ – $\mathsf{SAT} \in \mathsf{P/poly}$ combined with Item 2 implies that EF is p-bounded and $\mathsf{NP} = \mathsf{coNP}$ implies the desired lower bound using Kannan's lower bound. It is possible that Item 2 implies that EF proves efficiently $\mathsf{tt}(f, n^{O(1)})$ for some f, but it is not immediate: The outlined proof of $\mathsf{NP} \not\subseteq \mathsf{Circuit}[n^\ell]$ involves a case analysis yielding $\mathsf{tt}(\mathsf{SAT}_{n^k}, n^\ell)$ or $\mathsf{tt}(f', n^k)$ where f' has inputs of length n. Such disjunctions have, however, trivial proofs of polynomial size in the size of the formula (i.e. in $2^{O(n^k)}$). Nevertheless, Item 2 implies $poly(u)$-size EF-proofs of $\mathsf{lb}_A(\mathsf{SAT}, s) \vee \mathsf{tt}(h, s')$, for each $h \notin \mathsf{Circuit}[s']$, where the truth-table of h has length $< n$. Further, by truncating the inputs from A we get $poly(u)$-size EF-proofs of $\mathsf{tt}(h', s) \vee \mathsf{tt}(h, s')$, for some h' with the truth-table of size $\sim t$.[8]

Importantly, our theorem holds for stronger systems than EF as well – specifically, for systems EF_P from Section 1.3 – in case these are needed to satisfy Item 2. One way to interpret Theorem 11 is thus to say that if we can construct efficient adversaries for lower bounds in a sufficiently strong proof system which can prove constructively the existence of anti-checkers (e.g. by proving some circuit lower bound and reducing OWPs to $\mathsf{SAT} \notin \mathsf{P/poly}$), then we can separate the canonical pair of the system efficiently.

▶ Remark 13. Theorem 11 remains valid if the formula $\neg\mathsf{lb}_A(\mathsf{SAT}, s)$ in its conclusion is replaced by an arbitrary formula ϕ such that P has a $poly(u)$-size derivation of $\neg\mathsf{lb}_A(\mathsf{SAT}, s)|_\rho$ from ϕ, where $\psi|_\rho$ is a formula obtained from ψ by applying a substitution ρ which replaces some (possibly all) variables of ψ by $poly(s)$-size formulas built on variables of ϕ. In this case, the P-refutation is constructed as in the proof of Theorem 11 but applying ρ on formulas $\mathsf{lb}_A(\mathsf{SAT}, s)$, $\mathsf{SAT}(x, F(x, B))$ and using the fact that the assumption from Item 2 remains valid under ρ. Moreover, in Case II, we use the short P-derivation of $\neg\mathsf{lb}_A(\mathsf{SAT}, s)|_\rho$ from ϕ. Similarly, $\neg\mathsf{tt}(\mathsf{SAT}, s)$ in Theorem 11 can be replaced by an arbitrary formula ϕ such that P has a $2^{O(n)}$-size derivation of $\neg\mathsf{tt}(\mathsf{SAT}, s)|_\rho$ from ϕ, where ρ substitutes some variables by $poly(s)$-size (not $2^{O(n)}$-size) formulas built on variables of ϕ.

As noted by Moritz Müller, Theorem 11 holds also if we substitute $\neg\mathsf{lb}_A(\mathsf{SAT}, s)$ both in the conclusion and in the assumption by an arbitrary formula.

---

[8] If we assume the existence of efficient EF-proofs of $\mathsf{tt}(h', s) \vee \mathsf{tt}(h, s')$, for some h', h and suitable t, s, s', then a version of EF-natural proofs from [48] (defined as the existence of efficient EF-proofs of the formulas encoding that "EF does not prove efficiently $\mathsf{tt}(h', s) \vee \mathsf{tt}(f, s'/3)$", for many functions f) breaks superbits. In order to see this, note that short EF-proofs of $\mathsf{tt}(h', s) \vee \mathsf{tt}(h, s')$ yield short EF-proofs of $\mathsf{tt}(h', s) \vee \mathsf{tt}(h \oplus f, s'/3)$, for every $f \in \mathsf{Circuit}[s'/3]$.

3 Efficient adversaries for known lower bounds

3.1 Width and expansion

The strongest known lower bound on the length of proofs of tt-tautologies is due to Razborov [64, Theorem 2.13], who showed that (a concrete natural encoding of) $\mathsf{tt}(f_n,t)$, $n^2 \leq t \leq 2^n$, and in fact also $\mathsf{lb}_W(f_n,t)$, for any $W \subseteq \{0,1\}^n$ and any f_n, requires $\mathsf{Res}(k)$-proofs of size $2^{t^{\Omega(1)}}$, if $k = \epsilon \log t$ and $\epsilon > 0$ is a sufficiently small constant. The lower bound combines a small-restriction switching lemma, which turns short proofs into narrow ones, with a width lower bound which is obtained using expansion properties of the lb-formula (similarly as in the influential work of Ben-Sasson and Wigderson [7]). We show how to adapt the proof so that it yields efficient adversaries for formulas $\tau_W(A,b)$ described below. On the way, we will refer to the details of the proof (and definitions) from [64].

The reason why we focus on formulas τ instead of lb is that Razborov's lower bound for $\mathsf{lb}_W(f_n,t)$ is obtained by showing that every S-size $\mathsf{Res}(k)$-refutation of $\neg\mathsf{lb}_W(f_n,t)$ can be turned into an $S2^{O(k)}$-size $\mathsf{Res}(2k)$-refutation of $\tau_W(A,b)$ and then proving that $\tau_W(A,b)$ does not have such refutations. In fact, as we will observe, $\mathsf{Res}(2)$ admits $poly(|W|,t)$-size derivations of each clause of $\neg\mathsf{lb}_W(f_n,t)|_\rho$ from $\tau_W(A,b)$, where ρ substitutes some variables of $\neg\mathsf{lb}_W(f_n,t)$ by a function of at most two variables of $\tau_W(A,b)$. On the other hand, by Remark 13, Theorem 11 holds even if $\neg\mathsf{lb}_W(\mathsf{SAT},s)$ is replaced by an arbitrary formula ϕ such that EF has a $poly(s)$-size derivation of $\neg\mathsf{lb}_W(\mathsf{SAT},s)|_\rho$ from ϕ.[9] The focus on τ-formulas allows us to avoid the problem of encoding the transformation of refutations of $\neg$lb to refutations of τ in the strategy of the adversary.

In the following paragraphs defining formulas $\tau_W(A,b)$ we refer mainly to [64, Sections 6 and 8], where the reduction of lower bounds for $\neg\mathsf{lb}_W(f_n,t)|_\rho$ to lower bounds for $\tau_W(A,b)$ is described for the case $W := \{0,1\}^n$. We start with specifying formulas

$$\tau_{\leq}(A,b) =: \tau_{\{0,1\}^n}(A,b)$$

from [64], for A,b such that the desired short derivation of $\neg\mathsf{tt}(f_n,t)|_\rho$ from $\tau_{\leq}(A,b)$ exists. Formulas $\tau_{\leq}(A,b)$ are defined generally in [64, Definition 2.2] but we will consider only their particular instantiations obtained from $\tau(C^{n+1}_{A,\leq},b)$ in [64, Section 8] with two caveats:

1. We adopt the convention from the beginning of [64, Section 5] introducing axioms $\neg y^i_\emptyset$ and $y^i_{\{j\}} \vee \neg x_j, \neg y^i_{\{j\}} \vee x_j$ (instead of identifying x_j with $y^i_{\{j\}}$);
2. A,b in our $\tau_{\leq}(A,b)$ is not the same as A,b in $\tau(C^{n+1}_{A,\leq},b)$ from [64, Section 8]. In order to specify our A,b in $\tau_{\leq}(A,b)$, we need to express $\tau(C^{n+1}_{A,\leq},b)$ from [64, Section 8] as in the proof of [64, Theorem 2.12] which in turn refers to the proof of [64, Theorem 2.10], where $\tau(C^{n+1}_{A,\leq},b)$ is interpreted as a linear system $\hat{A}X = \hat{b}$ with constant right-hand side – this $\hat{A},\hat{b}$ is our A,b in $\tau_{\leq}(A,b)$ and our formula $\tau_{\leq}(A,b)$ is the formula $\tau_{\leq}(\hat{A},\hat{b})$ in the proof of [64, Theorem 2.10] obtained from $\tau(C^{n+1}_{A,\leq},b)$ considered in [64, Section 8].

The formulas $\neg\tau_{\leq}(A,b)$ express that a bit-string b of the length $|b| = 2t_0(2^{n+1}-1)$, $t_0 = \Omega((t/n)^{1/2})$, is not in the range of an NW-generator based on a $|b| \times t_0(2^{n+1}-1)$ matrix A and the parity function. In other words, $\tau_{\leq}(A,b)$ is a set of clauses asserting that $A\overline{x} = b$ is a system of linear equations solved by a $(|b|/2)$-bit vector $\overline{x}$. The matrix

[9] If $\neg\phi$ is to be hard for a strong proof system, it should be different from $\neg\tau_W(A,b)$, because systems simulating Gaussian elimination can prove $\neg\tau_W(A,b)$ efficiently.

A is an $(r, d+1)$-lossless expander, with $r = t^\delta$ for an arbitrary sufficiently small $\delta > 0$ and a sufficiently big $d = O(1)$, corresponding to an iterated composition of a function $H : \{0,1\}^{t_0} \mapsto \{0,1\}^{2t_0}$ (iteratively applied on the first t_0 and the last t_0 of its output bits) as described in [64, Section 6] (H denotes a different object in [64]). Additionally, A (is constructed from an (r,d)-lossless expander in [64, Section 8] and so it) satisfies $|J_i(A)| \geq t^\delta$ for all $i \in [|b|]$.[10] The iterated composition of functions H forms a tree T_H of $2^{n+1}-1$ copies of H. The indices of b are in one-to-one correspondence with outputs of the copies of H in T_H. The string b has 0s on all positions corresponding to outputs of the "inner" copies of H. In order to get a short derivation of $\neg\mathsf{tt}(f_n,t)|_\rho$ from $\tau_{\leq}(A,b)$ one defines $b_{a,j} := f_n(a)$, for $a \in \{0,1\}^n, j \in [2t_0]$, where $b_{a,j}$ corresponds to the jth output of the ath "outer" copy of H, i.e. the copy of H forming the ath leaf in T_H. As explained in [64, Section 8], there is a t-size circuit $D_{n,x_1,\ldots,x_{t_0}}(a)$ computing $b_{a,1} = f_n(a)$ on $a \in \{0,1\}^n$. The circuit $D_{n,x_1,\ldots,x_{t_0}}(a)$ gradually computes all copies of H in T_H that lead to $b_{a,1}$. In particular, the ath outer copy of H is computed by $D_{n,x_1,\ldots,x_{t_0}}(a)$ as well.

Given $W \subseteq \{0,1\}^n$, the formula $\tau_W(A,b)$ is obtained from restricting $\tau_{\leq}(A,b)$ to (the axioms encoding) the equations from $A\overline{x} = b$ which are used by the computation of $D_{n,x_1,\ldots,x_0}(a)$, for $a \in W$. Consequently, the size of $\tau_W(A,b)$ is $poly(t, |W|)$.

By inspecting the reduction in [64, Section 8], it can be verified that each clause of formula $\neg\mathsf{lb}_W(f_n,t)|_\rho$ has a $\mathsf{Res}(2)$ derivation of size $O(1)$ from $\tau_W(A,b)$, where ρ substitutes some variables of $\neg\mathsf{lb}_W(f_n,t)$ by a function of at most two variables of $\tau_W(A,b)$ and b is defined from f_n as above.

In the rest of this section we give a proof of the following theorem in which we construct a $poly(t^{t^\delta}, m')$-size (randomized circuits defining the responses of an) adversary for $\tau_W(A,b)$.

▶ **Theorem 14** (Efficient adversary for Razborov's $\mathsf{Res}(k)$ lower bound)**.** *Let $\gamma < 1$, $a \in \mathbb{N}$, let $\delta > 0$ be sufficiently small, $d \in \mathbb{N}$ be sufficiently big and $\epsilon > 0$. There is n_0 such that for all $n \geq n_0$ the following holds. Suppose A, b, t are as above (i.e. $n^2 \leq t \leq 2^n$, f_n specifying b is arbitrary, A is an $(r, d+1)$-lossless expander with $r = t^\delta$ and dimensions $|b| \times t_0(2^{n+1}-1)$), $k = \epsilon \log t$ and $W \subseteq \{0,1\}^n$ has size $|W| \leq t^a$. There is a $poly(t^{rad}, m')$-size randomized $\mathsf{Res}(k)$-adversary C for $\tau_W(A,b)$ such that for each $\frac{1-\gamma}{8}2^{r/(2^{O(k(d+1))})}$-size $\mathsf{Res}(k)$-refutation π of any set of clauses Φ, with probability $\geq \gamma$ over the randomness of C, π does not catch C. Here, m' refers to the input length of circuits from C.*

Let $A^-\overline{x} = b^-$ be the system of $m := |b^-|$ equations encoded by $\tau_W(A,b)$. That is, A^- results from A by erasing some rows and b^- is the corresponding restriction of b. Since $m \leq |\tau_W(A,b)| = poly(t, |W|)$ and $|W| \leq t^a$, such A^- satisfies

$$|J_i(A^-)| = t^\delta \geq C'_\gamma(d+1)(k + \log m), \text{ for all } i \in [m], \tag{1}$$

for arbitrary constant C'_γ, if n is sufficiently big.

On high level, the strategy of our adversary is to transform π using a transformation R and a random restriction ρ into a new $\mathsf{Res}(O(k))$-refutation which is practically equivalent to π restricted by ρ, but which contains only DNFs with small height and thus can be simulated by a Res-refutation of a small width. The small height/width will allow the adversary to find a partial solution of $A^-\overline{x} = b^-$ and respond to any formula from the original proof π so that each axiom of $\tau_W(A,b)$ is assigned 1 and the consistency w.r.t. $\mathsf{Res}(k)$-derivation rules is maintained (implying that π will not catch the adversary). We proceed with a more formal description.

[10] The existence of such A follows from the proof of [64, Theorem 2.5] and an application of Chernoff's bound, if $t \geq n_0$ for some n_0 depending on δ and d.

Moving to protected DNFs. Given a k-DNF F from π, the adversary first sets all variables in F which do not appear in $\tau_W(A,b)$ to 0 and applies the transformation R from the proof of [64, Claim 5.8] which produces a "protected" $O(k(d+1))$-DNF $R(F)$. However, our adversary adjusts the parameters in order to accommodate a more "sparse" ρ. Specifically, in [64, Definition 5.3] in Item 2, $3d$ is increased to $5d$ and in Item 3, $20kd$ is increased to $32kd$.[11] Further, in [64, Definition 5.4], $6d$ is increased to $10d$.[12] Consequently, $14kd$ in [64, Claim 5.5 a)] becomes $22kd$. In the proof of [64, Claim 5.7], $3d$ becomes $5d$ and $6kd$ becomes $10kd$. Recall also that d in [64, Definitions 5.3 & 5.4] is $d+1$ in our setting of parameters.

R (together with assigning 0's outside τ) transforms π into a set of $O(k(d+1))$-DNFs which can be completed into a $|\pi|2^{O(k(d+1))}$-size $\mathsf{Res}(O(k(d+1)))$-refutation of some Φ' such that each axiom of $\tau_W(A,b)$ gets converted to a DNF which logically follows from $\tau_W(A,b)$. The completion exists by the proof of [64, Claim 5.8] and by the proof of [64, Claim 5.7] (which guarantees that for each axiom F' in $\tau_W(A,b)$, we have $F' \Leftrightarrow R(F')$ assuming $\tau_W(A,b)$), but our adversary will construct only a part of it – we postpone the details. The definition of R exploits the property (1), which as we already explained can be assumed to hold for the relevant submatrix A^- of A.

The transformation R can be implemented by $poly(t, 2^{k(d+1)}, |W|, m')$-size circuits. This follows straightforwardly by inspection of its definition. We provide A^- as a non-uniform advice of the adversary. The occurrence of $|W|$ is enforced by the necessity of generating literals $(y^i_{J_i(A)})^{b_i}$ (referred to as "cosmetic") as the only way for our adversary to access bits b_i is to have the bits $b_{u,1}$ with $u \in W$ stored in a non-uniform advice.[13]

Applying a random restriction. Next, the adversary applies the random restriction ρ from [64, Definition 5.9], but the parameter $2d$ in [64, Definition 5.9] is increased to $4d$ – again, note that in our setting d in [64, Definition 5.9] becomes $d+1$. In more detail, if $R(F)$ contains a variable y^i_Σ, the adversary constructs $\tilde{Y}_i$ and a $poly(t_0)$-size part of J needed to determine Y_i (the part of J in $J_i(A)$). After determining Y_i, the adversary constructs the part of ρ needed to determine $R(F)|_\rho$ (using bits $b_{u,1}$ again to assign the cosmetic variable $y^i_{J_i(A)}$). Similarly, if $R(F)$ contains a variable x_j. The fact that circuits $C_{i,j}$ from our adversary share a random string is used to ensure that in future rounds our adversary will re-produce the same ρ which was used in previous rounds.

In the definition of sparsity, [64, Definition 5.1], $2d$ is increased to $4d$. Then, by [64, Claim 5.10], the probability that ρ is sparse is $\geq 1/2$, but the same proof shows that the probability is $\geq 1-(1-\gamma)/2$, if C'_γ is big enough. Here we use (1) again. To see that the proof of [64, Claim 5.10] goes through even with our adjusted parameters, it suffices to implement the following changes in it: $9d$ becomes $18d$, $s \geq C \log m/9$ becomes $s \geq C'_\gamma \log m/18$, $4d$ becomes $8d$ and $|\Delta^c_v| = d$ becomes $|\Delta^c_v| = 2d$.

As explained in the proof of [64, Claim 5.2], for any sparse ρ, $\tau_W(A,b)|_\rho$ can be transformed using a variable substitution ρ' into clauses encoding a linear system of equations $A^+\overline{x} = b^+$, where A^+ is an $(r, d+1)$-lossless expander with $|J_i(A^+)| \geq 4(d+1)$. (If we did not adjust the parameters above, we would get only $|J_i(A^+)| \geq 2(d+1)$, which would not suffice to prove Lemma 15 guarantying the existence of suitable partial solutions of $A^+\overline{x} = b^+$.)

[11] For clarification, we remark that $L_{i,1}, R_{i,1}$ in Item 1 of [64, Definition 5.3] denote L_{i1}, R_{i1} and $Conv(L_{i\nu}, R_{i\nu})$ in Item 3 is a subset of $J_i(A)$.

[12] Again, in order to avoid a potential ambiguity, we stress that intervals $(r_{i,\nu}, r_{i,\nu+1}]$ in [64, Definition 5.4] are subsets of $J_i(A)$.

[13] For this reason we would get at best only $poly(t^{t^\delta}, 2^n, m')$-size $\mathsf{Res}(k)$-adversaries for $\neg\mathsf{tt}(f_n, t) = \neg\mathsf{lb}_W(f_n, t)$, if $W = \{0,1\}^n$.

[64, Claim 5.14] remains valid for our adjusted parameters as well. To see that, it suffices to increase $6d$ to $10d$ in the proof of [64, Claim 5.12] and $2d$ to $4d$ in the proof of [64, Claim 5.13]. Therefore, by [64, Claim 5.14], ρ collapses the height of $R(F)$ with high probability,

$$\Pr_{\rho}[h(R(F)|_{\rho}) \geq h] \leq 1/e^{h/2^{O(k(d+1))}},$$

where $h := r/(C'_h k(d+1))$, for a big enough constant C'_h. Since $4|\pi| \leq (1-\gamma)2^{h/2^{O(k(d+1))}}/2$, with probability $\geq 1-(1-\gamma)/2$, every $F \in \pi \cup \pi'$ was transformed to a formula $R(F)|_{\rho}$ with height $\leq h$. Here, π' is a set of k-DNFs of size $|\pi'| \leq 3|\pi|$ which will be specified later. The adversary will produce formulas in π' from the formulas in π it receives and the transformation R will be defined on π'.

The last three paragraphs imply that with probability $\geq \gamma$, the applications of R (with assigning 0s outside τ) and ρ transform π into (a set of formulas that can be completed to) a $\mathsf{Res}(O(k(d+1)))$-refutation of some Φ'' such that each axiom of $\tau_W(A,b)|_{\rho}$ is converted to a DNF which follows from axioms which can be further interpreted (using ρ') as axioms encoding $A^+\overline{x} = b^+$, where A^+ is an $(r, d+1)$-lossless expander satisfying $|J_i(A^+)| \geq 4(d+1)$, for all $i \in [|b^+|]$. Additionally, $\forall F \in \pi \cup \pi'$, $h(R(F)|_{\rho}) \leq h$.

In the rest of the proof we define the responses of the adversary in the case that R and ρ ensured the properties above. If this does not happen and the adversary is led to respond to an axiom of $\tau_W(A,b)$ by 0, it recognizes that the transformation failed and (despite failing in this case) overwrites its response to 1 in order to ensure that it is *for* $\tau_W(A,b)$.

Finding partial solutions. Our adversary already produced $R(F)|_{\rho}$, for a given k-DNF F. To complete the proof of Theorem 14 it remains to explain how the adversary solves the system of equations corresponding to $R(F)|_{\rho}$ in a consistent fashion (so that it responds to each axiom of $\tau_W(A,b)$ by 1 and preserves the $\mathsf{Res}(k)$-derivation rules). For this purpose we adapt a proof of a width lower bound presented by Krajíček [34, Lemma 13.4.5]. Our proof differs from Krajíček's presentation in that instead of using $(r,\cdot)$-expansion we use $(r, d+1)$-lossless expansion.

For a set of rows $I \subseteq [|b^+|]$, let A^+_I be the matrix obtained from A^+ by deleting all rows in I and all columns in $\bigcup_{i\in I} J_i$, where $J_i := J_i(A^+)$.

▶ **Lemma 15.** *Let A^+ be an $(r, d+1)$-lossless expander as above satisfying $|J_i| \geq 4(d+1)$, for all $i \in [|b^+|]$. For any set of rows $I \subseteq [|b^+|]$ of size $|I| \leq r/2$, there is $\hat{I} \supseteq I$, $|\hat{I}| \leq 2|I|$ such that*

$$\text{For each } i \notin \hat{I}, \quad \left| J_i \setminus \bigcup_{u\in\hat{I}} J_u \right| \geq |J_i| - 2(d+1). \tag{2}$$

Moreover, for any $\hat{I}$ satisfying (2), for any nonempty set K of $\leq r$ rows in $A^+_{\hat{I}}$, $|\partial_{A^+_{\hat{I}}}(K)| > 0$.

Proof. Assume $|I| \leq r/2$. Put $I_0 := I$. Add in consecutive steps $t = 0, \ldots$ to I_t any one row i as long as

$$\left| J_i \cap \bigcup_{u\in I_t} J_u \right| > 2(d+1). \tag{3}$$

We claim that the process stops before t reaches $|I|$. Assume not, and let $I' := I_{|I|}$. Then by (3),

$$\sum_{i\in I'} |J_i| - \partial_{A^+}(I') > \sum_{i\in I'} |J_i| - \left(\sum_{i\in I} |J_i| + \sum_{i\in I'\setminus I} (|J_i| - 2(d+1))\right) = \sum_{i\in I'\setminus I} 2(d+1) = (d+1)|I'|.$$

This contradicts the expansion property of A^+.

Let $\hat{I}$ be the last I_t in the process, so $t < r/2$, $|\hat{I}| \le 2|I|$ and $\hat{I}$ has the property (2).

For the "moreover" part, note that

$$\partial_{A^+_{\hat{I}}}(K) = \partial_{A^+}(K) \setminus \bigcup_{i\in\hat{I}} J_i.$$

As for all $i \in K\setminus\hat{I}$ we have $|J_i \cap \bigcup_{u\in\hat{I}} J_u| \le 2(d+1)$, this implies that

$$|\partial_{A^+_{\hat{I}}}(K)| \ge |\partial_{A^+}(K)| - \sum_{i\in K} 2(d+1) \ge \sum_{i\in K} |J_i| - (d+1)|K| - 2(d+1)|K| > 0.$$

The last inequality uses $|J_i| \ge 4(d+1)$. ◀

Any $I \subseteq [|b^+|]$ satisfying (2) from Lemma 15 will be referred to as a *safe set of rows*. A partial assignment α to $\overline{x}$ assigning to 0 or 1 variables from a set $dom(\alpha)$ is *safe with support* I if $\{j \mid x_j \in dom(\alpha)\} \subseteq \bigcup_{i\in I} J_i$ for some safe I. A safe partial assignment α with support I is a *safe partial solution* of $A^+x = b^+$ *with support* I if and only if $\bigoplus_{j\in J_i} \alpha(x_j) = b^+_i$, for all $i \in I$.

▶ **Lemma 16.** *Assume $I \subseteq I' \subseteq [|b^+|]$ are safe sets and $|I'\setminus I| \le r$. Assume further that α is a safe assignment with support I, and that $c_i \in \{0,1\}$, for $i \in I'\setminus I$.*

Then the assignment α can be extended to a safe assignment α' with support I' such that

$$\bigoplus_{j\in J_i} \alpha'(x_j) = c_i,$$

for all $i \in I'\setminus I$.

Proof. By Lemma 15, each subset of $I'\setminus I$ has a non-empty border in A^+_I. In particular, no subset of $I'\setminus I$ can be a linearly dependent set of rows of A^+_I. Hence the system

$$\bigoplus_{j\in J_i\setminus dom(\alpha)} x_j = c_i \oplus \bigoplus_{j\in J_i\cap dom(\alpha)} \alpha(x_j), \text{ for } i \in I'\setminus I,$$

has a solution θ. Put $\alpha' := \alpha \cup \theta$. ◀

The last lemma we need is a result of Segerlind, Buss and Impagliazzo [68, Theorem 5.1] which translates Res(k)-derivations with lines of small height to Res-derivations of small width. Our formulation is slightly more general but it follows directly from the proof presented in [68]. Given a k'-DNF M of height $h(M) \le h'$, denote by $\mathcal{M}^{h'}_M$ the set of clauses M' of width h' built on variables of M such that $\neg M'$ falsifies all terms in M.

▶ **Lemma 17** ([68])**.** *If there is a* Res(k')*-derivation of a k'-DNF M_0 that has height $h(M_0) \le h'$ from a set of k'-DNFs $\mathcal{M}$ such that each $M_1 \in \mathcal{M}$ has height $h(M_1) \le h'$ and each k'-DNF M in the derivation has height $h(M) \le h''$, where $h'' \ge h'$, then every clause $M'_0 \in \mathcal{M}^{h'}_{M_0}$ can be derived from $\bigcup_{M_1\in\mathcal{M}} \mathcal{M}^{h'}_{M_1}$ by a* Res*-derivation of width $\le h''k'$ (possibly using the weakening rule).*

We are finally ready to finish the proof of Theorem 14.

In each round $j \geq 1$, given the jth k-DNF F, our adversary will construct an h-CNF $\overline{F} := \mathcal{M}^h_{R(F)|_\rho}$ equivalent to $R(F)|_\rho$ by going through all possible clauses M of width h and adding M to $\overline{F}$ if $\neg M$ falsifies all terms in $R(F)|_\rho$. The h-CNF exists and the construction succeeds because $h(R(F)|_\rho) \leq h$. Since there are only $poly(t, |W|)$ variables in $R(F)|_\rho$ (recall that all variables outside $\tau_W(A, b)$ are automatically set to 0), $\overline{F}$ can be constructed by a $poly((t + |W|)^h, |R(F)|_\rho|)$-size circuit. Further, the adversary will produce $\alpha_1, I_1, \ldots, \alpha_j, I_j$, where $\alpha_{j'}$, $j' \in [j]$, is a safe partial solution of $A^+\overline{x} = b^+$ with support $I_{j'}$ of size $|I_{j'}| \leq 2h$. The partial solution α_j will be used to evaluate F: either a clause in $\overline{F}$ will be falsified by $\overline{\rho \circ \alpha_j}$, in which case the response of the adversary to F is 0, or otherwise the response to F is 1. Here (and in the rest of the proof) we abuse the notation and use ρ' to identify variables of $R(F)$ with variables encoding $A^+\overline{x} = b^+$ – note that the relevant part of ρ' can be computed by $poly(t, |R(F)|, |W|, m')$-size circuits from $R(F)$ (the access to m' ensures that the adversary can use new random bits when constructing ρ). The partial assignment $\overline{\rho \circ \alpha_j}$ is the extension of $\rho \circ \alpha_j$ which assigns also the variables y^i_Σ whose values are determined by $\rho \circ \alpha_j$ and $A^+\overline{x} = b^+$.

Response to axioms. Let us first observe that the response of such an adversary to each axiom F in $\tau_W(A, b)$ in each round $j \geq 1$ will be 1. Denote the system of equations of $A^+\overline{x} = b^+$ corresponding to rows from a set $I \subseteq [|b^+|]$ by $A^+\overline{x} = b^+|_I$. Let I^+ be a minimal set of rows in $[|b^+|]$ such that all variables of $R(F)|_\rho$ are in the clauses encoding the equations $A^+\overline{x} = b^+|_{I^+}$. As F has a constant width, [64, Claim 5.7 a)] implies that $|I^+| \leq O(k(d+1))$ and $|I_j \cup I^+| \leq 2h + O(k(d+1)) \leq r/2$. By Lemma 15, there is thus an extension of $I_j \cup I^+$ to a safe set of size $\leq r$ and by Lemma 16 there is a solution $\alpha \supseteq \alpha_j$ of $A^+\overline{x} = b^+|_{I_j \cup I^+}$. A closer look at the definition reveals that $R(F)$ results from rewriting some variables $y^i_{\Sigma^i_\nu}$ in F by linear forms in x_ℓ's which these variables $y^i_{\Sigma^i_\nu}$ represent according to the i'th equation in $\tau_W(A, b)$ and the sign of the "anchor" variable. It is not hard to check that $A^+\overline{x} = b^+|_{I^+}$ implies that the $y^i_{\Sigma^i_\nu}$'s (whose assignment is now determined by $\rho \circ \alpha$) are equivalent to these linear forms. More precisely, if $y^i_{\Sigma^i_{\nu_{i\beta}}}$ is the anchor variable and $\nu > \nu_{i\beta}$, then $A^+\overline{x} = b^+|_{I^+}$ implies

$$y^i_{\Sigma^i_\nu} \oplus \bigoplus_{j \in \Sigma^i_{\nu_+} \setminus \Sigma^i_\nu} x_j = \rho(y^i_{\Sigma^i_{\nu_+}}) = \rho(y^i_{\Sigma^i_{\nu_-}}) \oplus \bigoplus_{j \in (\Sigma^i_{\nu_+} \setminus \Sigma^i_{\nu_-}) \cap J_x(\rho)} \rho(x_j) \oplus \bigoplus_{v \in (\nu-, \nu+]} (b|_\rho)_{i,v}$$
$$= y^i_{\Sigma^i_{\nu_{i\beta}}} \oplus \bigoplus_{j \in \Sigma^i_{\nu_+} \setminus \Sigma^i_{\nu_{i\beta}}} x_j,$$

where $\Sigma^i_\nu \subseteq \Sigma^i_{\nu_+}$ is the minimal initial segment of $J_i(A)$ such that $y^i_{\Sigma^i_{\nu_+}} \in sup(\rho)$ and $\Sigma^i_{\nu_-} \subseteq \Sigma^i_{\nu_{i\beta}}$ is the maximal initial segment of $J_i(A)$ such that $y^i_{\Sigma^i_{\nu_-}} \in sup(\rho)$, so $y^i_{\Sigma^i_\nu} = y^i_{\Sigma^i_{\nu_{i\beta}}} \oplus \bigoplus_{j \in \Sigma^i_\nu \setminus \Sigma^i_{\nu_{i\beta}}} x_j$. If $\nu < \nu_{i\beta}$, we get the same equations but $\Sigma^i_{\nu_+}$ is defined so that $\Sigma^i_{\nu_{i\beta}} \subseteq \Sigma^i_{\nu_+}$ and $\Sigma^i_{\nu_-}$ so that $\Sigma^i_{\nu_-} \subseteq \Sigma^i_\nu$, which yields $y^i_{\Sigma^i_\nu} = y^i_{\Sigma^i_{\nu_{i\beta}}} \oplus \bigoplus_{j \in \Sigma^i_{\nu_{i\beta}} \setminus \Sigma^i_\nu} x_j$. Since $\overline{\rho \circ \alpha}$ satisfies F, $R(F)|_{\overline{\rho \circ \alpha_j}}$ is satisfiable and the adversary responds to F by 1.

Round 1 and Weakening. The first k-DNF our adversary receives is the empty clause and $\alpha_1 := \emptyset, I_1 := \emptyset$. Recall the derivation rules of $\mathsf{Res}(k)$ from [64, Definition 2.6]. If the jth DNF F is a premise of an application of the weakening rule whose conclusion is a DNF $F_{j'}$ to which the adversary responded by 0 in round $j' < j$, the adversary recognizes that $F_{j'}$ is derived using the weakening rule or the CUT-rule. Even if the rule is CUT, it pretends that it is Weakening and sets $\alpha_j := \alpha_{j'}, I_j := I_{j'}$. As $\overline{F_{j'}} \subseteq \overline{F}$, this guarantees the consistency w.r.t. the rule.

Round $j > 1$ for CUT, AND-elimination and AND-introduction. Suppose that the jth k-DNF $F \vee T$ is a premise of the CUT-rule, AND-elimination or AND-introduction whose conclusion is a DNF $F \vee G$ to which the adversary responded by 0 in round $j' < j$. Further, assume that the situation is not consistent with an application of the weakening rule. If the situation is consistent with an application of AND-elimination, the adversary pretends that the rule is AND-elimination. If this is not the case and $T = G$ is a single literal, the adversary pretends that the rule is AND-introduction with a single premise. In the remaining cases, the adversary does not necessarily recognize if $F \vee G$ is derived using CUT or AND-introduction. If this happens, it simulates both alternatives. For AND-introduction, the adversary knows all premises of the derivation step, which is not necessarily the case for CUT. When simulating CUT, the adversary pretends that the premises are $F \vee G \vee T$ and $F \vee G \vee \neg T$. The formulas $F \vee G \vee T$ and $F \vee G \vee \neg T$ – which are such that the transformation R is defined on them – are added to π'. This adds to π' at most two new formulas for each CUT in π, i.e. $\leq 2|\pi|$ formulas in total. Denote the set of the premises of the simulated rule (as pretended by the adversary) by Γ.

As explained in the proof of [64, Claim 5.8], the R-image of the $\mathsf{Res}(k)$-derivation step concluding $F \vee G$ can be completed to a $\mathsf{Res}(O(k(d+1)))$-derivation Π using the axioms of $\tau_W(A, b)$. [14] Now we finish the specification of π' by including in it also the non-principal formulas of each derivation step in π except CUTs (the non-principal formulas of simulated CUTs are already in π'). Here, a non-principal formula of a derivation step is the biggest subformula in its conclusion not intersecting the principal formula. This adds to π' at most $|\pi|$ formulas, making the total size of π' at most $|\pi'| \leq 3|\pi|$. Given that the height of the $R|_\rho$-image of the non-principal formula of the simulated rule is $\leq h$, it is easy to check that height of each formula in $\Pi|_\rho$ is $\leq h + O(k(d+1))$. Hence, by Lemma 17, the clause in $\overline{F \vee G}$ falsified by $\overline{\rho \circ \alpha_{j'}}$ can be derived from $\overline{\Phi_\Gamma}$'s, for $\Phi_\Gamma \in \Gamma$, and from $\tau_W(A, b)|_\rho$ by a Res-derivation Π' of width $w \leq O(k(d+1)h + (k(d+1))^2) \leq O(k(d+1)h)$. The adversary can construct Π' by listing all possible clauses of width $\leq w$ (there are $poly((t + |W|)^w)$ of them) and gradually deriving every clause of width $\leq w$ that can be derived from $\overline{\Phi_\Gamma}$'s, for $\Phi_\Gamma \in \Gamma$, and from $\tau_W(A, b)|_\rho$ in $poly((t + |W|)^w)$ steps.

Next, the adversary simulates its evaluation on Π'. In each step i of the evaluation, it stands on a clause $M_i \in \Pi'$ of width $\leq w$ and holds a safe partial solution $\alpha_{j',i}$ of $A^+\overline{x} = b^+$ with a support $I_{j',i}$ of size $|I_{j',i}| \leq 2w$ such that $\overline{\rho \circ \alpha_{j',i}}$ falsifies M_i. Initially $\alpha_{j',0} := \alpha_{j'}$, $I_{j',0} := I_{j'}$ and M_0 is the clause in $\overline{F \vee G}$ falsified by $\overline{\rho \circ \alpha_{j'}}$. If the adversary wants to move to a predecessor of M_i in Π', it identifies the variable ℓ which was resolved in the step in Π' concluding M_i and a row r_0 in $A^+\overline{x} = b^+$ such that ℓ occurs in the clauses encoding $A^+\overline{x} = b^+|_{\{r_0\}}$. Since $|I_{j',i}| + 1 \leq r/2$, for a sufficiently big C'_h, by Lemma 15, there exists a safe set of rows $\hat{I}_{j',i} \supseteq I_{j',i} \cup \{r_0\}$ of size $|\hat{I}_{j',i}| \leq r$. As $|J_u(A^+) \cap J_{u'}(A^+)| > 1$ only if u, u' are from the same copy of A_0 (in the construction of A^+), the algorithm from the proof of Lemma 15 producing $\hat{I}_{j',i}$ can be implemented by a circuit of size $poly(t_0)$. By Lemma 16 there exists a solution $\hat{\alpha}_{j',i} \supseteq \alpha_{j',i}$ of $A^+\overline{x} = b^+|_{\hat{I}_{j',i}}$. Our adversary finds $\hat{\alpha}_{j',i}$ efficiently using Gaussian elimination (and the bits $b_{u,1}$ with $u \in W$ stored in the non-uniform advice). The adversary then proceeds to the predecessor $M_{i+1} \in \Pi'$ of M_i falsified by $\overline{\rho \circ \hat{\alpha}_{j',i}}$.

[14] Note that terms $R((y^i_{J_i(A)})^{b_i})$, which are by definition empty (by the proof of [64, Claim 5.8] R is defined on literals and by the proof of [64, Claim 5.7 a)] cosmetic literals are removed even if they appeared in the original term), are treated as valid and efficiently derivable. We prefer to set $R((y^i_{J_i(A)})^{b_i}) := (y^i_{J_i(A)})^{b_i}$ and $R(\neg y^i_\emptyset) := \neg y^i_\emptyset$. Formally, this is a new transformation R' which might not output a protected DNF, but whenever $R(F) \neq R'(F)$, $R'(F)|_\rho$ is trivialized because $\rho((y^i_{J_i(A)})^{b_i}) = \rho(\neg y^i_\emptyset) = 1$.

Since M_{i+1} has width $\leq w$, there exists a set $I^w_{j',i} \subseteq \hat{I}_{j',i}$ of w rows such that $\overline{\rho \circ \beta'}$ falsifies M_{i+1}, where β' is the restriction of $\hat{\alpha}_{j',i}$ to the variables with indices in $\bigcup_{i \in I^w_{j',i}} J_i$. The adversary can find $I^w_{j',i}$ efficiently. Finally, the adversary constructs the safe set of rows $I_{j',i+1} \supseteq I^w_{j',i}$, $I_{j',i+1} \subseteq \hat{I}_{j',i}$, of size $|I_{j',i+1}| \leq 2w$ using the algorithm from the proof of Lemma 15 again and sets $\alpha_{j',i+1}$ to be the restriction of $\hat{\alpha}_{j',i}$ to the variables with indices in $\bigcup_{i \in I_{j',i+1}} J_i$. If M_{i+1} is an initial clause of Π', it sets $\alpha_j := \alpha_{j',i+1}$ and $I_j := I_{j',i+1}$. Note that in such case, $|I_j| \leq 2h$ can be ensured.

The adversary cannot arrive in an axiom of $\tau_W(A, b)$, it arrives in a clause of $\overline{\Phi_\Gamma}$, for some $\Phi_\Gamma \in \Gamma$. Hence, if the simulated rule was AND-elimination or AND-introduction with a single premise, the adversary responds by 0 to $F \vee T$ and remains consistent. If the rule was CUT (but not Weakening) or AND-introduction with more premises and in at least one of these cases the simulation ended up in a clause of $\overline{F \vee T}$, the adversary outputs 0. The simulation of a rule is independent of the premise received by the adversary and in the case of CUT we have $\overline{F \vee G \vee T} \subseteq \overline{F \vee T}$. Therefore, the adversary responds by 0 to at least one of the premises and remains consistent w.r.t. every rule.

Efficiency. The least inefficient part of the algorithm implemented by our adversary was the construction of the Res-derivation Π' which took $poly(t^{rad}, m')$ steps. In each round j, the adversary uses a circuit of size $j \cdot poly(t^{rad}, m') \leq m' \cdot poly(t^{rad}, m')$ to produce $\alpha_{j'}, I_{j'}$, for all $j' \in [j]$. The complexity of the adversary is thus $poly(t^{rad}, m')$ and the proof of Theorem 14 is completed.

3.2 Pseudowidth

We present a constructive version of the lower bound for the weak pigeonhole principle PHP^m_n based on the method of pseudowidth, cf. [54, 61, 62, 63, 20]. The lower bound proceeds in two steps. In the first step, a supposed short Res refutation of PHP^m_n is transformed into a Res refutation of small pseudowidth. In the second step, it is shown that the resulting small pseudowidth Res refutation cannot exist. In order to make the lower bound constructive, we modify the second step.

$\neg\mathsf{PHP}^m_n$ is a CNF formula using variables $x_{i,j}$, $i \in [m], j \in [n]$. The intended meaning of $x_{i,j}$ is that the pigeon i is mapped to the hole j. $\neg\mathsf{PHP}^m_n$ consists of the following clauses:

- (pigeon axioms) $\bigvee_{j \in [n]} x_{i,j}$, for $i \in [m]$;
- (hole axioms) $\neg x_{i,j} \vee \neg x_{i',j}$, for $i \neq i' \in [m], j \in [n]$.

For a clause C and pigeon i,

$$N_C(i) := \{j \in [n] \mid C \text{ is satisfied if } x_{i,j} = 1 \wedge \bigwedge_{j' \neq j} x_{i,j'} = 0\}.$$

Further, define $\Delta_C(i) := |N_C(i)|$.

Let $d = (d_1, \ldots, d_m)$ be a vector such that $d_i \in \mathbb{N} \setminus \{0\}, \delta \in \mathbb{R}$ and $d_i > \delta$ for all $i \in [m]$. Then

$$P_d(C) := \{i \in [m] \mid \Delta_C(i) \geq d_i\},$$
$$P_{d,\delta}(C) := \{i \in [m] \mid \Delta_C(i) \geq d_i - \delta\}.$$

The pseudowidth of C is $w_{d,\delta}(C) := |P_{d,\delta}(C)|$. For a set of clauses $\mathcal{A}$ and a Res refutation π, the pseudowidth of $\pi \setminus \mathcal{A}$ is $w_{d,\delta}(\pi \setminus \mathcal{A}) := max_{C \in \pi \wedge C \notin \mathcal{A}} w_{d,\delta}(C)$. A clause C such that $|P_d(C)| \geq w_0$ is referred to as a (w_0, d)-*subaxiom*.

In this section, we include the weakening rule in Res, i.e. a clause C can be derived from a clause D if $D \subseteq C$. We denote the number of clauses in a Res refutation π by L_π.

The first step of the lower bound based on the pseudowidth method is captured in the following lemma. We formulate it in a stronger way than usual: Instead of concluding the existence of a vector d, we claim that a random vector has the desired properties. This is, in fact, what the standard proof of the lemma shows, cf. [20, Lemma 4.2].

▶ **Lemma 18** (Pseudowidth reduction). *Let $\delta := n/\log m$, $3 < m, n \in \mathbb{N}$. There is a distribution of m-dimensional integer vectors $\boldsymbol{d}$ depending only on m, n such that each vector $d = (d_1, \ldots, d_m)$ sampled according to $\boldsymbol{d}$ satisfies $\delta < d_i \leq n$, for all $i \in [n]$, and such that for each Res refutation π of a set of clauses Φ built on variables of PHP^m_n, for each $w_0 \in [m]$ and a random vector $d = (d_1, \ldots, d_m)$ sampled according to $\boldsymbol{d}$, with probability $\geq 1 - 1/L_\pi$, we have a set $\mathcal{A}$ of (w_0, d)-subaxioms with $|\mathcal{A}| \leq L_\pi$ and a Res refutation $\pi' \subseteq \pi$ of $\Phi \cup \mathcal{A}$ satisfying*

$$w_{d,\delta}(\pi' \setminus \mathcal{A}) = O(w_0 + \log L_\pi).$$

▶ **Lemma 19** (Pseudowidth lower bound). *Let $m, n, r \in \mathbb{N} \setminus \{0\}$ and $4 \leq \delta \in \mathbb{R}$. Suppose that $d = (d_1, \ldots, d_m)$ is an integer vector such that $\delta < d_i \leq n$, for all $i \in [m]$. Further, let $\mathcal{A}$ be an arbitrary set of $(1, d)$-subaxioms such that $\mathcal{A} \cap \neg\mathsf{PHP}^m_n = \emptyset$. Then for every Res refutation π of $\neg\mathsf{PHP}^m_n \cup \mathcal{A}$ satisfying $L_\pi < 2^{r\delta/8n}$, we have*

$$w_{d,\delta}(\pi \setminus \mathcal{A}) > \delta/6r.$$

Proof. For the sake of contradiction assume that π is a Res refutation of $\neg\mathsf{PHP}^m_n \cup \mathcal{A}$ with $w_{d,\delta}(\pi \setminus \mathcal{A}) \leq \delta/6r$. W.l.o.g. $|\mathcal{A}| \leq L_\pi$ since all unused $(1, d)$-subaxioms can be erased from $\mathcal{A}$. For the simplification of the notation we assume that for all $C \in \mathcal{A}$ we have $P_{d,\delta}(C) = P_d(C)$ and $P_d(C)$ consists of a single pigeon. (If there are more pigeons in $P_{d,\delta}(C)$, we choose one from $P_d(C)$ in an arbitrary way.)

To reach a contradiction we trace π from its last clause to an axiom. During the trace, when we are at a clause C, we maintain a partial multi-matching ρ_C on $P_{d,\delta}(C)$ which does not satisfy C. For each pigeon i from the domain of ρ_C, ρ_C sets $x_{i,j} = 1$ for r not necessarily different j's and $x_{i,j} = 0$ for all the remaining j's. Additionally, for all $i \neq i'$ in the domain of ρ_C, for all j, ρ_C sets $x_{i,j} = 0$ or $x_{i',j} = 0$. Each ρ_C will be generated using a randomized procedure.

If we reach an axiom C, we are in one of the following three cases:

1. C is a hole axiom $\neg x_{i,j} \vee \neg x_{i',j}$: Then $i, i' \in P_{d,\delta}(C)$, so ρ_C maps both i, i' into different holes and C is satisfied.
2. C is a pigeon axiom $\bigvee_j x_{i,j}$: Then $i \in P_{d,\delta}(C)$, so ρ_C maps i into some hole and C is satisfied.
3. $C \in \mathcal{A}$.

The desired contradiction will be obtained by showing that with high probability over the randomness used to generate ρ_C, $C \in \mathcal{A}$ is satisfied by ρ_C.

Denote the assignment that we have at the ℓth step by ρ_ℓ. Initially, ρ_1 is empty. Assume that in step $\ell > 1$ we have $\rho_\ell = \rho_C$ and that C_0, C_1 are the predecessors of C in π, i.e. C is derived by resolving on C_0, C_1. First, we extend ρ_ℓ to ρ'_ℓ in the following way:

- For each $i \in (P_{d,\delta}(C_0) \cup P_{d,\delta}(C_1)) \setminus P_{d,\delta}(C)$, choose r (not necessarily different) holes $j_1, \ldots, j_r$ uniformly at random from $[n] \setminus (N_C(i) \cup R)$, map i to $j_1, \ldots, j_r$ (i.e. set $x_{i,j_1}, \ldots, x_{i,j_r}$ to 1 and the remaining $x_{i,j}$'s to 0) and add $j_1, \ldots, j_r$ to R. Here, R is initially the set of holes occupied by pigeons assigned by ρ_ℓ.

During the construction of ρ'_ℓ, the size of R does not exceed $3wr \leq \delta/2$, where $w := w_{d,\delta}(\pi \setminus \mathcal{A}) \geq 1$.[15] Moreover, as $i \notin P_{d,\delta}(C)$, we have $\Delta_C(i) < d_i - \delta$, which leaves at least $n - 3wr - d_i + \delta \geq 1$ free holes for each pigeon. So the construction of ρ'_ℓ is well-defined. Since i was mapped to j's such that $j \notin N_C(i)$, the resulting ρ'_ℓ does not satisfy C and thus it does not satisfy at least one of the clauses C_0, C_1. We proceed to the clause which is not satisfied, w.l.o.g. it is C_0, and define $\rho_{\ell+1} = \rho_{C_0}$ as ρ'_ℓ restricted to $P_{d,\delta}(C_0)$. If C is derived via the weakening rule, the construction of $\rho_{\ell+1}$ is analogous. The partial multi-matchings ρ_C satisfy the required properties except that we need to show that with high probability, $C \in \mathcal{A}$ is satisfied by $\rho_\ell = \rho_C$ for $\ell \leq L_\pi$.

Suppose $D \in \mathcal{A}$. We want to bound the probability that $\exists t < \ell, E_t^D$ and D is not satisfied by $\rho_\ell = \rho_C$ at a step $\ell \leq L_\pi$, where E_t^D is the event that the pigeon $i \in P_d(D)$ is assigned in the process of generating ρ_{t+1} from ρ_t.

Consider ρ_t such that E_t^D takes place. During the construction of ρ_{t+1}, while assigning the pigeon i the r'th hole $j_{r'}$, $r' \leq r$, we have at least $n - 3wr - d_i + \delta$ possible choices among the free holes. Denote the set of free holes by $H^{r'}_{\text{free}}$. On the other hand, since $i \in P_d(D)$, there are at least $d_i - 1$ different holes such that sending i to one of these holes will satisfy D (If sending i to a hole j_i satisfies D because $x_{i,j'} = 0$ for some $j' \neq j_i$, the hole j_i becomes "blocked" and sending i to the remaining free holes will still satisfy D even as a multi-matching). Denote the set by $H^{r'}_{\text{sat}}$. Therefore, the probability of the event $B^D_{\rho_t}$ that assigning the pigeon i in the process of generating ρ_{t+1} from ρ_t does not satisfy any literal in D is

$$\begin{aligned}\Pr[B^D_{\rho_t}] &= \prod_{r' \leq r} \left(1 - \frac{|H^{r'}_{\text{free}} \cap H^{r'}_{\text{sat}}|}{|H^{r'}_{\text{free}}|}\right) \leq \prod_{r' \leq r} \left(1 - \frac{|H^{r'}_{\text{free}}| + |H^{r'}_{\text{sat}}| - n}{n}\right) \\ &\leq \left(1 - \frac{\delta - 3wr - 1}{n}\right)^r \leq \left(1 - \frac{\delta}{4n}\right)^r.\end{aligned}$$

The estimation of $\Pr[B^D_{\rho_t}]$ is independent of ρ_t and the assignment of other pigeons than i in the process of constructing ρ_{t+1} from ρ_t. Hence, for each $\ell \leq L_\pi$,

$$\Pr\left[\exists t < \ell, E_t^D \cap B_t^D\right] \leq \sum_{t \leq \ell} \left(1 - \frac{\delta}{4n}\right)^r \leq L_\pi \left(1 - \frac{\delta}{4n}\right)^r \leq L_\pi 2^{-r\delta/4n},$$

where B_t^D is defined like $B^D_{\rho_t}$ but in the probability space of all ρ_ℓ's, and

$$\Pr[C \in \mathcal{A} \wedge C \text{ is not satisfied by } \rho_C] \leq \sum_{D \in \mathcal{A}} \Pr[\exists t < \ell, E_t^D \cap B_t^D] \leq |\mathcal{A}| L_\pi / 2^{r\delta/4n} < 1. \blacktriangleleft$$

Lemmas 18-19, with $m = \lfloor 2^{n^{1/5}} \rfloor$, $r = \lfloor n^{1/2} \rfloor$ and $\Phi = \neg\mathsf{PHP}^m_n$, imply that there is no Res refutation of PHP^m_n of size $2^{\epsilon n^{1/4}}$, for a sufficiently small ϵ.

Notably, our pseudowidth lower bound from Lemma 19 crucially relies on the assumption that the refutation π is short, unlike the lower bounds from [61, 62, 63, 20]. An advantage of our Res lower bound in comparison to [54, 61, 62, 63, 20] is that it constructs efficient Res-adversaries for $\neg\mathsf{PHP}^m_n$.

[15] We have $w \geq 1$ because either a cluase of $\neg\mathsf{PHP}^m_n$ is in π or every axiom in π is in $\mathcal{A}$ and there is a clause in π derived from clauses in $\mathcal{A}$ by weakening or by cut. In all cases it follows that $w \geq 1$.

We remark that $\mathsf{PHP}_s^{|W|}$ is closely related to the statement that a Boolean function f_n with n inputs cannot be computed by a perfect DNF (PDNF) of size s on inputs from $W \subseteq \{0,1\}^n$. Here, a PDNF of size s is a disjunction $\bigvee_{j \le s} K_j$, where each K_j is a conjunction of n literals such that each variable from $\{x_1, \ldots, x_n\}$ appears in K_j in the positive or negative form.

To see the connection, interpret the variable $x_{a,j}$ of $\mathsf{PHP}_s^{|W|}$ as the statement $K_j(a) = 1$ for $a \in W := f^{-1}(1)$. If we wanted to construct efficient adversaries for the encoding of PDNF lower bounds from [63], we would need to make our lower bound work for the "onto" version of the "functional" PHP_n^m.

▶ **Theorem 20** (Efficient adversary for the pseudowidth lower bound)**.** *Let $\gamma < 1$ be an arbitrary constant. There is $\epsilon > 0$ and $n_0 \in \mathbb{N}$ such that for all $n \ge n_0$ there is a poly(m, m')-size randomized* Res*-adversary C for $\neg\mathsf{PHP}_n^m$, $m = \lfloor 2^{n^{1/5}} \rfloor$, such that for each $2^{\epsilon n^{1/4}}$-size* Res *refutation π of any set of clauses, with probability $\ge \gamma$ over the randomness of C, π does not catch C. Here, m' is the input length of circuits in C.*

The proof of Theorem 20 follows by combining Lemma 18 with the proof of Lemma 19.

A complication is that when the adversary possesses ρ_C and receives a predecessor C_0 of C, it might not know what C_1 is. In such case the adversary pretends that C_1 is $C \vee \neg(C_0 \setminus C)$. More precisely, the adversary will automatically replace each clause $C_i \in \pi$, which was used in π to derive C from C_0 and C_1, by $C \vee \neg(C_{1-i} \setminus C)$. The transformation maps π into a sequence of clauses which can be completed into a valid Res refutation $\rho^1(\pi)$ of axioms of π: For each clause $C \vee \neg(C_{1-i} \setminus C)$ replacing a clause C_i, $\rho^1(\pi)$ contains a weakening step deriving $C \vee \neg(C_{1-i} \setminus C)$ from C_i. Since $\rho^1(\pi)$ introduces at most two new clauses for each derivation step in π, we have $L_{\rho^1(\pi)} \le 3L_\pi$.

Given a clause $C \in \rho^1(\pi)$, the adversary sets all variables that do not appear in $\neg\mathsf{PHP}_n^m$ to 0. This substitution, denoted ρ^2, transforms $\rho^1(\pi)$ into $\rho^2(\rho^1(\pi))$. Next, the adversary chooses a random vector d according to the distribution $\boldsymbol{d}$ from Lemma 18. Since the circuits defining the adversary share a random string, the adversary can reproduce the same d in each round. By Lemma 18, with probability $\ge 1 - 1/L_{\rho^1(\pi)}$, $\rho^2(\rho^1(\pi))$ contains a Res-refutation π' of $\Phi \cup \mathcal{A}$ such that $w_{d,\delta}(\pi' \setminus \mathcal{A}) \le O(n^{1/4})$, where $\mathcal{A}$ is a set of $(1,d)$-subaxioms and Φ is a subset of axioms of $\rho^2(\rho^1(\pi))$. W.l.o.g. $1 - 1/L_{\rho^1(\pi)} \ge \gamma + (1-\gamma)/2$ as otherwise we can extend π using the weakening rule on the last clause sufficiently many times.

The adversary then simulates the construction of partial multi-matchings ρ_C from the proof of Lemma 19 on π', with $r = \lfloor n^{1/2} \rfloor$, and responds to $C \in \pi$ by 0 if and only if it ends up in C when traversing π'. Since there might be axioms in π' which are not in $\neg\mathsf{PHP}_n^m \cup \mathcal{A}$, we might not have $w \ge 1$, but the size of R is still bounded by $3r \cdot max\{1, w\} \le \delta/2$, for our setting of parameters. The transformation ρ^1 and the shared randomness of the circuits defining the adversary ensure that independently of whether the adversary received a premise C_0 or C_1 of a cut in π, the adversary constructs the same ρ'_ℓ when tracing the corresponding cut in π' and responds consistently on the cut in π. The property that the adversary responds 1 to the axioms of $\neg\mathsf{PHP}_n^m$ is preserved and the probability of responding 0 to the clauses in $\mathcal{A}$ is $\le (1-\gamma)/2$, if n is big enough.

This completes the proof of Theorem 20.

3.3 k-evaluations

One of the most beautiful lower bound methods in proof complexity is the method of k-evaluations which was used to prove a lower bound on the length of proofs of the pigeonhole principle PHP_n (mapping $n+1$ pigeons to n holes) in constant-depth Frege systems [1, 38, 51].

In the rest of the subsection we prove the following theorem constructing efficient AC^0-Frege-adversaries for $\neg\mathsf{PHP}_n$.

▶ **Theorem 21** (Efficient adversary for AC^0-Frege lower bound)**.**
Let P be an AC^0_d-Frege system in the language consisting of $\vee, \neg, 0, 1$. Further let $\delta < 1/9^d$ and $\gamma < 1$. There is $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ there is a poly(n^{n^δ}, m')-size randomized P-adversary C for $\neg\mathsf{PHP}_n$ such that for each 2^{n^δ}-size P-refutation π of any set of formulas, with probability $\geq \gamma$ over the randomness of C, π does not catch C. Here m' is the input length of circuits in C.

We will refer to the presentation of the AC^0_d-Frege lower bound for PHP_n in [34, Chapter 15].

For a sequence of formulas $\{\theta_i\}$, denote by $\Gamma_d\{\theta_i\}$, the set of all depth d formulas in $\{\theta_i\}$ and all their subformulas.

In each round $j \geq 1$, our adversary will produce a k-evaluation (S^j, H^j) of $\Gamma_d\{\phi_i|_{\rho'\circ\rho_d}\}_{i\in[j]}$, where $\phi_i \in \pi$ is the formula received by the adversary in round i and $k = n^\delta$. The partial restriction ρ' sets all variables that do not appear in PHP_n to 0. In order to simplify the notation, we will ignore ρ' in the rest of our proof. The restriction ρ_d will be the restriction ρ_d from the sequence of restrictions $\rho_0 \subseteq \cdots \subseteq \rho_d$ in the proof of [34, Lemma 15.2.3] but with $\epsilon_0^d = \epsilon < 1/9^d$ instead of $\epsilon < 1/5^d$. We assume $\delta < \epsilon$. Since the circuits defining the adversary share a random string, the adversary can re-produce the same ρ_d in each round. The response of the adversary to the j-th formula ϕ_j from a 2^{n^δ}-size P-refutation π is 1 if $S^j_{\phi_j|_{\rho_d}}(H^j_{\phi_j|_{\rho_d}}) = H^j_{\phi_j|_{\rho_d}}$ and 0 otherwise.

The construction will ensure that there exists a k-evaluation (S, H) of $\Gamma_d\pi|_{\rho_d}$ such that for each j, (S^j, H^j) is (S, H) restricted to $\Gamma_d\{\phi_i|_{\rho_d}\}_{i\in[j]}$. In fact, (S, H) will not depend on $\{\phi_i\}_{i\in[j]}$, i.e., independently of whether the adversary traces π along the path $\phi_1, \ldots, \phi_j$ or along a different path $\phi'_1, \ldots, \phi'_j$, it constructs a restriction of a single "global" k-evaluation (S, H). Therefore, by [34, Lemma 15.1.7], the adversary will be consistent w.r.t. all derivation steps in π. Moreover, since ρ_d is a partial bijection, $\neg\mathsf{PHP}_n|_{\rho_d}$ is an instance of $\neg\mathsf{PHP}_{n^\epsilon}$ (possibly with some tautological clauses), so by [34, Lemma 15.1.6] the adversary will assign 1 to each clause of $\neg\mathsf{PHP}_n$. This means that π will not catch the adversary.

It remains to show how the adversary constructs (S^j, H^j).

The adversary will construct gradually $(S^{j,t}, H^{j,t})$, for each $t \in [d]$, where $(S^{j,t}, H^{j,t})$ is a k-evaluation of $\Gamma_t\{\phi_i|_{\rho_t}\}_{i\in[j]}$, and set $(S^j, H^j) := (S^{j,d}, H^{j,d})$. Recall that ρ_t comes from [34, Lemma 15.2.3] with the adjusted parameter.

The definition of k-evaluation gives an explicit algorithm constructing $(S^{j,t}, H^{j,t})$, if t is the depth of constants $1, 0$ or atoms $p_{i,j}$.

Suppose our adversary already constructed $(S^{j,t-1}, H^{j,t-1})$. $(S^{j,t}, H^{j,t})$ is obtained from $(S^{j,t-1}, H^{j,t-1})$ as follows. First, ρ_t is applied on $(S^{j,t-1}, H^{j,t-1})$, which by [34, Lemma 15.2.1] results in a k-evaluation (S', H') of $\Gamma_{t-1}\{\phi_i|_{\rho_t}\}_{i\in[j]}$. Then all formulas in the set $\Gamma_t\{\phi_i|_{\rho_t}\}_{i\in[j]} \setminus \Gamma_{t-1}\{\phi_i|_{\rho_t}\}_{i\in[j]}$ are gradually "k-evaluated" (i.e. added to the domain of (S', H')) in an arbitrary order. The k-evaluation obtained at the end of this process is $(S^{j,t}, H^{j,t})$.

If the adversary wants to k-evaluate $\neg\psi$ for some ψ that has been k-evaluated already, the construction of the corresponding k-tree and its subset follows directly from the definition of k-evaluation. The only case which requires a more sophisticated treatment is when the adversary needs to k-evaluate $\phi := \bigvee_i \psi_i$, for ψ_i's that have been already k-evaluated.

In the remaining case, the adversary already possesses $(S'_{\psi_i}, H'_{\psi_i})$ given by the k-evaluation of ψ_i's. The adversary constructs an $n^{O(k)}$-size k-tree S_ϕ such that $\bigcup_i H'_{\psi_i} \triangleleft S_\phi$ by fixing a strategy I_{fix} on $\bigcup_i H'_{\psi_i}$ and simulating all possible moves of player II in the proof of [34, Lemma 15.2.2]. The set H_ϕ is defined as the projection of $\bigcup_i H'_{\psi_i}$ on S_ϕ.

[34, Lemma 15.2.2], for $\epsilon < 1/9$, does not only show the existence of some ρ such that the desired trees S_i exist, but in fact it shows that these trees exist for a random ρ (chosen from a set of partial matchings) with probability at least $1 - (1-\gamma)/d$. In more detail, in the proof of the claim that for some ρ, the number of critical pairs is $\leq k/2$, we could assume that for $(1-\gamma)/d$ of all ρ's there is a play with at least $k/2$ critical pairs. This would lead to the inequality $(1-\gamma)a \leq dsb(2k^{1/2}n^\epsilon)^k$ which contradicts the setting of the parameters. As a consequence, the probability that the restrictions $\rho_1, \ldots, \rho_d$ guarantee that for each relevant ϕ the desired k-tree S_ϕ exists (and the probability that the adversary will not be caught) is $\geq \gamma$.

W.l.o.g. the construction of the k-evaluation of a formula depends only on the formula itself and on $\rho_0 \subseteq \cdots \subseteq \rho_d$. Since the circuits defining the adversary share randomness, it can be guaranteed that in each round the adversary uses the same $\rho_0, \ldots, \rho_d$, independently of how it traces the proof π, i.e., independently of $\phi_1, \ldots, \phi_j$. Therefore, there exists a k-evaluation (S, H) such that for every path $\phi_1, \ldots, \phi_j$ in π, the adversary constructs (S, H) restricted to $\Gamma_d\{\phi_i|_{\rho_d}\}_{i\in[j]}$.

Finally, generating $(S^{j,t}, H^{j,t})$ requires to k-evaluate $\leq |\pi|$ formulas and for each relevant subformula ϕ, the construction of $(S^{j,t}_\phi, H^{j,t}_\phi)$ can be done by a circuit of size $n^{O(k)}$. The complexity of the adversary is thus $poly(n^{n^\delta}, m')$. This completes the proof of Theorem 21.

References

1 Miklós Ajtai. The Complexity of the Pigeonhole Principle. *Comb.*, 14(4):417–433, 1994. doi:10.1007/BF01302964.

2 Noel Arteche, Albert Atserias, Susanna F. de Rezende, and Erfan Khaniki. The Proof Analysis Problem. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 2555–2576, 2025. doi:10.1109/FOCS63196.2025.00133.

3 Noel Arteche, Erfan Khaniki, Ján Pich, and Rahul Santhanam. From Proof Complexity to Circuit Complexity via Interactive Protocols. In *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, Tallinn, Estonia, July 8-12, 2024*, pages 12:1–12:20, 2024. doi:10.4230/LIPIcs.ICALP.2024.12.

4 Albert Atserias and Moritz Müller. Automating Resolution is NP-Hard. *J. ACM*, 67(5):31:1–31:17, 2020. doi:10.1145/3409472.

5 Per Austrin and Kilian Risse. Sum-Of-Squares Lower Bounds for the Minimum Circuit Size Problem. In *38th Computational Complexity Conference, CCC 2023, Warwick, UK, July 17-20, 2023*, pages 31:1–31:21, 2023. doi:10.4230/LIPIcs.CCC.2023.31.

6 Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. In *Current Trends in Theoretical Computer Science, Entering the 21th Century*, pages 42–70. World Scientific, 2001.

7 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001. doi:10.1145/375827.375835.

8 Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. Non-Automatizability of Bounded-Depth Frege Proofs. *Comput. Complex.*, 13(1-2):47–68, 2004. doi:10.1007/S00037-004-0183-5.

9 Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On Interpolation and Automatization for Frege Systems. *SIAM J. Comput.*, 29(6):1939–1967, 2000. doi:10.1137/S0097539798353230.

10 Samuel R. Buss. *Bounded arithmetic*. Princeton University, 1985.

11 Samuel R. Buss. On model theory for intuitionistic bounded arithmetic with applications to independence results. In *Feasible Mathematics: A Mathematical Sciences Institute Workshop, Ithaca, New York, June 1989*, pages 27–47, 1990.

12 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning Algorithms from Natural Proofs. In *31st Conference on Computational Complexity, CCC 2016, Tokyo, Japan, May 29 - June 1, 2016*, pages 10:1–10:24, 2016. doi:10.4230/LIPIcs.CCC.2016.10.

13 Lijie Chen, Shuichi Hirahara, Igor Carboni Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. Beyond Natural Proofs: Hardness Magnification and Locality. *J. ACM*, 69(4):25:1–25:49, 2022. doi:10.1145/3538391.

14 Lijie Chen, Jiatu Li, and Igor C. Oliveira. Reverse Mathematics of Complexity Lower Bounds. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 505–527, 2024. doi:10.1109/FOCS61266.2024.00040.

15 Lijie Chen, Ron D. Rothblum, and Roei Tell. Fiat-Shamir in the Plain Model from Derandomization (Or: Do Efficient Algorithms Believe that NP = PSPACE?). In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 977–985, 2025. doi:10.1145/3717823.3718177.

16 Stephen A. Cook. Feasibly Constructive Proofs and the Propositional Calculus. In *Proceedings of the 7th Annual ACM Symposium on Theory of Computing, May 5-7, 1975, Albuquerque, New Mexico, USA*, pages 83–97, 1975.

17 Stephen A. Cook and Jan Krajícek. Consequences of the provability of $NP \subseteq P/poly$. *J. Symb. Log.*, 72(4):1353–1371, 2007. doi:10.2178/JSL/1203350791.

18 Stephen A. Cook and Alasdair Urquhart. Functional Interpretations of Feasibly Constructive Arithmetic. *Ann. Pure Appl. Log.*, 63(2):103–200, 1993. doi:10.1016/0168-0072(93)90044-E.

19 Ben Davis and Robert Robere. Res(log) Proves Bounded-Depth Frege Lower Bounds. *ECCC*, TR26-055, 2026.

20 Susanna F. de Rezende, Jakob Nordström, Kilian Risse, and Dmitry Sokolov. Exponential Resolution Lower Bounds for Weak Pigeonhole Principle and Perfect Matching Formulas over Sparse Graphs. *TheoretiCS*, 4, 2025. doi:10.46298/THEORETICS.25.9.

21 Mika Göös. *Communication Lower Bounds via Query Complexity*. University of Toronto (Canada), 2016.

22 Joshua A. Grochow and Toniann Pitassi. Circuit Complexity, Proof Complexity, and Polynomial Identity Testing: The Ideal Proof System. *J. ACM*, 65(6):37:1–37:59, 2018. doi:10.1145/3230742.

23 Svyatoslav Gryaznov and Navid Talebanfard. Bounded Depth Frege Lower Bounds for Random 3-CNFs via Deterministic Restrictions. *arXiv*, 2024. doi:10.48550/arXiv.2403.02275.

24 Shuichi Hirahara, Rahul Ilango, and Ryan Williams. Beating Brute Force for Compression Problems. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 659–670, 2024. doi:10.1145/3618260.3649778.

25 Emil Jerábek. Dual weak pigeonhole principle, Boolean complexity, and derandomization. *Ann. Pure Appl. Log.*, 129(1-3):1–37, 2004. doi:10.1016/J.APAL.2003.12.003.

26 Ondřej Ježil and Dimitrios Tsintsilidas. Parallelism and Adaptivity in Student-Teacher Witnessing. *arXiv*, 2026. arXiv:2602.19934.

27 Erfan Khaniki. Nisan-Wigderson Generators in Proof Complexity: New Lower Bounds. In *37th Computational Complexity Conference, CCC 2022, Philadelphia, PA, USA, July 20-23, 2022*, pages 17:1–17:15, 2022. doi:10.4230/LIPIcs.CCC.2022.17.

28 Erfan Khaniki. Jump Operators, Interactive Proofs and Proof Complexity Generators. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 573–593, 2024. doi:10.1109/FOCS61266.2024.00044.

29 Jan Krajicek. *Bounded arithmetic, propositional logic and complexity theory*, volume 60. Cambridge University Press, 1995.

30 Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170:123–140, 2001.

31 Jan Krajícek. Implicit proofs. *J. Symb. Log.*, 69(2):387–397, 2004. doi:10.2178/JSL/1082418532.

32 Jan Krajícek. On the Proof Complexity of the Nisan-Wigderson Generator based on a Hard NP ∩ coNP function. *J. Math. Log.*, 11(1), 2011. doi:10.1142/S0219061311000979.

33 Jan Krajíček. On the computational complexity of finding hard tautologies. *Bulletin of the London Mathematical Society*, 46(1):111–125, 2014.

34 Jan Krajíček. *Proof complexity*, volume 170. Cambridge University Press, 2019.

35 Jan Krajíček. *Proof Complexity Generators*. Cambridge University Press, 2025.

36 Jan Krajíček and Pavel Pudlák. Propositional provability and models of weak arithmetic. In *International Workshop on Computer Science Logic*, pages 193–210, 1989.

37 Jan Krajícek and Pavel Pudlák. Some Consequences of Cryptographical Conjectures for S_2^1 and EF. *Inf. Comput.*, 140(1):82–94, 1998.

38 Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random structures & algorithms*, 7(1):15–39, 1995. doi:10.1002/RSA.3240070103.

39 Jiatu Li and Igor C. Oliveira. Unprovability of Strong Complexity Lower Bounds in Bounded Arithmetic. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1051–1057, 2023. doi:10.1145/3564246.3585144.

40 Jiawei Li, Yuhao Li, and Hanlin Ren. Finding Bugs in Short Proofs: The Metamathematics of Resolution Lower Bounds . *arXiv*, 2026. arXiv:2411.15515.

41 Richard J. Lipton and Neal E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 734–740, 1994. doi:10.1145/195058.195447.

42 Jiaqi Lu, Rahul Santhanam, and Iddo Tzameret. $AC^0[p]$-Frege Cannot Efficiently Prove That Constant-Depth Algebraic Circuit Lower Bounds Are Hard. In *17th Innovations in Theoretical Computer Science Conference, ITCS 2026, Bocconi University, Milan, Italy, January 27-30, 2026*, pages 99:1–99:25, 2026.

43 Noam Mazor and Rafael Pass. The Non-Uniform Perebor Conjecture for Time-Bounded Kolmogorov Complexity Is False. In *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, Berkeley, CA, USA, January 30 - February 2, 2024*, pages 80:1–80:20, 2024. doi:10.4230/LIPIcs.ITCS.2024.80.

44 Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Ann. Pure Appl. Log.*, 171(2), 2020. doi:10.1016/J.APAL.2019.102735.

45 Igor C. Oliveira. SIGACT News Complexity Theory Column 124 Meta-Mathematics of Computational Complexity Theory. *SIGACT News*, 56(1):41–68, 2025. doi:10.1145/3726856.3726862.

46 Ján Pich. Circuit lower bounds in bounded arithmetics. *Ann. Pure Appl. Log.*, 166(1):29–45, 2015. doi:10.1016/J.APAL.2014.08.004.

47 Ján Pich. Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic. *Log. Methods Comput. Sci.*, 11(2), 2015. doi:10.2168/LMCS-11(2:8)2015.

48 Ján Pich and Rahul Santhanam. Why are Proof Complexity Lower Bounds Hard? In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1305–1324, 2019. doi:10.1109/FOCS.2019.00080.

49 Ján Pich and Rahul Santhanam. Strong co-nondeterministic lower bounds for NP cannot be proved feasibly. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 223–233, 2021. doi:10.1145/3406325.3451117.

50 Jan Pich and Rahul Santhanam. Towards $P \neq NP$ from Extended Frege lower bounds. *J. ACM*, 73(2), 2026.

51 Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential Lower Bounds for the Pigeonhole Principle. *Comput. Complex.*, 3:97–140, 1993. doi:10.1007/BF01200117.

52 Pavel Pudlák. Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi:10.2307/2275583.

53 Pavel Pudlák and Samuel R. Buss. How to Lie Without Being (Easily) Convicted and the Length of Proofs in Propositional Calculus. In *Computer Science Logic, 8th International Workshop, CSL '94, Kazimierz, Poland, September 25-30, 1994, Selected Papers*, pages 151–162, 1994. doi:10.1007/BFB0022253.

54 Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *J. ACM*, 51(2):115–138, 2004. doi:10.1145/972639.972640.

55 Alexander Razborov. Lower bounds on the monotone complexity of some Boolean function. In *Soviet Math. Dokl.*, volume 31, pages 354–357, 1985.

56 Alexander Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

57 Alexander Razborov. On the Method of Approximations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 167–176, 1989. doi:10.1145/73007.73023.

58 Alexander Razborov. *On provably disjoint NP-pairs.* Aarhus Universitet. Basic Research in Computer Science [BRICS], 1994.

59 Alexander Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya: mathematics*, 59(1):205–227, 1995.

60 Alexander Razborov. Lower Bounds for the Polynomial Calculus. *Comput. Complex.*, 7(4):291–324, 1998. doi:10.1007/S000370050013.

61 Alexander Razborov. Improved Resolution Lower Bounds for the Weak Pigeonhole Principle. *ECCC*, TR01-055, 2001. URL: https://eccc.weizmann.ac.il/eccc-reports/2001/TR01-055/index.html.

62 Alexander Razborov. Resolution lower bounds for the weak functional pigeonhole principle. *Theor. Comput. Sci.*, 303(1):233–243, 2003. doi:10.1016/S0304-3975(02)00453-X.

63 Alexander Razborov. Resolution lower bounds for perfect matching principles. *J. Comput. Syst. Sci.*, 69(1):3–27, 2004. doi:10.1016/J.JCSS.2004.01.004.

64 Alexander Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, pages 415–472, 2015.

65 Alexander Razborov and Steven Rudich. Natural Proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997. doi:10.1006/JCSS.1997.1494.

66 Robert Robere. *Unified lower bounds for monotone computation.* University of Toronto (Canada), 2018.

67 Rahul Santhanam and Iddo Tzameret. Iterated Lower Bound Formulas: A Diagonalization-Based Approach to Proof Complexity. *SIAM Journal on Computing*, pages STOC21–313, 2025.

68 Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. A Switching Lemma for Small Restrictions and Lower Bounds for k-DNF Resolution. *SIAM J. Comput.*, 33(5):1171–1200, 2004. doi:10.1137/S0097539703428555.

69 Roman Smolensky. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82, 1987. doi:10.1145/28395.28404.

70 Ryan Williams. Strong ETH Breaks With Merlin and Arthur: Short Non-Interactive Proofs of Batch Evaluation. In *31st Conference on Computational Complexity, CCC 2016, Tokyo, Japan, May 29 - June 1, 2016*, pages 2:1–2:17, 2016. doi:10.4230/LIPIcs.CCC.2016.2.

Hardness of Computing Nondeterministic Kolmogorov Complexity

Jinqiao Hu
University of Warwick, UK

Zhenjian Lu
University of Victoria, Canada

Igor C. Oliveira
University of Warwick, UK

Abstract

Meta-complexity investigates the complexity of computational problems and tasks that are themselves about computations and their complexity. Understanding whether such problems can capture the hardness of NP is a central research direction. A longstanding open problem in this area is to establish the NP-hardness of MINKT (Ker-I Ko, 1991 [27]), the problem of estimating time-bounded Kolmogorov complexity.

We contribute to this research direction by studying nK^t, a natural variant of Kolmogorov complexity that captures the complexity of representing a string using time-bounded *nondeterministic* computations [4]. Let MINnKT denote the task of estimating $\mathsf{nK}^t(x)$ of a given input string x. We prove that MINnKT $\in$ BPP if and only if NP $\subseteq$ BPP. This can be interpreted as a solution to Ko's question in the setting of nondeterministic time-bounded Kolmogorov complexity.

Crucial to the proof of this result is the investigation of a new notion of *probabilistic nondeterministic* time-bounded Kolmogorov complexity called pnK^t. This measure can be seen as an extension of pK^t complexity [9] obtained by replacing K^t with nK^t. We establish unconditionally that pnK^t has nearly all key properties of (time-unbounded) Kolmogorov complexity, such as language compression, conditional coding, and a form of symmetry of information. Finally, we show that the corresponding meta-computational problem MINpnKT also captures the hardness of NP, and that extending this result to the closely related problem Gap-MINpnKT would imply the exclusion of PH-Heuristica.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases meta-complexity, average-case complexity, Kolmogorov complexity

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.9

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2025/203/ [21]

Funding This work received support from the UKRI Frontier Research Guarantee Grant EP/Y007999/1 and the Centre for Discrete Mathematics and its Applications (DIMAP) at the University of Warwick.

Acknowledgements We thank the anonymous reviewers for their comments and suggestions about the presentation.

1 Introduction

1.1 Overview

Meta-complexity investigates the complexity of computational problems and tasks that are themselves about computations and their complexity. Two central problems in meta-complexity are MCSP [25], the task of estimating the circuit complexity of a given function,

© Jinqiao Hu, Zhenjian Lu, and Igor C. Oliveira;
licensed under Creative Commons License CC-BY 4.0

and MINKT [27], the task of estimating the time-bounded Kolmogorov complexity of a given string. The study of the computational complexity of these and related problems has had a significant impact in theoretical computer science over the last decade, with influential developments in areas such as learning theory [6], cryptography [30], and average-case complexity [15].

A major open problem is to show the NP-hardness of MCSP and MINKT. Beyond the intrinsic interest in establishing the hardness of natural problems, recent results have highlighted the broader significance of this research direction for algorithms and complexity theory. In particular, NP-hardness results in meta-complexity offer a promising approach toward excluding Heuristica [24], i.e., the quest to establish an equivalence between the worst-case and average-case complexities of NP (see [16]). In more detail, under the average-case easiness of NP, certain "gap" versions of MINKT and MCSP admit worst-case to average-case reductions. As a result, if solving Gap-MINKT in the worst case implies that NP is easy in the worst case, then the average-case easiness of NP (which entails the average-case easiness of MINKT and, via the worst-case to average-case reduction, the worst-case easiness of Gap-MINKT) would also lead to the worst-case easiness of NP. Ruling out Heuristica would constitute a major breakthrough in complexity theory and a crucial step toward basing cryptography on the worst-case assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$.

In a celebrated result, Hirahara [17] established the NP-hardness of "partial" versions of MINKT and MCSP, where the input string $x \in \{0,1,\star\}^n$ is allowed to contain unspecified bits encoded by "$\star$". Unfortunately, for such computational problems worst-case to average-case reductions remain unknown. This further motivates the task of establishing the hardness of "total" meta-computational problems, such as MINKT and its variants, which tend to be susceptible to worst-case to average-case reductions.

Summary of Contributions. We explore nK^t complexity, a natural variant of Kolmogorov complexity that considers the complexity of describing a string using time-bounded *nondeterministic* computations. We employ this extension of time-bounded Kolmogorov complexity to develop the meta-complexity of nondeterministic Kolmogorov complexity. In more detail, our main contributions can be summarized as follows:

- We show that the task of computing $\mathsf{nK}^t(x)$ of a given string $x \in \{0,1\}^n$ captures the worst-case hardness of NP. This hardness result holds for standard strings and does not rely on the use of "$\star$". In order to prove this result, we introduce conceptual and technical ideas that are likely to find further applications.
- In particular, we introduce a useful notion of probabilistic nondeterministic Kolmogorov complexity called pnK^t, and establish that it has nearly all key properties of (time-unbounded) Kolmogorov complexity. This means that in many applications we can replace Kolmogorov complexity by a time-bounded measure with similar features, thereby retaining information about the complexity of computations.
- Finally, we describe a striking connection between the complexity of computing $\mathsf{nK}^t(x)$ and its variants and average-case complexity. More precisely, we show that a seemingly minor extension of our hardness results would have a significant consequence to the average-case complexity of the polynomial hierarchy.

Next, we explain our results and contributions in detail.

1.2 Results

1.2.1 Capturing the Hardness of NP via Nondeterministic Kolmogorov Complexity

For $x \in \{0,1\}^*$ and $t \in \mathbb{N}$, the *t-time-bounded nondeterministic Kolmogorov complexity of x* is defined as

$$\mathsf{nK}^t(x) := \min_{\text{program } \Pi} \left\{ |\Pi| \;\middle|\; \begin{array}{l} \bullet\ \forall w \in \{0,1\}^t,\ \Pi(w) \text{ outputs } x \text{ or } \perp \text{ in } t \text{ steps} \\ \bullet\ \exists w \in \{0,1\}^t,\ \Pi(w) \text{ outputs } x \text{ in } t \text{ steps} \end{array} \right\}.$$

We extend this definition to $\mathsf{nK}^t(x \mid y)$, the conditional version of nK^t, in the natural way (see Section 2.3). The complexity measure nK^t was studied under the name CND^t in [4] and CN^t in [5].[1]

Let MINnKT be the following problem: Given $(x, 1^s, 1^t)$, where $x \in \{0,1\}^*$ and $s, t \in \mathbb{N}$, decide whether $\mathsf{nK}^t(x) \leq s$. Note that $\mathsf{MINnKT} \in \Sigma_2^p$, the second level of the polynomial hierarchy.

Theorem 1 stated below requires a computational model that behaves well with respect to the time and description length overhead of composing computations. We refer to Section 1.3 and Section 2 for more details about the mild conditions imposed on the encodings of programs. (We note that the conditions we need are often implicitly assumed in the literature on meta-complexity and time-bounded Kolmogorov complexity.)

Main Contribution. We are ready to state the main result of this paper.

▶ **Theorem 1** (Characterizing NP $\not\subseteq$ BPP by the Worst-Case Hardness of nK^t)**.** *The following equivalence holds.*

$$\mathsf{MINnKT} \in \mathsf{BPP} \iff \mathsf{NP} \subseteq \mathsf{BPP}.$$

Recall that it is a longstanding open problem to show the hardness of MINKT [27]. Theorem 1 can be interpreted as a solution to this problem in the setting of nondeterministic time-bounded Kolmogorov complexity.

In order to establish Theorem 1, we rely on a new approach that explores a delicate regime of parameters of the MINnKT instances. We explain this in more detail in Section 1.2.3, where we also discuss its relevance to average-case complexity. We note that, due to the use of non-black-box techniques from meta-complexity, our result does not provide a Karp-reduction from an NP-complete problem to MINnKT. In any case, the equivalence established in Theorem 1 is sufficient when exploring implications to average-case complexity (see Section 1.2.3).

Related Results. Next, we show that if MINnKT is average-case easy under the uniform distribution (see Section 2.7 for a precise statement), then NP can be solved in polynomial time in the average case.

▶ **Theorem 2** (Average-Case Hardness of nK^t)**.** *The following holds.*

$$(\mathsf{MINnKT}, \mathcal{U}) \in \mathsf{AvgBPP} \implies \mathsf{DistNP} \subseteq \mathsf{AvgBPP}.$$

[1] While the definition of the measure presented in these papers is slightly different than the one given here, it is not hard to see that they are equivalent.

It is instructive to compare Theorem 1 and Theorem 2 with some related results from [12, 14, 18, 31, 22]. Let $\mathsf{MINcKT}^{\mathsf{SAT}}$ denote the SAT-oracle version of conditional MINKT. The following results are known.

- **(i)** Worst-Case Hardness: $\mathsf{MINcKT}^{\mathsf{SAT}} \in \mathsf{BPP} \iff \mathsf{NP} \subseteq \mathsf{BPP}$ [14].[2] Moreover, computing sublinear conditional time-bounded Kolmogorov complexity is NP-hard [18, 31], while computing conditional time-bounded Kolmogorov complexity is NP-hard under a strong cryptographic assumption [22].
- **(ii)** Average-Case Hardness: $\mathsf{MINKT}^{\mathsf{SAT}} \in \mathsf{BPP} \implies \mathsf{DistNP} \subseteq \mathsf{HeurBPP}$ [14, Theorem 3.5].
- **(iii)** PH Average-Case Hardness: $\mathsf{GapMINKT}^{\mathsf{PH}} \in \mathsf{P} \implies \mathsf{DistPH} \subseteq \mathsf{AvgP}$ [12].

In comparison with the results in Item (i), Theorem 1 exhibits the hardness of a *non-conditional* meta-computational problem, i.e., for instances without a conditional string. On the other hand, in comparison with Item (ii), our proof of Theorem 2 establishes a stronger conclusion under a weaker assumption, as explained next. For the DistNP-hardness of $\mathsf{MINKT}^{\mathsf{SAT}}$ stated in Item (ii), it is enough to have an efficient algorithm that accepts the set of strings with low $\mathsf{K}^{\mathsf{poly},\mathsf{NP}}$-complexity while rejecting a large fraction of random strings. In contrast, in Theorem 2 we only need an efficient algorithm that accepts strings with low $\mathsf{nK}^{\mathsf{poly}}$-complexity while rejecting a large fraction of random strings. Since the set of strings with low $\mathsf{nK}^{\mathsf{poly}}$-complexity is a subset of those with low $\mathsf{K}^{\mathsf{poly},\mathsf{NP}}$-complexity, the actual assumption required in Theorem 2 is implied by the one needed in Item (ii). Moreover, in Theorem 2 we obtain the stronger consequence that NP is average-case easy in the errorless setting, rather than in the error-prone setting considered in Item (ii). Finally, in Item (iii), [12] shows the DistPH-hardness of a generalization of a gap formulation of MINKT that considers compression with access to a PH-oracle. While it is possible to use a weaker assumption to derive the weaker conclusion $\mathsf{DistNP} \subseteq \mathsf{AvgP}$, similarly to the discussion about Item (ii) the corresponding meta-computational problem is harder than the one in the assumption of Theorem 2 (see [12, Theorem 1.16]).

In our next result, we show that if MINnKT is average-case easy under the uniform distribution, then NP can be solved in sub-exponential time in the worst case.

▶ **Theorem 3** (Worst-Case to Average-Case Reduction). *The following holds.*

$$(\mathsf{MINnKT}, \mathcal{U}) \in \mathsf{AvgBPP} \implies \mathsf{NP} \subseteq \mathsf{RTIME}\left[2^{O(n/\log n)}\right].$$

Previous works [15, 9] have shown that if Σ_2^{P} is average-case easy, then NP can be solved in time $2^{O(n/\log n)}$. Theorem 3 obtains the same conclusion under the weaker assumption that MINnKT is average-case easy. Indeed, MINnKT can be solved in NP using two non-adaptive calls to an NP oracle, so it is in $\mathsf{NP}^{\mathsf{NP}} = \Sigma_2^{\mathsf{P}}$. Hence, if Σ_2^{P} is easy on average, so is MINnKT. However, the reverse direction remains unclear. Also, note that if MINnKT is average-case easy, then by Theorem 2, the same holds for NP. Yet, it is unknown if Σ_2^{P} is average-case easy under the assumption that NP is average-case easy.

1.2.2 Probabilistic Nondeterministic Kolmogorov Complexity Is (Essentially) All You Need

The proof of Theorem 1 relies on certain useful properties of a probabilistic variant of nK^t called pnK^t. First, we review the following definition, which will be useful when comparing

[2] In fact, [14] showed the stronger result that P^{NP} reduces to $\mathsf{MINcKT}^{\mathsf{SAT}}$ under deterministic polynomial-time reductions.

our results with prior work. For $x \in \{0,1\}^*$ and $t \in \mathbb{N}$, the *t-time-bounded randomized nondeterministic Kolmogorov complexity of x* is defined as

$$\mathsf{rnK}^t(x) := \min_{\text{program } \Pi}\left\{|\Pi| \;\middle|\; \Pr_{r\sim\{0,1\}^t}\begin{bmatrix}\forall w \in \{0,1\}^t, \Pi(w;r) \text{ outputs } x \text{ or } \perp \text{ in } t \text{ steps;}\\ \exists w \in \{0,1\}^t, \Pi(w;r) \text{ outputs } x \text{ in } t \text{ steps}\end{bmatrix} \geq \frac{2}{3}\right\}.$$

The complexity measure rnK^t was studied under the name CAM^t in [5].

pnK^t Complexity. Inspired by pK^t complexity [9] and its properties (see, e.g., [32]), we introduce the following notion of probabilistic nondeterministic Kolmogorov complexity. For $x \in \{0,1\}^*$ and $t \in \mathbb{N}$, the *t-time-bounded probabilistic nondeterministic Kolmogorov complexity of x* is defined as

$$\mathsf{pnK}^t(x) := \min\left\{k \in \mathbb{N} \;\middle|\; \Pr_{r\sim\{0,1\}^t}\begin{bmatrix}\exists \text{ program } \Pi \text{ of size at most } k \text{ s.t.}\\ \quad \text{the following conditions hold:}\\ \forall w \in \{0,1\}^t,\ \Pi(w;r) \text{ outputs } x \text{ or } \perp \text{ in } t \text{ steps;}\\ \exists w \in \{0,1\}^t,\ \Pi(w;r) \text{ outputs } x \text{ in } t \text{ steps}\end{bmatrix} \geq \frac{2}{3}\right\}.$$

Equivalently,

$$\mathsf{pnK}^t(x) := \min\left\{k \in \mathbb{N} \;\middle|\; \Pr_{r\sim\{0,1\}^t}\left[\mathsf{nK}^t(x \mid r) \leq k\right] \geq \frac{2}{3}\right\}.$$

In other words, while pK^t corresponds to K^t complexity in the presence of a random string, analogously, pnK^t is simply nK^t complexity in the presence of a random string.

We extend the definitions of rnK^t and pnK^t to their corresponding oracle versions in the natural way, i.e., by giving to the program oracle access to the corresponding set $\mathcal{O} \subseteq \{0,1\}^*$. We write $\mathsf{rnK}^{t,\mathcal{O}}$ and $\mathsf{pnK}^{t,\mathcal{O}}$, respectively.

We show that pnK^t admits nearly all key properties of (time-unbounded) Kolmogorov complexity, such as a general form of language compression, (conditional) coding, and useful variants of symmetry of information. As a consequence, in many scenarios, pnK^t can replace Kolmogorov complexity while still providing useful bounds on the complexity of computations.

Below we discuss the properties of pnK^t in more detail and contrast our results with prior work.

Language Compression for pnK^t. As established by [28, Theorem 4.1.4] (improving [5]), nondeterminism allows us to get non-trivial language compression in the time-bounded setting. In other words, there is a polynomial $p(\cdot)$ such that for every set $A := \{A_n \subseteq \{0,1\}^n\}_{n\in\mathbb{N}}$, and for all $n \in \mathbb{N}$ and $x \in A_n$,

$$\mathsf{nK}^{p(n),A_n}(x) \leq \log|A_n| + O(\sqrt{\log|A_n|} \cdot \log n).$$

[5] showed that an improved language compression theorem holds if we employ randomness in addition to nondeterminism. More precisely, there is a polynomial $p(\cdot)$ such that for every set $A := \{A_n \subseteq \{0,1\}^n\}_{n\in\mathbb{N}}$, and for all $n \in \mathbb{N}$ and $x \in A_n$,

$$\mathsf{rnK}^{p(n),A_n}(x) \leq \log|A_n| + \log^3 p(n).$$

This result is still suboptimal due to the additive $O(\log^3 n)$ term. This super-logarithmic overhead can lead to super-polynomial running times in applications.

In contrast to the aforementioned results, we observe that an optimal language compression theorem holds for pnK^t. This result plays a key role in the proof of Theorem 1.

▶ **Theorem 4** (Language Compression for pnK^t). *There is a polynomial $p_{\mathsf{LC}}(\cdot)$ such that for every set $A := \{A_n \subseteq \{0,1\}^n\}_{n\in\mathbb{N}}$, and for all $n \in \mathbb{N}$ and $x \in A_n$,*

$$\mathsf{pnK}^{p_{\mathsf{LC}}(n),A_n}(x) \leq \log |A_n| + \log p_{\mathsf{LC}}(n).$$

On the other hand, we provide evidence that nondeterminism is indeed crucial for language compression, even when restricted to predicates in P.

▶ **Theorem 5.** *Item 1 implies Item 2 in the statements appearing below.*

1. (Average-Case pK^t Language Compression for P.) *For any set $A := \{A_n \subseteq \{0,1\}^n\}_{n\in\mathbb{N}}$ that is in P, there exists a constant $c > 0$ such that for all $n \in \mathbb{N}$*

$$\Pr_{x\sim A_n}\left[\mathsf{pK}^{n^c}(x) \leq \log |A_n| + c \cdot \log n\right] \geq \frac{1}{n^c}.$$

2. $\mathsf{Unary\text{-}NP} \subseteq \mathsf{BPP}$.

Theorem 5 complements a result from [5] showing that for all $n, t, k \in \mathbb{N}$ such that $k \leq n$, there exists a set A such that $\log |A_n| = k$ and for every $x \in A_n$,

$$\mathsf{rK}^{t,A_n}(x) \geq n - \log |A_n| - \log t - O(1).$$

This bound on rK^{t,A_n} complexity rules out the possibility of getting a language compression theorem without nondeterminism for *every* set A, while Theorem 5 rules out language compression without nondeterminism (even on average) for sets decidable in polynomial time (under the assumption that $\mathsf{Unary\text{-}NP} \not\subseteq \mathsf{BPP}$). The latter case is of interest in the proof of Theorem 1.

Conditional Coding for pnK^t. Since pnK^t is a more "powerful" notion than pK^t, and the latter admits a coding theorem [33], it follows that pnK^t also has this property. Here, we show that pnK^t possesses an even stronger *conditional* coding property.

▶ **Theorem 6** (Conditional Coding for pnK^t). *For every polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ supported over $\{0,1\}^n \times \{0,1\}^n$, there exists a polynomial p such that for all $n \in \mathbb{N}$,*

$$\mathsf{pnK}^{p(n)}(x \mid y) \leq \log \frac{1}{\mathcal{D}_n(x \mid y)} + \log p(n).$$

In contrast, conditional coding does not hold for pK^t under the assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$ [19]. This gives another natural setting where nondeterminism seems essential and leads to an advantage of pnK^t over pK^t.

Symmetry of Information for pnK^t. Finally, we show that non-trivial forms of the symmetry of information principle hold for pnK^t complexity. The results discussed below improve the parameters of analogous results established for rnK^t complexity in [29] (see also [28, Theorem 6.3.3]).

▶ **Theorem 7** (Semi-Symmetry of Information for pnK^t). *There exist polynomials $p_{\mathsf{Sol}}(\cdot)$ and $p_0(\cdot)$ such that for all sufficiently large $x, y \in \{0,1\}^*$ and every $t \geq p_0(|x| + |y|)$,*

$$\mathsf{pK}^t(x, y) \geq \mathsf{pnK}^{p_{\mathsf{Sol}}(t)}(y \mid x) + \mathsf{pnK}^{p_{\mathsf{Sol}}(t)}(x) - \log p_{\mathsf{Sol}}(t).$$

The reason we call Theorem 7 semi-symmetry of information is because on the left-hand side of the inequality we obtain $\mathsf{pK}^t(x,y)$ as opposed to $\mathsf{pnK}^t(x,y)$. We are able to overcome this limitation in the average-case setting, which is sufficient in many applications.

▶ **Theorem 8** (Average-Case Symmetry of Information for pnK^t). *For every polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ supported over $\{0,1\}^n \times \{0,1\}^n$, and for every polynomial q, there exists a polynomial p such that for every computable time bound $t \geq p(n)$ and for all large enough n,*

$$\Pr_{(x,y)\sim\mathcal{D}_n}\left[\mathsf{pnK}^t(x,y) \geq \mathsf{pnK}^t(y \mid x) + \mathsf{pnK}^t(x) - c\cdot\log t\right] \geq 1 - \frac{1}{q(n)},$$

where $c > 0$ is a universal constant.

In contrast, average-case symmetry of information does not hold for pK^t under the existence of one-way functions [19].

To sum up, pnK^t satisfies nearly all pillars of time-unbounded Kolmogorov complexity (as discussed in [28]), which makes the new measure attractive in applications of meta-complexity in time-bounded settings. The only missing property is *worst-case* symmetry of information, which we leave as an interesting open problem. It is worth pointing out that there is an oracle world where symmetry of information for nK^t does not hold (see [29, Corollary 1] and [28, Corollary 6.5.2]).

Using a straightforward adaptation of the proof of Theorem 1, we also establish the hardness of $\mathsf{MINpnKT}$, the problem of computing pnK^t complexity (we refer to [21, Appendix C] for the precise definition of this problem and the minor modifications needed in the proof).

▶ **Theorem 9** (Characterizing $\mathsf{NP} \not\subseteq \mathsf{BPP}$ by the Worst-Case Hardness of pnK^t). *The following holds.*

$$\mathsf{MINpnKT} \in \mathsf{prBPP} \iff \mathsf{NP} \subseteq \mathsf{BPP}.$$

Similarly to Theorem 1, Theorem 9 requires the use of a computational model and encodings for which the composition of programs does not yield a significant overhead in running time and program length, as explained in Section 1.3 and Section 2.2.

1.2.3 Gap Problems in Meta-Complexity and Connections to Average-Case Complexity

In this section, we elaborate on some aspects of our hardness results and their connections to average-case complexity. In more detail, we discuss a striking phenomenon where a mild extension of Theorem 9 would have a significant impact on our understanding of the computational complexity of the polynomial hierarchy.

Related Work. [18] shows that if NP is average-case easy, then a specific gap version of computing *conditional* K^t is also easy. More precisely, given $(x, y, 1^s, 1^t)$, one can efficiently decide whether $\mathsf{K}^t(y \mid x) \leq s$ or $\mathsf{K}^{\mathsf{poly}(t,|x|)}(y \mid x) > s + O(\log(t+|x|)) + \mathsf{K}^t(x) - \mathsf{K}^{\mathsf{poly}(t,|x|)}(x)$. Thus, proving the NP-hardness of this problem would suffice to rule out Heuristica. However, the gap in this problem involves the quantity $\mathsf{K}^t(x) - \mathsf{K}^{\mathsf{poly}(t,|x|)}(x)$ (known as the computational depth of x), which can be large. This makes it more challenging to establish the NP-hardness of this class of instances. In fact, there remains a significant gap between existing NP-hardness results for computing conditional K^t and what is needed to exclude Heuristica (see [22] for a detailed discussion).

Shifting focus to the *non-conditional* case, if NP is average-case easy, then a simpler gap version of computing K^t is worst-case easy: given $(x, 1^s, 1^t)$, one can efficiently decide whether $\mathsf{K}^t(x) \leq s$ or $\mathsf{K}^{\mathsf{poly}(t)}(x) > s + O(\log t)$ [11, 13]. Thus, showing that this "small-gap" version (known as Gap-MINKT) is NP-hard would also rule out Heuristica. However, currently we do not know even if MINKT is hard, let alone Gap-MINKT.

We also mention related results from [17], which show that obtaining a significantly better hardness result for a meta-computational problem called MINLT is sufficient to rule out Heuristica.

Theorem 1 and Theorem 9 make progress on this front by presenting non-conditional meta-computational problems whose hardness characterize $\mathsf{NP} \not\subseteq \mathsf{BPP}$. Furthermore, existing worst-case to average-case reduction techniques can be adapted to the setting of MINpnKT. As explained below, these results bear an intriguing connection to the longstanding problem of excluding PH-Heuristica, i.e., showing that if DistPH $\subseteq$ AvgBPP then PH $\subseteq$ BPP (see, e.g., [12] for more information about this problem).[3]

Consider the following statements:

- **(i)** If MINpnKT $\in$ prBPP then NP $\subseteq$ BPP (Theorem 9).
- **(ii)** If (MINnKT,$\mathcal{U}$) $\in$ AvgBPP then Gap-MINpnKT $\in$ prBPP (see Proposition 58 in Appendix B).

Recall that, as alluded to above, we have MINnKT $\in \Sigma_2^{\mathsf{P}}$. As a consequence, under the assumption that Dist$\Sigma_2^{\mathsf{P}} \subseteq$ AvgBPP, it follows from Item (ii) that Gap-MINpnKT $\in$ prBPP. Therefore, if we could replace MINpnKT with Gap-MINpnKT in Item (i) (or replace Gap-MINpnKT with MINpnKT in Item (ii)), it would follow that NP $\subseteq$ BPP (and consequently PH $\subseteq$ BPP). In other words, the seemingly minor distinction between MINpnKT and Gap-MINpnKT is currently the only issue preventing us from excluding PH-Heuristica by showing that if Dist$\Sigma_2^{\mathsf{P}} \subseteq$ AvgBPP then PH $\subseteq$ BPP (see Corollary 59 in Appendix B).

Interpretation. There are two ways of interpreting these results. An optimistic perspective on the scenario described above is that we are close to excluding PH-Heuristica, since this merely requires the hardness of Gap-MINpnKT. In other words, under this perspective, the hope is that MINpnKT might be similar to Gap-MINpnKT, even if our current techniques distinguish these two problems. Moreover, a few existing hardness results, such as [17], extend to the gap version of the problem, perhaps indicating that a new idea or different proof of Theorem 9 might suffice to generalize the existing result. On the other hand, a pessimistic interpretation is that we remain far from excluding PH-Heuristica, and our results and techniques simply suggest a fundamental distinction between establishing the hardness of MINnKT and MINpnKT compared with the variants of these problems with a small gap. In particular, under this interpretation, our hardness results shed light on why existing worst-case to average-case reductions in meta-complexity inherently produce a gap between positive and negative instances.

While we are currently unable to exclude PH-Heuristica by establishing the hardness of computing nondeterministic Kolmogorov complexity with a small gap, we make progress by showing that the *conditional* version of Gap-MINnKT is hard. More precisely, for $\tau\colon \mathbb{N} \to \mathbb{N}$, let Gap$_\tau$-Cond-MINnKT be the following promise problem: Given $(x, y, 1^s, 1^t)$, where $x, y \in \{0,1\}^*$ and $s, t \in \mathbb{N}$, decide whether $\mathsf{nK}^t(y \mid x) \leq s$ or $\mathsf{nK}^{\tau(t,|x|)}(y \mid x) > s + \log \tau(t + |x|)$. We say that Gap-Cond-MINnKT $\in$ prBPP if there is a probabilistic polynomial-time algorithm that solves Gap$_\tau$-Cond-MINnKT for some polynomial τ.

[3] Recall that PH $\subseteq$ BPP if and only if NP $\subseteq$ BPP. Consequently, excluding PH-Heuristica is a necessary step toward excluding Heuristica (i.e., proving that if DistNP $\subseteq$ AvgBPP then NP $\subseteq$ BPP).

▶ **Theorem 10.** *The following holds.*

$$\mathsf{Gap\text{-}Cond\text{-}MINnKT} \in \mathsf{prBPP} \iff \mathsf{NP} \subseteq \mathsf{BPP}.$$

Thus, as a corollary of our results (Theorem 1 and Theorem 10), we relate the complexities of MINnKT and Gap-Cond-MINnKT, exhibiting an interesting link between the non-conditional version and the conditional version. Formally, it is possible to establish the following slightly more general equivalence.

▶ **Corollary 11.** *The following holds.*

$$\mathsf{MINnKT} \in \mathsf{BPP} \iff \mathsf{Cond\text{-}MINnKT} \in \mathsf{BPP} \iff \mathsf{Gap\text{-}Cond\text{-}MINnKT} \in \mathsf{prBPP}.$$

1.3 Techniques

In this section, we describe some ideas employed in the proof of our results. We primarily focus on the proof that if MINnKT $\in$ BPP then NP $\subseteq$ BPP (Theorem 1). Beyond being our main contribution, its proof provides insights into the techniques employed in other parts of the paper and highlights the usefulness of pnK^t complexity. (For technical reasons, we will introduce and use a further complexity measure in the proof, as described later in this section.)

In order to establish the hardness of MINnKT, we explore different properties of non-determinism Kolmogorov complexity and its probabilistic extension. The first component of the argument is the optimal language compression theorem for pnK^t (Theorem 4).

1. Language compression for pnK^t. We provide two proofs that pnK^t admits language compression with optimal parameters. Our first discovered proof, which relies on techniques from [5] and [15], served as the inspiration for the definition of pnK^t complexity. This proof appears in Section 3.2. Below we present a different, simpler argument that nicely illustrates the benefits of combining *randomness* and *nondeterminism* for string compression. A similar technique is used in [34] and [4].

Let $A_n \subseteq \{0,1\}^n$, $x \in A_n$, and set $k = \lceil \log |A_n| \rceil + 2$. We employ an efficiently computable family $H \subseteq \{0,1\}^n \to \{0,1\}^k$ of hash functions such that $\mathbf{Pr}_{h\sim H}[h(x) = h(y)] \leq 2^{-k}$ for every $y \in A_n \setminus \{x\}$. By a union bound and our choice of k,

$$\mathop{\mathbf{Pr}}_{h\sim H}[\exists y \in A_n \setminus \{x\} \text{ s.t. } h(x) = h(y)] \leq 2^{-k} \cdot |A_n| \leq \frac{1}{4}. \tag{1}$$

The idea is that, for a randomly sampled $h \sim H$ (corresponding to the *randomness* in our pnK^t description), we can store the integers n and k and the value $h(x) \in \{0,1\}^k$ in the program (which has access to h) using $k + O(\log n)$ bits, then use *nondeterminism* to guess x. In more detail, let v be the hash value stored in the description, and let x' be a nondeterministic input string of length n. If $h(x') = v$ and $x' \in A_n$, we output x'; otherwise we output $\perp$. By Equation (1), with probability at least $3/4$, we sample a good h such that the value $h(x)$ is *unique* among the elements in A_n. This shows that, for a good choice of h, there is an efficient nondeterministic program of length at most $k + O(\log n) = \log |A_n| + O(\log n)$ that (i) outputs x on at least one computation path, and (ii) outputs either x or $\perp$ on every computation path.

It follows from the argument described above that there is a polynomial $p_{\mathsf{LC}}(\cdot)$ such that, for every set $A := \{A_n \subseteq \{0,1\}^n\}_{n\in\mathbb{N}}$ and for every $n \in \mathbb{N}$ and $w \in A_n$,

$$\mathsf{pnK}^{p_{\mathsf{LC}}(n),A_n}(w) \leq \log |A_n| + \log p_{\mathsf{LC}}(n), \tag{2}$$

as desired.

Next, we explain the connection between language compression and the complexity of NP employed in the proof of Theorem 1. The argument is inspired by ideas from [1] (see also [19] for an equivalence between compression and the easiness of NP).

2. Compressing witnesses and the easiness of NP. Let $L \in \mathsf{NP}$. Fix a verifier $V(\cdot,\cdot)$ for L. Let $x \in L$, where $|x| = n$, and consider the set $W^x = \{w \in \{0,1\}^m \mid V(x,w) = 1\}$, where $m = \mathsf{poly}(n)$. Note that, by language compression (Equation (2)) applied to the set W^x, for every $w \in W^x$,

$$\mathsf{pnK}^{\mathsf{poly}(n),W^x}(w) \leq \log|W^x| + O(\log n).$$

Moreover, since given x the set W^x can be decided in polynomial time, we have

$$\mathsf{pnK}^{\mathsf{poly}(n)}(w \mid x) \leq \log|W^x| + O(\log n). \tag{3}$$

Due to the use of the nondeterministic measure $\mathsf{pnK}^{\mathsf{poly}(n)}$, this bound is not yet sufficient to allow us to efficiently find a witness $w \in W^x$ given x. However, let us suppose for a moment that, under the assumption that $\mathsf{MINnKT} \in \mathsf{BPP}$, we could show that for every pair (y,x) of strings of length $\mathsf{poly}(n)$ the following holds:

$$\mathsf{pK}^{\mathsf{poly}(t)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + \lambda(n), \tag{4}$$

for some function $\lambda(n)$.[4] In other words, assume that conditional pnK^t complexity collapses to conditional $\mathsf{pK}^{\mathsf{poly}(t)}$ with an additive overhead term $\lambda(n)$.

Under the hypothesis that Equation (4) holds, we obtain together with Equation (3) that

$$\mathsf{pK}^{\mathsf{poly}(n)}(w \mid x) \leq \log|W^x| + O(\log n) + \lambda(n) \tag{5}$$

for every $w \in W^x$. Roughly speaking, this means that every witness $w \in W^x$ can be compressed to just $v(n) := \log|W^x| + O(\log n) + \lambda(n)$ bits. Since each witness $w \in W^x$ admits its own compressed representation of this length, it is not hard to show that given x one can find a witness $w \in W^x$ with probability at least $1/(\mathsf{poly}(n) \cdot 2^{\lambda(n)})$ simply by running a randomly chosen program of length at most $v(n)$ for $\mathsf{poly}(n)$ steps (see Lemma 32). Consequently, repeating this procedure $\mathsf{poly}(n) \cdot 2^{\lambda(n)}$ times provides an algorithm that finds a witness for x with high probability, whenever one exists. This shows that $L \in \mathsf{BPP}$ if we can guarantee that $\lambda(n) = O(\log n)$.

Note that it is crucial for this argument to work that we obtain an upper bound on $\mathsf{pK}^{\mathsf{poly}(n)}$ complexity in Equation (5) instead of $\mathsf{pnK}^{\mathsf{poly}(n)}$ complexity. The latter does not provide an efficient algorithm that allows us to recover a string from its (nondeterministic) compressed representation.

Given the discussion above, in order to show that $\mathsf{NP} \subseteq \mathsf{BPP}$ it is sufficient to establish Equation (4) with $\lambda(n) = O(\log n)$, under the assumption that $\mathsf{MINnKT} \in \mathsf{BPP}$.

3. Easiness assumptions and collapses in time-bounded Kolmogorov complexity. It will be instructive to compare how different easiness assumptions allow us to relate pK and pnK complexities. For any large enough $t = \mathsf{poly}(n)$, the following implications can be established via a careful adaptation of existing techniques from meta-complexity to the setting of nondeterministic Kolmogorov complexity (see Lemma 55 and Lemma 52):

[4] Recall that $\mathsf{pK}^t(y)$ is the minimum k such that with probability at least $2/3$ over the choice of a random string $r \in \{0,1\}^t$, we have $\mathsf{K}^t(y \mid r) \leq k$ [32].

$$\mathsf{Gap\text{-}Cond\text{-}MINnKT} \in \mathsf{prBPP} \implies \mathsf{pK}^{\mathsf{poly}(t)}(y \mid x) \leq \mathsf{pnK}^{t}(y \mid x) + O(\log n).$$

$$\mathsf{Gap\text{-}MINnKT} \in \mathsf{prBPP} \implies \mathsf{pK}^{\mathsf{poly}(t)}(y \mid x) \leq \mathsf{pnK}^{t}(y \mid x) + \mathsf{pK}^{t}(x) - \mathsf{pK}^{\mathsf{poly}(t)}(x) + O(\log n).$$

The conclusion in the first implication is precisely what we would like to show in Equation (4). However, it assumes the easiness of the problem of estimating *conditional* nondeterministic Kolmogorov complexity. On the other hand, while the assumption in the second implication is sufficient for our purposes (as it is implied by $\mathsf{MINnKT} \in \mathsf{BPP}$), the inequality in the conclusion has the extra term $\mathsf{pK}^{t}(x) - \mathsf{pK}^{\mathsf{poly}(t)}(x)$, known as the *time-bounded computational depth*. This quantity might be significantly larger than $O(\log n)$ for some strings x and choices of t. On the other hand, it is known to be at most $O(\log n)$ for an average string x, and at most $O(n/\log n)$ for every x for some moderately bounded choice of t. We remark that this notion of computational depth, with a polynomial gap in the time bounds, appears in many prior works in meta-complexity and has been an infamous obstacle that forces results to hold only in the average-case setting or leads to suboptimal worst-case running times of the form $2^{O(n/\log n)}$ rather than polynomial time (see, e.g., [15, 7, 9, 20]).

In order to achieve our goal, an important idea is to aim instead for an implication of the following form:

$$\mathsf{MINnKT} \in \mathsf{BPP} \implies \mathsf{pK}^{\mathsf{poly}(t)}(y \mid x) \leq \mathsf{pnK}^{t}(y \mid x) + \mathsf{pnK}^{t}(x) - \mathsf{pnK}^{t+p(n)}(x) + O(\log n), \quad (6)$$

where $p(n)$ is a polynomial independent of t. As before, assuming Equation (6), we can conclude the proof if we could show that $\mathsf{pnK}^{t}(x) - \mathsf{pnK}^{t+p(n)}(x) = O(\log n)$. While this might not hold for a particular value of t, it is possible to argue via a telescoping sum trick that this bound does hold for some t' that is not too large.[5] In more detail, for any fixed time bound t, let $t_i = t + i \cdot p(n)$, where $i \in \{0, 1, \ldots, n\}$. First, note that $\mathsf{pnK}^{t_0}(x) - \mathsf{pnK}^{t_n}(x) \leq 2n$, since any n-bit string x can be efficiently described with at most $2n$ bits. Therefore,

$$\sum_{i=0}^{n-1} \left(\mathsf{pnK}^{t_i}(x) - \mathsf{pnK}^{t_{i+1}}(x)\right) = \mathsf{pnK}^{t_0}(x) - \mathsf{pnK}^{t_n}(x) \leq 2n.$$

Consequently, there is an index i such that $\mathsf{pnK}^{t_i}(x) - \mathsf{pnK}^{t_{i+1}}(x) \leq 2$. In particular, for some time bound $t' \leq t + n \cdot p(n)$, we have $\mathsf{pnK}^{t'}(x) - \mathsf{pnK}^{t'+p(n)}(x) = O(\log n)$. It turns out that this is sufficient to establish Equation (4) with $\lambda(n) = O(\log n)$ under the assumption that $\mathsf{MINnKT} \in \mathsf{BPP}$, as desired.[6]

Unfortunately, it is not clear how to prove Equation (6) exactly. A key difficulty is that in many arguments involving meta-complexity, we typically make use of some form of time-bounded symmetry of information (SoI). However, for the highly efficient bounds we aim to achieve, we need symmetry of information with an *additive* polynomial overhead in the running time of the form $t + p(n)$, as opposed to an overhead of the form $\mathsf{poly}(t)$ (see, e.g., [18, 8, 20] and the weak symmetry of information for pK^{t} from [23]). It is unclear to us how to establish such a result.

[5] This idea is not new and a similar trick was employed by [15] and subsequent papers. A key difference is that we are able to explore *additive* overheads in the running time, as opposed to polynomial and quasi-linear overheads.

[6] The benefits of obtaining improved bounds on the running time in applications of time-bounded Kolmogorov complexity have also been explored in [7].

To illustrate the issue, let us briefly recall the approach of [18, 8, 20], which shows symmetry of information for pK^t under the assumption that MINKT is easy. Concretely,

$$\mathsf{pK}^t(x,y) \gtrsim \mathsf{pK}^{\mathsf{poly}(t)}(x) + \mathsf{pK}^{\mathsf{poly}(t)}(y \mid x).$$

Note the $\mathsf{poly}(t)$ time bound on the right-hand side of the inequality, particularly for the term $\mathsf{pK}^{\mathsf{poly}(t)}(x)$.

The argument compares the following three distributions over strings:

$$\mathcal{D}_1 := \mathsf{DP}_k(x,\mathcal{U}_{nk}) \circ \mathsf{DP}_{k'}(y,\mathcal{U}_{nk'}) \circ \mathcal{U}_t,$$
$$\mathcal{D}_2 := \mathsf{DP}_k(x,\mathcal{U}_{nk}) \circ \mathcal{U}_{nk'+k'} \circ \mathcal{U}_t,$$
$$\mathcal{D}_3 := \mathcal{U}_{nk+k} \circ \mathcal{U}_{nk'+k'} \circ \mathcal{U}_t,$$

where $\mathsf{DP}_k \colon \{0,1\}^n \times \{0,1\}^{nk} \to \{0,1\}^{nk+k}$ is the (k-wise) direct product generator [15] (see Definition 21), and $\mathcal{U}_m$ denotes the uniform distribution over $\{0,1\}^m$. Roughly speaking, if the K^t-complexities of these distributions are close, then we can upper bound the K^t-complexity of $\mathcal{D}_1$ by $\mathsf{pK}^t(x,y)$ plus $nk + nk' + t$, and we can lower bound the K^t-complexity of $\mathcal{D}_3$ by $nk + nk' + t + k + k'$, so it follows that

$$\mathsf{pK}^t(x,y) \gtrsim k + k'.$$

(We omit the details here; see [20, Appendix A] for a full exposition.)

A useful property of the direct product generator is that if one can distinguish its output instantiated with a string x, namely $\mathsf{DP}_k(x,\mathcal{U}_{nk})$, from the uniform distribution $\mathcal{U}_{nk+k}$, then one can efficiently *reconstruct* x with only k bits of advice. By choosing $k \approx \mathsf{pK}^{\mathsf{poly}(t)}(x)$ and $k' \approx \mathsf{pK}^{\mathsf{poly}(t)}(y \mid x)$ for the above distributions, one can argue that, with high probability, the K^t-complexities of these distributions are indeed close. Otherwise, an efficient algorithm for MINKT would serve as a distinguisher, and the reconstruction property of the direct product generator would yield either $\mathsf{pK}^{\mathsf{poly}(t)}(x) \lesssim k$ or $\mathsf{pK}^{\mathsf{poly}(t)}(y \mid x) \lesssim k'$, contradicting the definitions of k and k'. Note that, since the reconstruction algorithm need to invoke MINKT, which runs in time $\mathsf{poly}(t)$ by our assumption, the $\mathsf{poly}(t)$ time bound in the choice $k \approx \mathsf{pK}^{\mathsf{poly}(t)}(x)$ seems unavoidable in this argument.

By adapting the above proof of symmetry of information with the assumption that MINnKT is easy, one can show that $\mathsf{pnK}^t(x,y) \gtrsim \mathsf{pK}^{\mathsf{poly}(t)}(x) + \mathsf{pK}^{\mathsf{poly}(t)}(y \mid x)$, which can then be used to obtain a weaker version of Equation (6) (see also Lemma 52):

$$\mathsf{MINnKT} \in \mathsf{BPP} \implies \mathsf{pK}^{\mathsf{poly}(t)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + \mathsf{pK}^t(x) - \mathsf{pK}^{\mathsf{poly}(t)}(x) + O(\log n).$$

However, the polynomial time overhead remains inherent.

A potentially useful observation in this context is that, instead of the standard formulation of SoI, it is possible to work directly with a weaker variant – referred to as weak SoI – in which the SoI inequality holds for most choices of one of the strings. This approach can be used to avoid the overhead associated with the direct-product reconstruction of x. That said, existing proofs of weak SoI still seem to suffer from essentially the same overhead issue, typically stemming from the use of coding theorems or related arguments (see [23]). Moreover, even an overhead of the form $O(t)$ is insufficient in our setting, which makes it particularly challenging.

Crucially, by defining a new notion of time-bounded Kolmogorov complexity, we manage to obtain an SoI-style result without incurring a large overhead in the time bound, which suffices for our purpose.

4. A new measure of complexity that achieves additive time overhead. We now give a simplified overview of our new approach, which overcomes this barrier and yields only an additive $\mathsf{poly}(n)$ overhead in the time bound. The key idea is to introduce a new complexity measure called ℓ-nK^t. Given a time bound t and a parameter $\ell \in \mathbb{N}$, we define ℓ-$\mathsf{nK}^t(x)$ as

$$\ell\text{-}\mathsf{nK}^t(x) = \min\ s \text{ s.t. } \Pr_{r\sim\mathcal{U}_\ell}\left[\mathsf{nK}^t(x\circ r) \le s+\ell\right] \ge \frac{2}{3},$$

where $x \circ r$ denotes the concatenation of the strings x and r. In a sense, ℓ-$\mathsf{nK}^t(x)$ is related to $\mathsf{pnK}^t(x)$: while $\mathsf{pnK}^t(x)$ captures the nK^t-complexity of x *conditioned* on a random string, ℓ-$\mathsf{nK}^t(x)$ captures the nK^t-complexity of x *concatenated* with a random string of length ℓ.

First, one aspect of our new approach that helps avoid the undesired polynomial overhead in the time bound is that it avoids invoking the direct-product reconstruction of x. More specifically, we define two distributions of strings:

$$\mathcal{D}_1 := x \circ \mathsf{DP}_k(y,\mathcal{U}_{nk}) \circ \mathcal{U}_\ell,$$
$$\mathcal{D}_2 := x \circ \mathcal{U}_{nk+k} \circ \mathcal{U}_\ell,$$

where $\ell \ge \mathsf{poly}(n)$ is a parameter smaller than t. Note that, unlike in the earlier approach, we do not apply the direct product generator to x here.

Next, by unpacking the definition of ℓ-nK, we can upper bound the nK^t-complexity of the first distribution as follows:

$$\mathsf{nK}^t(x \circ \mathsf{DP}_k(y,\mathcal{U}_{nk}) \circ \mathcal{U}_\ell) \lesssim (\ell+nk)\text{-}\mathsf{nK}^{t-\mathsf{poly}(n)}(x\circ y) + \ell + nk.$$

Moreover, for the second distribution, we are able to show that, with high probability,

$$\mathsf{nK}^t\big(x \circ \mathcal{U}_{nk+k} \circ \mathcal{U}_\ell\big) \gtrsim (\ell+nk+k)\text{-}\mathsf{nK}^t(x) + \ell + nk + k.$$

(The formal statement appears in Lemma 43.[7]) Notably, the time bound appearing on the right-hand side remains t. In particular, this inequality can be viewed as a form of weak SoI that incurs no overhead in the time bound. This is essentially the key benefit of working with the more flexible notion of ℓ-nK^t, which allows us to avoid the large overhead present in traditional time-bounded SoI results.

Now, by choosing

$$k \approx (\ell+nk)\text{-}\mathsf{nK}^{t-\mathsf{poly}(n)}(x\circ y) - (\ell+nk+k)\text{-}\mathsf{nK}^t(x),$$

the algorithm for MINnKT distinguishes between these two distributions.[8] As a result, given x, we can distinguish $\mathsf{DP}_k(y,\mathcal{U}_{nk})$ from $\mathcal{U}_{nk+k}$. Hence, by the reconstruction property of the direct product generator, we obtain

$$\begin{aligned}\mathsf{pK}^{\mathsf{poly}(t)}(y \mid x) &\lesssim (\ell+nk)\text{-}\mathsf{nK}^{t-\mathsf{poly}(n)}(x\circ y) - (\ell+nk+k)\text{-}\mathsf{nK}^t(x)\\ &\lesssim \mathsf{pnK}^{\mathsf{poly}(n)}(y \mid x) + (\ell+nk)\text{-}\mathsf{nK}^{t-\mathsf{poly}(n)}(x) - (\ell+nk+k)\text{-}\mathsf{nK}^t(x),\end{aligned}$$

[7] Also, in the formal proof, ℓ-nK is defined using different probability parameters rather than the fixed value $2/3$, which is omitted here for simplicity. The probability parameters in the two bounds differ by $1/\mathsf{poly}(n)$.

[8] For simplicity, k appears on the right-hand side of the definition here, so it is not immediately clear that such a k exists. In the actual proof, by observing that ℓ-nK^t is roughly non-increasing in ℓ, choosing $k \approx \ell\text{-}\mathsf{nK}^{t-\mathsf{poly}(n)}(x,y) - (\ell+\mathsf{poly}(n))\text{-}\mathsf{nK}^t(x)$ suffices.

where the second inequality uses the fact that, to print x, y, and a random string, it suffices to first print x and the random string, and then print y given x and the random string; under our assumption on the computation model (see the discussion below), these two programs can be composed efficiently. Note that the difference between $(\ell + nk)$-$\mathsf{nK}^{t-\mathsf{poly}(n)}(x)$ and $(\ell + nk + k)$-$\mathsf{nK}^{t}(x)$ is only additive $\mathsf{poly}(n)$, both in the time bound and in the length of the random string.[9] Thus, we can apply the telescoping trick to average out their difference, as explained earlier.

Theorem 1 and Theorem 9 hold with respect to any computational model that behaves nicely with respect to the composition of programs. This is sometimes implicitly assumed in the literature on (time-bounded) Kolmogorov complexity. For instance, it is often used that if a function f on an input x is computed by a program Π_f in time t_f, and a function g on an input $y = f(x)$ is computed by a program Π_g in time t_g, then $g(f(x))$ can be computed in time $t_f + t_g +$ "lower order terms" by a program of description length $|\Pi_f| + |\Pi_g| +$"lower order terms".[10] Since our results explore a novel regime of parameters and depend in an important way on claims of this form, for precision and clarity of the presentation, we explicitly postulate in Section 2.2 the necessary properties of the computational model and encodings.

2 Preliminaries

2.1 Basic Notation

We use ϵ to denote the empty string. The length of a string x is denoted $|x|$. Given strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$, we let $x \circ y \in \{0,1\}^{n+m}$ denote their concatenation.

We write $x \sim \mathcal{D}$ to denote that x is sampled according to the distribution $\mathcal{D}$. For an element a in the support of $\mathcal{D}$, we let $\mathcal{D}(a)$ denote its probability. We use $\mathcal{U}_\ell$ to denote the uniform distribution over $\{0,1\}^\ell$.

2.2 Strongly Efficient Computational Models

Even though all standard machine models (single-tape, multi-tape, random-access, etc.) are equivalent up to a polynomial difference in running time, to prove Theorem 1 and Theorem 9 we require a computational model and a corresponding encoding of programs that behave well with respect to the time and description length overhead when composing computations.

In this section, we introduce in a semi-formal way the notion of *strongly efficient* computational model. This is sufficient for the purpose explained above. While the properties that we require are fairly natural and intuitively should hold for any carefully designed computational model and encoding, for concreteness we describe in Appendix A an explicit model that satisfies the relevant properties.

Strongly Efficient Computational Models. We assume an underlying machine model (e.g, Turing machines, RAM, etc.). A *program* Π is a partial function mapping three strings (corresponding to input, non-determinism, and randomness, respectively) to another string. Unless stated otherwise, we use r to denote the string corresponding to the randomness, and

[9] Also the difference in the probability parameters is only $1/\mathsf{poly}(n)$.

[10] As an illustrative example, think about the necessary properties required to establish the easy direction of symmetry information, which states that $\mathsf{K}^t(x,y) \leq \mathsf{K}^{t_1}(x) + \mathsf{K}^{t_2}(y \mid x) + O(\log(|x| + |y|))$ for $t \approx t_1 + t_2$.

w to denote the string corresponding to the non-determinism. A program has two attributes: a non-negative integer *time* (think of the program computing over the empty string ϵ or over an explicitly described input), and a binary string *description*. The description of a program Π uniquely specifies the program. We abuse notation in that Π can refer to either the function or the description depending on the context. We define the *size* of a program Π to be the length of its description, written as $|\Pi|$.

We say that a program Π computes $(x;r) \to y$ *non-deterministically* in time t, if

- $\forall w \in \{0,1\}^t$, $\Pi(x;w;r)$ outputs either y or $\perp$ in time t, and
- $\exists w \in \{0,1\}^t$, $\Pi(x;w;r)$ outputs y in time t.

If Π has size s, we say $(x;r) \to y$ can be computed non-deterministically in time t by a program of size s, or in short, $(x;r) \to y$ in non-deterministic time t and size s. When $r = \epsilon$, we may omit it and use $x \to y$ as short for $(x;\epsilon) \to y$.[11]

We are ready to introduce the concept of strongly efficient computational model.

▶ **Definition 12** (Strongly Efficient Computational Model)**.** *We say that a computational model and corresponding program encoding are* strongly efficient *if the following properties hold:*[12]

(P1) *Efficient Simulation: The computational model can be efficiently simulated by Turing machines. More precisely, there exists a polynomial p and a Turing machine M with four input tapes and one output tape, such that for any program Π, if $\Pi(x;w;r) = y$ in time t, then $M(\Pi;x;w;r)$ outputs y in $p(t + |\Pi|)$ steps.*

(P2) *Efficient Universality: The computational model can efficiently simulate Turing machines. More precisely, for every Turing machine M with four input tapes and one output tape, there is a polynomial p_M such that if $M(\alpha;x;w;r)$ outputs y in t steps, then there exists a program Π of size at most $|\alpha| + \log p_M(|\alpha|)$ such that $\Pi(x;w;r) = y$ in time $p_M(t)$.*

(P3) *Randomness and Non-Determinism Extension: For any program Π, suppose $\Pi(x;w;r) = y$ in time t. If $|w| \geq t$, then for any $w' \in \{0,1\}^*$, $\Pi(x;w \circ w';r) = y$ in time t. Similarly, if $|r| \geq t$, then for any $r' \in \{0,1\}^*$, $\Pi(x;w;r \circ r') = y$ in time t.*

(P4) *Efficient Composition: There exists a polynomial p such that the following holds. For any $x, y, z, r \in \{0,1\}^*$, if $(x;r) \to y$ can be computed non-deterministically in time t_1 by a program of size s_1, and $y \to z$ can be computed non-deterministically in time t_2 by a program of size s_2, then $(x;r) \to z$ can be computed non-deterministically in time $t_1 + p(t_2) + p(|x| + |y| + |z|)$ by a program of size $s_1 + s_2 + \log p(|x| + |y| + |z|)$.*

As alluded to above, for completeness, in Appendix A we provide an example of a computational model that satisfies the aforementioned properties.

How is this different from other papers in meta-complexity/time-bounded Kolmogorov complexity? First, we stress that properties (P1), (P2), (P3), and (P4) stated above are not stringent. This seems clear in the case of (P1), (P2), and (P3), so we comment on the case of (P4). In all our proofs, we have $|x|, |y|, |z| \leq \mathsf{poly}(n)$ and $t_1, t_2 \leq 2^n$ (indeed t_1 and t_2 are large enough polynomials in n in our arguments), where n is the relevant input length. Moreover, since under our definition the time-bounded Kolmogorov complexity of every string of length n is at most $n + O(\log n)$,[13] (P4) is actually weaker than the requirement that under

[11] Note that, by design, the aforementioned notation using arrows coincide with the notion of conditional nK, which is formally introduced in Section 2.3.

[12] In fact, our results only need relaxed forms of these properties. However, we describe the properties in this way because they are natural and simplify the presentation.

[13] This bound easily follows from (P2).

composition the final running time is at most $t_1 + t_2 + \mathsf{poly}(|x| + |y| + |z| + \log t_1 + \log t_2)$ and the final encoding length is at most $s_1 + s_2 + O(\log(|x| + |y| + |z| + s_1 + s_2))$. We believe these are widely adopted conventions in the literature (even if implicit in proofs), which we make explicit in our exposition. (Formally, in (P4) we even allow the second computation to run in time $\mathsf{poly}(t_2)$ after composition, which is sufficient in our proofs. For this reason, one can think of it as a "weak" form of efficient composition with respect to the running time.)

The main difference in our case compared with some other works is that in order to have precise control over running times, we measure time complexity as the number of steps in the execution of a program Π, as opposed to the number of steps that a universal machine U takes to simulate Π for t steps (which could be for instance of order $t \cdot \mathsf{poly}(\log t)$). This distinction is almost always irrelevant, and perhaps for this reason it is often not explicitly noted in the literature,[14] but it is relevant in our arguments.

Of course, if a program Π expects a *description* of a program Π' as an input string, and Π' runs in time t', then there is a time overhead during the simulation of Π' by Π. This is also the reason why we developed the notion of ℓ-$\mathsf{nK}^t(x)$ complexity discussed in Section 1.3, as it avoids the use of weak symmetry of information for the standard formulation of time-bounded Kolmogorov complexity, whose known proofs lead to a simulation overhead. (Note that, under (P4), there will be almost no overhead when composing two explicit programs, since in this case the programs are directly *executed* instead of *simulated*.)

Finally, we are not aware of results from the literature that would not remain valid under the conventions adopted above.

2.3 Time-Bounded Kolmogorov Complexity

▶ **Definition 13** (K^t). *For $x, y \in \{0,1\}^*$, $t \in \mathbb{N}$, and an oracle $\mathcal{O}$, we define the* t-time-bounded Kolmogorov complexity of x conditioning on y and given oracle $\mathcal{O}$ *as*

$$\mathsf{K}^{t,\mathcal{O}}(x \mid y) := \min_{\Pi \in \{0,1\}^*} \left\{ |\Pi| \;\middle|\; \Pi^{\mathcal{O}}(y; \epsilon; \epsilon) \text{ outputs } x \text{ within } t \text{ steps} \right\}.$$

We can consider the case where the oracle calls are *non-adaptive*. In this case, we will write $\mathsf{K}^{t,\|\mathcal{O}}(x \mid y)$.

Next, we will define other variants of the time-bounded Kolmogorov complexity measure. For simplicity, we will define only the basic versions without oracles. These can easily be generalized to settings where an oracle is present.

▶ **Definition 14** (pK^t). *For $x, y \in \{0,1\}^*$, $\lambda \in [0,1]$, $t \in \mathbb{N}$, the* probabilistic t-time-bounded Kolmogorov complexity of x conditioning on y *is defined as*

$$\mathsf{pK}^t_\lambda(x \mid y) := \min \left\{ s \in \mathbb{N} \;\middle|\; \Pr_{r \sim \{0,1\}^t} [\exists \Pi \in \{0,1\}^s \text{ s.t. } \Pi(y; \epsilon; r) \text{ outputs } x \text{ within } t \text{ steps}] \geq \lambda \right\}.$$

We define $\mathsf{pK}^t_\lambda(x)$ *as* $\mathsf{pK}^t_\lambda(x \mid \epsilon)$. *We omit the subscript* λ *when* $\lambda = 2/3$.

▶ **Definition 15** (nK^t). *For $x, y, r \in \{0,1\}^*$ and $t \in \mathbb{N}$, the* nondeterministic t-time-bounded Kolmogorov complexity of x conditioning on $(y; r)$ *is defined as*

$$\mathsf{nK}^t(x \mid y; r) : \min_{\Pi \in \{0,1\}^*} \left\{ |\Pi| \;\middle|\; \begin{array}{l} \forall w \in \{0,1\}^t, \Pi(y; w; r) \text{ outputs } x \text{ or } \perp \text{ in } t \text{ steps}; \\ \exists w \in \{0,1\}^t, \Pi(y; w; r) \text{ outputs } x \text{ in } t \text{ steps} \end{array} \right\}.$$

We define $\mathsf{nK}^t(x \mid y)$ *as* $\mathsf{nK}^t(x \mid y; \epsilon)$, *and* $\mathsf{nK}^t(x)$ *as* $\mathsf{nK}^t(x \mid \epsilon)$.

[14] We observe that some references do adopt a similar convention when defining time-bounded Kolmogorov complexity by using "time" as the number of steps of the program instead of the universal machine, such as [31, Section 2.3].

▶ **Definition 16** (rnK^t). *For $x, y \in \{0,1\}^*$, $\lambda \in [0,1]$ and $t \in \mathbb{N}$, the* randomized nondeterministic t-time-bounded time-bounded Kolmogorov complexity of x conditioning on y *is defined as*

$$\mathsf{rnK}^t_\lambda(x \mid y) : \min_{\Pi \in \{0,1\}^*} \left\{ |\Pi| \;\middle|\; \Pr_{r \sim \{0,1\}^t} \begin{bmatrix} \forall w \in \{0,1\}^t, \Pi(y; w; r) \text{ outputs } x \text{ or } \perp \text{ in } t \text{ steps}; \\ \exists w \in \{0,1\}^t, \Pi(y; w; r) \text{ outputs } x \text{ in } t \text{ steps} \end{bmatrix} \geq \lambda \right\}.$$

We define $\mathsf{rnK}^t_\lambda(x)$ as $\mathsf{rnK}^t_\lambda(x \mid \epsilon)$. We omit the subscript λ when $\lambda = 2/3$.

▶ **Definition 17** (pnK^t). *For $x, y \in \{0,1\}^*$, $\lambda \in [0,1]$ and $t \in \mathbb{N}$, the* probabilistic nondeterministic t-time-bounded time-bounded Kolmogorov complexity of x conditioning on y *is defined as*

$$\mathsf{pnK}^t_\lambda(x \mid y) := \min \left\{ s \in \mathbb{N} \;\middle|\; \Pr_{r \sim \{0,1\}^t} \begin{bmatrix} \exists \Pi \in \{0,1\}^{\leq s} \text{ such that the following conditions hold:} \\ \forall w \in \{0,1\}^t, \Pi(y; w; r) \text{ outputs } x \text{ or } \perp \text{ in } t \text{ steps}; \\ \exists w \in \{0,1\}^t, \Pi(y; w; r) \text{ outputs } x \text{ in } t \text{ steps} \end{bmatrix} \geq \lambda \right\}.$$

Equivalently,

$$\mathsf{pnK}^t_\lambda(x \mid y) := \min \left\{ s \in \mathbb{N} \;\middle|\; \Pr_{r \sim \{0,1\}^t} \left[\mathsf{nK}^t(x \mid y; r) \leq s \right] \geq \lambda \right\}.$$

We define $\mathsf{pnK}^t_\lambda(x)$ as $\mathsf{pnK}^t_\lambda(x \mid \epsilon)$. We omit the subscript λ when $\lambda = 2/3$.

In the remainder of this subsection, we state some useful tools about Kolmogorov complexity.

▶ **Lemma 18** (Following [9, Lemma 18]). *There exists a universal constant $c > 0$ such that for any $x \in \{0,1\}^*$ and time bound $t \in \mathbb{N}$, the following hold.*

- $\mathsf{K}(x) \leq \mathsf{pK}^t(x) + c \cdot \log t$.
- $\mathsf{K}(x) \leq \mathsf{pnK}^t(x) + c \cdot \log t$.

▶ **Lemma 19** (See [19, Lemma 9]). *There exists a universal constant $b > 0$ such that for any distribution family $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$, where each $\mathcal{D}_n$ is over $\{0,1\}^n$, and $\gamma \in \mathbb{N}$,*

$$\Pr_{x \sim \mathcal{D}_n} \left[\mathsf{K}(x) < \log \frac{1}{\mathcal{D}_n(x)} - \gamma \right] < \frac{n^b}{2^\gamma}.$$

▶ **Theorem 20** (Efficient Coding Theorem [33]). *For every polynomial-time samplable distribution family $\{\mathcal{D}_n\}_{n \in \mathbb{N}}$, where each $\mathcal{D}_n$ is supported over $\{0,1\}^n$, there exists a polynomial p such that for every $x \in \mathsf{Support}(\mathcal{D}_n)$,*

$$\mathsf{pK}^{p(n)}(x) \leq \log \frac{1}{\mathcal{D}_n(x)} + \log p(n).$$

2.4 Direct Product Generator

For $x, y \in \{0,1\}^n$, we denote their inner product by $x \cdot y = \bigoplus_{i=1}^n x_i y_i$.

▶ **Definition 21** (Direct Product Generator (DPG) [15]). *For $n, k \in \mathbb{N}$, the k-wise direct product generator is the function*

$$\mathsf{DP}_k \colon \{0,1\}^n \times \{0,1\}^{nk} \to \{0,1\}^{nk+k},$$

defined by

$$\mathsf{DP}_k(x, z_1 \circ \cdots \circ z_k) = (z_1 \circ \cdots \circ z_k \circ x \cdot z_1 \circ \cdots \circ x \cdot z_k).$$

The reconstruction lemma for the direct product generator states that, given access to an oracle distinguishing between $\mathsf{DP}_k(x,\mathcal{U}_{nk})$ and $\mathcal{U}_{nk+k}$, along with k bits of advice, we can recover x with $1/\mathsf{poly}(n)$ probability.

▶ **Lemma 22** (DPG Reconstruction [15]). *There exists a pair of algorithms* $\mathsf{Recon}^{\mathcal{O}}$ *and* Ad*, and a polynomial* p*, such that for any* $n,m,k\in\mathbb{N}^+$*, we have the following.*

1. *For any* $r\in\{0,1\}^{p(nmk)}$*, any* $a\in\{0,1\}^k$*, and any function* $D\colon\{0,1\}^{nk+k}\to\{0,1\}$*,* $\mathsf{Recon}^D(1^n,1^m,1^k,r,a)$ *runs in at most* $p(nmk)$ *steps, makes at most* $p(nmk)$ *queries to* D*, and outputs a string* $x\in\{0,1\}^n$*. Furthermore, the queries to* D *are non-adaptive; that is,* Recon *generates the full list of positions to query before receiving any answers.*
2. *For any* $x\in\{0,1\}^n$ *and any* $r\in\{0,1\}^{p(nmk)}$*,* $\mathsf{Ad}(1^n,1^m,1^k,x,r)$ *runs in at most* $p(nmk)$ *steps and outputs a string* $a\in\{0,1\}^k$*.*
3. *For any* $x\in\{0,1\}^n$ *and any function* D *that* $(1/m)$*-distinguishes between* $\mathsf{DP}_k(x,\mathcal{U}_{nk})$ *and* $\mathcal{U}_{nk+k}$*, we have*

$$\Pr_{r\sim\mathcal{U}_{p(nmk)}}\Big[\mathsf{Recon}^D(1^n,1^m,1^k,r,\mathsf{Ad}(1^n,1^m,1^k,x,r))=x\Big]\geq\frac{1}{p(nmk)}.$$

Proof Sketch. The proof works by first converting the distinguisher into a next-bit predictor using Yao's hybrid argument [35]. It then employs the Goldreich–Levin local list-decoding algorithm for the Hadamard code to obtain a polynomial-sized list containing x. We observe that both Yao's argument and the Goldreich–Levin algorithm can be implemented using only non-adaptive queries (see, e.g., [2, Sections 9.3.1 and 9.3.2]). Consequently, our final reconstruction algorithm also relies solely on non-adaptive queries. ◀

▶ **Lemma 23** (pK^t Reconstruction Lemma). *There is a polynomial* p_{DP} *such that the following holds. For every* $n,m,k\in\mathbb{N}^+$*,* $x\in\{0,1\}^n$*, let* D *be a function that* $(1/m)$*-distinguishes* $\mathsf{DP}_k(x;\mathcal{U}_{nk})$ *from* $\mathcal{U}_{nk+k}$*. Then*

$$\mathsf{pK}^{p_{\mathsf{DP}}(nmk),\|D}(x)\leq k+\log p_{\mathsf{DP}}(nmk).$$

Proof. The lemma follows easily from the reconstruction property of the direct product generator (Lemma 22) and the success amplification property of pK^t (Lemma 29). ◀

▶ **Corollary 24** (pK^t Reconstruction with a Randomized Distinguisher). *There is a polynomial* p *such that the following holds. For every* $n,m,k,t\in\mathbb{N}^+$*, and any string* $x\in\{0,1\}^n$*, let* D *be a function* $D\colon\{0,1\}^t\times\{0,1\}^{nk+k}\to\{0,1\}$ *satisfying*

$$\mathop{\mathbb{E}}_{\substack{w\sim\mathcal{U}_t\\ r\sim\mathcal{U}_{nk}\\ r'\sim\mathcal{U}_{nk+k}}}\Big[D(w,\mathsf{DP}_k(x;r))-D(w,r')\Big]\geq\frac{1}{m}.$$

Then we have

$$\mathsf{pK}^{p(nmkt),\|D}(x)\leq k+\log p(nmkt).$$

Proof. Using Markov's inequality, one can argue that with probability at least $\frac{1}{2m}$ over the $w\sim\mathcal{U}_t$, $D(w,\cdot)$ $\frac{1}{2m}$-distinguishes $\mathsf{DP}_k(x;\mathcal{U}_{nk})$ from $\mathcal{U}_{nk+k}$. Therefore, we can first sample $w\sim\mathcal{U}_t$, then make all queries to $D(w,\cdot)$. By Lemma 23, there exists some large enough polynomial p_0 such that

$$\mathsf{pK}^{p_0(nmkt),\|D}_{1/(3m)}(x)\leq k+\log p_0(nmkt).$$

By Lemma 29, we can then amplify the success probability to be 2/3. Thus, when p is a large enough polynomial, we obtain

$$\mathsf{pK}^{p(nmkt),\|D}(x) \leq k + \log p(nmkt),$$

as desired. ◀

2.5 Universal Time-Bounded Sampler

▶ **Definition 25** (Universal Time-Bounded Sampler). *Let $m, t \in \mathbb{N}$ and $x \in \{0,1\}^*$. The universal sampler $\mathsf{USamp}(1^m, 1^t, x)$ does the following.*

1. *Pick a uniformly random $k \sim [O(m)]$.*
2. *Pick a uniformly random $r \sim \{0,1\}^t$.*
3. *Pick a uniformly random $\Pi \sim \{0,1\}^k$.*
4. *Let y be the output of $\Pi(x; r)$ after running for t steps.*
5. *Output y.*

Note that USamp runs in polynomial time. The following proposition follows easily from the definitions of pK^t and USamp.

▶ **Proposition 26.** *For every $m, t, k \in \mathbb{N}$, $y \in \{0,1\}^m$ and $x \in \{0,1\}^*$, if $\mathsf{pK}^t(y \mid x) \leq k$, then $\mathsf{USamp}(1^m, 1^t, x)$ outputs y with probability $\Omega\big(1/(m \cdot 2^k)\big)$, where USamp is the universal sampler defined in Definition 25.*

2.6 Hash Functions

▶ **Definition 27** (Pairwise-Independent Hash Family). *For any $n, m \in \mathbb{N}$, a set of functions $H \subseteq \{0,1\}^n \to \{0,1\}^m$ is called a* pairwise-independent hash family *if for any $x, y \in \{0,1\}^n$ such that $x \neq y$ and any $r, s \in \{0,1\}^m$,*

$$\Pr_{h \sim H}[h(x) = r \wedge h(y) = s] = 2^{-2m}.$$

▶ **Proposition 28.** *There exists a polynomial p and a randomized algorithm A such that, for every $n, m \in \mathbb{N}$ and every random string $r \in \{0,1\}^{p(nm)}$, $A(1^n, 1^m, r, \cdot)$ computes a function $\{0,1\}^n \to \{0,1\}^m$ in time $p(nm)$. Furthermore, $H := \big\{A(1^n, 1^m, r, \cdot) \mid r \in \{0,1\}^{p(nm)}\big\}$ is a pairwise-independent hash family.*

Proof. We can construct pairwise-independent hash families using finite fields $\mathrm{GF}(2^n)$. When $n = m$, for any $x \in \{0,1\}^n$, we sample $a, b \sim \mathcal{U}_n$, and compute $h_{a,b}(x) = a \cdot x + b$, where addition and multiplication are done over $\mathsf{GF}(2^n)$. Such a computation can be done in time $\mathsf{poly}(n)$, see e.g. [2, Appendix A.4]. Then for every $y \in \{0,1\}^n \setminus \{x\}$ and any $r, s \in \{0,1\}^n$, there exists exactly one pair of strings (a, b) satisfying $a \cdot x + b = r$ and $a \cdot y + b = s$. In other words, $\{h_{a,b} \mid a, b \in \{0,1\}^n\}$ is a pairwise-independent hash family. For $m < n$, we can simply discard the first $(n - m)$ bits of $h_{a,b}(x)$. For $m > n$, we can define $h'_{a,b}(x) = a \cdot (0^{m-n} \circ x) + b$. ◀

2.7 Average-Case Complexity

Recall that a pair $(L, \mathcal{D})$ is a *distributional problem* if $L \subseteq \{0,1\}^*$ and $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$ is a distribution family, where each $\mathcal{D}_n$ is a distribution over $\{0,1\}^*$.

We let DistNP denote the set of distributional problems $(L, \mathcal{D})$ such that $L \in \mathsf{NP}$ and $\mathcal{D}$ is polynomial-time samplable. Here, a distribution family $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$ is *polynomial-time samplable* if there exists a randomized polynomial-time algorithm A such that for every $n \geq 1$, the output of $A(1^n)$ is distributed according to $\mathcal{D}_n$.

A distributional problem $(L, \mathcal{D})$ is said to admit a (errorless) *heuristic scheme* if there exists a probabilistic polynomial-time algorithm A such that the following holds for every $n, k \in \mathbb{N}$:

1. For every $x \in \mathsf{Support}(\mathcal{D}_n)$,

$$\Pr_A \left[A(x, 1^n, 1^k) \in \{L(x), \perp\}\right] \geq \frac{4}{5},$$

2. and

$$\mathbf{Pr}_{x \sim \mathcal{D}_n} \left[\Pr_A[A(x, 1^n, 1^k) = \perp] < \frac{1}{5}\right] \geq 1 - \frac{1}{k}.$$

We let AvgBPP denote the set of distributional problems that admit a heuristic scheme. For more information about average-case complexity, we refer to [3].

Also, by slightly abusing notation, we say that MINnKT is average-case easy under the uniform distribution, or $(\mathsf{MINnKT}, \mathcal{U}) \in \mathsf{AvgBPP}$, if the following holds.

There exist a polynomial-time algorithm A and a polynomial ρ such that the following holds for all $n, s, k \in \mathbb{N}$, and all $t \geq \rho(n)$.

1. For all $x \in \{0,1\}^n$,

$$\mathbf{Pr}_A \left[A(x, 1^s, 1^t, 1^k) \in \{\mathsf{MINnKT}(x, 1^s, 1^t), \perp\}\right] \geq \frac{4}{5},$$

2. and

$$\mathbf{Pr}_{x \sim \{0,1\}^n} \left[\mathbf{Pr}_A[A(x, 1^s, 1^t, 1^k) = \perp] < 1/5\right] \geq 1 - \frac{1}{k}.$$

3 Properties of pnK^t

3.1 Basic Properties of pnK^t

▶ **Lemma 29** (Success Amplification). *For any string $x \in \{0,1\}^n$, time bound $t \in \mathbb{N}$, any $0 < \alpha < \beta < 1$ and any oracle $\mathcal{O}$, we have*

$$\mathsf{pnK}_\beta^{O(kt),\mathcal{O}}(x) \leq \mathsf{pnK}_\alpha^{t,\mathcal{O}}(x) + O(\log k),$$

where $k := \left\lceil \frac{\log(1-\beta)}{\log(1-\alpha)} \right\rceil$. Moreover, the above also holds in the setting where the oracle access is non-adaptive. That is,

$$\mathsf{pnK}_\beta^{O(kt),\|\mathcal{O}}(x) \leq \mathsf{pnK}_\alpha^{t,\|\mathcal{O}}(x) + O(\log k).$$

Proof. Let $l := \mathsf{pnK}_\alpha^{t,\mathcal{O}}(x)$. We say that a string $r \in \{0,1\}^t$ is *good* if there exists a program $\Pi \in \{0,1\}^l$ such that

- For every $w \in \{0,1\}^t$, $\Pi^{\mathcal{O}}(w; r)$ outputs either x or $\perp$ in t steps;
- There exists some $w \in \{0,1\}^t$ such that $\Pi^{\mathcal{O}}(w; r)$ outputs x in t steps.

By the definition of pnK and l, we have

$$\mathbf{Pr}_{r \sim \{0,1\}^t}[r \text{ is good}] \geq \alpha.$$

We can then take $k = \left\lceil \frac{\log(1-\beta)}{\log(1-\alpha)} \right\rceil$ independent samples $r_1, \ldots, r_k$ from $\mathcal{U}_t$. The probability of getting at least one good sample among $r_1, \ldots, r_k$ is at least $1 - (1-\alpha)^k \geq \beta$. If we have a good sample, then we can store the index i such that r_i is good using $\lceil \log k \rceil$ bits,

and there exists a program Π such that $\Pi^{\mathcal{O}}(w; r_i)$ outputs x. Hence, we conclude that $\mathsf{pnK}_{\beta}^{O(kt),\mathcal{O}}(x) \leq l + O(\log k)$. Notice that if the original program makes parallel queries to $\mathcal{O}$, then so does the new program.

The above proof can be easily extended to the setting of non-adaptive oracle access, which shows the "moreover" part of the lemma. ◀

Since for any $0 < \alpha < 1$, $-1/\log(1-\alpha) \leq 1/\alpha$ holds, we have the following corollary:

▶ **Corollary 30.** *For any string $x \in \{0,1\}^n$, time bound $t \in \mathbb{N}$, any $0 < \alpha < 1$ and any oracle $\mathcal{O}$, we have*

$$\mathsf{pnK}^{O(2t/\alpha),\mathcal{O}}(x) \leq \mathsf{pnK}_{\alpha}^{t,\mathcal{O}}(x) + O(\log(1/\alpha)).$$

Moreover, the above also holds in the setting where the oracle access is non-adaptive, i.e.

$$\mathsf{pnK}^{O(2t/\alpha),\|\mathcal{O}}(x) \leq \mathsf{pnK}_{\alpha}^{t,\|\mathcal{O}}(x) + O(\log(1/\alpha)).$$

▶ **Lemma 31.** *For any string $x, y \in \{0,1\}^*$, time bound $t \in \mathbb{N}$, any oracle $\mathcal{O}$, we have*

$$\mathsf{pnK}^{\mathsf{poly}(t),\mathcal{O}}(x \mid y) \leq \mathsf{pK}^{t,\|\mathsf{NP}^{\mathcal{O}}}(x \mid y) + O(\log t).$$

Proof. Suppose $\mathsf{pK}^{t,\|\mathsf{NP}^{\mathcal{O}}}(x \mid y) = s$. Then, with probability at least $2/3$ over the randomness $r \sim \{0,1\}^t$, there exists a program of size s that, given r, y and *non-adaptive* access to some oracle $A \in \mathsf{NP}^{\mathcal{O}}$, runs in t steps and outputs x. Suppose A is the following oracle:

$$A(q) = 1 \iff \exists y \in \{0,1\}^{\mathsf{poly}(|q|)} \text{ such that } V^{\mathcal{O}}(q, w) = 1,$$

where V is some polynomial-time verifier. We will replace oracle access to A with access only to $\mathcal{O}$ by guessing the answers to the (non-adaptive) queries to A and verifying them using $\mathcal{O}$, with the help of an additional small amount of advice.

More specifically, fix a randomness r and a program Π (which may depend on r). Let $q_1, q_2, \ldots, q_\ell$, where $\ell \leq t$, be the (non-adaptive) queries made by $\Pi(r)$ to the oracle A. Also, let p be the number of positive answers to these queries, i.e., $p := \sum_{i=1}^{\ell} A(q_i)$. We then non-deterministically guess ℓ witnesses $(w_1, w_2, \ldots, w_\ell) =: y$ and count the number of indices i for which $V^{\mathcal{O}}(q_i, w_i) = 1$. Denote this number by u_w. If u_w is not equal to p, we output $\perp$. Otherwise, for each w_i that satisfies the verifier V, we set the answer to the query q_i to be 1, and we let the rest of the answers be 0. We then run $\Pi^A(r)$ while simulating the oracle A using the answers obtained as described above.

Note that, given the number p, which can be specified using $\lceil \log \ell \rceil \leq \lceil \log t \rceil$ bits, the above procedure can be implemented to run in time $\mathsf{poly}(t)$ with only oracle access to $\mathcal{O}$. Also, it is easy to observe that there exists some guess of $w = (w_1, w_2, \ldots, w_\ell)$ that satisfies $u_w = p$. Moreover, for any guess of w that satisfies $u_w = p$, the answers obtained are the correct answers to the queries $(q_1, q_2, \ldots, q_\ell)$ for A. It follows that, with high probability, there exists a program of size at most $s + O(\log t)$ that, given y and oracle access to $\mathcal{O}$, runs in time $\mathsf{poly}(t)$ and outputs x non-deterministically. ◀

3.2 Language Compression for pnK^t

3.2.1 Proof of Theorem 4

We provide two different proofs for the optimal language compression property of pnK^t. The first proof, given below, is based on an approach from [5] and relies on the direct product generator (Definition 21).

First Proof of Theorem 4. Let A_n be any non-empty subset of $\{0,1\}^n$. We set $k := \lceil \log |A_n| \rceil + 1$. We define

$$B_n := \left\{ y \in \{0,1\}^{nk+k} \mid \exists x \in A_n, z \in \{0,1\}^{nk} \text{ such that } y = \mathsf{DP}_k(x,z) \right\}.$$

By a simple counting argument, we have that $|B_n| \leq |A_n| \cdot 2^{nk}$. Also, by our definition of k, $|A_n| \leq 2^{k-1}$, and hence $|B_n| \leq \frac{1}{2} \cdot 2^{nk+k}$. By viewing $B_n \colon \{0,1\}^{nk+k} \to \{0,1\}$ as the characteristic function of the set B_n, we have

$$\mathbf{Pr}[B_n(\mathcal{U}_{nk+k}) = 1] \leq \frac{1}{2}.$$

On the other hand, from the definition of B_n, we have that for any $x \in A_n$,

$$\mathbf{Pr}[B_n(\mathsf{DP}_k(x, \mathcal{U}_{nk})) = 1] = 1.$$

Therefore, B_n $\frac{1}{2}$-distinguishes $\mathsf{DP}_k(x, \mathcal{U}_{nk})$ and $\mathcal{U}_{nk+k}$.

By the pK^t reconstruction lemma (Lemma 23), we get that there exists some polynomial p such that

$$\mathsf{pK}^{p(n),\|B_n}(x) \leq k + \log p(n). \tag{7}$$

Now, observe that the membership of B_n can be decided in NP^{A_n}. Indeed, given $y \in \{0,1\}^{nk+k}$, we can guess $x \in \{0,1\}^n$, $z \in \{0,1\}^{nk}$, and then check whether both $x \in A_n$ and $y = \mathsf{DP}_k(x,z)$. [15] Now, by applying Lemma 31 to Equation (7), we get that

$$\mathsf{pnK}^{\mathsf{poly}(n),A_n}(x) \leq k + O(\log n) = \lceil \log |A_n| \rceil + 1 + O(\log n).$$

It follows that there exists a polynomial p_{LC} such that

$$\mathsf{pnK}^{p_{\mathsf{LC}}(n),A_n}(x) \leq \log |A_n| + \log p_{\mathsf{LC}}(n),$$

as desired. ◀

Next, we give a second proof that relies on hash functions.

Second Proof of Theorem 4. Let A_n be any non-empty subset of $\{0,1\}^n$, and let x be any string in A_n. We set $k := \lceil \log |A_n| \rceil + 2$. Let $H \subseteq \{0,1\}^n \to \{0,1\}^k$ be a family of pairwise-independent hash functions. Since H is pairwise-independent, we have

$$\mathop{\mathbf{Pr}}_{h \sim H}[h(x) = h(y)] = 2^{-k}$$

for any $y \in A_n \setminus \{x\}$. Thus, by a union bound, we obtain

$$\mathop{\mathbf{Pr}}_{h \sim H}[\exists y \in A_n \setminus \{x\} \text{ such that } h(x) = h(y)] \leq 2^{-k} \cdot |A_n| \leq \frac{1}{4}. \tag{8}$$

[15] Strictly speaking, we cannot say that B_n is decided in NP^{A_n} since checking the membership of a given string in B_n requires knowing the numbers n and k. One way to address this issue is to consider the set $B := \left\{ (y, n', k') \mid |y| = n'k' + k' \text{ and } \exists x \in \{0,1\}^{n'}, z \in \{0,1\}^{n'k'} \text{ s.t. } x \in A_n \text{ and } y = \mathsf{DP}_{k'}(x,z) \right\}$. We can then replace the oracle B_n with B while replacing every query y with (y,n,k), where n and k are fixed integers as specified in the proof and can be hardcoded in the description. For simplicity of presentation, we ignore this technicality.

For $h \sim H$, we include $v := h(x)$ as part of the description used to output x non-deterministically. Note that v can be specified using $\log |A_n| + 2$ bits. We can then guess $x' \in \{0,1\}^n$ and check whether $h(x') = v$ and $x' \in A_n$. If so, we output x'; otherwise, we output $\perp$.

By Equation (8), with probability at least $3/4$, we sample a good h such that $h(x)$ is unique in A_n, ensuring that the above procedure outputs x non-deterministically. It follows that

$$\mathsf{pnK}^{p_{\mathsf{LC}}(n),A_n}(x) \leq \log |A_n| + \log p_{\mathsf{LC}}(n)$$

for a sufficiently large polynomial p_{LC}. ◀

3.2.2 Proof of Theorem 5

Before presenting the proof of Theorem 5, we need the following lemma.

▶ **Lemma 32.** *Let $m, t, \theta \in \mathbb{N}$, $A \subseteq \{0,1\}^m$, and $x \in \{0,1\}^*$. Suppose for every $y \in A$, it holds that*

$$\mathsf{pK}^t(y \mid x) \leq \log |A| + \theta.$$

Then $\mathsf{USamp}(1^m, 1^t, x)$ outputs some string in A with probability at least $\frac{1}{\mathsf{poly}(m) \cdot 2^\theta}$.

Proof. By Proposition 26, for *each* $y \in A$, $\mathsf{USamp}(1^m, 1^t, x)$ outputs y with probability at least

$$\frac{1}{O(m) \cdot 2^{\log |A| + \theta}} = \frac{1}{O(m) \cdot |A| \cdot 2^\theta}.$$

Hence the probability that $\mathsf{USamp}(1^m, 1^t, x)$ outputs *some* $y \in A$ is at least

$$|A| \cdot \frac{1}{O(m) \cdot |A| \cdot 2^\theta} \geq \frac{1}{O(m) \cdot 2^\theta},$$

as desired. ◀

Proof of Theorem 5. Suppose we have average-case pK^t language compression for P. We show how to solve any unary NP problem in randomized polynomial time.

Let $L \in \mathsf{Unary\text{-}NP}$. Without loss of generality, we assume that for every instance 1^n, the set of L-witnesses of 1^n (with respect to some fixed verifier) has a length of exactly m, where $m := m(n) = \mathsf{poly}(n)$. Let A denote the set of L-witnesses. Note that the membership of A is decidable in polynomial time. Then by assumption on average-case pK^t language compression, it holds that for every m, with probability at least $1/m^c$ over $y \sim A_m$,

$$\mathsf{pK}^{m^c}(x) \leq \log |A_m| + c \cdot \log m.$$

Let A'_m be the set of strings in A_m for which the above condition holds. Note that

$$|A'_m| \geq |A_m|/m^c.$$

Then for each $y \in A'_m$, we have

$$\begin{aligned} \mathsf{pK}^{n^c}(y) &\leq \log |A_m| + c \cdot \log m \\ &\leq \log |A'_m| + c \cdot \log m + c \cdot \log m \\ &= \log |A'_m| + 2c \cdot \log m. \end{aligned}$$

Now consider the algorithm that, given as input 1^n, simply runs $\mathsf{USamp}\big(1^{m(n)}, 1^{m(n)^c}, \epsilon\big)$. By Lemma 32, this algorithm outputs some string in A'_m, which consists of L-witnesses and is non-empty if $1^n \in L$, with probability at least $1/\mathsf{poly}(n)$. By standard amplification, this yields an efficient randomized algorithm for solving L with high probability. ◀

3.3 Conditional Coding for pnK^t: Proof of Theorem 6

Proof of Theorem 6. Let $\{\mathcal{D}_n\}$ be a polynomial-time samplable distribution family. Fix $n \in \mathbb{N}$ and $(x, y) \in \mathsf{Support}(\mathcal{D}_n)$.

Let $\mathcal{E} := \mathcal{D}_n(x \mid y)$. Define

$$B := \left\{ z \in \{0,1\}^n : \mathcal{D}_n(z \mid y) \geq \frac{\mathcal{E}}{16} \right\}.$$

Note that $|B| \leq 16/\mathcal{E}$. Now consider a family of pairwise-independent hash functions $H \subseteq \{0,1\}^n \to \{0,1\}^k$, where $k := \lceil \log \frac{1}{\mathcal{E}} \rceil + 7$. By a union bound, with probability at least $7/8$ over $h \sim H$, we have $h(x) \neq h(z)$ for any $z \in B \setminus \{x\}$. Hence, we can try to include $v := h(x)$ in our description for outputting x. Then we guess x' and output it if and only if $h(x') = v$.

In order for the above to work, we also need to make sure that the guessed string x' is not in B, since otherwise we could have $h(x') = h(x)$ but $x' \neq x$. Next, we describe how to exclude strings outside of B. The argument is similar to a construction from [10].

Let A be a sampler for $\{\mathcal{D}_n\}$ that has a running time $q(n)$, where q is some polynomial. For any $a, b \in \{0,1\}^n$, define $R_{a,b} := \{r \in \{0,1\}^{q(n)} : A(1^n; r) = (a, b)\}$. Now let $k' = \lfloor \log |R_{x,y}| \rfloor - 1$. Then we define another pairwise-independent hash family $H' \subseteq \{0,1\}^{q(n)} \to \{0,1\}^{k'}$. Then by Chebyshev's inequality,

$$\Pr_{h' \sim H'} \left[\exists r \in R_{x,y} \text{ such that } h'(r) = 0^{k'} \right] \geq \frac{1}{2}.$$

However, for any $z \notin B$, by definition, we have $|R_{z,y}| \leq |R_{x,y}|/16$, implying that $|R_{z,y}| \leq 2^{k'}/4$. Therefore, by a union bound, we have

$$\Pr_{h' \sim H'} \left[\exists r \in R_{z,y} \text{ such that } h'(r) = 0^{k'} \right] \leq \frac{1}{4}.$$

We can see that there is a gap between the cases of $z = x$ and of $z \notin B$, so our next step is to amplify this gap: We set $l = C \cdot n$ for some large enough constant C, and take $l = C \cdot n$ samples from H' to obtain $h'_1, \ldots, h'_l$, and use non-determinism to guess $r_1, \ldots, r_l \in \{0,1\}^{q(n)}$. We also use non-determinism to guess $z \in \{0,1\}^n$. Then we do the following checks. If any of these checks fail, halt and output $\perp$; otherwise, output z:

1. $A(1^n; r_1) = A(1^n; r_2) = \cdots = A(1^n; r_l) = (z, y)$.
2. The number of $i \in [l]$ such that $h'_i(r_i) = 0^{k'}$ is at least $l/3$.
3. $h(z)$ is the same as the string $v := h(x)$ specified in the description.

By the Chernoff bound and a union bound, when C is a large enough constant, with probability at least $7/8$ over $h'_1, \ldots, h'_l$, for any $z \notin B$, no sequence of $r_1, \ldots, r_l$ can pass the first and second checks. While with probability at least $1 - 2^{-n}$, there exists some sequence of $r_1, \ldots, r_l$ such that x can pass the check. Also, with probability at least $7/8$ over h, any $z \in B \setminus \{x\}$ cannot pass the third check. Therefore, by a union bound, with probability at least $1 - 1/8 - 1/8 - 2^{-n} \geq 2/3$, only x can pass the three checks. It is not hard to see that the entire reconstruction procedure runs in time polynomial in n, and we only need to specify $h(x)$ in this reconstruction algorithm, as well as the description of the sampler A. Hence, we conclude that there exists some polynomial p such that $\mathsf{pnK}^{p(n)}(x \mid y) \leq \log \frac{1}{\mathcal{D}_n(x|y)} + \log p(n)$. ◀

3.4 Symmetry of Information for pnK^t: Proofs of Theorem 7 and Theorem 8

In this section, we prove the results related to symmetry of information for pnK^t (Theorem 7 and Theorem 8).

Proof Sketch of Theorem 7. The proof follows a similar approach to that used in [8, 18] to establish symmetry of information for pK^t under the assumption that MINKT is easy (see also [20, Appendix A]). This approach employs the direct product generator and its reconstruction property (Lemmas 22 and 23) while using MINKT (which is in NP) as a distinguisher. In fact, by adapting this approach and using the fact that the reconstruction procedure of the direct product generator makes only non-adapative queries, one can show that

$$\mathsf{pK}^t(x,y) \geq \mathsf{pK}^{\mathsf{poly}(t),\|\mathsf{NP}}(y \mid x) + \mathsf{pK}^{\mathsf{poly}(t),\|\mathsf{NP}}(x) + O(\log t).$$

Now note that the semi-symmetry of information property for pnK^t follows by combining the above with Lemma 31. ◀

Proof of Theorem 8. Let $\mathcal{D} = \{\mathcal{D}_n\}$ be any polynomial-time samplable distribution family and q be any polynomial. Let $p$and r be sufficiently large polynomials specified later. Fix $n \in \mathbb{N}$ and $t \geq p(n)$.

By Theorem 7, we have that for every $x, y \in \{0,1\}^n$, we have

$$\mathsf{pK}^{r(n)}(x,y) \geq \mathsf{pnK}^{p_{\mathsf{Sol}}(r(n))}(y \mid x) + \mathsf{pnK}^{p_{\mathsf{Sol}}(r(n))}(x) - \log p_{\mathsf{Sol}}(r(n)), \tag{9}$$

where p_{Sol} is the polynomial in Theorem 7.

First, by the coding theorem for pK^t (Theorem 20) and by letting r be a sufficiently large polynomial, we have

$$\mathsf{pK}^{r(n)}(x,y) \leq \log \frac{1}{\mathcal{D}_n(x,y)} + \log r(n). \tag{10}$$

On the other hand, by Lemma 19, we have

$$\Pr_{x \sim \mathcal{D}_n}\left[\mathsf{K}(x) \geq \log \frac{1}{\mathcal{D}_n(x)} - O(\log n) - \log q(n)\right] \geq 1 - \frac{1}{q(n)}. \tag{11}$$

It follows that with probability at least $1 - 1/q(n)$ over $(x,y) \sim \mathcal{D}_n$, we get

$$\begin{aligned}
\mathsf{pK}^{r(n)}(x,y) &\leq \log \frac{1}{\mathcal{D}_n(x,y)} + \log r(n) && \text{(by Equation (10))}\\
&\leq \mathsf{K}(x,y) + O(\log n) + \log q(n) + \log r(n) && \text{(by Equation (11))}\\
&\leq \mathsf{pnK}^t(x,y) + O(\log n) + \log q(n) + c' \cdot \log t, && \text{(by Lemma 18)}
\end{aligned}$$

where $c' > 0$ is a universal constant. Combining the above with Equation (9), we get

$$\mathsf{pnK}^t(x,y) \geq \mathsf{pnK}^{p_{\mathsf{Sol}}(r(n))}(y \mid x) + \mathsf{pnK}^{p_{\mathsf{Sol}}(r(n))}(x) - \log p_{\mathsf{Sol}}(r(n)) - O(\log n) - \log q(n) - c' \cdot \log t.$$

By letting p be a sufficiently large polynomial, the above yields

$$\mathsf{pnK}^t(x,y) \geq \mathsf{pnK}^t(y \mid x) + \mathsf{pnK}^t(x) - c \cdot \log t,$$

for some universal constant $c > 0$. ◀

4 Worst-Case Hardness of MINnKT: Proof of Theorem 1

In this section we prove Theorem 1.

Proof of Theorem 1. The theorem follows directly from Theorem 33 and Theorem 34, stated and proved in Section 4.1 and Section 4.2, respectively. ◀

4.1 The Easy Direction

Here we prove the following.

▶ **Theorem 33.** *If* $\mathsf{NP} \subseteq \mathsf{BPP}$*, then* $\mathsf{MINnKT} \in \mathsf{BPP}$*.*

Proof. By [26], we know that $\mathsf{NP} \subseteq \mathsf{BPP}$ implies $\mathsf{PH} \subseteq \mathsf{BPP}$. Therefore, it suffices to prove that $\mathsf{MINnKT} \in \mathsf{PH}$. By definition, we have

$$\begin{aligned}(x, 1^s, 1^t) \in \mathsf{MINnKT} &\iff \mathsf{nK}^t(x) \leq s \\ &\iff \exists \Pi, |\Pi| \leq s, \begin{cases} \forall w \in \{0,1\}^t, \Pi(\epsilon; w; \epsilon) \text{ outputs } x \text{ or } \perp \text{ in } t \text{ steps}, \\ \exists w \in \{0,1\}^t, \Pi(\epsilon; w; \epsilon) \text{ outputs } x \text{ in } t \text{ steps}. \end{cases}\end{aligned}$$

The last condition can be checked by a Σ_2^{p}-machine in time $\mathsf{poly}(s,t)$. Therefore, $\mathsf{MINnKT} \in \Sigma_2^{\mathsf{p}} \subseteq \mathsf{PH} \subseteq \mathsf{BPP}$. ◀

4.2 The Hard Direction

Throughout this subsection, we will assume that the underlying computational model is *strongly efficient*, as defined in Section 2.2. We will prove the hard direction of Theorem 1.

▶ **Theorem 34.** *If* $\mathsf{MINnKT} \in \mathsf{BPP}$*, then* $\mathsf{NP} \subseteq \mathsf{BPP}$*.*

We will first define a technical measure of Kolmogorov complexity, namely ℓ-nK. We then prove several properties of ℓ-nK. Finally, we use the measure and its properties to establish Theorem 34.

4.2.1 Technical Tool: ℓ-nK^t_γ

▶ **Definition 35** (ℓ-nK^t_γ)**.** *For any* $t, \ell \in \mathbb{N}$*, any* $\gamma \in (0,1)$*, and any string* $x \in \{0,1\}^*$*, we define* ℓ-$\mathsf{nK}^t_\gamma(x)$ *as*

$$\ell\text{-}\mathsf{nK}^t_\gamma(x) = \min\left\{ s \in \mathbb{Z} \;\middle|\; \Pr_{r \sim \mathcal{U}_\ell}\left[\mathsf{nK}^t(x \circ r) \leq s + \ell\right] \geq \gamma \right\}.$$

From the definition, it is not immediately clear whether ℓ-$\mathsf{nK}^t_\gamma(x)$ is non-negative.[16] Using a simple counting argument, we can bound its value from below by $(-1 - \lceil \log 1/\gamma \rceil)$, as stated in the following lemma.[17]

▶ **Lemma 36** (Lower bound for ℓ-nK^t_γ)**.** *For any* $\ell, t \in \mathbb{N}$*, any string* $x \in \{0,1\}^*$ *and any* $\gamma \in (0,1)$*, we have*

$$\ell\text{-}\mathsf{nK}^t_\gamma(x) \geq -\left\lceil \log \frac{1}{\gamma} \right\rceil - 1.$$

Proof. For the sake of contradiction, suppose that ℓ-$\mathsf{nK}^t_\gamma(x) \leq -\lceil \log 1/\gamma \rceil - 2$. By the definition of ℓ-nK, we have

$$\Pr_{r \sim \mathcal{U}_\ell}\left[\mathsf{nK}^t(x \circ r) \leq \ell - \left\lceil \log \frac{1}{\gamma} \right\rceil - 2\right] \geq \gamma.$$

[16] Consider for instance this definition when γ is exponentially small.

[17] Jumping ahead, in the proof of Theorem 34 we only need to consider ℓ-$\mathsf{nK}^t_\gamma(x)$ for $\gamma = \Omega(1)$, and additive constants do not affect the argument.

Now define the set A as

$$A = \left\{ x \circ r \;\middle|\; r \in \{0,1\}^{\ell},\ \mathsf{nK}^{t}(x \circ r) \leq \ell - \lceil \log 1/\gamma \rceil - 2 \right\}.$$

Then $|A| \geq \gamma \cdot 2^{\ell}$. However, the number of programs of length at most $(\ell - \lceil \log 1/\gamma \rceil - 2)$ is bounded by $2^0 + \cdots + 2^{\ell - \lceil \log 1/\gamma \rceil - 2} \leq 2^{\ell - \lceil \log 1/\gamma \rceil - 1} \leq \gamma \cdot 2^{\ell - 1}$. Since $|A| \geq \gamma \cdot 2^{\ell} > \gamma \cdot 2^{\ell-1}$, by the pigeonhole principle, there must be a program computing two strings in A, which is a contradiction. ◀

▶ **Lemma 37** (Upper Bound for ℓ-nK^t_γ)**.** *There exists a polynomial p such that the following holds. For any $\ell \in \mathbb{N}$, any string $x \in \{0,1\}^*$ and any $\gamma \in (0,1)$, we have*

$$\ell\text{-}\mathsf{nK}^{p(|x|+\ell)}_{\gamma}(x) \leq |x| + \log p(|x| + \ell).$$

Proof. Let M be the Turing machine satisfying the following.

For any strings $y, w \in \{0,1\}^*$, $M(y;\epsilon;w;\epsilon) = y$ in $O(|y|)$ steps.

Hence by efficient universality (Definition 12), when p is a large enough polynomial, for any $r \in \{0,1\}^{\ell}$, $\epsilon \to (x \circ r)$ is in time $p(|x|+\ell)$ and size $(|x| + \ell + \log p(|x| + \ell))$. Therefore, we get

$$\Pr_{r \sim \mathcal{U}_\ell}\left[\mathsf{nK}^{p(|x|+\ell)}(x \circ r) \leq |x| + \ell + \log p(|x| + \ell) \right] = 1.$$

By the definition of ℓ-nK, we conclude that for any $\gamma \in (0,1)$,

$$\ell\text{-}\mathsf{nK}^{p(|x|+\ell)}_{\gamma}(x) \leq |x| + \log p(|x| + \ell).$$

This completes the proof. ◀

▶ **Lemma 38** (Monotonicity of ℓ-$\mathsf{nK}^t_\gamma(x)$ on γ and t)**.** *For any $\gamma_1, \gamma_2 \in (0,1)$ satisfying $\gamma_1 \leq \gamma_2$, any $\ell, t_1, t_2 \in \mathbb{N}^+$ satisfying $t_1 \geq t_2$, and any string $x \in \{0,1\}^*$, we have*

$$\ell\text{-}\mathsf{nK}^{t_1}_{\gamma_1}(x) \leq \ell\text{-}\mathsf{nK}^{t_2}_{\gamma_2}(x).$$

Proof. Let $s = \ell\text{-}\mathsf{nK}^{t_2}_{\gamma_2}(x)$. Then we have

$$\Pr_{r \sim \mathcal{U}_\ell}\left[\mathsf{nK}^{t_2}(x \circ r) \leq s \right] \geq \gamma_2.$$

By non-determinism extension (Definition 12), for any $r \in \{0,1\}^{\ell}$, we have $\mathsf{nK}^{t_1}(x \circ r) \leq \mathsf{nK}^{t_2}(x \circ r)$. Hence we get

$$\Pr_{r \sim \mathcal{U}_\ell}\left[\mathsf{nK}^{t_1}(x \circ r) \leq s \right] \geq \gamma_2 \geq \gamma_1.$$

Hence by the definition of ℓ-nK, $\ell\text{-}\mathsf{nK}^{t_1}_{\gamma_1}(x) \leq s$. Substituting s by $\ell\text{-}\mathsf{nK}^{t_2}_{\gamma_2}(x)$ finishes the proof. ◀

▶ **Lemma 39** (Monotonicity of ℓ-$\mathsf{nK}^t_\gamma(x)$ on ℓ and x)**.** *There exists a polynomial p such that the following holds. For any $\ell, \ell' \in \mathbb{N}$ with $\ell \leq \ell'$, any $\gamma \in (0,1)$, any strings $x, y \in \{0,1\}^*$, and any $t \in \mathbb{N}$, we have*

$$\ell'\text{-}\mathsf{nK}^{t+p(|x|+|y|+\ell')}_{\gamma}(x) \leq \ell\text{-}\mathsf{nK}^{t}_{\gamma}(x \circ y) + \log p(|x| + |y| + \ell')$$

Proof. Let $s = \ell\text{-}\mathsf{nK}^t_\gamma(x \circ y)$. For a string $r \in \{0,1\}^\ell$, we say r is good if $\mathsf{nK}^t(x \circ y \circ r) \leq s + \ell$. Then by the definition of ℓ-nK, with probability at least γ over $r \sim \mathcal{U}_\ell$, r is good. For such a good r, $\epsilon \to (x \circ y \circ r)$ can be non-deterministically computed in time t and size $s + \ell$. For any integer i, let $\widetilde{i}$ denote the binary encoding of i with each bit duplicated, concatenated with 01. Let M be the Turing machine satisfying the following property.

For strings $x, y, r, r', w \in \{0,1\}^*$,

$$M(\widetilde{|x|} \circ \widetilde{|y|} \circ r'; x \circ y \circ r; w; \epsilon) = x \circ r \circ r'$$

in $\mathsf{poly}(|x|, |y|, |r|, |r'|)$ steps.

Hence by efficient universality (Definition 12), for any $r' \in \{0,1\}^{\ell' - \ell}$, $(x \circ y \circ r) \to (x \circ r \circ r')$ is in non-deterministic time $\mathsf{poly}(|x| + |y| + \ell')$ and size $(\ell' - \ell + O(\log(|x| + |y| + \ell')))$. By efficient composition (Definition 12), when p is some large enough polynomial, for any good r, $\epsilon \to (x \circ r \circ r')$ is in non-deterministic time $t + p(|x| + |y| + \ell')$ and size $(s + \ell' + \log p(|x| + |y| + \ell'))$. Therefore, we have

$$\Pr_{\substack{r \sim \mathcal{U}_\ell \\ r' \sim \mathcal{U}_{\ell' - \ell}}} \left[\mathsf{nK}^{t + p(|x|+|y|+\ell')}(x \circ r \circ r') \leq s + \ell' + \log p(|x| + |y| + \ell') \right] \geq \Pr_{r \sim \mathcal{U}_\ell}[r \text{ is good}] \geq \gamma$$

In other words, $\ell'\text{-}\mathsf{nK}^{t + p(|x|+|y|+\ell')}_\gamma(x) \leq s + \log p(|x| + |y| + \ell')$. Substituting s by $\ell\text{-}\mathsf{nK}^t_\gamma(x \circ y)$ finishes the proof. ◀

Similarly to MINnKT, we can define the $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$ problem for ℓ-nK.

▶ **Definition 40** ($\mathsf{MIN}\ell\text{-}\mathsf{nKT}$). *We define* $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$ *as the promise problem* $(\mathcal{YES}, \mathcal{NO})$ *such that*

$$\begin{cases} \ell\text{-}\mathsf{nK}^t_{b/a}(x) \leq s \implies (x, 1^s, 1^t, 1^\ell, 1^a, 1^b) \in \mathcal{YES}, \\ \ell\text{-}\mathsf{nK}^t_{(b-1)/a}(x) > s \implies (x, 1^s, 1^t, 1^\ell, 1^a, 1^b) \in \mathcal{NO}. \end{cases}$$

A basic property of $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$ is that its easiness follows from the easiness of MINnKT.

▶ **Lemma 41.** *If* $\mathsf{MINnKT} \in \mathsf{BPP}$*, then* $\mathsf{MIN}\ell\text{-}\mathsf{nKT} \in \mathsf{prBPP}$.

Proof. Suppose we are given an instance $(x, 1^s, 1^t, 1^\ell, 1^a, 1^b)$ that is in the promised set of inputs for $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$, and we want to decide whether it is in $\mathcal{YES}$ or $\mathcal{NO}$. If it is in $\mathcal{YES}$, then we have $\ell\text{-}\mathsf{nK}^t_{b/a}(x) \leq s$, and by the definition of ℓ-nK, we get

$$\Pr_{r \sim \mathcal{U}_\ell} \left[\mathsf{nK}^t(x \circ r) \leq s + \ell \right] \geq \frac{b}{a}.$$

On the other hand, if the instance is in $\mathcal{NO}$, then we have $\ell\text{-}\mathsf{nK}^t_{(b-1)/a}(x) > s$, and by the definition of ℓ-nK, we get

$$\Pr_{r \sim \mathcal{U}_\ell} \left[\mathsf{nK}^t(x \circ r) \leq s + \ell \right] < \frac{b-1}{a}.$$

Therefore, in order to solve $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$ it suffices to use an algorithm for MINnKT to distinguish the two cases described above. While this is a standard argument, for completeness, we describe the reduction and its analysis below.

Let A be a probabilistic polynomial-time algorithm that decides MINnKT with error $1/3$. Let A^m denote the algorithm that runs A for m times and takes a majority vote. By Chernoff bound, the error of A^m on any given input is bounded by $2^{-m/18}$. Our algorithm B for

deciding $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$ works as follows: it first defines $k = 18a^2$, then it takes k independent samples $R_1, \ldots, R_k \sim \mathcal{U}_\ell$. Next, for each $i \in [k]$, it computes $X_i = A^{\lceil 36 \log k \rceil}(x \circ R_i, 1^s, 1^t)$. Finally, it computes $V = (\sum_{i=1}^{k} X_i)/k$, and accepts if and only if $V \geq (b - 1/2)/a$.

It is not hard to see that B runs in time $\mathsf{poly}(|x|, t, s, \ell, a)$. We claim that B computes $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$ with error at most $1/3$. Let $Y_i = \mathsf{MINnKT}(x \circ R_i, 1^s, 1^t)$. First, since the error of A^m is bounded by $2^{m/18}$, by a union bound, we have

$$\mathbf{Pr}[\exists i \in [k], X_i \neq Y_i] \leq k \cdot 2^{-36 \log k/18} = \frac{1}{k}. \tag{12}$$

That is, with probability $(1 - 1/k)$, all $\mathsf{MINnKT}(x \circ R_i, 1^s, 1^t)$ are computed correctly by $A^{\lceil 36 \log k \rceil}$. Next, if the input is in $\mathcal{YES}$, then by the definition of $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$, we have $\mathbb{E}[\sum_{i=1}^{k} Y_i] \geq k \cdot b/a$. Also, Y_i are independent identically distributed random variables in $\{0, 1\}$. Therefore, using Chernoff bound, we get

$$\mathbf{Pr}\left[\sum_{i=1}^{k} Y_i < k \cdot \frac{b - 1/3}{a}\right] \leq \exp\left(-2 \cdot \frac{1}{9a^2} \cdot k\right) = e^{-4}. \tag{13}$$

Combining Equations (12) and (13) with a union bound, we have

$$\mathbf{Pr}\left[\sum_{i=1}^{k} X_i < k \cdot \frac{b - 1/3}{a}\right] \leq e^{-4} + \frac{1}{k} \leq \frac{1}{3}.$$

That is, the probability of B rejecting an input from $\mathcal{YES}$ is at most $1/3$. Similarly, one can show that the probability of B accepting an input from $\mathcal{NO}$ is at most $1/3$. Hence we conclude that $\mathsf{MIN}\ell\text{-}\mathsf{nKT} = (\mathcal{YES}, \mathcal{NO}) \in \mathsf{prBPP}$. ◀

4.2.2 Useful Bounds for $\ell\text{-}\mathsf{nK}^t_\gamma$

In this section, we state and prove the following lemmas for $\ell\text{-}\mathsf{nK}$, which are important ingredients in the proof of the hard direction of Theorem 1.

▶ **Lemma 42** (DP computation overhead). *There exists a polynomial p such that the following holds. For any $n, m, k, t, \ell \in \mathbb{N}^+$, any $\gamma \in (0, 1)$ and any strings $x \in \{0,1\}^n, y \in \{0,1\}^m, z \in \{0,1\}^{mk}$, we have*

$$\ell\text{-}\mathsf{nK}^{t+p(nmk\ell)}_\gamma(x \circ \mathsf{DP}_k(y; z)) \leq \ell\text{-}\mathsf{nK}^t_\gamma(x \circ y) + mk + \log p(nmk\ell).$$

Proof. Let $s = \ell\text{-}\mathsf{nK}^t_\gamma(x \circ y)$. We say that a given $r \in \{0,1\}^\ell$ is good if $\mathsf{nK}^t(x \circ y \circ r) \leq s + \ell$. Then by the definition of $\ell\text{-}\mathsf{nK}$, with probability at least γ over $r \sim \mathcal{U}_l$, r is good. For a good r, $\epsilon \to (x \circ y \circ r)$ is in non-deterministic time t and size $(s + \ell)$. For any integer i, let $\widetilde{i}$ denote the binary encoding of i with each bit duplicated, concatenated with 01. Let M be the Turing machine satisfying the following.

> For any $n, m, k, \ell \in \mathbb{N}^+$, and for any strings $x \in \{0,1\}^n, y \in \{0,1\}^m, z \in \{0,1\}^{mk}, r \in \{0,1\}^\ell, w \in \{0,1\}^*$,
>
> $$M(\widetilde{n} \circ \widetilde{m} \circ \widetilde{k} \circ \widetilde{\ell} \circ z; x \circ y \circ r; w; \epsilon) = (x \circ \mathsf{DP}_k(y; z) \circ r)$$
>
> in $\mathsf{poly}(nmk\ell)$ steps.

By efficient universality (Definition 12), $(x \circ y \circ r) \to (x \circ \mathsf{DP}_k(y;a) \circ r)$ is in non-deterministic time $\mathsf{poly}(nmk\ell)$ and size $(mk + O(\log(nmk\ell)))$. Then by efficient composition (Definition 12), when p is some large enough polynomial, $\epsilon \to (x \circ \mathsf{DP}_k(y;a) \circ r)$ is in non-deterministic time $(t + p(nmk\ell))$ and size $(s + \ell + mk + \log p(nmk\ell))$. Therefore, for any good $r \in \{0,1\}^\ell$, we have

$$\mathsf{nK}^{t+p(nmk\ell)}(x \circ \mathsf{DP}_k(y;a) \circ r) \leq s + \ell + mk + \log p(nmk\ell).$$

Because the fraction of good $r \sim \mathcal{U}_\ell$ is at least γ, we conclude that

$$\ell\text{-}\mathsf{nK}^{t+p(nmk\ell)}_\gamma(x \circ \mathsf{DP}_k(y;a)) \leq s + mk + \log p(nmk\ell).$$

Substituting s by $\ell\text{-}\mathsf{nK}^t_\gamma(x \circ y)$ completes the proof. ◀

▶ **Lemma 43** (Highly efficient weak symmetry of information surrogate for ℓ-nK)**.** *For any $n, m, t, \ell \in \mathbb{N}^+$, any $0 < \alpha < \gamma < 1$, and any string $x \in \{0,1\}^n$, we have*

$$\Pr_{w \sim \mathcal{U}_m}\left[\ell\text{-}\mathsf{nK}^t_\gamma(x \circ w) \geq (\ell+m)\text{-}\mathsf{nK}^t_{\gamma-\alpha}(x) + m\right] \geq \alpha.$$

Proof. Let s be the smallest integer satisfying

$$\Pr_{w \sim \mathcal{U}_m}\left[\ell\text{-}\mathsf{nK}^t_\gamma(x \circ w) \leq s\right] \geq 1 - \alpha.$$

Then by the definition of ℓ-nK, we have

$$\Pr_{w \sim \mathcal{U}_m}\left[\Pr_{r \sim \mathcal{U}_\ell}\left[\mathsf{nK}^t(x \circ w \circ r) \leq s + \ell\right] \geq \gamma\right] \geq 1 - \alpha.$$

Therefore, we have

$$\Pr_{\substack{w \sim \mathcal{U}_m \\ r \sim \mathcal{U}_\ell}}\left[\mathsf{nK}^t(x \circ w \circ r) \leq s + \ell\right] \geq \gamma(1-\alpha) \geq \gamma - \alpha.$$

By the definition of ℓ-nK, we get

$$(\ell+m)\text{-}\mathsf{nK}^t_{\gamma-\alpha}(x) \leq s - m.$$

In other words, $s \geq (\ell+m)\text{-}\mathsf{nK}^t_{\gamma-\alpha}(x) + m$. Now by minimality of s, we get

$$\Pr_{w \sim \mathcal{U}_m}\left[\ell\text{-}\mathsf{nK}^t_\gamma(x \circ w) \leq s - 1\right] < 1 - \alpha.$$

In other words, $\mathbf{Pr}_{w \sim \mathcal{U}_m}[\ell\text{-}\mathsf{nK}^t_\gamma(x \circ w) \geq s] > \alpha$. Substituting s, we get

$$\Pr_{w \sim \mathcal{U}_m}\left[\ell\text{-}\mathsf{nK}^t_\gamma(x \circ w) \geq (\ell+m)\text{-}\mathsf{nK}^t_{\gamma-\alpha}(x) + m\right] \geq \alpha.$$

This completes the proof. ◀

▶ **Lemma 44** (Chain-rule style relation between ℓ-nK and pnK)**.** *There exists a polynomial p such that the following holds. For strings $x, y \in \{0,1\}^*$ and parameters $\ell, t, t_1, t_2 \in \mathbb{N}^+$ and $\gamma, \gamma', \lambda \in (0,1)$ satisfying the following constraints*:

- $t_2 \leq \ell$,
- $t \geq t_1 + p(|x| + |y| + \ell)$,
- $(1-\gamma) \geq (1-\gamma') + (1-\lambda)$,

we have

$$\ell\text{-}\mathsf{nK}^t_\gamma(x \circ y) \leq \ell\text{-}\mathsf{nK}^{t_1}_{\gamma'}(x) + \mathsf{pnK}^{t_2}_\lambda(y \mid x) + \log p(|x| + |y| + \ell).$$

Proof. The general idea is that, if $t_2 \leq \ell$, then to non-deterministically compute $x \circ y$ we can use the random ℓ-bit string generated along with x via $\ell\text{-}\mathsf{nK}^{t_1}_{\gamma'}(x)$ to obtain y from $\mathsf{pnK}^{\ell}_{\lambda}(y \mid x)$. Let $\ell\text{-}\mathsf{nK}^{t_1}_{\gamma'}(x) = s_1$, and $\mathsf{pnK}^{\ell}_{\lambda}(y \mid x) = s_2$. Without loss of generality, we assume that $s_2 \leq |x| + |y|$, because otherwise, by Lemma 37, when p is a large enough polynomial, we have $\ell\text{-}\mathsf{nK}^{p(|x|+|y|+\ell)}_{\gamma}(x \circ y) \leq |x| + |y| + \log p(|x| + |y| + \ell)$, which finishes the proof. Now for any string $r \in \{0,1\}^{\ell}$, we define the following conditions:

(C1) r satisfies $\mathsf{nK}^{t_1}(x \circ r) \leq s_1 + \ell$.

(C2) r satisfies $\mathsf{nK}^{\ell}(y \mid x; r) \leq s_2$.

Then we have the following claim:

▷ Claim 45. If r satisfies both (C1) and (C2), then for a large enough polynomial p, we have $\mathsf{nK}^{t}(x \circ y \circ r) \leq s_1 + s_2 + \ell + \log p(|x| + |y| + \ell)$.

Proof of Claim 45. By (C1), we have

$$\epsilon \to (x \circ r) \text{ in non-deterministic time } t_1 \text{ and size } s_1 + \ell. \tag{14}$$

By (C2), we have

$$(x; r) \to y \text{ in non-deterministic time } \ell \text{ and size } s_2. \tag{15}$$

Let Π be the program of Equation (15). Let M be the Turing machine which gives efficient simulation for the computational model (Definition 12). Then, for a large enough polynomial p_0, since $s_2 \leq |x| + |y|$, we have

- $\forall w \in \{0,1\}^{\ell}$, $M(\Pi; x; w; r)$ outputs either y or $\perp$ in $p_0(|x| + |y| + \ell)$ steps,
- $\exists w \in \{0,1\}^{\ell}$, $M(\Pi; x; w; r)$ outputs y in $p_0(|x| + |y| + \ell)$ steps.

For any integer i, let $\widetilde{i}$ denote the binary encoding of i with each bit duplicated, concatenated with 01. Let M' be the Turing machine satisfying the following condition.

> For any $\Pi, x, w, r \in \{0,1\}^*$ and any $\ell \in \mathbb{N}$ satisfying $\ell \leq |w|$, on input $(\widetilde{|x|} \circ \widetilde{\ell} \circ \Pi; x \circ r; w; \epsilon)$, M' runs $M(\Pi; x; w_\ell; r)$ to obtain y, where w_ℓ denotes the length-ℓ prefix of w. If $y \neq \perp$, it outputs $x \circ y \circ r$; otherwise it outputs $\perp$.

Then for a large enough polynomial p_1, for any $t' \geq t$ we have

- $\forall w \in \{0,1\}^{t'}$, $M'(\widetilde{|x|} \circ \widetilde{\ell} \circ \Pi; x \circ r; w; \epsilon)$ outputs either $(x \circ y \circ r)$ or $\perp$ in $p_1(|x| + |y| + \ell)$ steps,
- $\exists w \in \{0,1\}^{t'}$, $M'(\widetilde{|x|} \circ \widetilde{\ell} \circ \Pi; x \circ r; w; \epsilon)$ outputs $(x \circ y \circ r)$ in $p_1(|x| + |y| + \ell)$ steps.

By efficient universality (Definition 12), when p_2 is a large enough polynomial, we get

$$(x \circ r) \to (x \circ y \circ r) \text{ in non-deterministic time } p_2(|x| + |y| + \ell) \text{ and size } (s_2 + \log p_2(|x| + |y| + \ell)). \tag{16}$$

By applying efficient composition (Definition 12) to Equations 14 and 16, when p is a large enough polynomial, we get

$$\epsilon \to (x \circ y \circ r) \text{ in non-deterministic time } (t_1 + p(|x| + |y| + \ell)) \text{ and size } (s_1 + s_2 + \ell + \log p(|x| + |y| + \ell)).$$

Since $t \geq t_1 + p(|x| + |y| + \ell)$, we have

$$\mathsf{nK}^{t}(x \circ y \circ r) \leq s_1 + s_2 + \ell + \log p(|x| + |y| + \ell).$$

This completes the proof of the claim. ◁

By the definition of ℓ-nK$_{\gamma'}$ and pnK$_\lambda$, for $r \sim \mathcal{U}_\ell$, r satisfies (C1) with probability at least γ', and r satisfies (C2) with probability at least λ. By a union bound, with probability at least $1-(1-\gamma')-(1-\lambda) \geq \gamma$, r satisfies both (C1) and (C2). Hence by the definition of ℓ-nK and Claim 45, we have

$$\ell\text{-nK}^t_\gamma(x \circ y) \leq \ell\text{-nK}^{t_1}_{\gamma'}(x) + \text{pnK}^\ell_\lambda(y \mid x) + \log p(|x|+|y|+\ell).$$

Since $t_2 \leq \ell$, by randomness and non-determinism extension (Definition 12), we have $\text{pnK}^\ell_\lambda(x) \leq \text{pnK}^{t_2}_\lambda(x)$. Hence we get

$$\ell\text{-nK}^t_\gamma(x \circ y) \leq \ell\text{-nK}^{t_1}_{\gamma'}(x) + \text{pnK}^{t_2}_\lambda(y \mid x) + \log p(|x|+|y|+\ell),$$

as desired. ◀

4.2.3 Main Lemma: Bounding Conditional pK

▶ **Lemma 46** (Main Lemma). *Suppose* MINnKT $\in$ BPP. *Then there exists a polynomial p and an integer $N_0 \in \mathbb{N}^+$ such that the following holds. For any $n, m, \tau, \ell, t, b \in \mathbb{N}^+$ and any string $x \in \{0,1\}^n, y \in \{0,1\}^m$ satisfying the following constraints*:

- $2/3 < b/n^3 < 1$,
- $N_0 \leq n \leq m \leq \tau$.
- $\tau \leq \ell \leq \tau^4$,
- $t \geq 2p(\tau)$,

we have

$$\text{pK}^{p(t)}(y \mid x) \leq \text{pnK}^\tau_{1-1/n^3}(y \mid x) + \ell\text{-nK}^{t-p(\tau)}_{(b+1)/n^3}(x) - (\ell+\tau^3)\text{-nK}^{t+p(\tau)}_{(b-2)/n^3}(x) + \log p(t). \quad (17)$$

We will follow the proof outline as described in Section 1.3. Roughly speaking, we want to define

$$k \approx (\ell+nk)\text{-nK}^{t-\text{poly}(n)}(x \circ y) - (\ell+nk+k)\text{-nK}^t(x), \quad (18)$$

which would allow us to distinguish $\text{DP}_k(y, \mathcal{U}_{nk})$ from $\mathcal{U}_{nk+k}$ when given x and access to MINℓ-nKT. Then by reconstruction of direct product generator, we can upper bound $\text{pK}^{\text{poly}(n)}(y \mid x)$ by approximately k. However, we note that there are several technical subtleties that complicate the proof.

- First, Equation (18) is not a well-formed definition for k, as k appears in both sides. To avoid this, we define k differently (Equation (19)), and prove a lower bound for k (Equation (23)) that allows us to separate the ℓ-nK of $\text{DP}_k(y, \mathcal{U}_{nk})$ from $\mathcal{U}_{nk+k}$ at a threshold of some s (Equation (24)). Then we use the upper bound for k (Equation (22)) to obtain Equation (17).
- Second, since we need to apply a telecoping argument of n rounds based on this lemma, our lemma needs to work for a number of different t's, ℓ's and probability of ℓ-nK, which is why our lemma contains many parameters.
- Finally, because we cannot afford a polynomial blow-up in t, we need to carefully keep track of the running time and all the polynomials from lemmas in the previous sections (p_1, p_2, p_3, p_4, etc.).

Proof of Lemma 46. Let p_1, p_2, p_3, p_4 be the polynomials defined in Lemmas 37, 39, 42, and 44 respectively. Without loss of generality, we assume they are all monotone. We define k as

$$k := \ell\text{-nK}^{t-p_3(\tau^8)}_{b/n^3}(x \circ y) - (\ell+\tau^3)\text{-nK}^{t+p_2(\tau^6)}_{(b-2)/n^3}(x) + \lceil \log p_3(\tau^8) \rceil + \lceil \log p_2(\tau^6) \rceil + 1. \quad (19)$$

First, we argue that k is a positive integer. By our assumptions on the parameters, when N_0 is large enough, we have $n+m+\ell+\tau^3 \le \tau^6$. Hence we have

$$\begin{aligned} k &\ge \ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)}_{b/n^3}(x\circ y) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+p_2(\tau^6)}_{(b-2)/n^3}(x) + \log p_2(\tau^6) + 1 \\ &\ge \ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)}_{b/n^3}(x\circ y) - \ell\text{-}\mathsf{nK}^{t}_{(b-2)/n^3}(x\circ y) + 1 && \text{(By Lemma 39)} \\ &\ge \ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)}_{b/n^3}(x\circ y) - \ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)}_{b/n^3}(x\circ y) + 1 && \text{(By Lemma 38)} \\ &= 1. && (20) \end{aligned}$$

Next, we upper bound and lower bound the value of k. We will then use the DPG reconstruction lemma to finish the proof.

Upper Bound for k. Since $n \le m \le \tau \le \ell$, by Lemma 44, we get

$$\ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)}_{b/n^3}(x\circ y) \le \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)-p_4(\tau^6)}_{(b+1)/n^3}(x) + \log p_4(\tau^6). \tag{21}$$

Therefore, by combining Equations (19) and (21), we have

$$\begin{aligned} k \le \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)-p_4(\tau^6)}_{(b+1)/n^3}(x) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+p_2(\tau^6)}_{(b-2)/n^3}(x) \\ + \log p_3(\tau^8) + \log p_2(\tau^6) + \log p_4(\tau^6) + 3. \end{aligned}$$

Therefore, when q_1 is a large enough polynomial, we get

$$k \le \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \ell\text{-}\mathsf{nK}^{t-q_1(\tau)}_{(b+1)/n^3}(x) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+q_1(\tau)}_{(b-2)/n^3}(x) + \log q_1(\tau). \tag{22}$$

Lower Bound for k. We first show a loose upper bound on k when p and N_0 are large enough. Based on this, we give a lower bound for k. When p is a large enough polynomial such that $p(z) \ge p_3(z^8)+p_1(z^6)$, $t \ge p(\tau) \ge p_3(\tau^8)+p_1(\tau^6)$. By Lemma 37, $\ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)}_{b/n^3}(x\circ y) \le n+m+\log p_1(n+m+\ell) \le 2\tau + \log p_1(\tau^6)$. By Lemma 36, we also have $(\ell+\tau^3)\text{-}\mathsf{nK}^{t+p_2(\tau^6)}_{(b-2)/n^3}(x) \ge -2$. Therefore, we have a loose upper bound for k:

$$k \le 2\tau + \log p_1(\tau^6) + \log p_3(\tau^8) + \log p_2(\tau^6) + 5.$$

Since $\tau \ge N_0$, when N_0 is a large enough constant, we have $k \le 3\tau$. Therefore, we get $mk + k \le \tau^3$, and $\tau^6 \ge \ell + \tau^3 + n$. By Lemma 39, we get

$$\begin{aligned} (\ell+\tau^3)\text{-}\mathsf{nK}^{t+p_2(\tau^6)}_{(b-2)/n^3}(x) - \log p_2(\tau^6) &\le (\ell+\tau^3)\text{-}\mathsf{nK}^{t+p_2(\ell+\tau^3+n)}_{(b-2)/n^3}(x) - \log p_2(\ell+\tau^3+n) \\ &\le (\ell+mk+k)\text{-}\mathsf{nK}^{t}_{(b-2)/n^3}(x). \end{aligned}$$

We also have $\tau^8 \ge nmk\ell$, which gives us

$$\ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)}_{b/n^3}(x\circ y) + \log p_3(\tau^8) \ge \ell\text{-}\mathsf{nK}^{t-p_3(nmk\ell)}_{b/n^3}(x\circ y) + \log p_3(nmk\ell).$$

Therefore, we can now lower bound k:

$$\begin{aligned} k &= \ell\text{-}\mathsf{nK}^{t-p_3(\tau^8)}_{b/n^3}(x\circ y) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+p_2(\tau^6)}_{(b-2)/n^3}(x) + \lceil \log p_3(\tau^8) \rceil + \lceil \log p_2(\tau^6) \rceil + 1 \\ &\ge \ell\text{-}\mathsf{nK}^{t-p_3(nmk\ell)}_{b/n^3}(x\circ y) + \log p_3(nmk\ell) - (\ell+mk+k)\text{-}\mathsf{nK}^{t}_{(b-2)/n^3}(x) + 1. \end{aligned} \tag{23}$$

Distinguisher for the Direct Product Generator. By Equation (20), when N_0 is large enough, $k \geq 1$ always holds. Here we define an algorithm D to distinguish $\mathsf{DP}_k(y;\mathcal{U}_{mk})$ from $\mathcal{U}_{mk+k}$. We set the threshold

$$s = \ell\text{-}\mathsf{nK}^{t-p_3(nmk\ell)}_{b/n^3}(x \circ y) + mk + \lfloor \log p_3(nmk\ell) \rfloor. \tag{24}$$

By Lemma 41, $\mathsf{MINnKT} \in \mathsf{BPP}$ implies $\mathsf{MIN}\ell\text{-}\mathsf{nKT} \in \mathsf{prBPP}$. Let B be the polynomial-time algorithm that decides $\mathsf{MIN}\ell\text{-}\mathsf{nKT}$ with error $1/n^4$. We define the distinguisher D, which depends on x, as follows:

1. D gets input $z \in \{0,1\}^{mk+k}$.
2. D outputs $B(x \circ z, 1^s, 1^t, 1^\ell, 1^{n^3}, 1^b)$.

In fact, if $z \sim \mathsf{DP}_k(y;\mathcal{U}_{mk})$, then by Lemma 42 and Equation (24), we get

$$\Pr_{w \sim \mathcal{U}_{mk}}\left[\ell\text{-}\mathsf{nK}^t_{b/n^3}(x \circ \mathsf{DP}_k(y;w)) \leq s\right] = 1.$$

Therefore, we have

$$\Pr_{\substack{w \sim \mathcal{U}_{mk} \\ \text{Randomness of } D}}[D(\mathsf{DP}_k(y;w)) = 1] \geq 1 - \frac{1}{n^4}. \tag{25}$$

On the other hand, if $z \sim \mathcal{U}_{mk+k}$, then by Lemma 43, we get

$$\Pr_{w \sim \mathcal{U}_{mk+k}}\left[\ell\text{-}\mathsf{nK}^t_{(b-1)/n^3}(x \circ w) \geq (\ell + mk + k)\text{-}\mathsf{nK}^t_{(b-2)/n^3}(x) + mk + k\right] \geq \frac{1}{n^3}.$$

But by Equation (23) and Equation (24), we have

$$(\ell + mk + k)\text{-}\mathsf{nK}^t_{(b-2)/n^3}(x) + mk + k \geq s + 1.$$

Combining these two inequalities, we get

$$\Pr_{w \sim \mathcal{U}_{mk+k}}\left[\ell\text{-}\mathsf{nK}^t_{(b-1)/n^3}(x \circ w) \geq s + 1\right] \geq \frac{1}{n^3}.$$

Therefore, we have

$$\Pr_{\substack{w \sim \mathcal{U}_{mk+k} \\ \text{Randomness of } D}}[D(w) = 1] \leq 1 - \frac{1}{n^3} + \frac{1}{n^4}. \tag{26}$$

By comparing Equation (25) and Equation (26), we see that D distinguishes $\mathsf{DP}_k(y;\mathcal{U}_{mk})$ from $\mathcal{U}_{mk+k}$ with probability $1/n^3 - 2/n^4 \geq 1/n^4$. Then by Corollary 24, there exists some polynomial p_{DP} such that

$$\mathsf{pK}^{p_{\mathsf{DP}}(t),\|D}(y) \leq k + \log p_{\mathsf{DP}}(t).$$

We can store the program of D in the description, as well as t, n, m, k, s, ℓ, b, which takes no more than $O(\log t)$ bits in total. If we have x, then we can simulate D, which runs in time $\mathsf{poly}(t)$, and answer the oracle queries. Therefore, when q_2 is a large enough polynomial, we have

$$\mathsf{pK}^{q_2(t)}(y \mid x) \leq k + \log q_2(t). \tag{27}$$

Putting It All Together. To summarize the previous paragraphs, when p is a large enough polynomial and N_0 is a large enough constant, by Equation (22) there exists some polynomial q_1 such that

$$k \leq \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \ell\text{-}\mathsf{nK}^{t-q_1(\tau)}_{(b+1)/n^3}(x) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+q_1(\tau)}_{(b-2)/n^3}(x) + \log q_1(\tau).$$

Also, by Equation (27), there exists some polynomial q_2 such that

$$k \geq \mathsf{pK}^{q_2(t)}(y \mid x) - \log q_2(t).$$

Combining these two inequalities, we get

$$\mathsf{pK}^{q_2(t)}(y \mid x) \leq \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \ell\text{-}\mathsf{nK}^{t-q_1(\tau)}_{(b+1)/n^3}(x) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+q_1(\tau)}_{(b-2)/n^3}(x) + \log q_1(\tau) + \log q_2(t).$$

Therefore, when p is a large enough polynomial, we get

$$\mathsf{pK}^{p(t)}(y \mid x) \leq \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \ell\text{-}\mathsf{nK}^{t-p(\tau)}_{(b+1)/n^3}(x) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+p(\tau)}_{(b-2)/n^3}(x) + \log p(t),$$

as desired. ◀

4.2.4 Proof of Theorem 34

We first state and prove the following lemma, which is then used to establish Theorem 34.

▶ **Lemma 47** (Witness compression). *Let $L = \mathsf{SAT}$, and define $L_n = L \cap \{0,1\}^n$. Suppose $L \in \mathsf{NTIME}[n^c]$. For a given $x \in \{0,1\}^n$, define $A_x \subseteq \{0,1\}^{n^c}$ to be the set of witnesses for x. If $\mathsf{MINnKT} \in \mathsf{BPP}$, then there exists a polynomial p such that for any n, any $x \in L_n$ and any $y \in A_x$, we have*

$$\mathsf{pK}^{p(n)}(y \mid x) \leq \log |A_x| + \log p(n).$$

Proof. If $x \in L_n$, then by Theorem 4, there exists some polynomial p_{LC} such that for any $y \in A_x$, $\mathsf{pnK}^{p_{\mathsf{LC}}(n^c),A_x}(y \mid x) \leq \log |A_x| + \log p_{\mathsf{LC}}(n^c)$. Since A_x is decidable in time n^c given x, and by Lemma 29 we can amplify the success probability, we get that there exists some polynomial p_1 such that $p_1(z) \geq z^d$, for a large enough constant $d > 0$, and

$$\mathsf{pnK}^{p_1(n)}_{1-1/n^3}(y \mid x) \leq \log |A_x| + \log p_1(n). \tag{28}$$

Let p_2 be the polynomial stated in Lemma 46. Let $m = n^c, \tau = p_1(n)$. Then by Lemma 46, when n is large enough, for any $\ell, t, b \in \mathbb{N}^+$ satisfying $2/3 < b/n^3 < 1$, $\tau \leq \ell \leq \tau^4$, $t \geq p_2(\tau)$, and any string $x \in \{0,1\}^n$, $y \in \{0,1\}^{n^c}$, we have

$$\mathsf{pK}^{p_2(t)}(y \mid x) \leq \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \ell\text{-}\mathsf{nK}^{t-p_2(\tau)}_{(b+1)/n^3}(x) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+p_2(\tau)}_{(b-2)/n^3}(x) + \log p_2(t).$$

Now, setting t to be some polynomial of n, if we could bound $\left(\ell\text{-}\mathsf{nK}^{t-p_2(\tau)}_{(b+1)/n^3}(x) - (\ell+\tau^3)\text{-}\mathsf{nK}^{t+p_2(\tau)}_{(b-2)/n^3}(x)\right)$ by a constant, then using Equation (28) we would be done. Even though we don't know how to do this directly, we can use a telescoping sum trick. More specifically, let p_3 be the polynomial stated in Lemma 37. We define three sequences $\{\ell_i\}_{i=0}^n$, $\{t_i\}_{i=0}^n$ and $\{b_i\}_{i=0}^n$ recursively:

- $\ell_0 = \tau^3$; $\ell_i = \ell_{i-1} + \tau^3$, $\forall i \in [n]$.
- $t_0 = 2p_2(\tau) + p_3(\tau^6)$; $t_i = t_{i-1} + 2p_2(\tau)$, $\forall i \in [n]$.
- $b_0 = \lfloor 3n^3/4 \rfloor$; $b_i = b_{i-1} - 3$, $\forall i \in [n]$.

It is not hard to verify that for any $i \in \{0, \dots, n\}$, ℓ_i, t_i, b_i satisfy the conditions of Lemma 46. Hence we get

$$\mathsf{pK}^{p_2(t_i)}(y \mid x) \leq \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \ell_i\text{-}\mathsf{nK}^{t_i - p_2(\tau)}_{(b_i+1)/n^3}(x) - (\ell_i + \tau^3)\text{-}\mathsf{nK}^{t_i + p_2(\tau)}_{(b_i-2)/n^3}(x) + \log p_2(t_i). \tag{29}$$

Now using a telescoping argument, we get

$$\begin{aligned}
\ell_0\text{-}\mathsf{nK}^{t_0 - p_2(\tau)}_{(b_0+1)/n^3}(x) - \ell_n\text{-}\mathsf{nK}^{t_n - p_2(\tau)}_{(b_n+1)/n^3}(x) &= \sum_{i=0}^{n-1} \left(\ell_i\text{-}\mathsf{nK}^{t_i - p_2(\tau)}_{(b_i+1)/n^3}(x) - \ell_{i+1}\text{-}\mathsf{nK}^{t_{i+1} - p_2(\tau)}_{(b_{i+1}+1)/n^3}(x) \right) \\
&= \sum_{i=0}^{n-1} \left(\ell_i\text{-}\mathsf{nK}^{t_i - p_2(\tau)}_{(b_i+1)/n^3}(x) - (\ell_i + \tau^3)\text{-}\mathsf{nK}^{t_i + p_2(\tau)}_{(b_i-2)/n^3}(x) \right)
\end{aligned}$$

By Lemma 37, we have

$$\ell_0\text{-}\mathsf{nK}^{t_0 - p_2(\tau)}_{(b_0+1)/n^3}(x) \leq (\tau^3)\text{-}\mathsf{nK}^{p_3(\tau^6)}_{(b_0+1)/n}(x) \leq n + \log p_3(\tau^6),$$

which is at most $1.5n$ for all large enough n. On the other hand, it follows from Lemma 36 and the definition of b_n that $-\ell_n\text{-}\mathsf{nK}^{t_n - p_2(\tau)}_{(b_n+1)/n^3}(x) = O(1)$. This implies that

$$\ell_0\text{-}\mathsf{nK}^{t_0 - p_2(\tau)}_{(b_0+1)/n^3}(x) - \ell_n\text{-}\mathsf{nK}^{t_n - p_2(\tau)}_{(b_n+1)/n^3}(x) \leq 2n,$$

provided that n is large enough. By averaging, there must be some $i \in \{0, \dots, n-1\}$ such that

$$\left(\ell_i\text{-}\mathsf{nK}^{t_i - p_2(\tau)}_{(b_i+1)/n^3}(x) - (\ell_i + \tau^3)\text{-}\mathsf{nK}^{t_i + p_2(\tau)}_{(b_i-2)/n^3}(x) \right) \leq 2.$$

For such an i, by Equation (29) we have

$$\mathsf{pK}^{p_2(t_i)}(y \mid x) \leq \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \log p_2(t_i) + 2.$$

Note that $t_i \leq t_n$. Then the above yields

$$\mathsf{pK}^{p_2(t_n)}(y \mid x) \leq \mathsf{pnK}^{\tau}_{1-1/n^3}(y \mid x) + \log p_2(t_n) + 2.$$

Combining this with Equation (28) and using that $\tau = p_1(n)$, we get that for all large enough n, for any $x \in L_n$ and $y \in A_x$,

$$\mathsf{pK}^{p_2(t_n)}(y \mid x) \leq \log |A_x| + \log p_1(n) + \log p_2(t_n) + 2.$$

Notice that $t_n \leq (2n+2)p_2(\tau) + p_3(\tau^6) = (2n+2)p_2(p_1(n)) + p_3(p_1^6(n))$, which is polynomially bounded by n. Hence for a large enough polynomial p, we have

$$\mathsf{pK}^{p(n)}(y \mid x) \leq \log |A_x| + \log p(n),$$

which concludes the proof. ◀

Proof of Theorem 34 from Lemma 47. Our randomized polynomial-time algorithm B for deciding $L = \mathsf{SAT}$ works as follows:

1. B gets $x \in \{0,1\}^n$ as input.
2. B takes a sample y from $\mathsf{USamp}(1^{n^c}, 1^{p(n)}, x)$.
3. B checks if y is a witness for x.
4. B repeats steps 2 and 3 for $O(n^c \cdot p(n))$ rounds. If at any round, y is indeed a witness for x, B accepts; otherwise B rejects.

If $x \notin L_n$, then $A_x = \varnothing$, so B never accepts x. If $x \in L_n$, then by Lemma 47 and Proposition 26, for any $y \in A_x$, the probability that $\mathsf{USamp}(1^{n^c}, 1^{p(n)}, x) = y$ is at least $\Omega(1/(n^c \cdot p(n) \cdot |A_x|))$. Adding up the probabilities for each $y \in A_x$, we get

$$\Pr_{y \sim \mathsf{USamp}(1^{n^c}, 1^{p(n)}, x)}[y \in A_x] \geq \Omega\left(\frac{1}{n^c \cdot p(n)}\right).$$

Therefore, if we sample $O(n^c \cdot p(n))$ times, we can obtain some $y \in A_x$ with probability $\Omega(1)$. Hence B accepts x with probability $\Omega(1)$. Since this probability can be easily amplified, we conclude that $L \in \mathsf{BPP}$. Finally, since $L = \mathsf{SAT}$ is NP-complete, we conclude that $\mathsf{NP} \subseteq \mathsf{BPP}$. ◀

5 Average-Case Hardness of MINnKT

In this section, we show Theorem 2 and Theorem 3. We begin with some useful tools.

5.1 Technical Tools

We will in fact consider an assumption that is weaker than MINnKT being average-case easy. More specifically, let "$(\mathsf{coMINnKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}$" denote the following statement.

> There exists a constant $c > 0$, a polynomial ρ, and a probabilistic polynomial-time algorithm A such that the following hold for all sufficiently large n, all $t \geq \rho(n)$, and all $s \leq n - c \cdot \log t$:
> 1. For every $x \in \{0,1\}^n$ with $\mathsf{nK}^t(x) \leq s$, we have $\mathbf{Pr}_A[A(x, 1^s, 1^t) = 1] \geq \frac{2}{3}$.
> 2. With probability at least $1 - 1/t$ over $x \sim \{0,1\}^n$, we have $\mathbf{Pr}_A[A(x, 1^s, 1^t) = 0] \geq \frac{2}{3}$.

Also, let "$(\mathsf{coMINKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}$" denote the analogous statement where nK^t is replaced by K^t.

We first state some results that are implicit in prior work, e.g., [11, 13, 8, 18, 20]. We omit the details of the proofs since no new ideas are needed.

▶ **Lemma 48.** *The following holds.*

$$(\mathsf{MINnKT}, \mathcal{U}) \in \mathsf{AvgBPP} \implies (\mathsf{coMINnKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP} \implies (\mathsf{coMINKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}.$$

Proof Sketch. The proof of the first implication can be easily adapted from that of [20, Proposition 11]. The high-level idea is that if there exists an errorless heuristic scheme that computes MINnKT, then we can replace $\perp$ with 1 to obtain an algorithm that still accepts all strings with small nK^t-complexity while rejecting a large fraction of random strings. This follows from the fact that a random string has large nK^t-complexity, and the new algorithm will only err on a small fraction of these strings.

The second implication follows from the fact that $\mathsf{nK}^t(x) \leq \mathsf{K}^t(x)$ for every x and t. ◀

▶ **Lemma 49** (Symmetry of Information for pK^t; See, e.g., [20, Lemma 36])**.** *If* $(\mathsf{coMINKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}$*, then there exist polynomials* p_{Sol} *and* p_0 *such that for all sufficiently large* $x, y \in \{0,1\}^*$ *and every* $t \geq p_0(|x| + |y|)$*,*

$$\mathsf{pK}^{p_{\mathsf{Sol}}(t)}(y \mid x) \leq \mathsf{pK}^t(x, y) - \mathsf{pK}^{p_{\mathsf{Sol}}(t)}(x) + \log p_{\mathsf{Sol}}(t).$$

▶ **Lemma 50.** *If* $(\mathsf{coMINKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}$*, then there exist a constant* $c > 0$*, a polynomial* p_{dth} *and an algorithm* Approx-depth *that, on input* $(x, 1^{t_1}, 1^{t_2}, 1^k)$*, where* $x \in \{0,1\}^n$*,* $t_1, t_2, k \in \mathbb{N}$ *with* $t_1, t_2 \geq cn$*, runs in time* $\mathsf{poly}(n, t_1, t_2, k)$ *and with probability* $1 - 2^{-k}$ *outputs an integer* s *such that*

$$\mathsf{pK}^{p_{\mathsf{dth}}(t_1)}(x) - \mathsf{pK}^{t_2}(x) \leq s \leq \mathsf{pK}^{t_1}(x) - \mathsf{pK}^{p_{\mathsf{dth}}(t_2)}(x) + \log p_{\mathsf{dth}}(t_1) + \log p_{\mathsf{dth}}(t_2).$$

Proof Sketch. The proof ideas are similar to those in [20, Lemma 47]. The main difference is that, instead of using a generator with a rK^t-style reconstruction and sub-optimal advice complexity as in the proof of [20, Lemma 47], we employ the direct product generator, which has a pK^t-style reconstruction and optimal advice complexity. ◀

▶ **Lemma 51.** *If* $(\mathsf{coMINnKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}$*, then there exists a polynomial* p_1 *such that for all* $z \in \{0,1\}^*$ *and* $t \in \mathbb{N}$*,*

$$\mathsf{pK}^{p_1(t)}(z) \leq \mathsf{pnK}^t(z) + \log p_1(t).$$

Proof Sketch. The proof can be easily adapted from that of Lemma 56, by considering x to be the empty string in the statement of Lemma 56. ◀

The following is the main technical lemma for this section.

▶ **Lemma 52.** *If* $(\mathsf{coMINnKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}$*, then there exist polynomials* p *and* q *such that for all* $x, y \in \{0,1\}^*$ *and all* $t \geq q(|x|, |y|)$*,*

$$\mathsf{pK}^{p(t)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + \mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x) + \log p(t).$$

Proof. Assume $(\mathsf{coMINnKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}$. Fix $x, y \in \{0,1\}^*$ and $t \geq q(|x|, |y|)$, where q is a sufficiently large polynomial.

First of all, note that by Lemma 48 and Lemma 49, we get symmetry of information for pK^t. Now let p_1 be the polynomial from Lemma 51. We have

$$\begin{aligned}
\mathsf{pK}^{p_{\mathsf{Sol}}(p_1(2t))}(y \mid x) &\leq \mathsf{pK}^{p_1(2t)}(x, y) - \mathsf{pK}^{p_{\mathsf{Sol}}(p_1(2t))}(x) + \log p_{\mathsf{Sol}}(p_1(2t)) \\
&\leq \mathsf{pnK}^{2t}(x, y) - \mathsf{pK}^{p_{\mathsf{Sol}}(p_1(2t))}(x) + \log p_{\mathsf{Sol}}(p_1(2t)) \qquad \text{(by Lemma 51)} \\
&\leq \mathsf{pnK}^t(y \mid x) + \mathsf{pnK}^t(x) - \mathsf{pK}^{p_{\mathsf{Sol}}(p_1(2t))}(x) + \log p_{\mathsf{Sol}}(p_1(2t)) \\
&\leq \mathsf{pnK}^t(y \mid x) + \mathsf{pK}^t(x) - \mathsf{pK}^{p_{\mathsf{Sol}}(p_1(2t))}(x) + \log p_{\mathsf{Sol}}(p_1(2t)).
\end{aligned}$$

Let p be a sufficiently large polynomial, the above yields

$$\mathsf{pK}^{p(t)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + \mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x) + \log p(t),$$

as desired. ◀

5.2 Proof of Theorem 2

In this subsection, we prove Theorem 2.

Proof of Theorem 2. Let $L \in \mathsf{NP}$. For an instance $x \in \{0,1\}^n$, let A_x denote the set of witnesses of x (with respect to some fixed verifier). Without loss of generality, we assume that every string in A_x has a length of exactly m, where $m = \mathsf{poly}(n)$. Let $\mathcal{D} := \{\mathcal{D}_n\}_{n \in \mathbb{N}}$ be a polynomial-time samplable distribution family.

Assuming $(\mathsf{MINnKT}, \mathcal{U}) \in \mathsf{AvgBPP}$, we show an errorless heuristic scheme for solving L over $\mathcal{D}$. The idea is to employ Lemma 52 and show that $\mathsf{pK}^{p(t)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + O(\log n)$ *for most instances* $x \sim \mathcal{D}$. Then we can use language compression for pnK^t (Theorem 4)

and follow the argument as shown in the proof of Theorem 10 to obtain an witness for those instances, using the universal sampler. For this, we show that computational depth, $\mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x)$ as appear in Lemma 52, is small for most x. Moreover, in order to make the algorithm errorless, we need to be able to recognize the "bad" x's whose computational depth is not small. More specifically, we need the following depth certifying algorithm.

▷ Claim 53. Let p be the polynomial in Lemma 52. There exists a probabilistic polynomial-time algorithm $\mathsf{Certify}$, a polynomial p_1, and a constant $d > 0$ such that for every $n, k \in \mathbb{N}$ and $t \geq p_1(n)$, the following holds with probability $1 - 2^{-k}$ over the internal randomness of $\mathsf{Certify}$:

1. For every $x \in \{0,1\}^n$, if $\mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x) > d \cdot \log t + \log k$, then $\mathsf{Certify}(x, 1^t, 1^k) = 0$.
2. Also,

$$\Pr_{x \sim \mathcal{D}_n}[\mathsf{Certify}(x, 1^t, 1^k) = 1] \geq 1 - \frac{1}{2k}.$$

Proof of Claim 53. Let $\mathsf{Approx\text{-}depth}$ be the algorithm in Lemma 50, and let $d > 0$ be a constant specified later. We define $\mathsf{Certify}$ as follows.

> On input $(x, 1^t, 1^k)$, compute $s := \mathsf{Approx\text{-}depth}(x, 1^{p^{-1}_{\mathsf{dth}}(t)}, 1^{p(t)}, 1^{n+k})$. If $s \leq d \cdot \log t + \log k$, accept; otherwise, reject.

By the correctness of the algorithm $\mathsf{Approx\text{-}depth}$ (Lemma 50) and a union bound over the set of n-bit strings, it is easy to see that with probability at least $1 - 2^{-k}$, it holds that

$$\mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x) \leq s \leq \mathsf{pK}^{p^{-1}_{\mathsf{dth}}(t)}(x) - \mathsf{pK}^{p_{\mathsf{dth}}(p(t))}(x) + \log(t) + \log p_{\mathsf{dth}}(p(t)).$$

The first property of $\mathsf{Certify}$ stated in the claim follows directly from the lower bound of s above.

For the second property, we use the upper bound of s and show that with probability at least $1 - 1/k$ over $x \sim \mathcal{D}_n$, it holds that the quantify

$$\mathsf{pK}^{p^{-1}_{\mathsf{dth}}(t)}(x) - \mathsf{pK}^{p_{\mathsf{dth}}(p(t))}(x) + \log(t) + \log p_{\mathsf{dth}}(p(t)) \tag{30}$$

is at most $\leq d \cdot \log t + \log k$.

First of all, by the coding theorem for pK^t (Theorem 20), we have for every $x \in \mathsf{Support}(\mathcal{D}_n)$

$$\mathsf{pK}^{p^{-1}_{\mathsf{dth}}(t)}(x) \leq \log \frac{1}{\mathcal{D}_n(x)} + O(\log n), \tag{31}$$

provided that $t \geq p_1(n)$ for some sufficiently large polynomial p_1.

On the other hand, by Lemma 19, we have

$$\Pr_{x \sim \mathcal{D}_n}\left[\mathsf{K}(x) \geq \log \frac{1}{\mathcal{D}_n(x)} - O(\log n) - \log 2k\right] \geq 1 - \frac{1}{2k}. \tag{32}$$

It follows that with probability at least $1 - 1/(2k)$ over $x \sim \mathcal{D}_n$, we get

$$\begin{aligned}
&\mathsf{pK}^{p^{-1}_{\mathsf{dth}}(t)}(x) - \mathsf{pK}^{p_{\mathsf{dth}}(p(t))}(x) \\
&\leq \mathsf{pK}^{p^{-1}_{\mathsf{dth}}(t)}(x) - \mathsf{K}(x) + O(\log p_{\mathsf{dth}}(p(t))) && \text{(by Lemma 18)} \\
&\leq \left(\log \frac{1}{\mathcal{D}_n} + O(\log n)\right) - \left(\log \frac{1}{\mathcal{D}_n(x)} - O(\log n) - \log 2k\right) + O(\log p_{\mathsf{dth}}(p(t))) \\
& && \text{(by Equations (31) and (32))} \\
&\leq (d/2) \cdot \log t + \log k,
\end{aligned}$$

provided that d is sufficiently large. Finally, by plugging the above into Equation (30), we get that Equation (30) (and hence s) is upper bounded by $d \cdot \log t + \log k$. This completes the proof of the second property of $\mathsf{Certify}$ and hence the claim. ◁

Before describing our final heuristic scheme for solving L, we show how we can solve L for those x whose computational depth is small. Define the following procedure B.

> On input x, run $\mathsf{USamp}(1^m, 1^{p(p_2(n))}, x)$, where $p_2 \geq \max\{p_{\mathsf{LC}}, q\}$, p_{LC} is the polynomial in Theorem 4 and p, q are the polynomials in Lemma 52. Let y be the output of USamp, if y is an L-witness of x, accept. Otherwise, reject.

Consider the case when the following condition holds:

$$\mathsf{pK}^{p_2(n)}(x) - \mathsf{pK}^{p_1(p_2(n))}(x) + \log p(t) \leq d \cdot \log p_2(n) + \log k. \tag{33}$$

Then we have

$$\begin{aligned}
\mathsf{pK}^{p(p_2(n))}(y \mid x) &\leq \mathsf{pnK}^{p_2(n)}(y \mid x) + \mathsf{pK}^{p_2(n)}(x) - \mathsf{pK}^{p(p_2(n))}(x) + \log p(p_2(n)) \\
&\qquad \text{(by Lemma 52)} \\
&\leq \log |A_x| + \log p_2(n) + \mathsf{pK}^{p_2(n)}(x) - \mathsf{pK}^{p(p_2(n))}(x) + \log p(p_2(n)) \\
&\qquad \text{(by Theorem 4)} \\
&\leq \log |A_x| + \log p_2(n) + d \cdot \log p_2(n) + \log k + \log p(p_2(n)) \\
&\qquad \text{(by Equation (33))} \\
&\leq \log |A_x| + d' \cdot \log n + \log k,
\end{aligned}$$

where $d' > 0$ is a sufficiently large constant. Then by Lemma 32, we get that A solves L on x with probability at least $1/\mathsf{poly}(n)$. Using standard techniques, we can amply the success probability of B to be at least $9/10$.

We now describe an errorless average-case algorithm for solving L. Let A be the following algorithm.

> On input $(x, 1^n, 1^k)$, we first run $\mathsf{Certify}(x, 1^{p_2(n)}, 1^k)$. If it rejects, we output $\perp$. Otherwise, we run $B(x)$ and output its returned answer.

To see that the above algorithm is heuristic scheme for solving L, first note that with probability at least $1 - 2^{-k}$ over the internal randomness of $\mathsf{Certify}$, we either have that $\mathsf{Certify}$ rejects, in which case we output $\perp$, or $\mathsf{Certify}$ accepts and in the latter case, by the first property of $\mathsf{Certify}$ stated in Claim 53, the condition as specified in Equation (33) holds, and the algorithm B will correct decide L on x with probability at least $9/10$. It follows that with probability at least $4/5$, the above algorithm A outputs either $\perp$ or the $L(x)$.

Secondly, by the second property of $\mathsf{Certify}$ stated in Claim 53, with probability at least $1 - 2^{-k}$ over the internal randomness of the our algorithm, we output a value that is not $\perp$ with probability at least $1 - 1/(2k)$ over $x \sim \mathcal{D}_n$. By a simple averaging argument, we get that

$$\mathop{\mathbf{Pr}}_{x \sim \mathcal{D}_n}\left[\mathop{\mathbf{Pr}}_{A}[A(x, 1^n, 1^k) = \perp] < 1/5\right] \geq 1 - 1/k.$$

This completes the proof of Theorem 2. ◀

5.3 Proof of Theorem 3

In this subsection, we show Theorem 3.

Proof of Theorem 3. Let $L \in \mathsf{NP}$. For an instance $x \in \{0,1\}^n$, let A_x denote the set of witnesses of x (with respect to some fixed verifier). Without loss of generality, we assume that every string in A_x has a length of exactly m, where $m = \mathsf{poly}(n)$. Let $\mathcal{D} : \{\mathcal{D}_n\}_{n\in\mathbb{N}}$ be a polynomial-time samplable distribution family.

Assuming $\mathsf{MINnKT}^{\mathcal{U}} \in \mathsf{AvgBPP}$, we show an randomized algorithm for solving L in time $2^{O(n/\log n)}$. The idea is to employ Lemma 52 and show that for every instance x, $\mathsf{pK}^{p(t)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + O(n/\log n)$ *for some* $t \leq 2^{O(n/\log n)}$. Then we can use language compression pnK^t (Theorem 4) and follow the argument as shown in the proof of Theorem 10 to obtain an witness within the claimed time.

▷ Claim 54 ([15]). For every $\varepsilon > 0$, every non-decreasing polynomials β and p, and every large enough $x \in \{0,1\}^n$, there exists a time bound t such that $\beta(n) \leq t \leq 2^{n^\varepsilon}$ and

$$\mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x) \leq O\left(\frac{n}{\log n}\right).$$

Proof Sketch of Claim 54. The proof can be easily adapted from that of [20, Lemma 12], which follows from a simple argument using telescoping sum. ◁

Consider the following algorithm $\mathcal{A}$ for solving L.

> On input $x \in \{0,1\}^n$, run $\mathsf{USamp}(1^m, 1^{2^{c\cdot n/\log n}}, x)$, where $c > 0$ is a constant specified later. Let y be the output of USamp, if y is an L-witness of x, accept. Otherwise, reject.

It is easy to see that the above algorithm runs in time $2^{O(n/\log n)}$. Next, we argue its correctness.

Let p_{LC} be the polynomial in Theorem 4 and p, q be the polynomials in Lemma 52. By Claim 54, there exists some $\max\{p_{\mathsf{LC}}(n), q(n,m)\} \leq t \leq 2^{\sqrt{n}}$ such that

$$\mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x) \leq O\left(\frac{n}{\log n}\right). \tag{34}$$

In this case, we have

$$\begin{aligned}
\mathsf{pK}^{p(t)}(y \mid x) &\leq \mathsf{pnK}^t(y \mid x) + \mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x) + \log p(t) && \text{(by Lemma 52)}\\
&\leq \log|A_x| + \log t + \mathsf{pK}^t(x) - \mathsf{pK}^{p(t)}(x) + \log p(t) && \text{(by Theorem 4)}\\
&\leq \log|A_x| + \log t + O\left(\frac{n}{\log n}\right) + \log p(t) && \text{(by Equation (34))}\\
&\leq \log|A_x| + \frac{d'\cdot n}{\log n},
\end{aligned}$$

where $d' > 0$ is a sufficiently large constant. Then by Lemma 32 and letting c be a sufficiently large constant, we get that $\mathsf{USamp}(1^m, 1^{2^{c\cdot n/\log n}}, x)$ outputs a L-witness of x (if exists) with probability at least $1/2^{O(n/\log n)}$. It follows the algorithm $\mathcal{A}$ will reject with probability 1 on a no-instance and accept with probability at least $1/2^{O(n/\log n)}$ on a yes-instance. Using standard techniques, we can amply the success probability of such an algorithm to be at least $2/3$. ◀

6 Worst-Case Hardness of Gap-Cond-MINnKT: Proofs of Theorem 10 and Corollary 11

In this section, we prove Theorem 10 and Corollary 11. We first need the following lemma.

▶ **Lemma 55.** *If* Gap-Cond-MINnKT $\in$ prBPP*, then there exists a polynomial $p(\cdot)$ such that for all $x, y \in \{0,1\}^*$ and $t \in \mathbb{N}$,*

$$\mathsf{pK}^{p(t+|x|)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + \log p(t+|x|).$$

For the convenience of presenting the proofs in some later results, we will prove the following stronger lemma.

▶ **Lemma 56.** *Assume there exists a constant $c > 0$, a polynomial ρ, and a probabilistic polynomial-time algorithm A such that the following hold for all sufficiently large $n, m' \in \mathbb{N}$, all $x \in \{0,1\}^n$, all $t' \geq \rho(n+m')$, and all $s' \leq m' - c \cdot \log(t'+n)$:*

1. *For every $y' \in \{0,1\}^{m'}$ with $\mathsf{nK}^t(y' \mid x) \leq s'$, we have $\mathbf{Pr}_A[A(x, y', 1^{s'}, 1^{t'}) = 1] \geq \frac{2}{3}$.*
2. *With probability at least $1 - 1/t'$ over $y' \sim \{0,1\}^{m'}$, we have $\mathbf{Pr}_A[A(x, y', 1^{s'}, 1^{t'}) = 0] \geq \frac{2}{3}$.*

Then there exists a polynomial $p(\cdot)$ such that for all $x, y \in \{0,1\}^*$ and $t \in \mathbb{N}$,

$$\mathsf{pK}^{p(t+|x|)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + \log p(t+|x|).$$

Proof. Fix $x \in \{0,1\}^n$, $y \in \{0,1\}^m$ and $t \in \mathbb{N}$. Without loss of generality, we assume that $t \geq |y|$. Let DP_k be the direct generator, where $k > 0$ is specified later.

Let $c > 0$ be a constant, ρ be a polynomial and A be an algorithm as stated in the assumption of the lemma.

Let $s := \mathsf{pnK}^t(y \mid x)$ and $k := s + d \cdot \log t$ for a sufficiently large constant $d > c$. Then with probability at least $2/3$ over $r \sim \{0,1\}^t$, there is a program $\Pi \in \{0,1\}^s$ such that

- $\forall w \in \{0,1\}^t$, $\Pi(x; w; r)$ outputs y or $\perp$ in t steps, and
- $\exists w \in \{0,1\}^t$, $\Pi(x; w; r)$ outputs y in t steps.

Note that the above implies that with probability at least $2/3$ over $r \sim \{0,1\}^t$,

$$\mathsf{nK}^{\mathsf{poly}(t)}(y \circ r \mid x) \leq s + |r| + O(\log t).$$

Also, for every $z \in \{0,1\}^{mk}$, $\mathsf{DP}_k(y; z)$ is computable in time $\mathsf{poly}(m)$ given x. It follows that with probability at least $2/3$ over $r \sim \{0,1\}^t$,

$$\mathsf{nK}^{t'}(\mathsf{DP}_k(y; z) \circ r \mid x) \leq s + mk + t + O(\log t) := s',$$

where $t' := \max\{\rho(n+m'), \mathsf{poly}(t)\}$ and $m' := mk + k$. By the property of the algorithm A (Item 1 in the lemma), the above implies that for any choice of $z \in \{0,1\}^{mk}$,

$$\mathop{\mathbf{Pr}}_{\substack{r \sim \{0,1\}^t \\ A}} \left[A\left(\mathsf{DP}_k(y; z) \circ r, x, 1^{s'}, 1^{t'}\right) = 1 \right] \geq \frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9}. \tag{35}$$

On the other hand, by the property of the algorithm A (Item 2 in the lemma), we get that

$$\mathop{\mathbf{Pr}}_{\substack{u \sim \{0,1\}^{nk+k} \\ r \sim \{0,1\}^t \\ A}} \left[A\left(u \circ r, x, 1^{s'}, 1^{t'}\right) = 0 \right] > \frac{2}{3} \cdot (1 - 1/t') > \frac{4}{9}. \tag{36}$$

Comparing Equation (35) and Equation (36), we get a randomized distinguisher for $\mathsf{DP}_k(y; \mathcal{U}_{nk})$ with advantage $1/12$, defined by sampling $r \sim \{0,1\}^t$, and outputting $A(- \circ$

$r, x, 1^{s'}, 1^{t'}$). By the reconstruction property of DP_k (Lemma 23), there exists some polynomial q such that

$$\mathsf{pK}^{q(t'+|x|)}(y \mid x) \leq k + \log q(t' + |x|).$$

Recall that $k = \mathsf{pnK}^t(y \mid x) + O(\log t)$. It follows that

$$\mathsf{pK}^{p(t+|x|)}(y \mid x) \leq \mathsf{pnK}^t(y \mid x) + \log p(t + |x|),$$

where p is some polynomial. ◀

Proof of Lemma 55. Given Lemma 56, it suffices to show that if $\mathsf{Gap\text{-}Cond\text{-}MINnKT} \in \mathsf{prBPP}$, then the assumption in Lemma 56 is true.

Assuming $\mathsf{Cond\text{-}MINnKT} \in \mathsf{prBPP}$, let B be a probabilistic polynomial-time algorithm and τ be a polynomial such that, given $(x, y', 1^{s'}, 1^{t'})$, B accepts with probability at least $2/3$ if $\mathsf{nK}^t(y' \mid x) \leq s'$ and rejects with probability at least $2/3$ if $\mathsf{nK}^{\tau(t+|x|)}(y' \mid x) > s' + \log \tau(t' + |x|)$.

We claim that B satisfies the conditions of an algorithm in Lemma 56. The first condition is immediate. For the second condition, note that by a simple counting argument, we have that with probability at least $1 - 1/t'$ over $y' \sim \{0,1\}^{m'}$,

$$\begin{aligned}\mathsf{nK}^{\tau(t')}(y' \mid x) &\geq \mathsf{K}(y' \mid x) \\ &\geq m' - O(\log t') \\ &> s' + \log \tau(t' + |x|),\end{aligned}$$

where the last inequality holds as long as $s' \leq m' - c \cdot \log t'$ for some sufficiently large constant c. In this case, we have that B rejects with probability at least $2/3$. ◀

We are now ready to prove Theorem 10.

Proof of Theorem 10. It is easy to see that $\mathsf{Cond\text{-}MINnKT}$ can be solved in NP using two (non-adaptive) calls to an NP oracle. Therefore, if $\mathsf{NP} \subseteq \mathsf{BPP}$, then $\mathsf{PH} \subseteq \mathsf{BPP}$ [26], which implies $\mathsf{Cond\text{-}MINnKT} \in \mathsf{BPP}$. This also implies $\mathsf{Gap\text{-}Cond\text{-}MINnKT} \in \mathsf{prBPP}$.

Next, we show that if $\mathsf{Gap\text{-}Cond\text{-}MINnKT} \in \mathsf{prBPP}$, then $\mathsf{NP} \subseteq \mathsf{BPP}$. Let $L \in \mathsf{NP}$. For an instance $x \in \{0,1\}^n$, let A_x denote the set of L-witnesses of x (with respect to some fixed verifier). Without loss of generality, we assume that every string in A_x has a length of exactly m, where $m = \mathsf{poly}(n)$.

By language compression for pnK^t (Theorem 4), we get that there is a polynomial p_{LC} such that for every string $y \in A_x$,

$$\mathsf{pnK}^{p_{\mathsf{LC}}(n), A_x}(y) \leq \log |A_x| + \log p_{\mathsf{LC}}(n).$$

Note that given x, the membership of A_x is decidable in polynomial time. Hence, we can efficiently answer oracle calls to A_x given x. This implies that there exists some polynomial p_1 such that for every $y \in A_x$,

$$\mathsf{pnK}^{p_1(n)}(y \mid x) \leq \mathsf{pnK}^{p_{\mathsf{LC}}(n), A_x}(y).$$

Furthermore, by Lemma 55, we have

$$\mathsf{pK}^{p(n)}(y \mid x) \leq \mathsf{pnK}^{p_1(n)}(y \mid x)$$

for some polynomial p. It follows that

$$\mathsf{pK}^{p(n)}(y \mid x) \leq \log |A_x| + \log p(n). \tag{37}$$

By Lemma 32, $\mathsf{USamp}(1^m, 1^{p(n)}, x)$ outputs a L-witness of x with probability at least $1/\mathsf{poly}(n)$. By standard amplification, this yields an efficient randomized algorithm for solving L with high probability. ◀

Proof of Corollary 11. First, by Theorem 1 and Theorem 10, the two statements "$\mathsf{MINnKT} \in \mathsf{BPP}$" and "$\mathsf{Gap\text{-}Cond\text{-}MINnKT} \in \mathsf{prBPP}$" are equivalent to "$\mathsf{NP} \subseteq \mathsf{BPP}$". To include "$\mathsf{Cond\text{-}MINnKT} \in \mathsf{prBPP}$" in this equivalence, observe that $\mathsf{Cond\text{-}MINnKT} \in \Sigma_2^{\mathsf{P}}$. Since $\mathsf{NP} \subseteq \mathsf{BPP}$ implies $\Sigma_2^{\mathsf{P}} \subseteq \mathsf{BPP}$, it follows from the easiness of $\mathsf{Gap\text{-}Cond\text{-}MINnKT}$ that $\mathsf{Cond\text{-}MINnKT} \in \mathsf{BPP}$. Finally, if $\mathsf{Cond\text{-}MINnKT} \in \mathsf{BPP}$, then it trivially follows that $\mathsf{Gap\text{-}Cond\text{-}MINnKT} \in \mathsf{prBPP}$. ◀

References

1 Luis Filipe Coelho Antunes, Lance Fortnow, Alexandre Pinto, and Andre Souto. Low-depth witnesses are easy to find. *Comput. Complex.*, 21(3):479–497, 2012. doi:10.1007/s00037-011-0025-1.

2 Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.

3 Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Found. Trends Theor. Comput. Sci.*, 2(1), 2006. doi:10.1561/0400000004.

4 Harry Buhrman, Lance Fortnow, and Sophie Laplante. Resource-bounded Kolmogorov complexity revisited. *SIAM Journal on Computing*, 31(3):887–905, 2001. doi:10.1137/S009753979834388X.

5 Harry Buhrman, Troy Lee, and Dieter van Melkebeek. Language compression and pseudorandom generators. *Comput. Complex.*, 14(3):228–255, 2005. doi:10.1007/s00037-005-0199-5.

6 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *Conference on Computational Complexity* (CCC), pages 10:1–10:24, 2016. doi:10.4230/LIPIcs.CCC.2016.10.

7 Lijie Chen, Shuichi Hirahara, and Neekon Vafa. Average-Case Hardness of NP and PH from Worst-Case Fine-Grained Assumptions. In *Innovations in Theoretical Computer Science Conference* (ITCS), volume 215, pages 45:1–45:16, 2022. doi:10.4230/LIPIcs.ITCS.2022.45.

8 Halley Goldberg and Valentine Kabanets. A simpler proof of the worst-case to average-case reduction for polynomial hierarchy via symmetry of information. *Electron. Colloquium Comput. Complex.*, 7:1–14, 2022. URL: https://eccc.weizmann.ac.il/report/2022/007/.

9 Halley Goldberg, Valentine Kabanets, Zhenjian Lu, and Igor C. Oliveira. Probabilistic Kolmogorov complexity with applications to average-case complexity. In *Computational Complexity Conference* (CCC), pages 16:1–16:60, 2022. doi:10.4230/LIPIcs.CCC.2022.16.

10 Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *Symposium on Theory of Computing* (STOC), pages 59–68, 1986. doi:10.1145/12130.12137.

11 Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *Symposium on Foundations of Computer Science* (FOCS), pages 247–258, 2018. doi:10.1109/FOCS.2018.00032.

12 Shuichi Hirahara. Characterizing average-case complexity of PH by worst-case meta-complexity. In *Symposium on Foundations of Computer Science* (FOCS), pages 50–60, 2020. doi:10.1109/FOCS46700.2020.00014.

13 Shuichi Hirahara. Non-disjoint promise problems from meta-computational view of pseudorandom generator constructions. In *Conference on Computational Complexity* (CCC), pages 20:1–20:47, 2020. doi:10.4230/LIPIcs.CCC.2020.20.

14 Shuichi Hirahara. Unexpected hardness results for Kolmogorov complexity under uniform reductions. In *Symposium on Theory of Computing* (STOC), pages 1038–1051, 2020. doi:10.1145/3357713.3384251.

15 Shuichi Hirahara. Average-case hardness of NP from exponential worst-case hardness assumptions. In *Symposium on Theory of Computing* (STOC), pages 292–302, 2021. doi:10.1145/3406325.3451065.

16 Shuichi Hirahara. Meta-computational average-case complexity: A new paradigm toward excluding Heuristica. *Bull. EATCS*, 136, 2022. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/688.

17 Shuichi Hirahara. NP-hardness of learning programs and partial MCSP. In *Symposium on Foundations of Computer Science* (FOCS), pages 968–979, 2022. doi:10.1109/FOCS54457.2022.00095.

18 Shuichi Hirahara. Symmetry of information from meta-complexity. In *Computational Complexity Conference* (CCC), pages 26:1–26:41, 2022. doi:10.4230/LIPIcs.CCC.2022.26.

19 Shuichi Hirahara, Rahul Ilango, Zhenjian Lu, Mikito Nanashima, and Igor C. Oliveira. A duality between one-way functions and average-case symmetry of information. In *Symposium on Theory of Computing* (STOC), pages 1039–1050, 2023. doi:10.1145/3564246.3585138.

20 Shuichi Hirahara, Valentine Kabanets, Zhenjian Lu, and Igor C. Oliveira. Exact search-to-decision reductions for time-bounded Kolmogorov complexity. In *Computational Complexity Conference* (CCC), pages 29:1–29:56, 2024. doi:10.4230/LIPIcs.CCC.2024.29.

21 Jinqiao Hu, Zhenjian Lu, and Igor C. Oliveira. Hardness of computing nondeterministic kolmogorov complexity. *Electron. Colloquium Comput. Complex.*, TR25, 2025. URL: https://eccc.weizmann.ac.il/report/2025/203.

22 Yizhi Huang, Rahul Ilango, and Hanlin Ren. NP-hardness of approximating meta-complexity: A cryptographic approach. In *Symposium on Theory of Computing* (STOC), pages 1067–1075, 2023. doi:10.1145/3564246.3585154.

23 Rahul Ilango. SAT reduces to the minimum circuit size problem with a random oracle. In *Symposium on Foundations of Computer Science* (FOCS), pages 733–742, 2023. doi:10.1109/FOCS57990.2023.00048.

24 Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference*, pages 134–147, 1995. doi:10.1109/SCT.1995.514853.

25 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Symposium on Theory of Computing* (STOC), pages 73–79, 2000. doi:10.1145/335305.335314.

26 Ker-I Ko. Some observations on the probabilistic algorithms and NP-hard problems. *Inf. Process. Lett.*, 14(1):39–43, 1982. doi:10.1016/0020-0190(82)90139-9.

27 Ker-I Ko. On the complexity of learning minimum time-bounded Turing machines. *SIAM J. Comput.*, 20(5):962–986, 1991. doi:10.1137/0220059.

28 Troy Lee. *Kolmogorov complexity and formula lower bounds*. PhD thesis, University of Amsterdam, 2006.

29 Troy Lee and Andrei E. Romashchenko. Resource bounded symmetry of information revisited. *Theor. Comput. Sci.*, 345(2-3):386–405, 2005. doi:10.1016/j.tcs.2005.07.017.

30 Yanyi Liu and Rafael Pass. On one-way functions and Kolmogorov complexity. In *Symposium on Foundations of Computer Science* (FOCS), pages 1243–1254, 2020. doi:10.1109/FOCS46700.2020.00118.

31 Yanyi Liu and Rafael Pass. On one-way functions from NP-complete problems. In *Conference on Computational Complexity* (CCC), pages 36:1–36:24, 2022. doi:10.4230/LIPIcs.CCC.2022.36.

32 Zhenjian Lu and Igor C. Oliveira. Theory and applications of probabilistic Kolmogorov complexity. *Bull. EATCS*, 137, 2022. URL: `http://bulletin.eatcs.org/index.php/beatcs/article/view/700`.

33 Zhenjian Lu, Igor C. Oliveira, and Marius Zimand. Optimal coding theorems in time-bounded Kolmogorov complexity. In *International Colloquium on Automata, Languages, and Programming* (ICALP), pages 92:1–92:14, 2022. `doi:10.4230/LIPIcs.ICALP.2022.92`.

34 Michael Sipser. A complexity theoretic approach to randomness. In *Symposium on Theory of Computing* (STOC), pages 330–335, 1983. `doi:10.1145/800061.808762`.

35 Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *Symposium on Foundations of Computer Science* (FOCS), pages 80–91, 1982. `doi:10.1109/SFCS.1982.45`.

A An Explicit Computational Model

In this section, we describe one possible way of defining a computational model and a modified encoding of programs that satisfy the properties stated in Section 2.2.

We consider multi-tape Turing machines over the alphabet $\Sigma = \{0, 1, \perp\}$. In addition to the usual input tape and work tapes, we allow a machine to access three additional tapes: one holding an auxiliary string, one holding a non-deterministic guess, and one holding the randomness. We assume that the machine has *sequential access* (i.e., no random access) to the randomness tape and the non-deterministic tape.[18] This assumption will be useful to guarantee "Randomness and Non-Determinism Extension". The remaining details of the model can be arbitrary.

We also assume an encoding function $\langle\cdot\rangle$ for machines (which we omit for simplicity of notation) such that there is a time-efficient universal machine U which, given (the encoding of) a machine M, an input α, an auxiliary string x, a non-deterministic guess w, and the random string r, runs in time $\mathsf{poly}(|M|, t)$ and outputs the string produced by $M(\alpha; x; w; r)$, where $|M|$ denote the encoding length of M and t is the running time of M on $(\alpha; x; w; r)$. This assumption will be useful to guarantee "Efficient Simulation".

In several previous papers, a program is defined as a pair (M, α), where M is a machine and $\alpha \in \{0,1\}^*$ is an "advice" string. However, this definition of programs alone, which does not provide further details about the encoding of programs, might not satisfy the conditions for being "strongly efficient", as stated in Section 2.2. Instead, we present a generic way of obtaining the desired properties given an existing encoding, which will be useful to guarantee "Efficient Composition". We consider *a sequence of machine-advice pairs*. More formally, a program Π is defined as

$$\Pi := ((M_1, \alpha_1), \ldots, (M_\ell, \alpha_\ell)),$$

where each M_i is a valid encoding of a machine and $\alpha_i \in \{0,1\}^*$. The output of Π on $(x; w; r)$ (which corresponds to the auxiliary input, the non-deterministic guess, and the randomness), denoted as $\Pi(x; w; r)$, is obtained as follows:[19]

- Let y_1 be the output of $M_1(\alpha_1; x; w; r)$. Also, let t_1 be the number of steps $M_1(\alpha_1; x; w; r)$ takes to produce y_1.

[18] In order to avoid confusion, sequential access means that the tape head can remain at a given tape cell or move left or right in each time step of the computation. In particular, we do not require read-only access or one-way access to the tape.

[19] For a string z of length ℓ and a subset $S \subseteq [\ell]$, where $[\ell] = \{1, 2, \ldots, \ell\}$, we let w_S denote the substring of z obtained by concatenating the bits of z corresponding to indexes in S. For $a < b$, we let $[a : b]$ denote the set $\{a, a+1, \ldots, b\}$.

- Let y_2 be the output of $M_2\big(\alpha_2; x \circ y_1; w_{[t_1+1:|w|]}; r_{[t_1+1:|r|]}\big)$ and t_2 be the corresponding running time.

$$\vdots$$

- Let y_ℓ be the output of $M_\ell\big(\alpha_\ell; x \circ y_{\ell-1}; w_{[t_{\ell-1}+1:|w|]}; r_{[t_{\ell-1}+1:|r|]}\big)$ and t_ℓ be the corresponding running time.
- Output y_ℓ.

Additionally, in each of the above steps, if $\perp$ is produced, Π outputs $\perp$ and halts.

The running time of the program Π on input (x, w, r) is given by $\sum_{i=1}^{\ell} t_i$.

We now describe how the program Π is encoded. To encode a machine-advice pair (M, α), we use the following format:

$$s_1\, 01\, \langle M \rangle\, s_2\, 01\, \alpha,$$

where $\langle M \rangle$ is an encoding of the machine M,[20] and s_1 is (resp. s_2) the binary representation of the integer specifying the length of $\langle M \rangle$ (resp. α), with each bit duplicated. The full description of Π is obtained by concatenating the encodings of all pairs (M_i, α_i) for $i \in [\ell]$. The size of Π, denoted $|\Pi|$, is the total bit length of this description.

▶ **Proposition 57.** *The computational model described above is strongly efficient, as specified by Definition 12.*

Proof. The "Efficient Simulation" property follows from the fact that we use a time-efficient universal machine U that can simulate any machine with only a polynomial overhead.

The "Randomness Extension" property follows from the fact that a machine only has sequential access to the randomness tape. Indeed, if a program has a running time of at most t on a random string r with $|r| \geq t$, then since it only has sequential access to r, it will never reach the end of r during its execution, and appending any string to r will result in exactly the same output behavior. Similarly, the model also admits "Non-Determinism Extension".

For the "Efficient Universality" property, suppose we have a machine M and $\alpha_0, x, w, r \in \{0,1\}^*$ such that $M(\alpha_0; x; w; r)$ outputs y in t steps. Consider the program $\Pi := (M', \alpha)$, where M' and α are defined as follows:

$\alpha := (M, \alpha_0)$, and M' is a machine that performs the following operation: on input $(\alpha = (M, \alpha_0); x; w; r)$, it simulates $M(\alpha_0; x; w; r)$.

It is straightforward to verify that the description length of Π is at most $|\alpha_0| + c_M \cdot \log |\alpha_0|$ for some constant $c_M > 0$ that depends only on M. Moreover, it follows from the construction that $\Pi(x; w; r)$ outputs $M(\alpha_0; x; w; r) = y$ in time t^{c_M}, provided that c_M is chosen to be sufficiently large.

Finally, we argue that the "Efficient Composition" property also holds. Let $x, y, z, r \in \{0,1\}^*$. Suppose there is a program $\Pi_1 = ((M_1, \alpha_1), \ldots, (M_\ell, \alpha_\ell))$ of size s_1 that computes $(x; r) \to y$ non-deterministically in time t_1, and a program Π_2 of size s_2 that computes $y \to z$ non-deterministically in time t_2. We first define the following machine-advice pair (M, α):

$\alpha := (\Pi_2, m)$, where $m \in \mathbb{N}$ is the length of y, and M is a machine that does the following: given $(\alpha; x \circ y; w'; r')$, simulate $\Pi_2(y; w'; \epsilon)$.

[20] As noted above, we assume a fixed standard encoding for Turing machines.

Our final program Π for computing $(x;r) \to z$ is defined as Π_1 extended by (M,α), i.e.,

$$\Pi := ((M_1,\alpha_1),\dots,(M_\ell,\alpha_\ell),(M,\alpha))\,.$$

First of all, it is easy to verify that the description of Π has length at most $s_1 + s_2 + \log p(|y|+|z|)$ for some sufficiently large polynomial p. To see that Π computes $(x;r) \to z$, note that after the execution of Π_1 on $(x;r)$, we have non-deterministically computed y, and the running time up to this point is at most t_1. After that, $M(\alpha, x \circ y; w'; r')$ will be run, where w' is the remaining part of the string on the non-deterministic tape and r' is the remaining part on the randomness tape. By construction, the (sub-)program (M,α) on $x \circ y$ will (non-deterministically) output z, as desired. Also, note that due to efficient simulation, $M(\alpha, x \circ y; w'; r')$ runs in time at most $p(t_2) + p(|x|+|y|+|z|)$. Therefore, the running time of the program Π on $(x;r)$ is at most $t_1 + p(t_2) + p(|x|+|y|+|z|)$. ◀

B Worst-Case to Average-Case Reductions and Gap-MINpnKT

For $\tau\colon \mathbb{N} \to \mathbb{N}$, let Gap_τ–$\mathsf{MINpnKT}$ be the following problem: Given $(x, 1^s, 1^t)$, where $x \in \{0,1\}^*$ and $s,t \in \mathbb{N}$, decide whether $\mathsf{pnK}^t(x) \le s$ or $\mathsf{pnK}^{\tau(t)}(x) > s + \log \tau(t)$. We say that $\mathsf{Gap\text{-}MINpnKT} \in \mathsf{prBPP}$ if there is a probabilistic polynomial-time algorithm that solves $\mathsf{Gap}_\tau\text{-}\mathsf{MINpnKT}$ for some polynomial τ.

▶ **Proposition 58.** *The following holds.*

$$(\mathsf{MINnKT}, \mathcal{U}) \in \mathsf{AvgBPP} \implies \mathsf{Gap\text{-}MINpnKT} \in \mathsf{prBPP}.$$

Proof. First of all, suppose $(\mathsf{MINnKT}, \mathcal{U}) \in \mathsf{AvgBPP}$. Then by Lemma 48, we have that $(\mathsf{coMINnKT}, \mathcal{U}) \in \mathsf{Avg}^1\mathsf{BPP}$. In other words, there exist a constant $c > 0$, a polynomial ρ, and a probabilistic polynomial-time algorithm A such that the following hold for all sufficiently large n', all $t' \ge \rho(n')$, and all $s' \le n' - c \cdot \log t'$:

1. For every $x \in \{0,1\}^{n'}$ with $\mathsf{nK}^t(x) \le s'$, we have $\mathbf{Pr}_A[A(x, 1^{s'}, 1^{t'}) = 1] \ge 1 - \frac{1}{10t'}$.
2. With probability at least $1 - 1/t'$ over $x \sim \{0,1\}^{n'}$, we have $\mathbf{Pr}_A[A(x, 1^{s'}, 1^{t'}) = 0] \ge 1 - \frac{1}{10t'}$.

Let $d > 0$ be a sufficiently large constant and consider the following parameters.

- $k := s + d^3 \cdot \log t$.
- $n' := k + t^d$.
- $t' := t^{2d}$.
- $s' := s + O(\log\log t') + t^d + nk + d^2 \log t$.

Define an algorithm A' as follows:

On input $(x, 1^s, 1^t)$ with $x \in \{0,1\}^n$, sample $z \sim \{0,1\}^{nk}$ and $w \sim \{0,1\}^{t^d}$, and then output $A\Big(\mathsf{DP}_k(x;z) \circ w, 1^{s'}, 1^{t'}\Big)$.

Below, we show that A' solves $\mathsf{Gap}_\tau\text{-}\mathsf{MINpnKT}$ correctly with high probability, where τ is a polynomial specified later.

First, consider the case that $(x, 1^s, 1^t)$ is a yes-instance of $\mathsf{Gap}_\tau\text{-}\mathsf{MINpnKT}$, i.e., $\mathsf{pnK}^t(x) \le s$. By success amplification (Lemma 29), we get that

$$\mathsf{pnK}^{t^d}_{1-1/10n'}(x) \le s + O(\log\log t').$$

In other words, for at least $1 - 1/(10n')$ of the randomness $w \in \{0,1\}^{t^d}$, there exists a *non-deterministic* program M of length at most $s + O(\log\log t')$ that outputs x within t^d steps. Note that this implies that for any choice of $z \in \{0,1\}^{nk}$,

$$\Pr_{w\sim\{0,1\}^{t^d}}\left[\mathsf{nK}^{t^{2d}}(\mathsf{DP}_k(y;z)\circ w) \le s + O(\log\log t') + t^d + nk + O(d\log t)\right] \ge 1 - \frac{1}{10n'}. \quad (38)$$

Also, note that by letting d be a sufficiently large constant, we have

$$s + O(\log\log t') + t^d + nk + O(d\log t) \le s'. \quad (39)$$

Equation (38) and Equation (39) together imply that

$$\Pr_{w\sim\{0,1\}^{t^d}}\left[\mathsf{nK}^{t^{2d}}(\mathsf{DP}_k(y;z)\circ w \mid x) \le s'\right] \ge 1 - \frac{1}{10t'}.$$

Then by the property of A and a union bound, we have

$$\Pr_{w,z,A}\left[A(\mathsf{DP}_k(x;z)\circ w, 1^{s'}, 1^{t'}) = 1\right] \ge 1 - \frac{1}{5t'},$$

and so

$$\Pr_{A'}\left[A'(x, 1^s, 1^t) = 1\right] \ge 1 - \frac{1}{5t'}. \quad (40)$$

Now consider any $(x, 1^s, 1^t)$ that is a no-instance of $\mathsf{Gap}_\tau\text{-}\mathsf{MINpnKT}$, i.e., $\mathsf{pK}^{\tau(t)}(X) > s + \log\tau(t)$. We will show that

$$\Pr_{A'}\left[A'(x, 1^s, 1^t) = 1\right] \le 1 - \frac{2}{5t'}. \quad (41)$$

Note that by combining Equation (40) and Equation (41), A' yields a polynomial-time algorithm for solving $\mathsf{Gap}_\tau\text{-}\mathsf{MINpnKT}$ via standard success amplification techniques.

Suppose, for the sake of contradiction, Equation (41) is not true. Then by the definition of A', we have

$$\Pr_{w,z,A}\left[A(\mathsf{DP}_k(x;z)\circ w, x, 1^{s'}, 1^{t'}) = 1\right] > 1 - \frac{2}{5t'}. \quad (42)$$

On the other hand, by the property of A, we get

$$\Pr_{u,w,A}\left[A(u\circ w, 1^{s'}, 1^{t'}) = 0\right] \ge 1 - \frac{1}{2t'}. \quad (43)$$

Comparing Equation (42) and Equation (43), we get a randomized procedure defined as $B(-\circ \mathcal{U}_{t^d}, 1^{s'}, 1^{t'})$ that distinguishes $\mathsf{DP}_k(x;\mathcal{U}_{O(nk)})$ from $\mathcal{U}_m$ with advantage $1/(10t')$. By Lemma 23 and by letting τ be a sufficiently large polynomial, we get

$$\begin{aligned}\mathsf{pK}^{\tau(t)}(x) &\le k + O(\log t')\\ &\le s + d^3\cdot\log t + O(d\log t)\\ &\le s + \log\tau(t).\end{aligned}$$

This means $(x, 1^s, 1^t)$ is *not* a no-instance of $\mathsf{Gap}_\tau\text{-}\mathsf{MINpnKT}$, which gives the desired contradiction. ◀

▶ **Corollary 59.** *Suppose*

$$\mathsf{Gap\text{-}MINpnKT} \in \mathsf{prBPP} \implies \mathsf{NP} \subseteq \mathsf{BPP}.$$

Then

$$\mathsf{Dist}\Sigma_2^{\mathsf{P}} \subseteq \mathsf{AvgBPP} \implies \mathsf{NP} \subseteq \mathsf{BPP}.$$

Proof. If $\mathsf{Dist}\Sigma_2^{\mathsf{P}} \subseteq \mathsf{AvgBPP}$, then $(\mathsf{MINnKT}, \mathcal{U}) \in \mathsf{AvgBPP}$, since $\mathsf{MINnKT} \in \Sigma_2^{\mathsf{P}}$. By Proposition 58, we get $\mathsf{Gap\text{-}MINpnKT} \in \mathsf{prBPP}$. Finally, by the assumption, we get $\mathsf{NP} \subseteq \mathsf{BPP}$. ◀

Consequently, if Gap-MINpnKT also captures the hardness of NP (extending Theorem 9), a breakthrough result relating the average-case and worst-case complexities of the polynomial hierarchy would follow.

The Rate-Immediacy Barrier in Explicit Tree Code Constructions

Gil Cohen
Department of Computer Science, Tel Aviv University, Israel

Leonard J. Schulman
E&AS Division, California Institute of Technology, Pasadena, CA, USA

Piyush Srivastava
Tata Institute of Fundamental Research, Mumbai, India

Abstract

Since the introduction of tree codes by Schulman (STOC 1993), explicit construction of asymptotically good tree codes has remained a notorious challenge. A work by Cohen, Haeupler and Schulman (STOC 2018), as well as the state-of-the-art construction by Ben Yaacov, Cohen, and Yankovitz (STOC 2022) have achieved codes with rate $\Omega(1/\log\log n)$, exponentially improving upon the original rate $\Omega(1/\log n)$ construction of Evans, Klugerman and Schulman from 1994. All of these constructions rely, at least in part, on increasingly sophisticated methods of combining (block) error-correcting codes.

In this work, we identify a fundamental barrier to constructing tree codes using known techniques. We introduce a key property which we call *immediacy*, that, while not required by the original definition of tree codes, is shared by all known constructions and inherently arises in recursive combinations of error-correcting codes. Our main technical contribution is the proof of a *rate–immediacy trade-off*, which, in particular, implies that any tree code with constant distance and non-trivial immediacy must necessarily have vanishing rate. By applying our rate-immediacy trade-off to existing constructions, we establish that their known rate analyses are essentially optimal given their actual error-correction properties. More broadly, our work highlights the need for fundamentally new ideas – beyond the recursive use of error-correcting codes – to achieve substantial progress in explicitly constructing asymptotically good tree codes.

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

Keywords and phrases Tree codes, Information Theory

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.10

Related Version *Arxiv Version*: https://arxiv.org/abs/2504.09388

Funding *Gil Cohen*: Supported by ERC starting grant 949499.
Leonard J. Schulman: Supported in part by NSF Award 2321079.
Piyush Srivastava: Supported in part by the Department of Atomic Energy, Government of India, under project nos. RTI4001 and RTI4014, by SERB MATRICS grant number MTR/2023/001547, and by the Infosys-Chandrasekharan virtual center for Random Geometry

1 Introduction

Coding theory addresses the problem of communication over an imperfect channel. In the classic setting [19, 31], Alice aims to communicate a message to Bob via a channel that may introduce errors. The central question is: how should Alice encode her message so that Bob can accurately recover it, provided that the number of errors is limited? This scenario motivates the notion of an error-correcting code. Formally, a function $C\colon \Sigma^k \to \Sigma^n$ is a *block error-correcting code* with distance δ and block length k if, for every pair of distinct strings $x, y \in \Sigma^k$, their respective encodings $C(x)$ and $C(y)$ differ in at least a δ fraction

© Gil Cohen, Leonard J. Schulman, and Piyush Srivastava;
licensed under Creative Commons License CC-BY 4.0

of positions, with respect to the Hamming distance. The *rate* of information transmission $\rho = \frac{k}{n}$ and the fraction of errors corrected, $\frac{\delta}{2}$, are competing parameters. A fundamental problem in coding theory is to construct explicit *asymptotically good codes*, i.e., codes that maintain constant distance $\delta > 0$ and constant rate $\rho > 0$. Here, by "explicit" we mean that the encoding function C can be computed in polynomial time. Justesen [20] was the first to provide such an explicit construction. Since then, many explicit constructions have been developed (see, e.g., [32, 33]).

While error-correcting codes solve the problem of sending a single message from Alice to Bob, there are scenarios involving dynamic interaction, where messages exchanged depend on previously communicated information. Interactive coding addresses this more intricate problem of enabling reliable interactive communication over imperfect channels. Standard error-correcting codes are insufficient for this task. Readers interested in this rapidly growing research field are encouraged to consult the comprehensive survey by Gelles [13].

Tree codes are crucial combinatorial structures introduced in [28, 29] to facilitate deterministic interactive coding. Analogous to error-correcting codes in single-message scenarios, tree codes are trees equipped with a specific distance property. To define this formally, we first introduce some notation. Let T be an infinite complete rooted binary tree, with edges labeled from an alphabet Σ. For two vertices u, v of equal depth, let their least common ancestor be denoted w. Let ℓ be the number of edges on the path from u (or v) to w. Define $p_u, p_v \in \Sigma^\ell$ as the sequences of symbols labeling the edges on the paths from w to u and from w to v, respectively. The quantity $h(u, v)$ denotes the relative Hamming distance between p_u and p_v. Intuitively, $h(u, v)$ measures how distinct the sequences labeling the paths to u and v are, disregarding their common prefix. A tree code enforces a minimum bound on this quantity:

▶ **Definition 1** (Tree codes [28]). *Let $\delta \in [0, 1]$ and let T be an infinite complete rooted binary tree. A labeling of the edges of T from an alphabet Σ is called a* tree code with distance δ *if, for every pair of vertices u, v at the same depth, it holds that $h(u, v) \geq \delta$. The* rate *of the tree code, denoted by ρ, is defined as $\frac{1}{\log_2 |\Sigma|}$.*

An alternative definition found in the literature describes a tree code as a family $(T_n)_{n \in \mathbb{N}}$, where each T_n is a rooted binary tree of *finite* depth n. Such a family is called a tree code with distance δ if each T_n has a distance of at least δ, as previously defined. Clearly, an infinite tree code naturally induces such a finite family when truncated at any given depth. Conversely, it has been shown that the opposite direction also holds [2], with only a constant degradation in parameters. Hence, we use these two definitions interchangeably throughout this informal discussion.

Initially, it was not clear whether an asymptotically good tree code – one with both positive rate and positive distance – existed. Schulman provided three distinct proofs showing that, for any constant $\delta < 1$, a tree code exists with alphabet size $|\Sigma| = O_\delta(1)$ that achieves distance δ. More recently, by adapting one of these approaches, it was proved that there is a tree code with just $|\Sigma| = 4$ symbols, namely a rate-$\frac{1}{2}$ tree code, and positive distance (specifically, $\delta > 0.136$) [9]. It was also observed in [9] that 3 symbols are insufficient for guaranteeing positive distance. However, all known existence proofs of asymptotically good tree codes are non-explicit, relying on probabilistic methods in non-trivial ways. Despite significant interest, the explicit construction of asymptotically good tree codes remains a notorious open problem.

Given these difficulties, researchers have naturally considered constructing tree codes allowing for vanishing rate, where the objective is nonetheless to minimize the rate deterioration as a function of the depth n.[1] A trivial construction which encodes the entire path from the root on each edge, achieves $\delta = 1$ but rate $\frac{1}{n}$. In an unpublished manuscript, Evans, Klugerman, and Schulman [12] provided a construction with rate $\Omega(1/\log n)$. An improvement was made only fairly recently by Cohen, Haeupler and Schulman [8], who constructed tree codes with rate $\Omega(1/\log\log n)$. A decoding algorithm to the latter tree code construction was devised by Narayanan and Weidner [25], who also suggested alternative constructions. Connections between [8] and the work of Pudlák [26] (see also [34] for a related construction) were explored by Bhandari and Harsha [3]. Additionally, two distinct explicit constructions with constant rate were proposed by Moore and Schulman [23] and by Ben Yaacov, Cohen, and Narayanan [1], but their correctness hinges upon plausible but unproven conjectures.

The state-of-the-art construction by Ben Yaacov, Cohen, and Yankovitz [2] also achieves a rate of $\Omega(1/\log\log n)$ for constant distance δ, but it improves upon the dependence on the distance parameter. Specifically, while the construction of [8] has a rate that is upper bounded by $O(1/\log\log n)$ regardless of the value of δ, the construction of [2] achieves rate $\Omega(1/(\sqrt{\delta}\cdot\log\log n))$. In particular, it can attain *constant* rate by compromising on the distance, achieving a distance of $\delta = \Omega(1/(\log\log n)^2)$. This related, "dual" problem of constructing a constant-rate tree code with slowly deteriorating distance was first studied by Gelles, Haeupler, Kol, Ron-Zewi, and Wigderson [14], who showed how to achieve distance $\Omega(1/\log n)$.

1.1 Immediacy

Given that we still have no explicit construction of asymptotically good tree codes – despite significant efforts for over three decades – it is important to better understand why current techniques fall short. Identifying a barrier may help steer research away from potential dead ends. More specifically, since tree codes induce standard block error-correcting codes, while the latter are heavily relied upon in all known constructions (except for the two conjecture-based approaches [1, 23]), it is natural to ask: to what extent might constant-rate tree codes be constructed from block codes? Exploring this question may shed light on the fundamental nature of tree codes and clarify how "close" they truly are to their well-understood counterparts, block error-correcting codes.

The main contribution of this paper is identifying a fundamental barrier in the construction of tree codes. We observe that each of the existing explicit constructions of tree codes (except those relying on unproven conjectures) satisfies a stronger guarantee than strictly required for a tree code; we refer to this guarantee as *immediacy*. This stronger property arose inherently due to the recursive nature of existing constructions [2,8,12,25]. We prove that the immediacy property incurs a cost in rate. Specifically, we establish a quantitative *rate-immediacy trade-off*, precisely capturing the achievable rate given a code's immediacy. Our results show that existing tree code constructions essentially achieve an optimal rate-immediacy balance, thus clarifying why their rate inevitably vanishes. More broadly, our work indicates that the typically employed framework, that of tiling standard error-correcting block codes in progressively sophisticated ways, cannot yield asymptotically good tree codes. Such strategies inherently satisfy the immediacy property, constraining their potential rate.

[1] We sometimes also refer to the depth n of a tree code as its *transmission length*.

We formally define immediacy in detail later, as its precise definition is somewhat technical, involving a certain set system (see the definition of *immediacy codes* in Section 3). However, to capture the essence, we first provide an informal, simplified definition in which immediacy is represented by a monotone-increasing function $\mathrm{Imm} : \mathbb{N} \to \mathbb{N}$. A tree code C is said to have immediacy Imm if, for every pair of messages $x, y \in \Sigma^n$ (where n is the transmission length), every index s where $x_s \neq y_s$, and each integer $k \geq 1$, the relative Hamming distance between $C(x)$ and $C(y)$, restricted to the interval $[s, s + \mathrm{Imm}(k))$ (provided that $s + \mathrm{Imm}(k)$ is no larger than the depth of the tree, if finite), is bounded below by δ.

The definition of a standard tree code imposes no non-trivial immediacy constraint. Disagreements between corresponding codewords are required to accumulate only starting from the *first* index s at which the two messages x and y differ. In contrast, a code admitting an immediacy function requires that disagreements accumulate starting from *every* index s for which $x_s \neq y_s$, at interval lengths determined by the immediacy function Imm. Put differently, unlike a standard tree code, a tree code with immediacy function Imm supports *cold-start decoding*. That is, if during any interval $[s, s + \mathrm{Imm}(k))$ the noise rate is below $\delta/2$, then the receiver can decode the s-th bit – even if the noise prior to time s was high and bits prior to s cannot yet be reconstructed. While this property may be advantageous in certain applications (and is typical of weakly-encoded communications such as human speech or packetized transmissions), our results show that it is an "expensive" feature that cannot be afforded when aiming to construct asymptotically good tree codes.

To take an example, it can be shown by inspection that the first tree code construction [12], which has a rate of $\Omega(1/\log n)$ (where n is the depth of the tree), possesses a non-trivial immediacy property that corresponds roughly to $\mathrm{Imm}(k) = 2^{\Theta(k)}$, which we denote throughout this informal presentation as $\exp(k)$. At this informal stage, it is convenient to regard Imm as a real-valued function – which in this case is $\mathrm{Imm}(x) = \exp(x)$ – thus extending the original discrete definition. With this, our main result can be informally stated as follows:

▶ **Theorem 2** (Main result; informal)**.** *Let T be a depth-n tree code with constant distance and immediacy function* Imm*. Then, subject to some technical conditions*[2] *on* Imm*, that are satisfied for the cases* $\mathrm{Imm}(k) = \exp(k)$ *and* $\mathrm{Imm}(k) = \exp(\exp(k))$*, the rate of T satisfies*

$$\rho = O\left(\frac{1}{\mathrm{Imm}^{-1}(n)}\right).$$

Recall that for the construction given in [12], the immediacy corresponds roughly to $\mathrm{Imm}(x) = \exp(x)$ and so $\mathrm{Imm}^{-1}(x) = \Theta(\log x)$. Thus, Theorem 2 indicates that the rate of such a construction cannot exceed $O(1/\log n)$ – which matches its known rate, up to the constant. Theorem 2 also explains the rate achieved by the state-of-the-art tree code constructions [2, 8, 25]. Upon inspection, these constructions – similarly to [12] – also rely on composition of error-correcting codes and possess a non-trivial, though exponentially weaker, immediacy that corresponds roughly to $\mathrm{Imm}(x) = \exp(\exp(x))$. Thus, Theorem 2 indicates that the rate of these constructions cannot exceed $O(1/\log\log n)$, matching the proven rate of these codes.

[2] For instance, it turns out that exponential immediacy (i.e. $\mathrm{Imm}(k) = \exp(k)$ as described above) implies *linear* immediacy. Consequently, Theorem 2 should be interpreted as applying only to exponential (or faster-growing) immediacy.

Weaker immediacy properties

We emphasize here, however, that the precise immediacy properties enjoyed by the [12] and [8] constructions are slightly weaker than that implied by an immediacy function, and so Theorem 2 does not *formally* apply to them. Roughly speaking, these constructions only admit cold-start decoding after a "lag". We show in Section 5 that our general framework of *immediacy codes* and our general rate upper bound results in Theorems 8 and 9 discussed in Section 3.1 are powerful enough to extend to constructions with such weaker forms of immediacy. Thus, as shown in Section 5, the rate upper bounds indicated by Theorem 2 indeed apply to the above constructions. (Further, as we show in Section 3.2, constructions satisfying the simplified immediacy condition described here are also immediacy codes with appropriate parameters and our general rate upper bounds proved in Theorems 8 and 9 therefore also apply to them.)

More generally, immediacy should be kept in mind when constructing tree codes: immediacy properties of a proposed construction can be analyzed in order to obtain an upper bound on the rate it can achieve. (Further, the above discussion also shows that a rough analysis along the lines of Theorem 2 may serve as a useful guide before a formal analysis of the exact immediacy properties of a proposed construction.) It is worth noting that the two constructions whose analyses rely on unproven conjectures [1, 23] do not appear to exhibit non-trivial immediacy. Therefore, they are not excluded from being asymptotically good codes by this work.

Rate–distance–immediacy trade-off

We point out that our rate upper bounds apply also to codes with vanishing distance. In fact, for a general relative distance $\delta > 0$, the upper bound on the rate ρ stated in Theorem 2 exhibits, roughly, an inverse-linear dependence on δ. The precise details are more nuanced, and we direct the reader to the formal statement of our main result in Theorem 8, and to the computations leading to Equations (10) and (11) (that show that fast growing immediacy functions imply an appropriate immediacy codes condition). The key takeaway, however, is that Theorem 2, in its full generality, yields a *rate–distance–immediacy trade-off*. For ease of presentation, we choose to focus on the trade-off between rate and immediacy in this introductory section (see Section 1.3 for further discussion).

1.2 Related work

Randomized vs. deterministic encoding schemes

Randomized hashing schemes were employed for online and interactive communication [27] before deterministic schemes were made possible by tree codes. Randomized schemes do not suffer a trade-off between rate and immediacy. Computationally efficient encoding and decoding was shown for such schemes in [4, 6, 16]; in a considerable accomplishment Brakerski, Kalai, and Saxena [5] later showed that, with access only to explicit (i.e., polynomial-time encodable) tree codes, an interactive communication can be performed at the rate of those tree codes, while also performing all necessary decoding in polynomial time.

Tree code variants

The original motivation for tree codes came from the question of developing analogues of Shannon's channel coding theorem (for one-way communication) for situations involving interaction. Using the existence of tree codes, such an interactive coding theorem was proved

by Schulman [29, Theorem 2]. Later work by Gelles, Moitra, and Sahai [15] showed that such an interactive coding theorem can also be established using a weaker variant of tree codes that they called *potent tree codes*. On the other hand, stronger variants of tree codes featuring local testability were recently introduced by Moud, Rosen, and Rothblum [24]. It is interesting to compare this local testability guarantee with the immediacy property of tree codes, as immediacy also provides a form of local guarantee, albeit of a seemingly different nature. Lastly, we mention that a signal-processing analogue of tree codes was introduced by Schulman and Srivastava [30], and a variant of tree codes, dubbed *palette-alternating tree codes* which allows one to bypass the $\frac{1}{2}$-rate barrier was introduced by Cohen and Samocha [9].

1.3 Technical overview and organization of the paper

We follow the tradition of using information theoretic tools for bounding capabilities of codes. Information theory had its genesis in understanding fundamental limitations on source and channel coding in the stochastic error model. However, it has also been useful in understanding limitations on codes in the "Hamming model" of a bounded number of worst case errors, including in the setting of more modern coding-theoretic notions such as the notion of local decoding analyzed by Katz and Trevisan [22, Section 3.1], as well as its more delicate relaxed variant analyzed by Gur and Lachish [18] (see also [11, 17]). In our application to understanding immediacy properties of tree codes, the crucial ingredient turns out to be a careful accounting of the "common information" between different parts of the codeword, performed simultaneously at different length scales.

The main conceptual contribution of this paper, introduced in Section 3, is the notion of *immediacy codes*. The notion of immediacy codes provides a more robust formulation of the informal notion of immediacy functions discussed in the introduction (the close connection between the two notions is further explored in Section 3.2), and also allows us to smoothly carry out the accounting of common information alluded to above. The main information theoretic tool we need for this accounting is a form of the data processing inequality (Lemma 10)[3] to quantify "common information" between different parts of a code word. Our main technical result is that *immediacy codes cannot have high rates* (Theorem 8): this is proved in Section 4. The principle underlying the definition of immediacy codes and the proof of Theorem 8 is a careful accounting of the amount of "information" that needs to be duplicated in order to provide the immediacy guarantees that the code provides. We emphasize that once we have set up the definition of immediacy codes and the requisite form of the data processing inequality, the proof of Theorem 8 is technically quite simple.

Finally, in Section 5, we apply our framework to study the immediacy properties of the tree code constructions proposed in [12], [8] and [14]. As discussed earlier, the immediacy properties of these constructions differ slightly from the informal notion introduced above. For the first two constructions, we show that the rates achieved by these codes are optimal – up to constant factors – with respect to the immediacy guarantees they provide. In Section 5.1.1, we consider a method due to Braverman [7] for constructing depth-n good tree codes in time sub-exponential in n, and show that the rate-runtime trade-off obtained by the best currently known version of this construction is essentially the best possible given the immediacy guarantees achieved by the construction.

[3] This corollary of the (proof of the) data processing inequality also underlies the formulation of the *Gács-Körner common information* (see, e.g., [21, Section III]), but we do not need this connection for this paper.

The construction of [14] attempts to tackle a different trade-off: it achieves an $\Omega(1/\log n)$ distance with constant rate. We show that for the immediacy guarantee that this construction provides, no constant rate tree code can improve the distance to $\omega\left(\frac{\log\log n}{\log n}\right)$. This establishes that the rate/distance trade-off achieved by this construction is close to the best possible given the immediacy guarantees it achieves.

We do not carry out a full analysis of the immediacy properties of the constructions in [2, 25]. The construction of [25] employs the same "code tiling" structure as [8]; the main difference lies in the "outer" large-alphabet tree code used. Since immediacy is determined primarily by the code tiling aspect, the analysis of [25] proceeds similarly to our analysis of [8]. The construction of [2], in contrast, uses a somewhat different code-tiling strategy. Nevertheless, a variant of our analysis for [8] can still be adapted to study its immediacy properties. Lastly, we note that the construction of [14] is similar to that of [12] with respect to the code-tiling component. As such, our result – when viewed not as a rate–immediacy trade-off but rather as a distance–immediacy trade-off – can be used to establish an upper bound on the distance achievable by the construction.

We begin in the next section with some technical preliminaries.

2 Preliminaries

Notation

We will view strings s of length ℓ as indexed by integers in $\{1, 2, \ldots, \ell\} =: [\ell]$. Given such a string, and a subset $A \subseteq [\ell]$ we denote by s_A the sub-sequence of s obtained by taking the letters at the indices in A. We will sometimes overload the usual notation for real intervals (e.g. $[a, b], (a, b]$ etc.) to refer only to the *integers* in those intervals: this overloading should always be clear from the context. We denote $\log_2 x$ by $\lg x$ for any positive real x.

2.1 Information theory

We use standard information theoretic notation for entropy, conditional entropy and mutual information (see, e.g., [10]), which we now proceed to review. Given a random variable X taking values in some finite set Ω, its *entropy* $H(X)$ is defined as $-\sum_{\omega\in\Omega} \Pr[X = \omega] \cdot \lg \Pr[X = \omega]$ (so we measure entropy in bits), with the convention $0 \cdot \lg 0 = 0$. Given two random variables X and Y, both taking values in (possibly different) finite sets, the *conditional entropy* $H(X|Y)$ is defined as

$$H(X|Y) := -\sum_{x,y} \Pr[X = x, Y = y] \cdot \lg \Pr[X = x|Y = y]. \tag{1}$$

The *mutual information* $I(X : Y)$ is then defined as

$$I(X : Y) = H(X) - H(X|Y). \tag{2}$$

If Z is another random variable taking values in a finite set, the *conditional mutual information* $I(X : Y|Z)$ is defined as

$$I(X : Y|Z) = H(X|Z) - H(X|Y, Z). \tag{3}$$

We collect here some of the standard properties of these quantities, which we shall subsequently use without comment. Proofs of these properties can be found, e.g., in [10].

▶ **Proposition 3** (Properties of mutual information and entropy). *Let X, Y and Z be random variables taking values in finite sets $\Omega_1, \Omega_2, \Omega_3$ respectively. Then*

- *(Non-negativity of the entropy)* $0 \leq H(X) \leq \lg |\Omega_1|$.
- *(Non-negativity of the mutual information)* $I(X : Y) \geq 0$ *and* $I(X : Y|Z) \geq 0$.
- *(Conditioning reduces entropy)* $H(X|Y, Z) \leq H(X|Y) \leq H(X)$.
- *(Chain rule for entropy)* $H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$.
- *(Sub-additivity of entropy)* $H(X, Y) \leq H(X) + H(Y)$.
- *(Chain rule for mutual information)* $I(X, Y : Z) = I(X : Z) + I(Y : Z|X) = I(Y : Z) + I(X : Z|Y)$.
- *(Conditional entropy of deterministic functions is zero) If there is a function $f : \Omega_2 \to \Omega_1$ such that $f(Y) = X$, then $H(X|Y) = 0$.*

3 Immediacy codes

In this section, we first introduce the notion of *immediacy codes* (Definition 6), a formal abstraction of the informal notion of immediacy discussed in the introduction. To connect the two notions, we show in Section 3.2 how codes with the two kinds of immediacy functions discussed in the introduction satisfy the immediacy code property. In Section 4, we prove our main technical result (Theorem 8): a general rate upper bound for immediacy codes.

The motivation behind our notion of immediacy codes is to capture the *cold-start decoding* property implied by immediacy: if the noise in a given interval is small, then a code with immediacy properties should be able to decode at least part of the interval. For greater applicability of our framework, we would not like to impose such a condition on *all* intervals, and towards this goal we begin with a few combinatorial definitions. By a *partition* of a set U, we mean an ordered tuple of pairwise disjoint non-empty subsets of U whose union is U.

▶ **Definition 4** (Tagged partitions). *A* tagged partition *S of $[n]$ is a partition $(S_1, S_2, \ldots, S_k)$ of $[n]$, along with a partition $(\text{left}(S_i), \text{right}(S_i))$ of each S_i, $1 \leq i \leq k$, into two non-empty sets.*

▶ **Definition 5** (Laminar partitions). *An (α, ℓ)-*laminar partition *of n is a tuple $P = (P_0, P_1, P_2, \ldots, P_\ell)$ of partitions of $[n]$ into disjoint non-empty subsets, in which $P_1, \ldots, P_\ell$ are tagged partitions, and which satisfies the following properties:*

1. ***(Size property)*** *For each $1 \leq i \leq \ell$, and for each $B \in P_i$, $|\text{left}(B)| \geq \alpha |B|$. (Here, $\text{left}(\cdot)$ is as in the definition of a tagged partition).*
2. ***(The laminar property)*** *For each partition P_i with $i \geq 1$, and any subset $B \in P_i$, there exist sub-collections of P_{i-1} that are partitions of $\text{left}(B)$ and $\text{right}(B)$, respectively, into pairwise disjoint non-empty sets (see Figure 1 for an illustration).*

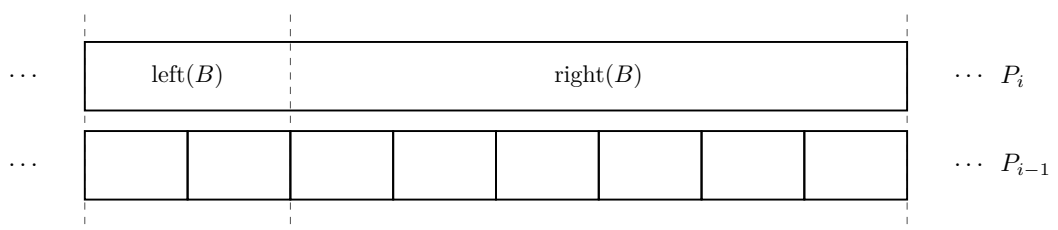


Figure 1 An illustration of laminar partitions.

▶ **Definition 6** (Immediacy codes). *An* (α, ℓ)-immediacy code *with code alphabet* Σ *and transmission length* n *is a function* $c : \Sigma_{\text{in}}^n \to \Sigma^n$ *along with an* (α, ℓ)*-laminar partition* $P = (P_0, P_1, P_2, \ldots, P_\ell)$ *of* $[n]$ *such that*

- ***(The code decodes a large neighborhood)*** *If* B *is a subset in a partition* $P_i \in P$ *for* $i \geq 1$*, then for all* $x, y \in \Sigma_{\text{in}}^n$*,* $x_{\text{left}(B)} \neq y_{\text{left}(B)}$ *implies that* $c(x)_{\text{right}(B)} \neq c(y)_{\text{right}(B)}$*. In other words, there is a function* $\phi_B : \Sigma^{|\text{ right}(B)|} \to \Sigma_{\text{in}}^{|\text{ left}(B)|}$ *such that* $\phi_B(c(x)_{\text{right}(B)}) = x_{\text{left}(B)}$ *for all* $x \in \Sigma_{\text{in}}^n$*.*

For some applications, the neighborhood decoding condition above may not hold for all subsets in all partitions. For handling these cases, we will also need the following weakening of the above definition.

▶ **Definition 7** (Deficient Immediacy code). *A* D*-deficient* (α, ℓ)*-immediacy code with code alphabet* Σ *and transmission length* n *is a function* $c : \Sigma_{\text{in}}^n \to \Sigma^n$ *along with an* (α, ℓ)*-laminar partition* $P = (P_0, P_1, P_2, \ldots, P_\ell)$ *of* $[n]$ *such that*

1. *There exist* deficiency subsets S_i *of* P_i*, for* $1 \leq i \leq \ell$*, such that the total size of all the subsets contained in the* S_i *is at most* D*. In symbols,*

$$\sum_{i=1}^{\ell} \sum_{B \in S_i} |B| \leq D. \tag{4}$$

2. ***(The code often decodes a large neighborhood)*** *If* B *is a subset in a partition* $P_i \in P$ *for* $i \geq 1$ *such that* B *is not an element of the corresponding deficiency set* S_i*, then for all* $x, y \in \Sigma_{\text{in}}^n$*,* $x_{\text{left}(B)} \neq y_{\text{left}(B)}$ *implies that* $c(x)_{\text{right}(B)} \neq c(y)_{\text{right}(B)}$*. In other words, there is a function* $\phi_B : \Sigma^{|\text{ right}(B)|} \to \Sigma_{\text{in}}^{|\text{ left}(B)|}$ *such that* $\phi_B(c(x)_{\text{right}(B)}) = x_{\text{left}(B)}$ *for all* $x \in \Sigma_{\text{in}}^n$*.*

3.1 Rate upper bounds for immediacy codes

We can now state our main technical results, which give rate upper bounds for (deficient) immediacy codes.

▶ **Theorem 8** (Rate upper bound for immediacy codes). *Let* n *be a positive integer, and suppose that* $c : \Sigma_{\text{in}}^n \to \Sigma^n$ *is an* $(\alpha(n), \ell(n))$*-immediacy code. Then,*

$$\lg |\Sigma| \geq \alpha(n)\ell(n) \lg |\Sigma_{\text{in}}| . \tag{5}$$

▶ **Theorem 9** (Rate upper bound for deficient immediacy codes). *Let* n *be a positive integer, and suppose that* $c : \Sigma_{\text{in}}^n \to \Sigma^n$ *is a* D*-deficient* $(\alpha(n), \ell(n))$*-immediacy code. Then,*

$$\lg |\Sigma| \geq \alpha(n) \cdot \left(\ell(n) - \frac{D}{n}\right) \cdot \lg |\Sigma_{\text{in}}| . \tag{6}$$

Remark

In Section 3.2, we describe how the notion of immediacy handled in the above theorems relates to the informal discussion in the Introduction in terms of immediacy functions. We then show how the informal statement of Theorem 2 can be formalized using the above theorems. Note, however, that the more abstract nature of the above theorems allows us to handle the subtler immediacy properties of the known tree code constructions we consider in Section 5.

3.2 Immediacy functions and immediacy codes

We now show that tree codes satisfying the immediacy conditions described in the introduction also satisfy the immediacy code condition, with parameters depending upon the distance and the immediacy function. In order to avoid technical issues such as divisibility, we make here some simplifying assumptions on the form of the immediacy function. We note that these assumptions apply to the examples of immediacy functions discussed in the introduction, and also that the framework of immediacy codes is flexible enough that one can work through a similar route even when they do not hold.

Let $T : \{0,1\}^n \to \Sigma^n$ be a tree code with immediacy function Imm, distance parameter $\delta \in (0,1)$, and suppose that its depth $n = 2 \cdot \mathrm{Imm}(\ell t)$, where ℓ and t are positive integers, with $t = t(\delta)$ possibly depending upon the distance δ. Let $\kappa := \lfloor \lg(2/\delta) \rfloor$ be a positive integer so that $2^{-\kappa} < \delta \leq 2^{-\kappa+1}$. We assume that t can be chosen so that for each $1 \leq j \leq \ell$,

$$\begin{aligned} &2^{-\kappa} \cdot \mathrm{Imm}(jt) \text{ is an integer, and} \\ &2 \cdot \mathrm{Imm}((j-1)t) \text{ divides } 2^{-\kappa} \cdot \mathrm{Imm}(jt) \end{aligned} \tag{7}$$

(We note that this is the main technical assumption that would need to be modified when working with immediacy functions other than those discussed in the introduction. For the two immediacy functions discussed in the introduction, we show how to choose such a t towards the end of this section.)

Given these preliminaries, we now show that T is an (α, ℓ)-immediacy code with $\alpha := 2^{-(\kappa+1)}$ and ℓ as above. We first show that with the divisibility condition in Equation (7), we naturally obtain an (α, ℓ) laminar partition $(P_0, P_1, P_2, \ldots, P_\ell)$. For each $0 \leq j \leq \ell$, partition $[n] = [2\mathrm{Imm}(\ell t)]$ into consecutive blocks of length $2\mathrm{Imm}(jt)$ each, and let these blocks constitute the tagged partition P_j. Given a block $B \in P_j$ for $j \geq 1$, we define $\mathrm{left}(B)$ to be the set of the leftmost $2^{-\kappa} \cdot \mathrm{Imm}(jt)$ integers in B and $\mathrm{right}(B)$ to be the rightmost $2\mathrm{Imm}(jt) - |\mathrm{left}(B)|$ integers in B. Given the divisibility conditions in Equation (7), we can then verify by a direct computation that both the laminar and size properties in the definition of an (α, ℓ)-laminar partition are satisfied by this construction. In particular, $\mathrm{left}(B)$ and $\mathrm{right}(B)$ are disjoint unions of blocks in P_{j-1}.

We now show that T is also an immediacy code with respect to the tagged partition $(P_0, P_1, \ldots, P_\ell)$. Fix any $1 \leq j \leq \ell$ and any block B in the tagged partition P_j. We only need to show that if $x, y \in \{0,1\}^n$ differ on some index in $\mathrm{left}(B)$, then $T(x)$ and $T(y)$ differ on some index in $\mathrm{right}(B)$. For any such x, y, let i be the smallest index in B on which x and y differ, and let S be the interval $[i, i + \mathrm{Imm}(jt))$ of length $\mathrm{Imm}(jt)$. By the choice of x and y, we have $i \in \mathrm{left}(B)$. Since $|\mathrm{left}(B)| = \mathrm{Imm}(jt) \cdot 2^{-\kappa}$, $|B| = 2\mathrm{Imm}(jt)$, and $\kappa \geq 1$, it then follows that S is contained in B, and intersects $\mathrm{right}(B)$. Further, by the definition of the immediacy function, the Hamming distance between $T(x)_S$ and $T(y)_S$ must be at least

$$\delta \cdot \mathrm{Imm}(jt) > 2^{-\kappa}\mathrm{Imm}(jt) = |\mathrm{left}(B)|.$$

Thus, since this Hamming distance is greater than the length of $\mathrm{left}(B)$, it must be the case that $T(x)_S$ and $T(y)_S$ differ also on some index in $\mathrm{right}(B)$, as we wanted to prove. This proves that T is an (α, ℓ)-immediacy code, with $\alpha = 2^{-(\kappa+1)}$, as chosen above.

Rate upper bounds with immediacy functions

Using the rate upper bound for (α, ℓ)-immediacy codes given by our main result (Theorem 8), we thus get that if Σ is the output alphabet of T, it must be the case that (recall that $n = 2 \cdot \mathrm{Imm}(\ell t(\delta))$)

$$\lg |\Sigma| \geq \frac{\ell}{2^{1+\kappa}} = \frac{\mathrm{Imm}^{-1}(n/2)}{t(\delta) \cdot 2^{1+\kappa}} \geq \frac{\delta}{4t(\delta)}\mathrm{Imm}^{-1}(n/2). \tag{8}$$

Put differently, the rate ρ of such a tree code satisfies

$$\rho \leq \frac{4t(\delta)}{\delta \cdot \mathrm{Imm}^{-1}(n/2)}. \tag{9}$$

(We remark here that since we work in the setting of constant distance δ, we have not attempted to optimize the dependence on δ in the above computation.)

Choosing $t(\delta)$. We now show how to choose $t(\delta)$ so as to satisfy the divisibility conditions of Equation (7), for the two immediacy functions considered in the introduction.

Case 1: $\mathbf{Imm(k) = 2^k}$. Put $t(\delta) = 1 + \kappa = \lfloor \lg(4/\delta) \rfloor$. A direct computation shows that the condition in Equation (7) holds with this choice. The rate upper bound in Equation (9) becomes

$$\rho \leq \frac{4 \lg(4/\delta)}{\delta \cdot \lg(n/2)}. \tag{10}$$

Case 2: $\mathbf{Imm(k) = 2^{2^k}}$. Put $t(\delta) := \lceil \lg(\kappa + 2) \rceil = \lceil \lg \lfloor \lg(8/\delta) \rfloor \rceil$. Again, a direct computation shows that the condition in Equation (7) holds with this choice. The rate upper bound in Equation (9) becomes

$$\rho \leq \frac{4 \cdot \lceil \lg \lg(8/\delta) \rceil}{\delta \cdot \lg \lg(n/2)}. \tag{11}$$

4 Rate upper-bound for immediacy codes

In this section, we prove our main result: a rate upper bound for immediacy codes (Theorem 8). The main ingredient is a simple information theoretic tool, that we describe next. As stated in the introduction, given this tool and the definition of immediacy codes, the proof of the main result becomes quite simple.

4.1 Data processing inequality

We will need the following simple consequence of (the usual proof of) the data processing inequality. Similar inequalities arise also in studies of the Gács-Körner common information (see, e.g., [21]).

▶ **Lemma 10** (A consequence of the data processing inequality). *Let A, B, C be discrete random variables such that $H(A|B) = H(A|C) = 0$. Then $I(B : C) \geq H(A)$. In particular,*

$$H(B) + H(C) \geq H(B, C) + H(A).$$

Proof. Since A, B, C are discrete, the associated entropies and conditional entropies are non-negative. Combining this with the non-negativity of mutual information and conditional mutual information, we thus have

$$0 \leq I(A : C|B) \leq H(A|B) = 0. \tag{12}$$

We also have $I(A : C) = H(A) - H(A|C) = H(A)$. The chain rule for mutual information along with the non-negativity of the conditional mutual information then gives

$$\begin{aligned} H(A) = I(A : C) &\leq I(A : C) + I(B : C|A) \\ &= I(A, B : C) = I(B : C) + I(A : C|B) \overset{\text{Equation (12)}}{=} I(B : C), \end{aligned}$$

which proves that $I(B : C) \geq H(A)$. The final claim follows since $H(B, C) = H(B) + H(C) - I(B : C)$. ◀

4.2 Proof of the main result

We now prove our main result (Theorem 8). We restate the result here for convenience of reference: please also see the remark at the end of Section 3.1 for further discussion.

▶ **Theorem 8** (Rate upper bound for immediacy codes). *Let n be a positive integer, and suppose that $c : \Sigma_{\text{in}}^n \to \Sigma^n$ is an $(\alpha(n), \ell(n))$-immediacy code. Then,*

$$\lg |\Sigma| \geq \alpha(n)\ell(n) \lg |\Sigma_{\text{in}}| . \tag{5}$$

Proof. To simplify notation, we shorten $\ell(n)$ to ℓ in the following. Let $P = (P_0, P_1, \ldots, P_\ell)$ be the laminar partition associated with the immediacy code c. Modify the code c so that it is systematic, in the sense that $c(x)_k$ determines x_k for each $1 \leq k \leq n$: this can be done by changing the alphabet to $\Sigma \times \Sigma_{\text{in}}$, and simply concatenating x_k to each $c(x)_k$. Note that even after the modification, the code c continues to be an immediacy code, with the same associated laminar partition P. However, the output alphabet of the code now is Σ' where $\Sigma' = \Sigma \times \Sigma_{\text{in}}$.

Let X be uniformly distributed over Σ_{in}^n, and define $Y = c(X)$. We then have $H(X) = H(Y) = n \cdot \lg |\Sigma_{\text{in}}|$, where we measure entropy in bits. Now, consider any subset B that is an element of one of the partitions P_i, for some $i \geq 1$. Since c is systematic, we then have $H(X_{\text{left}(B)}|Y_{\text{left}(B)}) = H(X_{\text{left}(B)}|c(X)_{\text{left}(B)}) = 0$. Further, by the neighborhood decoding property of the immediacy code c, we also have

$$H(X_{\text{left}(B)}|Y_{\text{right}(B)}) = H(\phi_B(c(X)_{\text{right}(B)})|c(X)_{\text{right}(B)}) = 0. \tag{13}$$

(Here, ϕ_B is as in the definition of the neighborhood decoding property of an immediacy code.) Applying Lemma 10, we then have

$$\begin{aligned} H(Y_B) = H(Y_{\text{left}(B)}, Y_{\text{right}(B)}) &\leq H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)}) - H(X_{\text{left}(B)}) && (14)\\ &= H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)}) - |\text{left}(B)| \cdot \lg |\Sigma_{\text{in}}| && (15)\\ &\leq H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)}) - \alpha(n)|B| \cdot \lg |\Sigma_{\text{in}}| , && (16) \end{aligned}$$

where the last inequality follows since $\text{left}(B) \geq \alpha(n)|B|$, by the size property of $(\alpha(n), \ell(n))$-laminar partitions. Note that Equation (16) holds for every subset B that is part of some partition P_i, $i \geq 1$, of P. Define, for $0 \leq i \leq \ell$

$$T_i := \sum_{B \in P_i} H(Y_B). \tag{17}$$

We then have, for $1 \leq i \leq \ell$,

$$\begin{aligned} T_i = \sum_{B \in P_i} H(Y_B) &\overset{\text{Equation (16)}}{\leq} -\alpha(n)n \cdot \lg |\Sigma_{\text{in}}| + \sum_{B \in P_i} \left(H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)}) \right) && (18)\\ &\leq -\alpha(n)n \cdot \lg |\Sigma_{\text{in}}| + \sum_{B' \in P_{i-1}} H(Y_{B'}) = T_{i-1} - \alpha(n)n \cdot \lg |\Sigma_{\text{in}}| . && (19) \end{aligned}$$

where the second inequality follows from the laminar property of the laminar partition P (which ensures that each of $\text{left}(B)$ and $\text{right}(B)$ are disjoint unions of sets in the partition P_{i-1}, for every set $B \in P_i$), along with the sub-additivity of entropy: for any two random variables Z_1 and Z_2, $H(Z_1, Z_2) \leq H(Z_1) + H(Z_2)$. By induction, Equation (19) thus gives

$$T_\ell \leq T_0 - \alpha(n)n\ell \cdot \lg |\Sigma_{\text{in}}| . \tag{20}$$

The subadditivity of entropy also gives $T_\ell \geq H(Y) = n \cdot \lg |\Sigma_{\text{in}}|$, and

$$T_0 \leq \sum_{i=1}^{n} H(Y_i) \leq n \cdot (\lg |\Sigma_{\text{in}}| + \lg |\Sigma|), \text{ since each } Y_i \text{ has support } \Sigma' = \Sigma \times \Sigma_{\text{in}}. \tag{21}$$

Substituting these in Equation (20) gives $\lg |\Sigma| \geq \alpha(n)\ell \cdot \lg |\Sigma_{\text{in}}|$. ◀

The proof above is robust to certain small perturbations to the definition of an immediacy code. In particular, it can be easily adapted to establish rate upper bounds for D-deficient immediacy codes as well, as we show below. We restate the corresponding result (Theorem 9) below for ease of reference.

▶ **Theorem 9** (Rate upper bound for deficient immediacy codes). *Let n be a positive integer, and suppose that $c : \Sigma_{\text{in}}^n \to \Sigma^n$ is a D-deficient $(\alpha(n), \ell(n))$-immediacy code. Then,*

$$\lg |\Sigma| \geq \alpha(n) \cdot \left(\ell(n) - \frac{D}{n}\right) \cdot \lg |\Sigma_{\text{in}}|. \tag{6}$$

Proof. The proof is a minor modification of the proof of Theorem 8, but we include all the details for completeness. Let $P = (P_0, P_1, \ldots, P_\ell)$ be the laminar partition associated with the D-deficient immediacy code c, and let the deficiency subsets $S_i \subseteq P_i$ be as in the definition (Definition 7). Modify the code c so that it is systematic, in the sense that $c(x)_k$ determines x_k for each $1 \leq k \leq n$: this can be done by changing the alphabet to $\Sigma \times \Sigma_{\text{in}}$, and simply concatenating x_k to each $c(x)_k$. Note that even after the modification, the code c continues to be a D-deficient immediacy code, with the same associated laminar partition P and the same associated deficiency sets. However, the output alphabet of the code now is Σ' where $\Sigma' = \Sigma \times \Sigma_{\text{in}}$.

Let X be uniformly distributed over Σ_{in}^n, and define $Y = c(X)$, so that $H(X) = H(Y) = n \cdot \lg |\Sigma_{\text{in}}|$. Now, consider any subset B that is an element of one of the partitions P_i, for some $i \geq 1$. Since c is systematic, we then have $H(X_{\text{left}(B)}|Y_{\text{left}(B)}) = H(X_{\text{left}(B)}|c(X)_{\text{left}(B)}) = 0$. Further, in case $B \notin S_i$, then by the neighborhood decoding property of the deficient immediacy code c, we also have

$$H(X_{\text{left}(B)}|Y_{\text{right}(B)}) = H(\phi_B(c(X)_{\text{right}(B)})|c(X)_{\text{right}(B)}) = 0. \tag{22}$$

(Here, ϕ_B is as in the definition of the neighborhood property of a deficient immediacy code.) Applying Lemma 10, we then have

$$\begin{aligned} H(Y_B) = H(Y_{\text{left}(B)}, Y_{\text{right}(B)}) &\leq H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)}) - H(X_{\text{left}(B)}) && (23)\\ &= H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)}) - |\text{left}(B)| \cdot \lg |\Sigma_{\text{in}}| && (24)\\ &\leq H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)}) - \alpha(n)|B| \cdot \lg |\Sigma_{\text{in}}|, && (25) \end{aligned}$$

where the last inequality follows since $\text{left}(B) \geq \alpha(n)|B|$, by the size property of $(\alpha(n), \ell(n))$-laminar partitions. Note that when $i \geq 1$, Equation (25) holds for every subset $B \in P_i \setminus S_i$. The only difference with the proof of Theorem 8 is that when $B \in P_i$ is an element of the deficiency set S_i, we however only have (by the sub-additivity of the entropy):

$$H(Y_B) = H(Y_{\text{left}(B)}, Y_{\text{right}(B)}) \leq H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)}). \tag{26}$$

Define, for $0 \leq i \leq \ell$

$$T_i := \sum_{B \in P_i} H(Y_B). \tag{27}$$

We then have, for $1 \le i \le \ell$,

$$T_i = \sum_{B \in P_i} H(Y_B) \overset{\text{Equations (25) and (26)}}{\le} -\alpha(n)n \cdot \lg|\Sigma_{\text{in}}| + \alpha(n) \cdot \lg|\Sigma_{\text{in}}| \cdot \sum_{B \in S_i} |B| + \sum_{B \in P_i} \left(H(Y_{\text{left}(B)}) + H(Y_{\text{right}(B)})\right) \tag{28}$$

$$\le -\alpha(n)n \cdot \lg|\Sigma_{\text{in}}| + \alpha(n) \cdot \lg|\Sigma_{\text{in}}| \cdot \sum_{B \in S_i} |B| + \sum_{B' \in P_{i-1}} H(Y_{B'}) \tag{29}$$

$$= T_{i-1} - \alpha(n)n \cdot \lg|\Sigma_{\text{in}}| + \alpha(n) \cdot \lg|\Sigma_{\text{in}}| \sum_{B \in S_i} |B| \,. \tag{30}$$

where the second inequality follows from the laminar property of the laminar partition P (which ensures that each of $\text{left}(B)$ and $\text{right}(B)$ are disjoint unions of sets in the partition P_{i-1}, for every set $B \in P_i$), along with the sub-additivity of entropy. By induction, Equation (30) thus gives

$$T_\ell \le T_0 - \alpha(n)n\ell \cdot \lg|\Sigma_{\text{in}}| + \alpha(n) \cdot \lg|\Sigma_{\text{in}}| \cdot \sum_{i=1}^{\ell} \sum_{B \in S_i} |B| \le T_0 - \alpha(n) \cdot \lg|\Sigma_{\text{in}}| \cdot (n\ell - D)\,, \tag{31}$$

where the last inequality uses the fact that the code is only D-deficient (Equation (4) of Definition 7). The subadditivity of entropy yields $T_\ell \ge H(Y) = n \cdot \lg|\Sigma_{\text{in}}|$, and also that

$$T_0 \le \sum_{i=1}^{n} H(Y_i) \le n \cdot (\lg|\Sigma_{\text{in}}| + \lg|\Sigma|), \text{ since each } Y_i \text{ has support } \Sigma' = \Sigma \times \Sigma_{\text{in}}. \tag{32}$$

Substituting these in Equation (31) gives $\lg|\Sigma| \ge \alpha(n) \cdot \left(\ell - \frac{D}{n}\right) \cdot \lg|\Sigma_{\text{in}}|\,.$ ◀

5 Rate upper-bounds for immediacy properties of known constructions

In this section, we describe the immediacy properties enjoyed by the tree code constructions of [8], [12] and [14]. The first two are similar to – but not the same as – the informal notion of immediacy discussed in the introduction. However, as we show below, our general framework of immediacy codes developed in Section 3 still applies, and allows us to conclude that the $\Omega(1/\log\log n)$ and $\Omega(1/\log n)$ rates achieved by these constructions are the best possible (up to constant factors) given the immediacy properties they achieve. The third construction, due to [14], tackles a different trade-off: what is the best (even if vanishing, as n increases) distance one can achieve for a tree code if one imposes the condition that the rate has to be constant. For this construction we show that the distance $\Omega(1/\log n)$ distance it achieves with a constant rate is tight up to a $\log\log n$ factor for the immediacy guarantee that this construction provides.

5.1 The CHS construction

We start with the construction of [8] (which for brevity we shall call the CHS construction). To describe the immediacy properties of this construction, we first import some of the notation set up in [8]. In the following, references to theorems, pages etc. in [8] refer to the ECCC version of [8]. Define a sequence of length scales (following the proof of Theorem 1.1 in [8, p. 16]) given by $\ell_1 := 2^{20}$ and $\ell_{i+1} := \ell_i^2/2^{10}$ for $i \ge 1$; thus $\ell_i = 2^{10} \cdot 32^{2^i}$ for every

positive integer i. We consider a transmission length n of the form ℓ_{m+1} for some positive integer m. Thus, n is divisible by each ℓ_i, $1 \le i \le m$, and no divisibility issues arise in the following description. We also note for future use that for $i \ge 1$

$$16\sqrt{\ell_i} \le \frac{3\ell_i}{8}. \tag{33}$$

For each length scale ℓ_i, $1 \le i \le m+1$ as above, we consider the partition P_{i-1} of $[n]$ into $2n/\ell_i$ consecutive disjoint intervals of length $\ell_i/2$ each. In agreement with the terminology in [8], we refer to the elements of the P_i as *blocks*. For a block $B \in P_i$ for $i \ge 1$, we denote by $\mathrm{left}(B)$ the set consisting of the first $|B|/4$ positions in B, and by $\mathrm{right}(B)$ the set consisting of the last $3|B|/4$ positions in B. With these definitions, $P := (P_0, P_1, P_2, \ldots, P_m)$ is a $(1/4, m)$-laminar partition of $[n]$. In particular, for any $B \in P_i$, $i \ge 1$, $\mathrm{left}(B)$ and $\mathrm{right}(B)$ are disjoint unions of blocks in P_{i-1}, because $\ell_i/2$ divides $\ell_{i+1}/8$.

The CHS immediacy condition

We can now describe the immediacy-like property satisfied by the CHS tree code construction. Denote their code by T. Let x and y be two strings in $\{0,1\}^n$ which differ at a position $s' \in [n]$. Then, for any i such that $2 \le i \le m+1$ such that s' is *not* in the rightmost block in P_{i-1} (i.e., such that $s' \le n - \ell_i/2$), the following is true. Let $s(i)$ denote the *leftmost* position in the unique block $B \in P_{i-1}$ containing s' such that $x_{s(i)} \ne y_{s(i)}$. Then, for any d satisfying

$$\ell_{i-1}/2 = 16\sqrt{\ell_i} \le d \le \ell_i/2, \tag{34}$$

it holds that

$$\Delta_{\mathrm{Hamming}}\left(T(x)_{I(d)}, T(y)_{I(d)}\right) \ge d/3, \tag{35}$$

where $I(d)$ denotes the interval

$$I(d) := [s(i), s(i)+d]\,. \tag{36}$$

The above condition is implicit in the proof given in [8]: in particular, it follows by substituting $s(i)$ above in the role played by the "split" s in the proof of Claim 6.4 of [8].

We then have the following consequence of the above immediacy condition.

▶ **Lemma 11.** *Let m be a positive integer. Let the sequence $\ell_1, \ell_2, \ldots$, the positive integer $n = \ell_{m+1}$, and the $(1/4, m)$-laminar partition $P = (P_0, P_1, P_2, \ldots, P_m)$ of n be as defined above. Let $T : \{0,1\}^n \to \Sigma^n$ be a code satisfying the CHS immediacy condition described above. Consider any block B in a partition $P_{i-1} \in P$, $2 \le i \le m+1$, so that B is not the rightmost block in P_{i-1}. Then, there exists a function ϕ_B such that for any $x \in \{0,1\}^n$, $\phi_B(c(x)_{\mathrm{right}(B)}) = x_{\mathrm{left}(B)}$. In other words, the prefix $x_{\mathrm{left}(B)}$ of x_B is uniquely determined given the suffix $c(x)_{\mathrm{right}(B)}$ of $c(x)_B$.*

Proof. It is sufficient to show that if $x, y \in \{0,1\}^n$ are such that $x_{\mathrm{left}(B)} \ne y_{\mathrm{left}(B)}$ then it must be the case that $c(x)_{\mathrm{right}(B)} \ne c(y)_{\mathrm{right}(B)}$. Consider therefore $x, y \in \{0,1\}^n$ such that $x_{\mathrm{left}(B)} \ne y_{\mathrm{left}(B)}$. Thus, there must be a position s' in $\mathrm{left}(B)$ such that $x_{s'} \ne y_{s'}$. Let $s(i)$ be the leftmost such position in $\mathrm{left}(B)$. Choose d so that the interval

$$I(d) = [s(i), s(i)+d] = B \cap [s, n],$$

i.e., so that $I(d)$ is the suffix of B starting at $s(i)$. Since $|B| = \ell_i/2$, this d is of the form $\ell_i/2 - k(i)$, where $k(i) \geq 1$ is the relative index, counting starting with one from the left, of $s(i)$ within B. Since $s(i) \in \text{left}(B)$ and $\text{left}(B)$ consists of the leftmost $|B|/4 = \ell_i/8$ positions in B, it is also the case that $k(i) \leq \ell_i/8$. Given Equation (33), d therefore satisfies the lower bound required in Equation (34) since $k(i) \leq \ell_i/8$ and $i \geq 2$. It also satisfies the upper bound required in Equation (34) by construction. Since B is, by hypothesis, not the rightmost block in P_{i-1}, Equation (35) then yields that

$$\Delta_{\text{Hamming}}\left(c(x)_{I(d)}, c(y)_{I(d)}\right) \geq d/3 = \frac{\ell_i}{8} + \frac{\ell_i/8 - k(i)}{3}. \tag{37}$$

Suppose, if possible, that $c(x)_{\text{right}(B)} = c(y)_{\text{right}(B)}$, so that $\Delta_{\text{Hamming}}\left(c(x)_{\text{right}(B)}, c(y)_{\text{right}(B)}\right) = 0$. Let S denote the suffix of $\text{left}(B)$ starting from the position $s(i)$, so that $I(d) = S \sqcup \text{right}(B)$. We then have

$$\Delta_{\text{Hamming}}\left(c(x)_{I(d)}, c(y)_{I(d)}\right) = \Delta_{\text{Hamming}}\left(c(x)_S, c(y)_S\right) + \Delta_{\text{Hamming}}\left(c(x)_{\text{right}(B)}, c(y)_{\text{right}(B)}\right) \tag{38}$$

$$= \Delta_{\text{Hamming}}\left(c(x)_S, c(y)_S\right) \leq |S| = \frac{\ell_i}{8} - k(i) + 1. \tag{39}$$

However, this contradicts Equation (37) since $\ell_i \geq 32$ and $k(i) \geq 1$. It must therefore be the case that $c(x)_{\text{right}(B)} \neq c(y)_{\text{right}(B)}$. ◀

It is an easy consequence of the above lemma that any binary tree code satisfying the CHS immediacy condition is a deficient immediacy code, with a small deficiency parameter.

▶ **Proposition 12.** *Let m be a positive integer. Let the sequence $\ell_1, \ell_2, \ldots$, the positive integer $n = \ell_{m+1}$, and the $(1/4, m)$-laminar partition $P = (P_0, P_1, P_2, \ldots, P_m)$ of n be as defined above. Let $T : \{0,1\}^n \to \Sigma^n$ be a code satisfying the CHS immediacy condition. Then T is an n-deficient $(1/4, m)$-immediacy code.*

Proof. For each $1 \leq i \leq m$ define S_i to consist only of the rightmost block in P_i. Then, it follows directly from Lemma 11 that T is a D-deficient $(1/4, m)$-immediacy code with deficiency sets $S_1, S_2, \ldots, S_m$, provided D satisfies

$$D \geq \sum_{i=1}^{m} \sum_{B \in S_i} |B|. \tag{40}$$

Now, the latter quantity can be bounded from above as

$$\frac{n}{2} \sum_{i=1}^{m} \frac{\ell_{i+1}}{\ell_{m+1}} \leq n, \tag{41}$$

using the growth condition on the ℓ_i. Thus the choice $D = n$ works. ◀

Applying Theorem 9, we thus get that the output alphabet size of any binary tree code satisfying the CHS immediacy condition with transmission length n as above satisfies

$$\lg |\Sigma| \geq \frac{m-1}{4} = \Omega(\lg \lg n), \tag{42}$$

since $n = 2^{10} \cdot 32^{2^{m+1}}$. Thus, the rate ρ of such a code must satisfy $\rho = O(1/\lg \lg n)$.

5.1.1 Subexponential time constructions of good tree codes

Braverman [7] presented a generic procedure to explicitly construct good tree codes in sub-exponential time. More formally, given $\epsilon \in (0,1)$, Braverman's procedure runs in time $\exp(O(n^\epsilon))$ and outputs a depth n tree code that has constant rate (depending upon ϵ) and fixed constant distance. At a high level, Braverman's construction works by starting with a brute force $\exp(O(n^\epsilon))$-time construction of a tree code of depth n^ϵ, and then using a sequence of (efficiently implementable) tilings of this intermediate tree code, augmented with error-correcting codes, in order to handle error-propagation at different length scales. However, as described in [7], this construction achieves a final alphabet size that is at least exponential in $1/\epsilon$, so that the rate achieved is no better than $O(1/\epsilon)$.

We first review here a refinement of the above construction, mentioned in passing in [8, remarks following Theorem 1.1], that constructs a constant distance tree code with rate $\Theta(1/\log(1/\epsilon))$ and depth n in time $\exp(O(n^\epsilon))$. We then describe the immediacy properties of this construction, and show that no code with these immediacy properties can have rate better than $\Omega(1/\log(1/\epsilon))$. The arguments are very similar to those carried out above for the immediacy properties of the CHS construction.

Review of a subexponential-time constructible good tree code

We use the notation from [8] imported above. Let k be a fixed positive integer parameter (corresponding roughly to $\log(1/\epsilon)$). Consider an n of the form $\ell_{m+1} = 2^{10} \cdot 32^{2^{m+1}}$ as above. The polynomial time (in n) construction of a tree code of depth n given in the proof of Theorem 1.1 of [8] consists principally of a concatenation of constant rate intermediate codes C_i, for each $1 \le i \le m+1$, where the ith code corresponds to length scale $\ell_i = 2^{10} \cdot 32^{2^i}$. In consequence, the resulting final code has rate $\Theta(1/\lg\lg n)$.

To get a better rate at the cost of an increase in the runtime of the construction, start by replacing the above construction by a concatenation of only the codes $C_{m-k+1}, C_{m-k+2}, \ldots, C_{m+1}$. Following the proof of Theorem 1.1 of [8], one then gets that this resulting intermediate code is, in the terminology of [8], a *lagged tree code* (i.e., where error propagation from an error starts after a certain lag) with lag $16\sqrt{\ell_{m-k+1}} = \ell_{m-k}/2$. Applying a variant of the idea from [7], one can then remove the lag as follows. First construct, in time $\exp(O(\ell_{m-k}))$, a constant rate and constant distance tree code of depth ℓ_{m-k+1}. Then, concatenate to the above intermediate construction two tilings of this good tree code staggered by a shift of $\ell_{m-k+1}/2$ (following, e.g., Lemma 5 of [7]). The resulting construction is a tree code of constant distance and rate $\Theta(1)/k$, obtained in time $\exp(O(\ell_{m-k+1})) = \exp(O(n^{1/2^k}))$. Reparameterizing with $\epsilon = 1/2^k$, we see that the rate is $\Theta(1)/\lg(1/\epsilon)$ and the runtime of the construction is $\exp(O(n^\epsilon))$.

Immediacy properties

We now consider the immediacy properties of the the above construction. The arguments before Lemma 11, when applied to above variant of the CHS construction, show that the immediacy conditions described in Equations (34)–(36) hold for this version for all i satisfying $m-k+2 \le i \le m+1$. (For the original CHS construction, these conditions held for all i such that $2 \le i \le m+1$.)

Rate upper bound

We now show that any tree code satisfying these immediacy properties cannot have rate better better than $O(1/k)$. To do this, we repeat the proof of Lemma 11 and Proposition 12 with the above modified version of the CHS immediacy condition to obtain that any such code must be an n-deficient $(1/4, k)$-immediacy code with respect to the $(1/4, k)$-laminar partition $(P_{m-k}, P_{m-k+1}, \ldots, P_m)$ (where the P_i are as defined in the above description of the original CHS construction, following Equation (33)). Applying Theorem 9, we therefore get that the output alphabet size of any depth n binary tree code satisfying the above variant of the CHS immediacy condition satisfies

$$\lg |\Sigma| \geq \frac{k-1}{4}. \tag{43}$$

Thus, the rate ρ of such a code must satisfy $\rho = O(1/k)$.

5.2 The EKS construction

We now turn to the construction of [12] (which for brevity we shall call the EKS construction.) A version of this construction can be described as follows. Let $\mathrm{ECC}_\ell : \{0,1\}^\ell \to \left\{\{0,1\}^b\right\}^\ell$ denote an error-correcting block code with distance $\delta \in (0,1)$. (It is known that there is a choice of $b = b(\delta)$, independent of the block-length ℓ, so that such a code exists for each positive integer ℓ, and is such that ECC_ℓ can be computed in time polynomial in ℓ: see, e.g., [8, Lemma 3.2 in the ECCC version].) Note that ECC_1 can be chosen to be the repetition code of length b. For simplicity, we describe the construction of the EKS tree code T when the depth $n = 2^k$ is a power of 2. For $0 \leq i \leq k$, let P_i be the partition of $[n]$ consisting of $n/2^i$ consecutive disjoint blocks of length 2^i each, and we write $P_{i,j} = \{(j-1)2^i + 1, \ldots, j \cdot 2^i\}$. For a message $x \in \{0,1\}^n$ we let $x_{P_{i,j}} \in \{0,1\}^{2^i}$ denote the restriction of x to the interval $P_{i,j}$. To describe the encoding $T(x)$ of a message $x \in \{0,1\}^n$, we form the following $(k+1) \times n$ table with entries in $\{0,1\}^b$:

1. The first row consists of the concatenation of $\mathrm{ECC}_1(x_j)$ for $j = 1, \ldots, n$.
2. The ith row, for $2 \leq i \leq k+1$, consists of $(0^b)^{2^{i-2}}$ followed by the concatenation of $\mathrm{ECC}_{2^{i-2}}(x_{P_{i-2,j}})$ for $j = 1, \ldots, n \cdot 2^{2-i} - 1$. Informally, the code blocks in the ith row are "right-shifted" by 2^{i-2}.

The encoding $T(x)$ is defined by letting its jth character, for $1 \leq j \leq n$, be the jth column (from the left) of this table. Note that because of the "right shift" operation in Item 2, this construction satisfies the "online" requirement for tree codes. The output alphabet of T is $\{0,1\}^{b(k+1)}$.

The EKS immediacy condition

From the construction, it is easy to see that the EKS construction satisfies the following version of immediacy. Denote the code by T, and let $x, y \in \{0,1\}^n$ be two strings that differ at a position $s' \in [n]$. Let $0 \leq \ell < k$ be any integer such that $s' \leq n - 2^\ell = 2^k - 2^\ell$. Define $s = s(\ell) := \left\lceil s'/2^\ell \right\rceil \cdot 2^\ell$ to be the smallest multiple of 2^ℓ that is at least as large as s'. Then

$$\Delta_{\mathrm{Hamming}}(T(x)_{(s,s+2^\ell]}, T(y)_{(s,s+2^\ell]}) \geq \delta \cdot 2^\ell > 0. \tag{44}$$

We now show that any code satisfying the immediacy condition described by Equation (44) must be an $(\Omega(1), \Omega(\log n))$-immediacy code. To see this, we consider the partitions P_i, $0 \leq i \leq k$, of $[n] = [2^k]$ considered above, and convert each of them into a tagged partition

by defining, for each $B \in P_i$ with $i \geq 1$, left(B) to be the leftmost 2^{i-1} entries in B and right(B) to be the rightmost 2^{i-1} entries in B. With this construction, it is immediate that $(P_0, P_1, P_2, \ldots, P_k)$ is a $(1/2, k)$-laminar partition. Since the code T satisfies the immediacy condition in Equation (44), it is also immediate that it satisfies the neighborhood decoding property for each $B \in P_i$ with $i \geq 1$. We thus see that any tree code T with depth $n = 2^k$ and satisfying the immediacy condition given by Equation (44) must be a $(1/2, k)$-immediacy code. Applying Theorem 8, we thus see that the output alphabet Σ of such a code must satisfy

$$|\Sigma| \geq \frac{k}{2} = \Omega(\lg n), \tag{45}$$

since $n = 2^k$. Thus the rate ρ of such a code must be $O(1/\lg n)$.

5.3 The GHKRZW construction

We now turn to a tree code construction by Gelles, Haeupler, Kol, Ron-Zewi and Wigderson [14] (which we shall refer to as the GHKRZW construction). While related to the EKS construction discussed in Section 5.2, this construction is qualitatively different from the previous constructions considered in this paper in that it insists upon a *constant*, non-vanishing rate even if at the cost of a vanishing *distance* parameter. We show here that for the immediacy guarantee the GHKRZW construction achieves, and given its requirement of a constant rate, the $\Omega(1/\log n)$ vanishing distance it achieves is tight up to a $\log \log n$ factor.

The GHKRZW construction (described in [14, Section 5.1]) demonstrates the following: there exists a positive integer k_0, such that for every $\epsilon \in (0, 1)$ there is a tree code T, with transmission length n, input alphabet $\Sigma_{\text{in}} := \{0,1\}^{(\lg n)/\epsilon}$, output alphabet $\Sigma := \{0,1\}^{(1+1/\epsilon)\cdot \lg n}$ (so that its rate is $1/(1+\epsilon)$) and distance parameter δ is $\frac{\epsilon}{32 k_0 \cdot \lg n}$. Here, it is convenient to assume (as we do now) that $\lg n, k_0$ and $1/\epsilon$ are all powers of two, and that $k_0 \geq 16$. (The notation and conventions here are essentially exactly those adopted in [14, Section 5.1]).

The GHKRZW immediacy condition

The distance proof for the GHKRZW construction, given in Section 5.2 of [14], in fact establishes the following immediacy property. Let k_0, ϵ be parameters not growing with the transmission length n (as above) and let $m := (k_0/\epsilon) \cdot \lg n$ be as defined in [14]; note that m is also a power of two by the assumptions adopted above. Let t be any integer satisfying $\lg m \leq t \leq \lg n - 1$. Suppose that $x, y \in \Sigma_{\text{in}}$ differ at a position $i \leq n - 2^t$, and Let $i_0 = i_0(t) := \lfloor (i-1)2^{-t} \rfloor \cdot 2^t$ be the largest multiple of 2^t that is smaller than i, as defined in [14, Section 5.2]. Then,

$$\Delta_{\text{Hamming}}\left(T(x)_{(i_0, i_0+2^{t+1}]}, T(y)_{(i_0, i_0+2^{t+1}]}\right) \geq \delta \cdot 2^{t+1}. \tag{46}$$

To prove this, one repeats the proof in Section 5.2 of [14], ignoring the parameter j in that proof, and also ignoring the condition, stipulated in the proof, that i is the *first* position on which x and y differ. The last paragraph on p. 1930 in [14], combined with the first display on p. 1931 (neither of which use the latter condition that i is the first position on which x and y differ), then yields (a slightly stronger version of) the above condition, after we note that δ' in that proof is at least 8δ, where δ, as above, is the distance parameter achieved by the GHKRZW construction.[4]

[4] For the convenience of the reader, we note here that there is a minor typographical error on p. 1930 of [14], where the "max" in the first display in Section 5.2 should actually be "min".

We now show that any constant rate tree code satisfying the immediacy condition specified by Equation (46) cannot have distance much better than the GHKRZW construction. The proof is similar to the case of an exponential immediacy function analyzed in Section 3.2.

As in that proof, our goal is to show that such a code is an immediacy code with appropriate parameters. We begin by defining the corresponding laminar partition. Let $\delta \in (0,1)$ be the distance parameter (possibly dependent upon n) of such a code. Let $\kappa := \lfloor \lg(2/\delta) \rfloor$ be a positive integer so that $2^{-\kappa} < \delta \leq 2^{-\kappa+1}$. Set

$$\ell := 1 + \lfloor (\lg(n/(2m)))/\kappa \rfloor . \tag{47}$$

Let P_0 be the partition of $[n]$ into n singletons, and for $1 \leq i \leq \ell$, define P_i to the partition of $[n]$ into blocks of consecutive integers, each of length $n/2^{\kappa\cdot(\ell-i)}$. Thus, for $1 \leq i \leq i+1$, any set B in P_i is an interval of the form $(b \cdot 2^{t+1}, (b+1)\cdot 2^{t+1}]$, where b is an integer and $t = \lg n - \kappa \cdot (\ell - i) - 1$. For $1 \leq i \leq \ell$, convert P_i into a tagged partition by defining, for each $B \in P_i$, $\mathrm{left}(B)$ to be the set of the smallest $2^{-\kappa} \cdot |B|$ elements in B (so that $\mathrm{right}(B) = B \setminus \mathrm{left}(B)$). It is easy to verify that the above construction gives a $(2^{-\kappa}, \ell)$-laminar partition of $[n]$.

We now show that any code satisfying the immediacy condition of Equation (46) must be a $(2^{-\kappa}, \ell)$-immediacy code. We use the laminar partition defined above. Fix $1 \leq j \leq \ell$ and $B \in P_j$. Let $t = \lg n - \kappa \cdot (\ell - j) - 1$ be as above (note that t satisfies $\lg m \leq t \leq \lg n - 1$), so that $B = (b \cdot 2^{t+1}, (b+1)\cdot 2^{t+1}]$ for some non-negative integer b. Consider any $x, y \in \Sigma_{\mathrm{in}}^n$ that differ on $\mathrm{left}(B) = (b \cdot 2^{t+1}, (b+2^{-\kappa})\cdot 2^{t+1}]$. Thus, there exists $i \in \mathrm{left}(B) \subseteq (b \cdot 2^{t+1}, (b+1/2)\cdot 2^{t+1}]$ such that $x_i \neq y_i$. The immediacy condition of Equation (46) then gives that

$$\Delta_{\mathrm{Hamming}}\left(T(x)_B, T(y)_B\right) \geq \delta \cdot 2^{t+1} > 2^{-\kappa} \cdot 2^{t+1} = |\mathrm{left}(B)| . \tag{48}$$

Here the second inequality uses $\delta > 2^{-\kappa}$. Thus, we see that the Hamming distance between $T(x)_B$ and $T(y)_B$ is strictly more than the size of $\mathrm{left}(B)$ whenever $x_{\mathrm{left}(B)}$ and $y_{\mathrm{left}(B)}$ differ. This implies that $T(x)_{\mathrm{right}(B)}$ and $T(y)_{\mathrm{right}(B)}$ must differ whenever $x_{\mathrm{left}(B)}$ and $y_{\mathrm{left}(B)}$ differ. This establishes that T is a $(2^{-\kappa}, \ell)$-immediacy code.

We now apply Theorem 8 to see that a tree code $T : \Sigma_{\mathrm{in}}^n \to \Sigma^n$ satisfying the immediacy condition of Equation (46) must therefore satisfy

$$\frac{\lg|\Sigma|}{\lg|\Sigma_{\mathrm{in}}|} \geq 2^{-\kappa} \cdot \ell \overset{\text{Equation (47)}}{\geq} \frac{\delta \cdot \lg \frac{n}{2m}}{2 \cdot \lg(2/\delta)}. \tag{49}$$

Recall that $m = (k_0/\epsilon) \cdot \lg n$, where k_0, ϵ do not grow with n. Thus, Equation (49) shows that any constant rate tree code satisfying the immediacy condition Equation (46) of the GHKRZW construction must have a distance parameter δ satisfying

$$\frac{\delta}{\log(1/\delta)} = O(1/\log n). \tag{50}$$

In particular, a code with the immediacy guarantee that the GHKRZW construction achieves cannot achieve a distance parameter δ that is $\omega\left(\frac{\log\log n}{\log n}\right)$.

References

1 I. Ben Yaacov, G. Cohen, and A. K. Narayanan. Candidate tree codes via Pascal determinant cubes. In *Proc. Approximation, Randomization, and Combinatorial Optimization, APPROX/RANDOM*, volume 207 of *LIPIcs*, pages 54:1–54:22, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.54.

2 I. Ben Yaacov, G. Cohen, and T. Yankovitz. Explicit binary tree codes with sub-logarithmic size alphabet. In *Proc. ACM Symposium on Theory of Computing (STOC)*, STOC, pages 595–608, 2022. doi:10.1145/3519935.3520033.

3 S. Bhandari and P. Harsha. A note on the explicit constructions of tree codes over polylogarithmic-sized alphabet, February 2020. arXiv:2002.08231.

4 Z. Brakerski and Y. T. Kalai. Efficient interactive coding against adversarial noise. In *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 160–166, 2012. doi:10.1109/FOCS.2012.22.

5 Z. Brakerski, Y. T. Kalai, and R. R. Saxena. Deterministic and efficient interactive coding from hard-to-decode tree codes. In *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 446–457, 2020. doi:10.1109/FOCS46700.2020.00049.

6 Z. Brakerski and M. Naor. Fast algorithms for interactive coding. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 443–456, 2013. doi:10.1137/1.9781611973105.32.

7 M. Braverman. Towards deterministic tree code constructions. In *Proc. Innovations in Theoretical Computer Science Conference (ITCS)*, pages 161–167, 2012. doi:10.1145/2090236.2090250.

8 G. Cohen, B. Haeupler, and L. J. Schulman. Explicit binary tree codes with polylogarithmic size alphabet. In *Proc. ACM Symposium on Theory of Computing (STOC)*, pages 535–544, 2018. Full version available on ECCC at https://eccc.weizmann.ac.il/report/2018/032/. doi:10.1145/3188745.3188928.

9 G. Cohen and S. Samocha. Palette-alternating tree codes. In *Proc. Computational Complexity Conference (CCC)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:29, 2020. doi:10.4230/LIPIcs.CCC.2020.11.

10 T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, second edition, 2006.

11 M. Dall'Agnol, T. Gur, and O. Lachish. A structural theorem for local algorithms with applications to coding, testing, and verification. *SIAM J. Comput.*, 52(6):1413–1463, 2023. doi:10.1137/21M1422781.

12 W. Evans, M. Klugerman, and L. J. Schulman. Postscript to "Coding for interactive communication". Unpublished work, 1994. URL: http://users.cms.caltech.edu/~schulman/Papers/intercodingpostscript.txt.

13 R. Gelles. Coding for interactive communication: A survey. *Foundations and Trends in Theoretical Computer Science*, 13(1-2):1–157, 2017. doi:10.1561/0400000079.

14 R. Gelles, B. Haeupler, G. Kol, N. Ron-Zewi, and A. Wigderson. Towards optimal deterministic coding for interactive communication. In *Proc. Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1922–1936, 2016. doi:10.1137/1.9781611974331.ch135.

15 R. Gelles, A. Moitra, and A. Sahai. Efficient and explicit coding for interactive communication. In *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 768–777, 2011. doi:10.1109/FOCS.2011.51.

16 M. Ghaffari and B. Haeupler. Optimal Error Rates for Interactive Coding II: Efficiency and List Decoding. In *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 394–403, 2014. doi:10.1109/FOCS.2014.49.

17 O. Goldreich. On the lower bound on the length of relaxed locally decodable codes. In *Computational Complexity and Local Algorithms : On the Interplay Between Randomness and Computation*, pages 151–192. Springer Nature Switzerland, 2025. Preliminary version: Electronic Colloquium on Computational Complexity (ECCC): TR23-064. doi:10.1007/978-3-031-88946-2_11.

18 T. Gur and O. Lachish. On the power of relaxed local decoding algorithms. *SIAM Journal on Computing*, 50(2):788–813, 2021. doi:10.1137/19M1307834.

19 R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950. doi:10.1002/j.1538-7305.1950.tb00463.x.

20 J. Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972. doi:10.1109/tit.1972.1054893.

21 S. Kamath and V. Anantharam. A new dual to the Gács-Körner common information defined via the Gray-Wyner system. In *Proc. Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1340–1346, September 2010. doi:10.1109/ALLERTON.2010.5707069.

22 J. Katz and L. Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proc. ACM Symposium on Theory of Computing (STOC)*, pages 80–86, May 2000. doi:10.1145/335305.335315.

23 C. Moore and L. J. Schulman. Tree codes and a conjecture on exponential sums. In *Proc. Innovations in Theoretical Computer Science Conference (ITCS)*, pages 145–154, 2014. doi:10.1145/2554797.2554813.

24 T. Moud, A. Rosen, and R. Rothblum. Locally testable tree codes. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5523–5559, 2025. doi:10.1137/1.9781611978322.189.

25 A. K. Narayanan and M. Weidner. On decoding Cohen-Haeupler-Schulman tree codes. In S. Chawla, editor, *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1337–1356, 2020. doi:10.1137/1.9781611975994.81.

26 P. Pudlák. Linear tree codes and the problem of explicit constructions. *Linear Algebra and its Applications*, 490:124–144, 2016. doi:10.1016/j.laa.2015.10.030.

27 L. J. Schulman. Communication on noisy channels: a coding theorem for computation. *Proc. IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 724–733, 1992. doi:10.1109/SFCS.1992.267778.

28 L. J. Schulman. Deterministic coding for interactive communication. In *Proc. ACM Symposium on Theory of Computing (STOC)*, pages 747–756, 1993. doi:10.1145/167088.167279.

29 L. J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. doi:10.1109/18.556671.

30 L. J. Schulman and P. Srivastava. Online codes for analog signals. *IEEE Transactions on Information Theory*, 65(10):6633–6649, 2019. doi:10.1109/TIT.2019.2919632.

31 C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948. doi:10.1002/j.1538-7305.1948.tb01338.x.

32 M. Sipser and D. A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996. doi:10.1109/18.556667.

33 M. A. Tsfasman, S. G. Vladut, and T. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, 1982. doi:10.1002/mana.19821090103.

34 T. Yankovitz. CHS-alike $1/O(\log \log n)$-rate tree codes from elementary binary shifts. Electronic Colloquium on Computational Complexity (ECCC): TR25-167, November 2025. URL: https://eccc.weizmann.ac.il/report/2025/167/.

Beyond Bilinear Complexity: What Works and What Breaks with Many Modes?

Cornelius Brand
University of Regensburg, Germany

Radu Curticapean
University of Regensburg, Germany
IT University of Copenhagen, Denmark

Petteri Kaski
Aalto University, Finland

Baitian Li
Columbia University, New York, NY, USA

Ian Orzel
University of Copenhagen, Denmark

Tim Seppelt
IT University of Copenhagen, Denmark

Jiaheng Wang
University of Regensburg, Germany
University of Helsinki, Finland

Abstract

The complexity of bilinear maps (equivalently, of 3-mode tensors) has been studied extensively, most notably in the context of matrix multiplication. While circuit complexity and tensor rank coincide asymptotically for 3-mode tensors, this correspondence breaks down for $d \geq 4$ modes. As a result, the complexity of d-mode tensors for larger fixed d remains poorly understood, despite its relevance, e.g., in fine-grained complexity. Our paper explores this intermediate regime.

First, we give a "graph-theoretic" proof of Strassen's $2\omega/3$ bound on the asymptotic rank exponent of 3-mode tensors. Our proof directly generalizes to an upper bound of $(d-1)\omega/3$ for d-mode tensors. Using refined techniques available only for $d \geq 4$ modes, we improve this bound beyond the current state of the art for ω. We also obtain a bound of $d/2+1$ on the asymptotic exponent of *circuit complexity* of generic d-mode tensors and optimized bounds for $d \in \{4,5\}$.

To the best of our knowledge, asymptotic circuit complexity (rather than rank) of tensors has not been studied before. To obtain a robust theory, we first ask whether low complexity of T and U imply low complexity of their Kronecker product $T \otimes U$. While this crucially holds for rank (and thus for circuit complexity in 3 modes), we show that assumptions from fine-grained complexity rule out such a *submultiplicativity* for the circuit complexity of tensors with many modes. In particular, assuming the Hyperclique Conjecture, this failure occurs already for $d = 8$ modes. Nevertheless, we can salvage a restricted notion of submultiplicativity.

From a technical perspective, our proofs heavily make use of the *graph tensors* T_H, as employed by Christandl and Zuiddam (*Comput. Complexity* 28 (2019) 27–56) and Christandl, Vrana and Zuiddam (*Comput. Complexity* 28 (2019) 57–111), whose modes correspond to the vertices of undirected graphs H. We make the simple but conceptually crucial observation that Kronecker products $T_G \otimes T_H$ are isomorphic to T_{G+H}, and that G and H may also be *fractional* graphs. By asymptotically converting generic tensors to specific graph tensors, we can use nontrivial results from algorithmic graph theory to study the rank and complexity of d-mode tensors for fixed d.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases arithmetic circuits, tensor rank, bilinear complexity, graph tensors

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.11

Related Version *Full Version*: https://arxiv.org/abs/2602.11975 [6]

Funding Funded by the European Research Council (ERC) under grant agreement no. 101077083 (CountHom). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. *Baitian Li*: Supported in part by NSF Grant CCF-2238221, a Packard Foundation Fellowship, and a Columbia SEAS Presidential Fellowship.



© Cornelius Brand, Radu Curticapean, Petteri Kaski, Baitian Li, Ian Orzel, Tim Seppelt, and Jiaheng Wang;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 11; pp. 11:1–11:23





Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Ian Orzel: Supported by the ERC under grant agreement no. 101125652 (ALBA).
Jiaheng Wang: Supported by the Helsinki Institute of Information Technology (HIIT).

Acknowledgements We thank the anonymous referees, whose comments substantially improved the presentation.

1 Introduction

In several fundamental computational problems, the input data is represented by two vectors u, v and the output is a vector $f(u, v)$ that captures a meaningful combination of the input data. Prominent examples are *matrix multiplication* (when the input matrices are flattened to vectors), *polynomial multiplication* (when the input polynomials are given as coefficient vectors), and more general *convolution* problems such as the *subset convolution* in parameterized and exact algorithms, which is essentially multiplication in the algebra of square-free multivariate polynomials.

Bilinear Maps and Rank. The above problems all ask to evaluate maps $f : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}^n$ that are *bilinear* over the field $\mathbb{F}$; they satisfy $f(\alpha x + \beta x', y) = \alpha f(x, y) + \beta f(x', y)$, with an analogous rule for the second argument. Bilinear maps enjoy a strong connection between computational complexity and *rank*, an algebraic complexity measure. The rank $\mathrm{R}(f)$ is the minimum number of terms required to express f as a linear combination of rank-1 maps, where a rank-1 map is a product of linear functions in disjoint variable sets.

Rank and Kronecker Products. Several extremely well-studied bilinear maps, such as matrix multiplication and subset convolution, have a very useful structure: They can be expressed as *Kronecker products* of lower-dimensional bilinear maps. To make this precise, given vectors $u, u' \in \mathbb{F}^n$, define $u \otimes u' \in \mathbb{F}^{n^2}$ by setting $(u \otimes u')_{i,j} = u_i u'_j$ and viewing the index set $[n]^2$ as $[n^2]$ under some bijection. For bilinear maps $f, f' : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}^n$, the *Kronecker product* $f \otimes f' : \mathbb{F}^{n^2} \times \mathbb{F}^{n^2} \to \mathbb{F}^{n^2}$ is defined by mapping

$$(u \otimes u', v \otimes v') \mapsto f(u, v) \otimes f'(u', v'),$$

and extending this rule bilinearly. For example, the Kronecker product of the bilinear map g_n for multiplication of two $n \times n$ matrices and the map g_m for multiplication of two $m \times m$ matrices is the map $g_n \otimes g_m \equiv g_{nm}$ for multiplication of two $nm \times nm$ matrices. Likewise, subset convolution over a universe of size n (i.e., square-free products of n-variate multilinear polynomials) is the Kronecker power of subset convolution over a universe of size 1.

Rank and Kronecker products interact very favorably: If f_1 and f_2 have rank $r_1 = \mathrm{R}(f_1)$ and $r_2 = \mathrm{R}(f_2)$, then $\mathrm{R}(f_1 \otimes f_2) \leq r_1 r_2$, since the rank decompositions for f_1 and f_2 can be combined to a rank decomposition for $f_1 \otimes f_2$. Unlike for linear maps, this combined rank decomposition of bilinear maps may however not be optimal, and we may have $\mathrm{R}(f_1 \otimes f_2) < r_1 r_2$. Consequently, we say that rank is *submultiplicative* under Kronecker products.

Asymptotic Rank and Circuit Complexity. The rank of bilinear maps like matrix multiplication and subset convolution of increasing dimensions can be understood by fixing a constant-dimension bilinear map $g : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}^n$ of rank $r = \mathrm{R}(g)$ and considering its *Kronecker power* $g^{\otimes k} = g \otimes \ldots \otimes g$, a bilinear map between spaces of dimension n^k. As discussed above, we then have $\mathrm{R}(g^{\otimes k}) \leq r^k$, but this bound may be exponentially loose: One of the simplest examples of this phenomenon is the subset convolution map $g : \mathbb{F}^2 \times \mathbb{F}^2 \to \mathbb{F}^2$

on 1-element universes, or what is essentially the same, the three-qubit W-state in quantum information theory. We have $\mathrm{R}(g) = 3$ and thus $\mathrm{R}(g^{\otimes k}) \leq 3^k$, while it is known that $\mathrm{R}(g^{\otimes k}) \leq O(2^k k)$. Combined with algorithmic ideas dating back to Yates [50], this nontrivial rank bound for $g^{\otimes k}$ translates into nontrivial $2^{k+o(k)}$ size circuits for subset convolution on k-element universes. Similar observations are crucial in the study of matrix multiplication.

This algorithmic connection between rank and algorithmic complexity of Kronecker powers motivates defining the *asymptotic rank exponent* $\omega(g)$ of a bilinear map $g : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}^n$ as the infimum over all $\beta > 0$ such that $\mathrm{R}(g^{\otimes k}) \leq O(n^{\beta k})$. Likewise, we define the *asymptotic circuit complexity exponent* $\eta(g)$ as the infimum over all $\beta > 0$ such that $g^{\otimes k}$ has arithmetic circuits of size $O(n^{\beta k})$. For bilinear maps, these two exponents are known to be *equal* (cf. the discussion following Theorem 8), so asymptotic circuit complexity can be understood entirely in terms of asymptotic rank, which in turn is amenable to techniques from pure mathematics.

Trivially, the asymptotic rank exponent of every bilinear map is bounded by 2, as $\mathrm{R}(g) \leq n^2$. However, Strassen [45] showed a surprising bound of $2\omega/3 < 2$, where $\omega < 3$ is the asymptotic rank exponent of matrix multiplication. This translates directly to improved algorithms, e.g., for convolution problems [7]. The *asymptotic rank conjecture* [46] postulates that this upper bound can be pushed further down to 1, i.e., that every bilinear map g among n-dimensional vector spaces satisfies $\mathrm{R}(g^{\otimes k}) \leq O(n^k)$. This would lead to near-linear time algorithms for evaluating any map of the form $g^{\otimes k}$, which directly implies that $n \times n$ matrices can be multiplied with $n^{2+o(1)}$ operations. Moreover, it was recently shown that the asymptotic rank conjecture implies breakthrough exponential-time algorithms, e.g., for the permanent [5], for the chromatic number of graphs [3], and for the set cover problem [4, 40].

More Modes. So far, we described *bi*-linear maps, which take *two* input vectors and produce an output vector. As a natural generalization, a *multi*-linear map takes d input vectors, called *modes*, to a single output vector, and is linear in each argument. Multilinear maps are mentioned as an open research direction, e.g., in the survey by Wigderson and Zuiddam [49, Sect. 13]. Beyond their intrinsic algebraic appeal, they capture multi-party states in quantum information theory and problems in fine-grained complexity.

A prominent example for a multilinear map with d modes is the *iterated matrix multiplication*, which takes as input d matrices $A_1, \ldots, A_d$ (flattened to vectors) and outputs their product $A_1 \ldots A_d$. Other examples include the *determinant* and *permanent* of an $n \times n$ matrix A, which are multilinear *forms* (rather than *maps*, i.e., they output scalars) with n modes corresponding to the columns of A. In fine-grained complexity, the k-hyperclique problem admits a natural formulation as a multilinear form with k modes [35]. The complexity of this problem is interesting even for small, fixed values of k, starting with $k = 4$, and has attracted significant attention in recent years, especially as a source of conditional hardness [32, 8, 24].

▶ Remark. We will only consider multilinear *forms* in the following, since maps with $d - 1$ modes correspond to forms with d modes by taking duals. After fixing coordinates, multilinear forms g with d modes can be viewed as set-multilinear polynomials: Denoting the entries of the input vector $\mathbf{x^{(i)}}$ at mode $i \in [d]$ by $x_1^{(i)}, \ldots, x_n^{(i)}$, we have

$$g(\mathbf{x^{(1)}}, \ldots, \mathbf{x^{(d)}}) = \sum_{i_1,\ldots,i_d} t_{i_1,\ldots,i_d}\, x_{i_1}^{(1)} \ldots x_{i_d}^{(d)}$$

with coefficients $t_{i_1,\ldots,i_d} \in \mathbb{F}$ that fully specify g. Arranging these coefficients in an array with d modes, we call this array the *tensor* of g, and overloading terminology further, we call g *itself* a tensor with d modes. After tacitly performing all required identifications, we can view bilinear maps as 3-mode tensors.

Rank versus Circuit Complexity for More Modes. Rank and circuit complexity can be generalized directly to tensors with an arbitrary number of modes. For $d \geq 4$ modes however, the correspondence between asymptotic circuit complexity and asymptotic rank breaks. As a folklore example of this, consider the tensor P_n with four n-dimensional modes defined by

$$P_n(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = \left(\sum_{j=1}^{n} x_j y_j \right) \left(\sum_{j=1}^{n} w_j z_j \right). \tag{1}$$

In algebraic complexity, P_n is known as a product of inner products (PIP). Since $P_n \otimes P_m \equiv P_{nm}$, the asymptotic rank exponent of every single P_n equals the exponent of the increasing sequence $\mathrm{R}(P_n)$ as $n \to \infty$. The defining formula for P_n of size $O(n)$ implies an asymptotic circuit complexity exponent of 1, while the asymptotic rank exponent of P_n is 2: Multiplying out (1) gives an upper bound of n^2, matched by a simple flattening lower bound.

Thus, unlike for $d = 3$ modes, asymptotic rank and asymptotic circuit complexity of 4-mode tensors for $d \geq 4$ need not agree. The separation can be exacerbated by generalizing the PIP tensor to a fixed number of $2d$ modes, which increases the asymptotic rank exponent to d, while the asymptotic circuit complexity exponent stays 1. As the example shows, the separation holds even for $\Pi\Sigma\Pi$-formula complexity, which morally still resembles rank in that rank decompositions can be viewed as $\Sigma\Pi\Sigma$-formulæ.

Full Version. Before explaining our results, we wish to highlight that an extended discussion as well as all omitted proofs can be found in the full version of the present paper [6].

Our Results: Phenomena of Tensors with More Modes

Starting from the observation that rank and circuit complexity need not agree for relevant tensors with $d \geq 4$ modes, we are led to systematically study asymptotic rank and asymptotic circuit complexity for d-mode tensors when $d \geq 3$ is fixed and moderately large or may tend to infinity. As part of this study, we obtain better upper bounds on asymptotic rank and asymptotic circuit complexity for generic d-mode tensors. Moreover, we investigate to which extent desirable properties like submultiplicativity fail for the circuit complexity of d-mode tensors with $d \geq 4$. The foundation to our results are *graph tensors* in the form used by Christandl and Zuiddam [15] and Christandl, Vrana and Zuiddam [14], which allow us to import techniques from graph theory.

We elaborate on these tensors later; for now, it suffices to note that the modes of the graph tensor T_H correspond to the vertices of a graph H. Graph tensors can be viewed as a special class of tensors that admit representation as a *tensor network*. Tensor networks have received extensive attention across a range of disciplines, each pursuing its own distinctive terminology and notation, from pure mathematics and physics [33, 37, 39] to the study of probabilistic graphical models in artificial intelligence and machine learning [30, 31, 38], and to the study of restricted circuit models in algebraic computation, where tensor networks give rise to a class of set-multilinear circuits for evaluating multilinear maps [1].

Generic Asymptotic Rank. To our surprise and best of knowledge, an analogue of Strassen's $2\omega/3$ bound on the asymptotic exponent of 3-mode tensors for d-tensors with $d \geq 4$ is not given in the literature. We give a simple proof for an upper bound of $(d-1)\omega/3$ on the asymptotic exponent of all d-mode tensors for $d \geq 3$ using the framework of graph tensors.

Our proof admits some freedom in the choice of low-rank tensor L to reduce to: While Strassen's original proof requires L to be the matrix multiplication tensor, which will turn out to be the graph tensor of a triangle K_3, our proof allows L to be the graph tensor of the complete graph K_s for $s \leq d$. In the case $d = 3$, we recover the aforementioned bound of

$(d-1)\omega/3$, but for $d \geq 4$, better upper bounds on the exponent of the graph tensor of K_d are known. Moreover, with a nontrivial application of Strassen's *laser method*, we improve the best known exponent on the graph tensor for K_4 and obtain:

▶ **Theorem 1** (Upper bound on asymptotic rank). *For every d-mode tensor with $d \geq 4$, the asymptotic rank exponent of T is at most* $0.772318(d-1)$.

For comparison, the PIP tensor presented above (1) establishes a lower bound of $\lfloor d/2 \rfloor$ on the rank exponent of d-mode tensors. Moreover, $\omega = 2$ would imply an upper bound of $2/3 \cdot (d-1)$. This is elaborated more formally in Remark 21.

Submultiplicativity. While the PIP tensor rules out a functional dependence between rank and circuit complexity for $d \geq 4$ modes, we ask whether circuit complexity is at least *submultiplicative*: From *rank* decompositions for tensors T and U, we can explicitly construct a *rank* decomposition for $T \otimes U$ of rank $\mathrm{R}(T) \cdot \mathrm{R}(U)$. If T and U are given by circuits of small size, can we analogously expect a circuit of small size for $T \otimes U$? Restricted circuit models are known to have submultiplicativity under Kronecker powering; in particular, tensor networks admit submultiplicativity of amortized cost via contraction trees [1]. But does *unrestricted* circuit complexity have submultiplicativity?

Our results strongly suggest a negative answer, but we obtain such statements only *conditionally*, as an unconditional result would imply strong circuit lower bounds. In the following, we write $\mathrm{C}(T)$ for the size of a smallest arithmetic circuit computing the tensor T.

▶ **Theorem 2** (Submultiplicativity of C implies VP = VNP). *There are explicit tensors $T_1, T_2, \ldots$ and $U_1, U_2, \ldots$, where T_d, U_d for $d \in \mathbb{N}$ are d-mode 0-1-tensors, such that the following holds: If $\mathrm{C}(T_d \otimes U_d) \leq \mathrm{poly}(d, \mathrm{C}(T_d), \mathrm{C}(U_d))$ for all $d \in \mathbb{N}$, then* VP = VNP.

Thus, the assumption VP $\neq$ VNP rules out submultiplicativity on d-mode tensors for all $d \in \mathbb{N}$, even when a polynomial slack on the complexity bound of the product tensor is allowed. Note that VP $\neq$ VNP implies super-polynomial circuit lower bounds for permanents [47]. If we assume even further that permanents require exponential-size circuits, then we can rule out submultiplicativity on d-mode tensors for concrete values of d.

▶ **Theorem 3** (Submultiplicativity of C implies faster permanents). *For all $0 < c < 1$, there are explicit $\lceil 4/c \rceil^2$-mode 0-1-tensors $T_1, T_2, \ldots$ and $U_1, U_2, \ldots$ such that the following holds: If $\mathrm{C}(T_n \otimes U_n) \leq O(\mathrm{C}(T_n) \cdot \mathrm{C}(U_n))$, then permanents have circuits of size $2^{cn} n^{O(1)}$.*

In particular, unless the permanent has circuits of size $2^{o(n)}$, there is a constant $D \in \mathbb{N}$ such that C is not submultiplicative for D-mode tensors. This assumption on the permanent is implied by a suitable non-uniform variant of the *exponential-time hypothesis*. For a more fine-grained approach, we can consider concrete values of $c > 0$ in Theorem 3. For example, if the permanent does not have circuits of size $2^{0.1n}$, then submultiplicativity fails for tensors with $d = \lceil 4/0.1 \rceil^2 = 40^2 = 1600$ modes. If even circuits of size $2^{0.8n}$ can be ruled out for the permanent, then submultiplicativity fails already on $d = 25$ modes, the smallest d for which Theorem 3 still yields a statement.

As for the plausibility of this assumption, Knuth asks in *The Art of Computer Programming* [29, Volume 2, §4.6.4, Problem 11] whether there is any way to evaluate the permanent of a general $n \times n$ matrix using fewer than 2^n arithmetic operations. This is still an open problem. Note that there is no asymptotic notation around 2^n. [1]

[1] Knuth's question is currently known to admit an affirmative answer only under the asymptotic rank conjecture [5]; more specifically, the asymptotic rank conjecture implies uniform arithmetic circuits of size 2^{cn} for the permanent where $c \approx 0.9183$.

Using another conjecture from fine-grained complexity theory, we approach $d = 4$, the smallest number of modes where rank and circuit complexity differ: The (h, k)-*hyperclique conjecture* [35] rules out $O(n^{k-\varepsilon})$-time algorithms for detecting k-hypercliques in h-uniform hypergraphs (see the full version for details).

▶ **Theorem 4** (Submultiplicativity of C implies faster hypercliques)**.** *For every even $d \geq 8$, there are explicit d-mode 0-1-tensors $T_1, T_2, \ldots$ and $U_1, U_2, \ldots$ such that: If $\mathrm{C}(T_n \otimes U_n) \leq O(\mathrm{C}(T_n) \cdot \mathrm{C}(U_n))$, then the non-uniform $(\frac{d}{2} - 1, \frac{d}{2})$-hyperclique conjecture fails.*

While our results suggest that circuit complexity is not submultiplicative, we can efficiently construct a circuit for $T \otimes U$ from a rank decomposition of T and a circuit for U.

▶ **Theorem 5** (Mixed asymptotic rank and circuit complexity)**.** *Given d-mode tensors T and U, where T has asymptotic rank exponent $\psi \geq 1$ and U has asymptotic circuit complexity exponent $\eta \geq 1$, the asymptotic circuit complexity exponent of $T \otimes U$ is at most $\eta + \psi$.*

Asymptotic Circuit Complexity. With tools like Theorem 5 and insights from graph theory, we adapt the graph-theoretic proof of our asymptotic rank bound in Theorem 1 to obtain stronger upper bounds for asymptotic circuit complexity rather than rank.

First, using a known nontrivial upper bound on the treewidth of line graphs of complete graphs [25], we obtain an upper bound on the circuit complexity of generic d-mode tensors:

▶ **Theorem 6** (Upper bound on asymptotic circuit complexity)**.** *The asymptotic circuit complexity exponent of every d-mode tensor T is less than $d/2 + 1$.*

While a lower bound of $d/2$ on the asymptotic *rank* exponent of d-mode tensors is immediate from (1), an unbounded *circuit complexity* exponent in d would constitute a strong circuit lower bound and hence seems exceedingly unlikely to be obtainable.

Next, we observe that Theorem 6 does not yield optimal upper bounds for small fixed d. For example, the theorem gives a circuit complexity exponent bound of $5/2$ for $d = 4$, but $5/2$ is even above the asymptotic *rank* upper bound of 2.317. To obtain meaningful upper bounds for small fixed d, we combine Theorem 5 with fractional decompositions of specific small graphs into low-rank and low-treewidth parts. These graph decompositions are found through computer-aided optimization. Our technique is applicable for arbitrary fixed d, albeit with higher computational load, and we obtain exemplary results for $d = 4, 5$:

▶ **Theorem 7** (Upper bound on asymptotic circuit complexity for small orders)**.** *For every 4-mode tensor T, the asymptotic circuit complexity exponent of T is at most 2.2928. For every 5-mode tensor, it is at most 2.8738.*

For comparison, the upper bounds on the asymptotic *rank* exponents from Theorem 1 for $d = 4, 5$ are about 2.317 and 3.0893, respectively. Table 1 provides a comprehensive overview of the kinds of results we obtain.

Our Techniques: New Insights into Graph Tensors

The protagonists in our proofs are so-called *graph tensors*, i.e., tensors with many modes that are composed from 2-mode tensors by using graphs as "composition blueprints." Such tensors appear in quantum information theory as *matrix-product states* and *projected entangled pairs* [16], in quantum machine learning as (restricted) *tensor networks* [26], and in counting complexity (under the right abstraction) as *Holant problems* [48, 12, 13]. For us, graph tensors act as a bridge between graph theory and multilinear algebra that allows us to import

Table 1 Bounds on asymptotic rank and circuit complexity exponents of generic d-mode tensors. The first row states rank upper bounds obtained via Theorem 1. The next two rows state upper bounds on asymptotic circuit exponents. For comparison, we include the known case $d = 3$ and the strongest known lower bounds.

Method	$d = 3$	$d = 4$	$d = 5$	$d = 6$	$d = 10$
$\underset{\sim}{\mathrm{R}}(T)$, asymptotic submult. (Theorem 1)	1.59	2.32	3.09	3.87	6.96
$\underset{\sim}{\mathrm{C}}(T)$, treewidth-based (Theorem 30)	2.00	2.50	3.20	3.67	5.80
$\underset{\sim}{\mathrm{C}}(T)$, specialized decompositions (Theorem 7)	–	2.30	2.88	–	–
Flattening lower bound on $\underset{\sim}{\mathrm{R}}(T)$ from (1)	1.00	2.00	2.00	3.00	5.00

nontrivial graph-theoretic concepts (e.g., graph decompositions and treewidth) and results (e.g., bounds on treewidth for specific graph classes) to the study of tensors with many modes. Moreover, graph tensors are closed under Kronecker products, which makes them particularly well-suited for studying the asymptotic behavior of rank and circuit complexity. They can be generalized further to tensors admitting representation by tensor networks or factor-graph models (e.g. [1, 31]), but graph tensors will be sufficient for our present purposes.

As a simple example, the outer tensor product of k tensors on b modes gives a tensor T with kb modes. In the extreme case of $b = 1$, the resulting tensor T has rank 1, but already for $b = 2$, examples of large rank such as the PIP tensor (1) emerge. Graph tensors are obtained by additionally allowing identification (with flattening) of modes. Such tensors admit a natural interpretation as generic instances of Holant problems; we discuss the former in the following. The interpretation as quantum states is discussed in the full version.

Graph Tensors in Terms of Holant Problems. To analyze the complexity of tensors $T_{H,n}$ for graphs H, we observe that they can be interpreted as *Holant problems*, which are very well-studied in counting complexity [48, 12, 10, 11, 44]: On input a graph $G = (V, E)$, a Holant problem asks to compute a weighted count of edge-assignments $s\colon E \to [r]$, with weights determined locally at vertices; the vast majority of the literature focuses on the case $r = 2$. As a concrete example, the number of *perfect matchings* in G is the number of edge-assignments $s\colon E(G) \to \{0, 1\}$ such that every vertex $v \in V$ has exactly one 1-labeled edge in the local restriction $s|_{I(v)}$ of the global assignment s.

In general Holant problems, the weights are determined by *signatures* $f_v\colon [r]^{I(v)} \to \mathbb{C}$ at the vertices $v \in V$. On input G and signatures $\{f_v\}_{v\in V}$, we then wish to determine

$$\mathrm{Holant}(G) = \sum_{s\colon E\to[r]} \prod_{v\in V} f_v(s|_{I(v)}). \tag{2}$$

Holant problems have been studied from a very different perspective in the literature: Usually, a set of possible signatures $\mathcal{F}$ is fixed, and the input is a graph G with signatures $f_v \in \mathcal{F}$. We consider the converse setting: The graph G is fixed, but the signatures can vary freely in that they are provided as input vectors to the $|V(G)|$ modes of the multilinear form $T_{G,n}$.

As an example of particular relevance for us, fix G as the 4-regular toroidal grid on $t \times t$ vertices. Then $T_{G,2}$ is a t^2-mode tensor with a copy of $(\mathbb{C}^2)^{\otimes 4}$ at each mode. By specifying a vertex signature $f_v : \{0, 1\}^4 \to \mathbb{C}$ for each grid vertex and inputting it as a vector $u_v \in \mathbb{C}^{16}$ into the multilinear form $T_{G,2}$, we can count perfect matchings, Eulerian subgraphs, or evaluate any Holant problem definable on the fixed grid G. In particular, for growing grid sizes, this includes VNP-hard problems: By appropriately engineering the

signatures, we express the permanent of arbitrary $n \times n$ matrices as a particular Holant problem on the fixed $n \times n$ *grid*, and we obtain hardness of the grid graph tensor. On the algorithmic side, known fixed-parameter tractable algorithms over low-treewidth graphs establish a nontrivial complexity upper bound of $n^{O(t)}$ on the evaluation of $T_{G,n}$, where our generic results would only yield an $n^{O(t^2)}$ bound. Such upper and lower complexity bounds informed by (parameterized) algorithmic graph theory are ubiquitous in our arguments.

2 Preliminaries

For a nonnegative integer n we write $[n] = \{1, 2, \ldots, n\}$. We write $\mathbb{N} = \{1, 2, \ldots\}$. The Kronecker delta δ_{ij} is 1 if $i = j$ and 0 else.

Algebraic Complexity

Tensors. For convenience, we work with tensors in coordinates and view them as set-multilinear polynomials with an underlying tuple of modes. Let $\mathbb{F}$ be a field, and $\mathbb{F}[X]$ the ring of polynomials over $\mathbb{F}$ in a tuple X of one or more indeterminates. We assume X is partitioned into d nonempty sets $X_1, X_2, \ldots, X_d$ called *modes*, each of *dimension* $|X_i|$. A monomial is *set-multilinear* if each mode has exactly one indeterminate of degree one in the monomial, and all others have degree zero. A polynomial is *set-multilinear* if all of its monomials are. A *d-mode tensor* (or a *d-tensor*) T is a set-multilinear polynomial together with a d-tuple $(X_1, X_2, \ldots, X_d)$ of modes. We call d the *degree* or *order* of the tensor. We suppress d and the modes when they are clear from context. Suppressing also the variable names, we write $(\mathbb{F}^n)^{\otimes d}$ for the set of d-tensors over $\mathbb{F}$ with each mode of dimension n.

Addition and scalar multiplication of tensors with identical tuples of modes are inherited from polynomial arithmetic. The *Kronecker product* of d-tensors $S \in \mathbb{F}[X]$ and $T \in \mathbb{F}[Y]$ with X, Y disjoint is the d-tensor $S \otimes T$ obtained from the polynomial product ST by viewing the Cartesian products $X_i \times Y_i$ for $i \in [d]$ as the d modes. When the sets X, Y are not disjoint, such as when taking a Kronecker power of a tensor, we tacitly assume a disjoint copy of one of the sets of indeterminates is formed before taking the Kronecker product.

Projection, Equivalence, and Restriction. A *simple substitution* for sets of indeterminates X, Y is a map $\sigma : Y \to \mathbb{F} \cup X$. For tensors $S \in \mathbb{F}[X]$ and $T \in \mathbb{F}[Y]$ we say that S is a *projection* of T and write $S \leq_{\mathrm{s}} T$ if there exists a simple substitution σ such that the polynomial identity $S = T^\sigma$ holds and the image of each mode Y_j of T under σ intersects at most one mode X_i of S. A simple substitution is *nonscalar* if it does not assume values in $\mathbb{F}$. We say that S and T are *equivalent* and write $S \equiv T$ if the polynomial identity $S = T^\sigma$ holds for a bijective nonscalar σ that is bijective on modes. For d-tensors S and T we say that T *restricts to* S and write $S \leq T$ if the polynomial identity $S = T^\mu$ holds for a substitution $\mu : Y \to \mathbb{F}[X]$ such that for all for all modes Y_j and all indeterminates $y \in Y_j$ the polynomial $\mu(y)$ is a linear polynomial in the indeterminates X_j.

Tensor Rank. A nonzero d-tensor T has *rank one* if there exist linear polynomials $\ell_j \in \mathbb{F}[X_j]$ for $j \in [d]$ such that the polynomial identity $T = \ell_1 \ell_2 \cdots \ell_d$ holds. The *rank* $\mathrm{R}(T)$ of a d-tensor T is the minimum number of rank one tensors whose sum is T. (Note that $d = 2$ recovers matrix rank.) For two d-tensors S and T, we have $\mathrm{R}(S \otimes T) \leq \mathrm{R}(S) \cdot \mathrm{R}(T)$, with equality for $d = 2$ but strict inequality may hold for $d \geq 3$. This property is commonly referred to as the *submultiplicativity* of tensor rank. Moreover, the rank of any tensor $T \in (\mathbb{F}^n)^{\otimes d}$ is obviously bounded by n^d, the number of set-multilinear monomials (and the

stronger bound n^{d-1} can easily be shown). Consequently, $T^{\otimes k}$ has rank at most $n^{(d-1)k}$. The *asymptotic rank* of $T \in (\mathbb{F}^n)^{\otimes d}$ is defined as the limit $\underset{\sim}{\mathrm{R}}(T) = \lim_{k\to\infty} \mathrm{R}(T^{\otimes k})^{1/k}$, which exists by submultiplicativity of rank and Fekete's Lemma (the result is folklore, but see [49, Lemma 2.10] and the references given there). Like rank, also the asymptotic rank satisfies submultiplicativity with respect to Kronecker products. We say that a d-tensor is *concise* if all of its d flattenings to a matrix with one mode forming the rows and all the other modes forming the columns have full rank. A concise tensor $T \in (\mathbb{F}^n)^{\otimes d}$ in particular satisfies both $\mathrm{R}(T) \geq n$ and $\underset{\sim}{\mathrm{R}}(T) \geq n$. The *exponent* of a concise tensor $T \in (\mathbb{F}^n)^{\otimes d}$ is defined by

$$\omega(T) = \inf\{\beta \geq 0 : \mathrm{R}(T^{\otimes k}) = O(n^{\beta k})\}$$

and we have

$$\underset{\sim}{\mathrm{R}}(T) = n^{\omega(T)}.$$

For example, the tensor $\mathrm{MM}_m \in \mathbb{F}^{m^2} \otimes \mathbb{F}^{m^2} \otimes \mathbb{F}^{m^2}$ that represents the bilinear map that multiplies two $m \times m$ matrices for any constant $m \geq 2$ is concise and recovers the exponent ω of square matrix multiplication either via the asymptotic rank identity $\underset{\sim}{\mathrm{R}}(\mathrm{MM}_m) = m^\omega$ or, equivalently, via the tensor exponent identity $\omega(\mathrm{MM}_m) = \omega/2$.

Arithmetic Circuits. Tensor rank fails to capture the arithmetic complexity of tensors for $d \geq 4$, see Example 11. In contrast, circuits model arithmetic complexity more faithfully. An *arithmetic circuit over* $\mathbb{F}[X]$ is a directed acyclic graph with sinks called *outputs*, sources called *inputs*, vertices called *gates*, and arcs called *wires.* Non-input gates are labeled with $+$ and $\times$, while inputs are labeled with elements from $\mathbb{F} \cup X$, and wires are labeled with elements of $\mathbb{F}$. A gate *computes* a polynomial in the obvious inductive manner, and the *size* of a circuit is the number of wires (see e.g. Kayal's [28] lecture notes for background on this particular implementation of the formalism). For a tensor $T \in (\mathbb{F}^n)^{\otimes d}$, we write $\mathrm{C}(T)$ for the size of a smallest circuit computing T, called the *arithmetic circuit complexity* of T.

A rank-r decomposition of $T \in (\mathbb{F}^n)^{\otimes d}$ yields an arithmetic circuit of size $O(d \cdot n \cdot r)$. This trivial bound $\mathrm{C}(T^{\otimes k}) \leq O(d \cdot n^{(d+1)k})$ allows us to define, in analogy with the tensor case, the *circuit exponent* $\eta(T)$ and the *asymptotic circuit complexity* $\underset{\sim}{\mathrm{C}}(T)$ of T via

$$\eta(T) = \inf\{\beta \geq 0 : \mathrm{C}(T^{\otimes k}) \leq O(n^{k\beta})\}, \quad \underset{\sim}{\mathrm{C}}(T) = n^{\eta(T)}.$$

Again, $\underset{\sim}{\mathrm{C}}(T) = \limsup_{k\to\infty} \mathrm{C}(T^{\otimes k})^{1/k}$ can be shown, but it is not clear if the limit exists. The bound $\mathrm{C}(T) = O(d \cdot n \cdot \mathrm{R}(T))$ shows $\eta(T) \leq \omega(T) + 1$, and a classic result of Yates [50] (cf. Knuth [29, §4.6.4]) shows that for $d \geq 2$, concise $T \in (\mathbb{F}^n)^{\otimes d}$, and all $k \geq 1$,

$$\mathrm{C}(T^{\otimes k}) = O(dk \cdot \mathrm{R}(T)^{k+1}). \tag{3}$$

The treatments in the literature we are aware of focus on the cases $d = 2, 3$ of Kronecker powers of matrices and 3-tensors, but the result easily generalizes to $d \geq 4$.

▶ **Theorem 8** (Yates's algorithm). *Let $d \geq 2$ and let $T \in (\mathbb{F}^n)^{\otimes d}$ be concise. Then,*

$$\eta(T) \leq \omega(T), \text{ and therefore } \underset{\sim}{\mathrm{C}}(T) \leq \underset{\sim}{\mathrm{R}}(T).$$

For $d = 3$, it is also true that $\underset{\sim}{\mathrm{C}}(T) \geq \underset{\sim}{\mathrm{R}}(T)$ holds, justifying tensor rank as a measure of arithmetic complexity. As mentioned, this behavior does not extend to $d \geq 4$.

Closure Properties. Since restrictions of rank-1 tensors are themselves rank-1 (or identically zero), $\mathrm{R}(S) \leq \mathrm{R}(T)$ whenever $S \leq T$, and $\mathrm{R}(S) = \mathrm{R}(T)$ if $S \equiv T$. The same statements hold for asymptotic rank. For circuits, it is only true that $\mathrm{C}(S) \leq \mathrm{C}(T)$ when $S \leq_{\mathrm{s}} T$ for $S, T \in (\mathbb{F}^n)^{\otimes d}$, but $S \leq T$ only implies $\mathrm{C}(S) \leq \mathrm{C}(T) + n^{d+1}$. Still, asymptotic circuit complexity admits $\underset{\sim}{\mathrm{C}}(S) \leq \underset{\sim}{\mathrm{C}}(T)$ if $S \leq T$ as a consequence of Yates's algorithm (3). Moreover, we do have that $\mathrm{C}(U) \leq \mathrm{C}(U \otimes V)$ provided $V \neq 0$. The questions of whether and when $\underset{\sim}{\mathrm{C}}$ and C are robust with respect to Kronecker products will be a main theme of the article.

A very general description of these properties in relation to tensor rank (and far beyond) is laid out in the work of Wigderson and Zuiddam [49]. For further background on algebraic complexity we refer to the book of Bürgisser, Clausen and Shokrollahi [9].

Graph Theory

A *graph* G consists of a finite set $V(G)$ of *vertices* and a finite set $E(G)$ of *edges* such that each edge is associated with a set of two vertices called the *end-vertices* or *ends* of the edge. We stress that multiple edges may have the same ends; such edges are called *parallel*. For a vertex $v \in V(G)$ we write $I_G(v) \subseteq E(G)$ for the set of all edges that have v as an end, and call $|I_G(v)|$ the *degree* of v. The maximum degree of a vertex in G is denoted by $\Delta(G)$.

For graphs G and H with disjoint edge sets, the *sum* $G + H$ is the graph defined by $V(G + H) = V(G) \cup V(H)$ and $E(G + H) = E(G) \cup E(H)$. If the edge sets are not disjoint, such as for $G + G$, we assume a disjoint copy of one of the edge sets is taken beforehand.

We call two graphs G and H *isomorphic*, written $G \equiv H$, if there are bijections $\phi\colon V(G) \to V(H)$, $\psi\colon E(G) \to E(H)$ such that every $e \in E(G)$ has the ends $\{v_1, v_2\} \subseteq V(G)$ if and only if $\psi(e)$ has the ends $\{\phi(v_1), \phi(v_2)\} \subseteq V(H)$. Equivalently, $\psi(I_G(v)) = I_H(\phi(v))$ should hold.

Subgraphs. A graph H is a *subgraph* of a graph G if $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$, and common edges of H and G have the same ends. A *subdivision* of a graph G is a graph obtained by applying the following operation zero or more times: select an edge e with end u and v, insert a new vertex w, delete the edge e, insert a new edge with ends u and w, and insert a new edge with ends v and w. A graph H is a *topological subgraph* of a graph G if there exists a subdivision of H that is isomorphic to a subgraph of G.

Treewidth. A *tree decomposition* of a graph G is a pair (T, β) where T is a tree and β assigns to each vertex $a \in V(T)$ a *bag* $\beta(a) \subseteq V(G)$ such that (i) the union of all bags is $V(G)$; (ii) both ends of each edge of G are contained in at least one bag; and (iii) for all $a, c \in V(T)$ and all $b \in V(T)$ on the path from a to c in T, $\beta(a) \cap \beta(c) \subseteq \beta(b)$ holds. The *width* of (T, β) is $\max_{a \in V(T)} |\beta(a)| - 1$. The *treewidth* $\mathrm{tw}(G)$ is the minimum width of a tree decomposition of G. More background on algorithmic graph theory, in particular algorithms using tree decompositions of low width, can be found in the textbook of Cygan et al. [19].

3 Graph Tensors and the Asymptotic Rank

This section develops our conventions for graph tensors and their basic properties in relation to the underlying graphs. We then prove our main theorem (Theorem 1) on bounding the asymptotic rank for d-mode tensors for $d \geq 4$, extending Strassen's result for $d = 3$.

3.1 Graph Tensors

Our graphs are undirected, loopless, and may have parallel edges. Let G be a graph and let $n \in \mathbb{N}$. For a mapping $f\colon E(G) \to [n]$ and a vertex $v \in V(G)$, let us write $f|_{I(v)}$ for the restriction of f to the set $I(v)$ of edges incident to v in G. For each vertex $v \in V(G)$, introduce a mode $X^{(v)} = \{x_g^{(v)} \mid g\colon I(v) \to [n]\}$.

▶ **Definition 9** (Graph tensor). *The* graph tensor $T_{G,n}$ *of the graph* G *with* length parameter n *has the modes* $(X^{(v)} : v \in V(G))$ *and is defined by the polynomial identity*

$$T_{G,n} = \sum_{f\colon E(G)\to[n]} \prod_{v\in V} x^{(v)}_{f|_{I(v)}}. \tag{4}$$

Note that $T_{G,n}$ *has* $|V(G)|$ *modes and the* length *of the mode* $X^{(v)}$ *for* $v \in V(G)$ *is* $|X^{(v)}| = n^{|I(v)|}$. *Graph tensors are concise.*

▶ Remark 10 (The Holant view). As laid out in the introduction, graph tensors and Holant problems are intimately connected. Namely, the tensor $T_{G,n}$ essentially captures all Holant problems definable on the fixed graph G by choosing concrete signatures $s_v : [n]^{I(v)} \to \mathbb{C}$ for the vertices $v \in V(G)$ and substituting $x_a^{(v)} \leftarrow s_v(a)$ for all $a \in [n]^{I(v)}$.

▶ **Example 11** (The k-matching tensor). Let us write H_k for the *k-matching graph*, i.e., the 1-regular graph consisting of vertices $u_1, \ldots, u_k$ and $v_1, \ldots, v_k$ and edges u_iv_i for all $i \in [k]$. For $n \in \mathbb{N}$, the *k-matching tensor* $T_{H_k,n}$ is known as the *product of inner products* in algebraic complexity theory. For $k \geq 2$, $T_{H_k,n}$ provides an example of the separation of algebraic complexity and tensor rank for tensors with at least four modes. Namely, we observe

$$T_{H_k,n} \equiv \sum_{f\colon [k]\to[n]} \prod_{i\in[k]} x^{(u_i)}_{i\mapsto f(i)} x^{(v_i)}_{i\mapsto f(i)} = \prod_{i\in[k]} \sum_{j\in[n]} x^{(u_i)}_{i\mapsto j} x^{(v_i)}_{i\mapsto j}\,,$$

holds as a polynomial identity, where the last formula shows $\mathrm{C}(T_{H_k,n}) \leq 2kn$, yet we have $\mathrm{R}(T_{H_k,n}) = n^k$. The first formula above gives $\mathrm{R}(T_{H_k,n}) \leq n^k$. The lower bound $\mathrm{R}(T_{H_k,n}) \geq n^k$ holds since tensor rank is lower-bounded by the matrix rank of any matrix flattening of the tensor. For instance, the $n^k \times n^k$ matrix flattening of $T_{H_k,n}$ defined by the modes $u_1, \ldots, u_k$ and $v_1, \ldots, v_k$ is a permutation of the $n^k \times n^k$ identity matrix.

Several useful connections between algebraic properties of $T_{G,n}$ and graph-theoretic properties of G can be shown. Crucially for us, the Kronecker product of graph tensors of graphs G and H is the graph tensor of the *sum graph* $G + H$. (Recall that the sum may introduce multiedges.) The elementary proof is given in the full version.

▶ **Lemma 12** (Product of graph tensors corresponds to graph sum). *Let* G *and* H *be graphs and let* $n \in \mathbb{N}$. *Then,* $T_{G,n} \otimes T_{H,n} \equiv T_{G+H,n}$.

▶ Remark 13 (Single-edge decomposition). Christandl and Zuiddam [15] used a special case of Lemma 12 to *define* $T_{G,n}$. Specifically, let $G = (V, E)$ with $E = \{e_1, \ldots, e_m\}$, and let $F_i = (V, \{e_i\})$ for $i \in [m]$ be the graph on vertex set V that contains only e_i as edge. Then $G = F_1 + \ldots + F_m$ and

$$T_{G,n} \equiv T_{F_1,n} \otimes \ldots \otimes T_{F_m,n}. \tag{5}$$

Lemma 12, which was not stated in [15], then follows by commutativity of $\otimes$ up to relabeling modes. The definition as a sum of single-edge graphs is more elegant, but the monomial expansion of $T_{G,n}$ in (4) directly connects to algebraic complexity, as witnessed in Example 11.

▶ Remark 14 (Length rule and sum rule). With the same proof as in Lemma 12, for $n_1, n_2 \in \mathbb{N}$ we have the length rule $T_{G,n_1} \otimes T_{G,n_2} \equiv T_{G,n_1 \cdot n_2}$. Indeed, the set of pairs of mappings (f_1, f_2) with $f_i : E(G) \to [n_i]$ is in bijective correspondence with the set of mappings $E(G) \to [n_1 \cdot n_2]$. Consequently, writing $2 \cdot G = G + G$ for any graph G, and more generally $k \cdot G = G + \cdots + G$ for the k-fold sum for $k > 2$, we have the sum rule $T_{k \cdot G,n} \equiv T_{G,n^k}$.

The following lemma is similarly elementary.

▶ **Lemma 15** (Projection under topological subgraphs). *Let G and H be graphs and let $n \in \mathbb{N}$. If H is isomorphic to a topological subgraph of G, then $T_{H,n} \leq_{\mathrm{s}} T_{G,n}$.*

Moreover, we will use *contractions* of graphs: Given a set $U \subseteq V(G)$, the graph G/U is obtained by replacing U by a single vertex w incident with all edges *leaving* U; this process can create multiedges. Contractions are efficient if they don't involve too many edges:

▶ **Lemma 16** (Complexity of contractions). *For every G and $n \in \mathbb{N}$ and $U \subseteq V(G)$, it holds that $\mathrm{C}(T_{G,n}) \leq \mathrm{C}(T_{G/U,n}) + |U| \cdot n^{a(U)}$, where $a(U) \coloneqq \sum_{u \in U} \deg_G(u) - |E(G[U])|$ is the number of edges incident with vertices in U.*

A proof is provided in the full version. Contractions for graph tensors have been extensively studied for probabilistic inference in factor graphs and Bayesian networks [2, 21, 31, 34, 38, 41, 42, 43, 52]. Even a contraction-based model of computation was studied [1].

3.2 Asymptotic Rank

We now prove the main result of this section, Theorem 1. In line with Christandl, Vrana and Zuiddam [14], define the *exponent* of a graph G as

$$\omega(G) \coloneqq \inf \{\beta \mid \mathrm{R}(T_{G,n}) = O(n^\beta)\}.$$

A related quantity is the *exponent per edge* of G, defined as

$$\tau(G) \coloneqq \omega(G)/|E(G)|.$$

The following useful property can be shown in a standard manner, see the full version.

▶ **Lemma 17** (Sum rule for exponents). *Let G be a graph and let $k \in \mathbb{N}$. Then,*

$$\omega(k \cdot G) = k \cdot \omega(G). \tag{6}$$

To motivate the use of graph tensors in our result, we recall Strassen's upper bound on the maximal asymptotic rank of bilinear maps (i.e., 3-mode tensors) together with a high-level intuition of its proof. This property is often referred to as asymptotic submultiplicativity.

▶ **Theorem 18** (Strassen [45]; Upper bound on asymptotic rank for 3-tensors). *Let $T \in (\mathbb{F}^n)^{\otimes 3}$ be an arbitrary 3-mode tensor. Then, $\underset{\sim}{\mathrm{R}}(T) \leq n^{2\omega/3}$.*

Strassen's proof of Theorem 18 works with the tensor corresponding to the universal bilinear map $U_3 : \mathbb{F}^n \times \mathbb{F}^n \to \mathbb{F}^n \otimes \mathbb{F}^n,\ (u, v) \mapsto u \otimes v$, which in our notation can be written as

$$U_3 = \sum_{i,j \in [n]} x_i y_j z_{ij}.$$

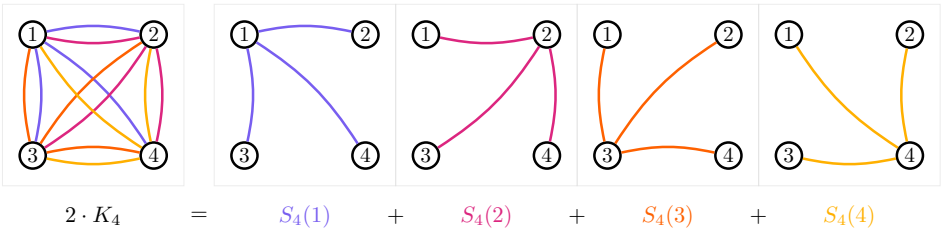


Figure 1 Decomposition of $2 \cdot K_d$ into d stars, (9), for $d = 4$.

In the language of graph tensors, U_3 can be represented as $U_3 \equiv T_{P_2,n}$, where P_2 is the path graph with two edges. This generalizes to $d > 3$ modes as follows. The universal $(d-1)$-linear map $U_d : \mathbb{F}^n \times \cdots \times \mathbb{F}^n \to (\mathbb{F}^n)^{\otimes d-1}$ is the d-mode tensor

$$U_d = \sum_{i_1,\ldots,i_d \in [n]} x^{(1)}_{i_1} \cdots x^{(d-1)}_{i_{d-1}} \cdot x^{(d)}_{i_1,\ldots,i_{d-1}} .$$

We observe that $U_d \equiv T_{S_d,n}$, where S_d is the star graph with $d-1$ edges, and $S_3 \equiv P_2$ in particular. This enables us to consider universality under Kronecker powers combinatorially via graph sums and graph decompositions (cf. Lemma 12) and thus arrive at our main theorem; all our other upper bounds also follow from this observation. To this end, let $d \geq 3$ and write K_d for the complete graph on d vertices. We recall that $\tau(K_d) = \omega(K_d)/\binom{d}{2}$, and write $\tau(d) := \tau(K_d)$ for brevity in what follows.

▶ Remark 19 (The exponents $\tau(d)$ for $d \geq 3$). Recall that we write ω for the exponent of square matrix multiplication. Since $T_{K_3,n}$ is equivalent to the tensor of $n \times n$ matrix multiplication, we have $\tau(3) = \omega/3$. Moreover, $\tau(d) \leq \tau(e)$ whenever $d \geq e$ [14, Proposition 1.31].

The following lemma generalizes Strassen's result through graph tensors.

▶ **Lemma 20** (Generalization of Strassen's upper bound to d modes). *Let $T \in (\mathbb{F}^n)^{\otimes d}$ be an arbitrary d-mode tensor with $d \geq 3$. Then, $\underset{\sim}{\mathrm{R}}(T) \leq n^{(d-1)\tau(d)}$.*

Proof. For $u \in [d]$, let us write $S_d(u)$ for the d-vertex star graph with vertex set $[d]$ such that u is the unique *center* vertex of degree $d-1$. An arbitrary d-mode tensor $T \in (\mathbb{F}^n)^{\otimes d}$ admits representation in coordinates as the polynomial

$$T = \sum_{i_1,\ldots,i_d \in [n]} T_{i_1,\ldots,i_d} x^{(1)}_{i_1} \cdots x^{(d)}_{i_d} = \sum_{i_1,\ldots,i_{d-1} \in [n]} x^{(1)}_{i_1} \cdots x^{(d-1)}_{i_{d-1}} \left(\sum_{i_d \in [n]} T_{i_1,\ldots,i_d} x^{(d)}_{i_d} \right) . \quad (7)$$

Explicitly, the linear substitution

$$x_{i_1,\ldots,i_{d-1}} \mapsto \sum_{i_d \in [n]} T_{i_1,\ldots,i_d} x^{(d)}_{i_d}$$

into the d-tensor $T_{S_d(d),n}$ shows by (7) and symmetry that

$$T \leq T_{S_d(u),n} \quad \text{for all } u \in [d] . \quad (8)$$

We now observe the graph-sum identity (see Figure 1)

$$S_d(1) + \cdots + S_d(d) \equiv 2 \cdot K_d , \quad (9)$$

where we recall that K_d is the complete graph on d vertices and $2 \cdot K_d$ is the graph obtained from K_d by taking two copies of each edge. From (8), Lemma 12, (9), and Remark 14 thus

$$T^{\otimes d} \leq T_{2\cdot K_d,n} \equiv T_{K_d,n^2}. \tag{10}$$

By definition of $\tau(d)$ and the properties of $\underset{\sim}{\mathrm{R}}$, it follows from (10) that

$$\underset{\sim}{\mathrm{R}}(T) \leq \underset{\sim}{\mathrm{R}}(T_{K_d,n})^{2/d} \leq n^{2\binom{d}{2}\tau(d)/d} = n^{(d-1)\tau(d)}. \qquad \blacktriangleleft$$

For $d = 3$, Lemma 20 replicates Strassen's upper bound from Theorem 18, and generalizes it for $d \geq 4$. Our main result of this section is the following upper bound for $d \geq 4$ that combines Lemma 20 with an improved upper bound for $\tau(d)$.

▶ **Theorem 1** (Upper bound on asymptotic rank)**.** *For every d-mode tensor with $d \geq 4$, the asymptotic rank exponent of T is at most* $0.772318(d-1)$.

Proof. Christandl, Vrana, and Zuiddam [14] showed that $\underset{\sim}{\mathrm{R}}(T_{K_d}) \leq n^{\binom{d}{2}\log_7(9/2)}$ for $d \geq 4$, which implies $\tau(d) < 0.7729$. By a careful analysis and modification of their argument, we obtain the sharper bound $\tau(d) < 0.772318$; see the appendix of the full version. The theorem then follows by Lemma 20. $\blacktriangleleft$

▶ Remark 21 (Bounds on $\tau(d)$ for $d \geq 4$). To obtain the constant 0.772318 in Theorem 1, an upper bound for $\tau(4)$ suffices since $\tau(d) \leq \tau(4)$ for $d \geq 4$. It is natural to ask whether better upper bounds can be obtained by finding better upper bounds on $\tau(d)$ for $d \geq 4$. The best-known lower bounds on $\tau(d)$ allow for a scenario in which $\tau(d) \leq 1/2 + o(1)$, implying $\underset{\sim}{\mathrm{R}}(T) \leq n^{(d-1)/2+o(d)}$ would follow for d-mode tensors. On the other hand, the current best-known upper bound on ω yields an exponent greater than the exponent $(d-1)\cdot 0.772318$ in Theorem 1, while $\omega = 2$ would imply $\tau(3) = \tau(4) = 2/3$. It is an open problem whether there is a d such that $\tau(d) < 2/3$. Recalling from Example 11 that the k-matching tensor $T_{H_k,n}$ has $d = 2k$ modes and its rank $\mathrm{R}(T_{H_k,n}) = n^k$ equals a matrix-flattening-rank lower bound, this lower bound applies also to asymptotic rank and thus $\underset{\sim}{\mathrm{R}}(T_{H_k,n}) = n^k$, implying $\tau(d) \geq 1/2$ by Theorem 1 for all $d \geq 4$.

4 Asymptotic Circuit Complexity

This section proceeds to study tensors with $d \geq 4$ modes from the standpoint of the asymptotic circuit complexity of a tensor viewed as a set-multilinear polynomial. Recalling Example 11 and our discussion in the introduction, for $d \geq 4$ the asymptotic circuit complexity and the asymptotic tensor rank are no longer tightly connected as is the case for $d = 3$. Yet tensor families of interest from the standpoint of algorithms and complexity (e.g. matrix permanent, hypercliques, general convolutions, iterated matrix multiplication, ...) can be captured as Kronecker powers of individual base tensors with $d \geq 4$ modes, making the asymptotic circuit complexity of these constant-size base tensors worth studying. Graph tensors provide a convenient tool for this study.

4.1 Complexity of Kronecker Powers

We ask whether tensors $U, T \in (\mathbb{F}^n)^{\otimes d}$ of low circuit complexity also have Kronecker products $U \otimes T$ of low circuit complexity. Recalling that tensor rank is submultiplicative for tensors of all orders, a natural first hope promoted by the $d = 3$ connection of circuit complexity to tensor rank would be to establish the *submultiplicativity* property $\mathrm{C}(T \otimes U) \leq \mathrm{C}(T) \cdot \mathrm{C}(U)$ of the complexity measure C for all orders $d \in \mathbb{N}$. However, already at $d = 4$ submultiplicativity would have a breakthrough consequence.

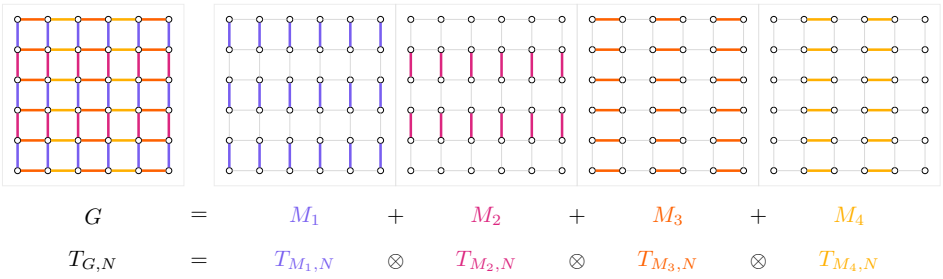


Figure 2 The grid G is the sum of four matchings, so Lemma 12 shows that its graph tensor $T_{G,N}$ is the Kronecker product of four graph tensors of matchings. But since grids are hard by Lemma 25 and matchings are easy by Example 11, we do not expect complexity to be submultiplicative.

The proofs in this section rely on the graph tensors $T_{M_k,n}$ of matchings M_k, for $k, n \in \mathbb{N}$, which are explicit 0-1 tensors. Recall their definition in Example 11, where it is also shown that the circuit complexity of $T_{M_k,n}$ is $O(kn)$. In a nutshell, if circuit complexity were submultiplicative, then $T_{G,n}$ would have low complexity for every graph G that can be decomposed into few matchings, which would contradict common beliefs. We make this statement precise in the remainder of the section.

▶ **Theorem 22** (Fast matrix multiplication under submultiplicativity). *Let M, M' be the matching graphs with $V(M) = V(M') = \{v_1, \dots, v_4\}$ and $E(M) = \{v_1v_2, v_3v_4\}$ and $E(M') = \{v_1v_3, v_2v_4\}$. If*

$$\mathrm{C}(T_{M,n} \otimes T_{M',n}) \leq O(\mathrm{C}(T_{M,n}) \cdot \mathrm{C}(T_{M',n})), \tag{11}$$

then $\omega = 2$.

Proof. The cycle graph C_4 is the sum of the two matchings M and M'. By virtue of Lemma 12 and Example 11, Equation (11) implies $\mathrm{C}(T_{C_4,n}) \leq O(n^2)$. Moreover, $T_{C_3,n}$ is a projection of $T_{C_4,n} \equiv \sum_{i,j,k,\ell} x_{ij}y_{jk}z_{k\ell}w_{\ell i}$ via $z_{k\ell} \mapsto \delta_{k\ell}$, see Lemma 15. Hence, $\mathrm{C}(T_{C_3,n}) \leq O(n^2)$ and $\omega = \omega(C_3) = 2$. ◀

The remainder of this subsection strengthens the case for the absence of submultiplicativity, illustrating the serendipity of graph tensors as a tool when working with more modes.

We proceed to reduce matrix permanent tensors to graph tensors of square grid graphs G. Since grids have large treewidth, related results [22, 18] suggest their graph tensors have large algebraic complexity. Indeed, we show in Lemma 25 that the $d \times d$ matrix permanent is a projection of the graph tensors of $(d+2) \times (d+2)$ grids, even for mode dimension 2. On the other hand, as shown in Figure 2, every grid G can be written as the sum

$$G \equiv M_1 + M_2 + M_3 + M_4 \tag{12}$$

of edge-disjoint matching graphs M_1, M_2, M_3, M_4. Hence, Lemma 12 implies for $N \in \mathbb{N}$ that

$$T_{G,N} \equiv T_{M_1,N} \otimes T_{M_2,N} \otimes T_{M_3,N} \otimes T_{M_4,N}. \tag{13}$$

Graph tensors of matchings have low complexity by Example 11. So, if C were submultiplicative, then $T_{G,N}$ would also have low complexity. More generally, this holds for every graph G of small edge-chromatic number, i.e., whenever G can be partitioned into few matchings:

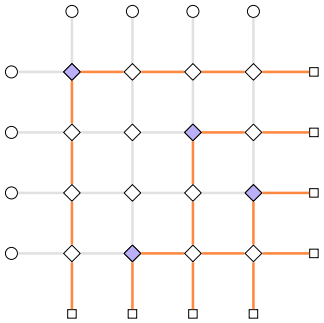


Figure 3 The reduction from the permanent to the graph tensors of grids in Lemma 25. The gray and orange edges are assigned 0 and 1, respectively. Each horizontal and each vertical path flips from 0 to 1 exactly once; whenever this happens at some vertex $v_{i,j}$ (a *flip vertex*, shown purple), we ensure that both the i-th horizontal and the j-th vertical path flip at $v_{i,j}$ (so purple vertices always lie on orange corners). Each flip vertex $v_{i,j}$ contributes $x_{i,j}$ to the weight of the assignment, others 1. The weight of the assignment shown here is $x_{11}x_{23}x_{34}x_{42} = \prod_i x_{i,\pi(i)}$ for $\pi = 1342$.

▶ **Lemma 23** (Graphs with small edge-chromatic number under submultiplicativity). *Let G be a graph such that $E(G)$ can be partitioned into t matchings $M_1, \ldots, M_t$ with maximum edge-multiplicity $b \in \mathbb{N}$. Let $N \in \mathbb{N}$. If*

$$\mathrm{C}(T_{M_1,N} \otimes \ldots \otimes T_{M_t,N}) \leq O(\mathrm{C}(T_{M_1,N}) \cdot \ldots \cdot \mathrm{C}(T_{M_t,N})), \tag{14}$$

then $\mathrm{C}(T_{G,N}) \leq O(2^t|V(G)|^t \cdot N^{tb})$.

Proof. Example 11 and Remark 14 give $\mathrm{C}(T_{M_i,N}) \leq 2|V(G)| \cdot N^b$. The assumed submultiplicativity of C, together with Lemma 12 gives $\mathrm{C}(T_{G,N}) \leq \prod_{i=1}^{t} \mathrm{C}(T_{M_i,N}) \leq 2^t|V(G)|^t \cdot N^{tb}$, thus proving the claim. ◀

Using (13), we obtain the following corollary for grids.

▶ **Corollary 24.** *Let G denote the $n \times m$ grid, for $n, m \in \mathbb{N}$, with maximum edge-multiplicity $b \in \mathbb{N}$. If Equation (14) holds for $t = 4$ matchings, each on $|V(M_1)| = \ldots |V(M_4)| = nm$ vertices, then $\mathrm{C}(T_{G,N}) \leq O(n^4m^4 \cdot N^{4b})$.*

Our results are obtained by invoking this corollary for various choices of grids G and dimensions N. We begin under the comparatively weak assumption $\mathsf{VP} \neq \mathsf{VNP}$ [47]. Towards this end, we first show how the permanent reduces to grid graph tensors.

▶ **Lemma 25** (The permanent tensor reduces to a grid-graph tensor). *The $n \times n$ permanent tensor per_n is a projection of $T_{G,2}$ for the simple $(n+2) \times (n+2)$ grid graph G.*

Proof sketch. The reduction is illustrated in Figure 3: G is obtained by attaching pendant vertices of degree 1 to the boundary of an $n \times n$ grid; the pendant vertices are needed to ensure boundary conditions. Each edge of the resulting graph carries a Boolean state, and we substitute values for the indeterminates of $T_{G,2}$ such that the resulting polynomial counts assignments a to the edges of the grid with the following properties:

1. We ensure that each horizontal and each vertical path P starts with a 0-edge and the edge states flip from 0 to 1 at exactly one vertex; we call this the *flip vertex* of P under a.
2. For all $i, j \in [n]$, we ensure that a vertex $v_{i,j}$ is the flip vertex of the i-th horizontal path if and only if the same vertex $v_{i,j}$ also is the flip vertex of the j-th vertical path.
3. Finally, we ensure that a flip vertex contributes a factor $x_{i,j}$ to the weight of the assignment, while non-flip vertices do not contribute (i.e., they contribute 1).

These conditions can be realized by a 4-leg graph tensor (signature): if we denote by t, r, b, ℓ the top, right, bottom and left input of the signature, then its weight is the corresponding entry in the matrix if $t = \ell = 0, r = b = 1$, is 1 if $t = b, r = \ell$, and 0 otherwise. If these conditions hold, the flip vertices under an assignment a induce an $n \times n$ permutation matrix P_π, and the weight of a is the product of the indeterminates $x_{i,\pi(i)}$ selected by P_π. Summing over valid assignments a gives per_n. ◀

▶ **Theorem 2** (Submultiplicativity of C implies VP = VNP). *There are explicit tensors $T_1, T_2, \ldots$ and $U_1, U_2, \ldots$, where T_d, U_d for $d \in \mathbb{N}$ are d-mode 0-1-tensors, such that the following holds: If $\mathrm{C}(T_d \otimes U_d) \leq \mathrm{poly}(d, \mathrm{C}(T_d), \mathrm{C}(U_d))$ for all $d \in \mathbb{N}$, then $\mathsf{VP} = \mathsf{VNP}$.*

Proof. Let $d \in \mathbb{N}$ be a perfect square, otherwise use padding. Let G be the $\sqrt{d} \times \sqrt{d}$ grid and $M_1, \ldots, M_4$ be its decomposition into matchings. Then

$$T_{G,2} \equiv \underbrace{T_{M_1,2} \otimes T_{M_2,2}}_{=:T_d} \otimes \underbrace{T_{M_3,2} \otimes T_{M_4,2}}_{=:U_d}.$$

Note that T_d and U_d are graph tensors, each corresponding to a disjoint union of paths.

Under the assumption of the theorem, Corollary 24 and Lemma 25 would imply $\mathrm{C}(T_{G,2}) \leq \mathrm{poly}(d)$, which would imply $\mathsf{VP} = \mathsf{VNP}$ due to the VNP-hardness of the permanent. ◀

▶ Remark 26. As pointed out to us by an anonymous referee, a weaker version of the preceding Theorem 2 can be obtained immediately by considering the so-called Hadamard product $\odot$ of tensors. This is simply the point-wise product of monomial coefficients and can be obtained as a simple substitution of the Kronecker product. In particular, $\mathrm{per}_n = \det_n \odot \det_n$. We note that the appearing tensors have d modes of dimension d, whereas Theorem 2 applies already to tensors with modes of constant dimension.

If we assume that permanents require exponential-size circuits, then we can obtain stronger bounds. Towards this, we first contract large grids into smaller grids with larger edge-multiplicities. (Larger edge-multiplicities correspond to larger mode dimensions: For every graph H and $b, k, n \in \mathbb{N}$ with k dividing n, and writing $B = b^{n/k}$, we have $T_{H,B} \equiv T_{n/k \cdot H, b}$.) Namely, first partition the $n \times n$ grid into $k \times k$ pieces of size $n/k \times n/k$ each, then contract each piece to a single vertex by Lemma 16. The complexity bound in Lemma 16 requires a little care in the choice of contraction sequence to obtain the desired lower bounds later.

▶ **Lemma 27** (Grid graph tensor contraction by subgrids). *Let k divide n and let H and G be the $k \times k$ and $n \times n$ grid graphs, respectively. Let $b \in \mathbb{N}$ and $B = b^{n/k}$. Then, $\mathrm{C}(T_{G,b}) \leq (2b^4 + b)n^2 \cdot b^{3(n/k)} + \mathrm{C}(T_{H,B})$.*

Proof. Partition the vertex set of G into $k \times k$ square blocks $A_{i,j}$, each of size $n/k \times n/k$. For each $i, j \in [k]$, we then perform the following:

1. For $t = 1, \ldots, n/k$, contract the vertices in row t of block $A_{i,j}$ by Lemma 16. Each contraction involves $3n/k + 1$ edges and thus incurs an additive cost term of $bn/k \cdot b^{3(n/k)}$ in Lemma 16. Block $A_{i,j}$ now consists of a vertical path of length n/k, with edge-multiplicity n/k between consecutive vertices, and with two additional edges per vertex. The total cost of the contractions in this step is $b(n/k)^2 \cdot b^{3(n/k)}$.
2. For $t = 1, \ldots, n/k - 1$, we contract vertices t and $t + 1$ on the path remaining in block $A_{i,j}$. This involves $3n/k + 4$ edges per contraction and incurs a cost term of $2b^4 \cdot b^{3(n/k)}$ in Lemma 16. Block $A_{i,j}$ now consists of a single vertex incident with all edges leaving $A_{i,j}$. The total cost of the contractions in this step is $2b^4(n/k) \cdot b^{3(n/k)}$.

In total over the k^2 cells, we incur cost at most $(2b^4+b)k^2(n/k)^2 \cdot b^{3(n/k)} = (2b^4+b)n^2 \cdot b^{3(n/k)}$ for all contractions and obtain the $k \times k$ grid with edge-multiplicity n/k from G. ◀

A lower bound now follows easily.

▶ **Theorem 3** (Submultiplicativity of C implies faster permanents). *For all $0 < c < 1$, there are explicit $\lceil 4/c \rceil^2$-mode 0-1-tensors $T_1, T_2, \ldots$ and $U_1, U_2, \ldots$ such that the following holds: If $\mathrm{C}(T_n \otimes U_n) \leq O(\mathrm{C}(T_n) \cdot \mathrm{C}(U_n))$, then permanents have circuits of size $2^{cn} n^{O(1)}$.*

Proof. Choose $k = \lceil 4/c \rceil$ so $4/k \leq c$, and let H be the $k \times k$ grid on k^2 vertices. By Remark 14 and Lemma 27 with $b = 2$, if the k^2-mode tensor $T_{n/k \cdot H,2}$ admits a circuit of size s, then the n^2-mode tensor $T_{G,2}$ for the $n \times n$ grid G admits a circuit of size $s' \leq n^{O(1)} \cdot 8^{n/k} + s$, and Lemma 25 gives a circuit of size $s' \cdot n^{O(1)}$ for per_n.

If submultiplicativity indeed held, then Corollary 24 would give a circuit of size $s \leq n^{O(1)} \cdot 2^{4n/k}$ for $T_{n/k \cdot H,2}$, so we obtain a circuit of size $s' \leq n^{O(1)} \cdot 2^{4n/k} \leq 2^{cn} n^{O(1)}$ for $T_{G,2}$ and the permanent. ◀

The above strategy for conditional arguments can be refined in different ways to rule out submultiplicativity on fewer modes, e.g., by choosing graph tensors other than grids. For instance, we show:

▶ **Theorem 4** (Submultiplicativity of C implies faster hypercliques). *For every even $d \geq 8$, there are explicit d-mode 0-1-tensors $T_1, T_2, \ldots$ and $U_1, U_2, \ldots$ such that: If $\mathrm{C}(T_n \otimes U_n) \leq O(\mathrm{C}(T_n) \cdot \mathrm{C}(U_n))$, then the non-uniform $(\frac{d}{2} - 1, \frac{d}{2})$-hyperclique conjecture fails.*

4.2 Upper Bounds via Line Graph Treewidth

The *line-treewidth* $\mathrm{ltw}(G)$ is the treewidth $\mathrm{tw}(L(G))$ of the *line graph* $L(G)$ of G, where the vertex set of $L(G)$ is $E(G)$ and two vertices corresponding to edges e, e' are adjacent in $L(G)$ if they share a vertex in G. Of special interest is the treewidth of the line graph of the complete graph, which has been determined exactly.

▶ **Theorem 28** (Line-treewidth of a complete graph [25, Theorem 1]). *For $d \geq 1$, we have*

$$\mathrm{ltw}(K_d) = \begin{cases} \left(\frac{d-1}{2}\right)^2 + d - 2, & \text{if } d \text{ is odd,} \\ \left(\frac{d-2}{2}\right)\left(\frac{d}{2}\right) + d - 2, & \text{if } d \text{ is even.} \end{cases}$$

Similar to standard dynamic programming arguments on tree-decompositions in, e.g., [51, Section 5.1], [20, Lemma 5.23], and [23, Example 3.4], we can show:

▶ **Lemma 29.** *For every H and $n \in \mathbb{N}$, we have $\mathrm{C}(T_{H,n}) \leq |V(H)| \cdot n^{\mathrm{ltw}(H)+1}$.*

As an immediate consequence the above lemma proves $\underset{\sim}{\mathrm{C}}(T_{H,n}) \leq n^{\mathrm{ltw}(H)+1}$ for every graph H. For general tensors, recall that we write $\eta(T)$ for the exponent of the asymptotic circuit complexity $\underset{\sim}{\mathrm{C}}(T) = n^{\eta(T)}$ of a tensor $T \in (\mathbb{F}^n)^{\otimes d}$. We then have the following implication, which gives a proof of Theorem 6 in a stronger version:

▶ **Theorem 30** (Upper bound on the asymptotic circuit complexity of a d-tensor). *Let $T \in (\mathbb{F}^n)^{\otimes d}$ be a d-mode tensor of dimension n. Then, we have*

$$\eta(T) \leq \frac{2}{d}(\mathrm{ltw}(K_d) + 1) = d/2 + 1 - \frac{7 + (-1)^d}{4d}.$$

The proofs of Theorem 30 and Lemma 29 are given in the full version.

▶ Remark 31. Our proof of Theorem 30 in fact constructs a monotone circuit, and so does the circuit constructed in Lemma 29, as it implements a standard dynamic programming approach through tensor contractions.

▶ Remark 32. It is natural to ask whether the exponent from Theorem 30 is asymptotically tight under standard assumptions from complexity theory. To the best of our knowledge, current techniques would only rule out exponents of the form $o(d/\log d)$.

For instance, known results on the complexity of subgraph isomorphism [36, 27, 17] imply $\eta(T_{H,n}) \in \Omega(d/\log d)$ for a sequence of d-vertex graphs H of maximum degree 3. It is also possible to adapt a recent bound by Pratt [40] to a higher number of modes, which would yield a lower bound under the set cover conjecture. Since all of the resulting lower bounds are fairly loose, we refrain from formally stating and proving them here.

5 Restricted Submultiplicativity

The results in Section 4.1 strongly suggest the absence of circuit complexity submultiplicativity for d-mode tensors with $d \geq 4$: For d-mode tensors T and U of low complexity, no useful upper bounds for the circuit complexity $\mathrm{C}(T \otimes U)$ as a function of $\mathrm{C}(T)$ and $\mathrm{C}(U)$ are known. However, we can construct small circuits for $T \otimes U$ from small circuits for T and U, provided that one of the circuits is sufficiently simple.

To this end, we will need the following property of tensor contraction. Let $\ell_1, \ldots, \ell_d \colon \mathbb{F}^m \to \mathbb{F}$ and $h_1, \ldots, h_d \colon \mathbb{F}^n \to \mathbb{F}$ be linear forms. We consider the Kronecker product of the corresponding rank-one tensors $L = \prod_{i=1}^d \ell_i(x^{(i)})$ and $H = \prod_{i=1}^d h_i(y^{(i)})$, where the i-th mode of $L \otimes H$ has coordinates $z^{(i)}_{jk} \coloneqq x^{(i)}_j \otimes y^{(i)}_k$, with $j \in [m], k \in [n]$. For brevity, we write $z^{(i)}$ for the $m \times n$ matrix with entries $z^{(i)}_{jk}$, and identify h_i with its vector of coefficients in $\mathbb{F}^n$. Then, one can check that

$$L \otimes H = \prod_{i=1}^{d} \ell_i(h_i(z^{(i)}_{11}, \ldots, z^{(i)}_{1n}), \ldots, h_i(z^{(i)}_{m1}, \ldots, z^{(i)}_{mn})) = L(z^{(1)} \cdot h_1, \ldots, z^{(d)} \cdot h_d).$$

By linearity and the fact that every tensor U can be decomposed into rank-1 tensors,

$$U \otimes H = U(z^{(1)} \cdot h_1, \ldots, z^{(d)} \cdot h_d) \tag{15}$$

holds for all tensors U with d modes in coordinates $z^{(i)}_{jk}$. We can use this to show:

▶ **Lemma 33.** *Let $T \in (\mathbb{F}^n)^{\otimes d}$ and $U \in (\mathbb{F}^m)^{\otimes d}$ with T of rank at most r, with $T = \sum_{i=1}^r \prod_{j=1}^d \ell_{ij}(x^{(j)})$. For $1 \leq j \leq d$, let $\Lambda^{(j)}$ be the $n \times r$ matrix of linear forms $(\ell_{ij})_i$ appearing in the j-th mode of the rank-decomposition of T, and we recall that $z^{(i)}$ is an $m \times n$ matrix of indeterminates $z^{(i)}_{j,k}$ for $1 \leq i \leq d$. Then,*

$$\mathrm{C}(U \otimes T) \leq r(\mathrm{C}(U) + 1) + \sum_{i=1}^{d} \mathrm{C}(z^{(i)} \cdot \Lambda^{(i)}) + r \cdot m \cdot d.$$

Proof. Distributing (15) linearly across the terms of the assumed rank-r decomposition of T, we see that

$$U \otimes T = \sum_{i=1}^{r} \left(U \otimes \prod_{j=1}^{d} \ell_{ij}(x^{(j)}) \right) = \sum_{i=1}^{r} U(z^{(1)} \cdot \ell_{i,1}, \ldots, z^{(d)} \cdot \ell_{i,d}). \tag{16}$$

On the right-hand side of (16), the j-th mode of the instance of U appearing in the i-th term in the sum obtains as an input the i-th column of $z^{(j)} \cdot \Lambda^{(j)}$. This is a column vector of dimension m. To implement (16) as a circuit, place a fan-in r $+$-gate at the top, contributing

r wires. The i-th input of this topmost $+$-gate is a copy of the circuit for U, contributing $r \cdot \mathrm{C}(U)$ to the overall size. In addition, we need to compute the inputs to the individual r copies of U (which becomes crucial in the asymptotic regime.) As mentioned, for this it suffices to compute the d matrix products $z^{(1)} \cdot \Lambda^{(1)}, \ldots, z^{(d)} \cdot \Lambda^{(d)}$. Finally, we wire each of the entries of $z^{(j)} \cdot \Lambda^{(j)}$ to the respective input of U, amounting to a total of $r \cdot m \cdot d$ wires. ◀

For our applications, we need a variant of Lemma 33 for higher Kronecker powers of T and U. It can be proven by applying Lemma 33 and combining it with Yates's algorithm.

▶ **Theorem 5** (Mixed asymptotic rank and circuit complexity)**.** *Given d-mode tensors T and U, where T has asymptotic rank exponent $\psi \geq 1$ and U has asymptotic circuit complexity exponent $\eta \geq 1$, the asymptotic circuit complexity exponent of $T \otimes U$ is at most $\eta + \psi$.*

The preceding Theorem 5 provides a useful tool to study more general graph decompositions to obtain asymptotic circuit complexity upper bounds. This can be done by generalizing the graph tensor framework to fractional graphs, and then performing computer-aided search for fractional decompositions under relaxed constraints. The details of this approach can be found in the full version. As a result, these techniques allow us to obtain Theorem 7.

References

1 Per Austrin, Petteri Kaski, and Kaie Kubjas. Tensor network complexity of multilinear maps. *Theory Comput.*, 18:1–54, 2022. doi:10.4086/TOC.2022.V018A016.

2 Bozhena Bidyuk and Rina Dechter. Cutset sampling for Bayesian networks. *J. Artif. Intell. Res.*, 28:1–48, 2007. doi:10.1613/JAIR.2149.

3 Andreas Björklund, Radu Curticapean, Thore Husfeldt, Petteri Kaski, and Kevin Pratt. Fast deterministic chromatic number under the asymptotic rank conjecture. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 2804–2818. SIAM, 2025. doi:10.1137/1.9781611978322.91.

4 Andreas Björklund and Petteri Kaski. The asymptotic rank conjecture and the set cover conjecture are not both true. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 859–870. ACM, 2024. doi:10.1145/3618260.3649656.

5 Andreas Björklund, Petteri Kaski, Tomohiro Koana, and Jesper Nederlof. Kronecker scaling of tensors with applications to arithmetic circuits and algorithms. *CoRR*, abs/2504.05772, 2025. doi:10.48550/arXiv.2504.05772.

6 Cornelius Brand, Radu Curticapean, Petteri Kaski, Baitian Li, Ian Orzel, Tim Seppelt, and Jiaheng Wang. Beyond bilinear complexity: What works and what breaks with many modes?, 2026. doi:10.48550/arXiv.2602.11975.

7 Cornelius Brand, Radu Curticapean, Baitian Li, and Kevin Pratt. Faster convolutions: Yates and Strassen revisited. In *Proceedings of the 2026 SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 328–339, 2026. doi:10.1137/1.9781611978964.25.

8 Karl Bringmann and Jasper Slusallek. Current algorithms for detecting subgraphs of bounded treewidth are probably optimal. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, Glasgow, Scotland (Virtual Conference), July 12-16, 2021*, volume 198 of *LIPIcs*, pages 40:1–40:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.40.

9 Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1997. With the collaboration of Thomas Lickteig. doi:10.1007/978-3-662-03338-8.

10 Jin-Yi Cai and Xi Chen. *Complexity Dichotomies for Counting Problems: Volume 1, Boolean Domain*. Cambridge University Press, 2017. doi:10.1017/9781107477063.

11 Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. *SIAM J. Comput.*, 45(5):1671–1728, 2016. doi:10.1137/15M1049798.

12 Jin-yi Cai and Pinyan Lu. Holographic algorithms: From art to science. *J. Comput. Syst. Sci.*, 77(1):41–61, 2011. doi:10.1016/J.JCSS.2010.06.005.

13 Jin-Yi Cai and Ben Young. Vanishing signatures, orbit closure, and the converse of the holant theorem. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference, ITCS 2026, Bocconi University, Milan, Italy, January 27-30, 2026*, volume 362 of *LIPIcs*, pages 32:1–32:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi:10.4230/LIPIcs.ITCS.2026.32.

14 Matthias Christandl, Péter Vrana, and Jeroen Zuiddam. Asymptotic tensor rank of graph tensors: beyond matrix multiplication. *Comput. Complex.*, 28(1):57–111, 2019. doi:10.1007/S00037-018-0172-8.

15 Matthias Christandl and Jeroen Zuiddam. Tensor surgery and tensor rank. *Comput. Complex.*, 28(1):27–56, 2019. doi:10.1007/S00037-018-0164-8.

16 J. Ignacio Cirac, David Pérez-García, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Rev. Mod. Phys.*, 93:045003, December 2021. doi:10.1103/RevModPhys.93.045003.

17 Radu Curticapean, Simon Döring, Daniel Neuen, and Jiaheng Wang. Can you link up with treewidth? In Olaf Beyersdorff, Michal Pilipczuk, Elaine Pimentel, and Kim Thang Nguyen, editors, *42nd International Symposium on Theoretical Aspects of Computer Science, STACS 2025, Jena, Germany, March 4-7, 2025*, volume 327 of *LIPIcs*, pages 28:1–28:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.STACS.2025.28.

18 Radu Curticapean and Dániel Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 130–139. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.22.

19 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

20 Anuj Dawar, Benedikt Pago, and Tim Seppelt. Symmetric algebraic circuits and homomorphism polynomials, 2025. arXiv:2502.06740v3.

21 Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artif. Intell.*, 113(1-2):41–85, 1999. doi:10.1016/S0004-3702(99)00059-4.

22 Praeteek Dwivedi, Benedikt Pago, and Tim Seppelt. Lower bounds in algebraic complexity via symmetry and homomorphism polynomials, 2026. doi:10.48550/arXiv.2601.09343.

23 Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM J. Comput.*, 33(4):892–922, 2004. doi:10.1137/S0097539703427203.

24 Egor Gorbachev and Marvin Künnemann. Combinatorial designs meet hypercliques: Higher lower bounds for Klee's measure problem and related problems in dimensions $d \geq 4$. In Erin W. Chambers and Joachim Gudmundsson, editors, *39th International Symposium on Computational Geometry, SoCG 2023, Dallas, Texas, USA, June 12-15, 2023*, volume 258 of *LIPIcs*, pages 36:1–36:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.SOCG.2023.36.

25 Daniel J. Harvey and David R. Wood. Treewidth of the line graph of a complete graph. *J. Graph Theory*, 79(1):48–54, 2015. doi:10.1002/JGT.21813.

26 William Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and E Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and Technology*, 4(2):024001, January 2019. doi:10.1088/2058-9565/aaea94.

27 Karthik C. S., Dániel Marx, Marcin Pilipczuk, and Uéverton S. Souza. Conditional lower bounds for sparse parameterized 2-CSP: A streamlined proof. In Merav Parter and Seth Pettie, editors, *2024 Symposium on Simplicity in Algorithms, SOSA 2024, Alexandria, VA, USA, January 8-10, 2024*, pages 383–395. SIAM, 2024. doi:10.1137/1.9781611977936.35.

28 Neeraj Kayal. Lecture 9: Lower bounds. Lecture notes, Topics in Complexity Theory, Indian Institute of Science, 2015. Accessed 2026-05-19. URL: https://www.csa.iisc.ac.in/~chandan/courses/arithmetic_circuits/notes/lec9.pdf.

29 Donald E. Knuth. *The Art of Computer Programming. Vol. 2.* Addison-Wesley, Reading, MA, third edition, 1998. Seminumerical Algorithms.

30 Daphne Koller and Nir Friedman. *Probabilistic Graphical Models.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2009. Principles and techniques.

31 Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, 2001. doi:10.1109/18.910572.

32 Marvin Künnemann. A tight (non-combinatorial) conditional lower bound for Klee's measure problem in 3D. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 555–566. IEEE, 2022. doi:10.1109/FOCS54457.2022.00059.

33 J. M. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics.* American Mathematical Society, Providence, RI, 2012. doi:10.1090/gsm/128.

34 S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 50(2):157–194, 1988. doi:10.1111/j.2517-6161.1988.tb01721.x.

35 Andrea Lincoln, Virginia Vassilevska Williams, and R. Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1236–1252. SIAM, 2018. doi:10.1137/1.9781611975031.80.

36 Dániel Marx. Can you beat treewidth? *Theory Comput.*, 6(1):85–112, 2010. doi:10.4086/TOC.2010.V006A005.

37 Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Ann. Physics*, 349:117–158, 2014. doi:10.1016/j.aop.2014.06.013.

38 Judea Pearl. Fusion, propagation, and structuring in belief networks. *Artif. Intell.*, 29(3):241–288, 1986. doi:10.1016/0004-3702(86)90072-X.

39 Roger Penrose and Wolfgang Rindler. *Spinors and Space-Time. Vol. 1.* Cambridge Monographs on Mathematical Physics. Cambridge University Press, Cambridge, 1987. Two-spinor calculus and relativistic fields. doi:10.1017/CBO9780511564048.

40 Kevin Pratt. A stronger connection between the asymptotic rank conjecture and the set cover conjecture. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 871–874. ACM, 2024. doi:10.1145/3618260.3649620.

41 Irina Rish and Rina Dechter. Resolution versus search: Two strategies for SAT. *J. Autom. Reason.*, 24(1/2):225–275, 2000. doi:10.1023/A:1006303512524.

42 Ross D. Shachter, Stig K. Andersen, and Peter Szolovits. Global conditioning for probabilistic inference in belief networks. In Ramón López de Mántaras and David Poole, editors, *UAI '94: Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence, Seattle, Washington, USA, July 29-31, 1994*, pages 514–522. Morgan Kaufmann, 1994. doi:10.1016/B978-1-55860-332-5.50070-5.

43 Glenn Shafer and Prakash P. Shenoy. Probability propagation. *Ann. Math. Artif. Intell.*, 2:327–351, 1990. doi:10.1007/BF01531015.

44 Shuai Shao and Jin-Yi Cai. A dichotomy for real boolean holant problems. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1091–1102. IEEE, 2020. doi:10.1109/FOCS46700.2020.00105.

45 V. Strassen. The asymptotic spectrum of tensors. *J. Reine Angew. Math.*, 384:102–152, 1988. doi:10.1515/crll.1988.384.102.

46 V. Strassen. Algebra and complexity. In *First European Congress of Mathematics, Vol. II (Paris, 1992)*, volume 120 of *Progr. Math.*, pages 429–446. Birkhäuser, Basel, 1994. doi:10.1007/978-3-0348-9112-7_18.

47 Leslie G. Valiant. Completeness classes in algebra. In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.

48 Leslie G. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008. doi:10.1137/070682575.

49 Avi Wigderson and Jeroen Zuiddam. Asymptotic spectra: Theory, applications and extensions. *Bull. Amer. Math. Soc.*, 63(2):199–331, 2026. doi:10.1090/bull/1880.

50 Frank Yates. *The Design and Analysis of Factorial Experiments*. Imperial Bureau of Soil Science. Technical Communication, no. 35. Imperial Bureau of Soil Science, Harpenden, 1937.

51 Yitong Yin and Chihao Zhang. Approximate counting via correlation decay on planar graphs. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 47–66. SIAM, 2013. doi:10.1137/1.9781611973105.4.

52 N. L. Zhang and D. L. Poole. A simple approach to Bayesian network computations. In *Proc. Tenth Canadian Conference on Artificial Intelligence*, pages 171–178, 1994.

Fine-Grained Complexity for Quantum Problems from Size-Preserving Circuit-To-Hamiltonian Constructions

Nai-Hui Chia
Department of Computer Science, Rice University, Houston, TX, USA
Ken Kennedy Institute, Rice University, Houston, TX, USA
Smalley-Curl Institute, Rice University, Houston, TX, USA

Atsuya Hasegawa
Graduate School of Mathematics, Nagoya University, Japan

François Le Gall
Graduate School of Mathematics, Nagoya University, Japan

Yu-Ching Shen
Department of Computer Science, Houston, Rice University, Houston, TX, USA

Abstract

The local Hamiltonian (LH) problem is the canonical QMA-complete problem introduced by Kitaev. In this paper, we show its hardness in a very strong sense: we show that the 3-local Hamiltonian problem on n qubits cannot be solved classically in time $O(2^{(1-\varepsilon)n})$ for any $\varepsilon > 0$ under the Strong Exponential-Time Hypothesis (SETH), and cannot be solved quantumly in time $O(2^{(1-\varepsilon)n/2})$ for any $\varepsilon > 0$ under the Quantum Strong Exponential-Time Hypothesis (QSETH). These lower bounds give evidence that the currently known classical and quantum algorithms for LH cannot be significantly improved.

Furthermore, we are able to demonstrate fine-grained complexity lower bounds for approximating the quantum partition function (QPF) with an arbitrary constant relative error. Approximating QPF with relative error is known to be equivalent to approximately counting the dimension of the solution subspace of QMA problems. We show the SETH and QSETH hardness to estimate QPF with constant relative error. We then provide a quantum algorithm that runs in $O(\sqrt{2^n})$ time for an arbitrary $1/\mathrm{poly}(n)$ relative error, matching our lower bounds and improving the state-of-the-art algorithm by Bravyi, Chowdhury, Gosset, and Wocjan (Nature Physics 2022) in the low-temperature regime.

To prove our fine-grained lower bounds, we introduce the first size-preserving circuit-to-Hamiltonian construction that encodes the computation of a T-time quantum circuit acting on N qubits into a $(d+1)$-local Hamiltonian acting on $N + O(T^{1/d})$ qubits. This improves the standard construction based on the unary clock, which uses $N + O(T)$ qubits.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Fine-grain complexity, SETH, QSETH, Local Hamiltonian problem, Quantum partition problem

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.12

Related Version *arXiv version*: https://arxiv.org/abs/2602.14379

Funding *Nai-Hui Chia*: NSF Award FET-2535028, NSF Award FET-2243659, NSF Career Award FET-2339116, Google Scholar Award, and DOE Quantum Testbed Finder Award DE-SC0024301.
Atsuya Hasegawa: JSPS KAKENHI grant No. 24H00071, 25K24674, 25K2446.
François Le Gall: JSPS KAKENHI grant No. 24H00071, 25K24674, 25K2446, MEXT Q-LEAP grant No. JPMXS0120319794, JST ASPIRE grant No. JPMJAP2302 and JST CREST grant No. JPMJCR24I4.
Yu-Ching Shen: NSF Award FET-2339116 and NSF Award FET-2243659.



© Nai-Hui Chia, Atsuya Hasegawa, François Le Gall, and Yu-Ching Shen;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 12; pp. 12:1–12:35



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Acknowledgements NHC and YCS thank Chunhao Wang and Christopher Ye for valuable discussions. FLG thanks Suguru Tamaki for discussions. The authors thank the anonymous referees for their comments.

1 Introduction

1.1 Background

Local Hamiltonian (LH) problem

Kitaev [34] introduced the k-local Hamiltonian (k-LH) problem, which asks to estimate the ground-state energy of a k-local Hamiltonian

$$H = \sum_{i=1}^{m} H_i \,,$$

acting on n qubits, where $m = \text{poly}(n)$ and $\|H\| \leq \text{poly}(n)$, with constant additive error.[1] Kitaev, at the same time, showed that the problem is QMA-complete, i.e., any efficient quantum verification procedure can be embedded into the local Hamiltonian problem. The embedding procedure is called the circuit-to-Hamiltonian construction, which is a quantum analog of the Cook-Levin theorem [16, 38]. The local Hamiltonian problem is in the interface between physics and computer science, and has been the central problem in quantum complexity theory. Subsequent work showed that, when we reduce the locality to 2, restrict the interaction of local terms to geometrically local, and consider some specific type of Hamiltonians, the problem is still QMA-hard [7, 17, 32, 45, 46].

The k-local Hamiltonian problem can be solved classically in $O^*(2^n)$ time using the power method or its variant the Lanczos method [36, 37], for $k = O(\log n)$.[2] On a quantum computer, we can use Grover search [25] combined with other techniques to achieve running time $O^*\left(2^{n/2}\right)$ [3, 22, 24, 33, 39, 42, 47].[3]

Approximating quantum partition function with relative error

Computing the quantum partition function is an even harder problem. The partition function is a function of the temperature and the Hamiltonian of the system. It is defined by

$$Z := \text{tr}[e^{-\beta H}],$$

where β is the inverse of the temperature and H is the Hamiltonian. The *Quantum Partition Function (QPF)* problem asks us to compute the value of Z for a given Hamiltonian at a specified temperature with relative error.[4] Intuitively, QPF is harder than LH because evaluating the partition function requires information about the *entire* spectrum of the Hamiltonian, whereas solving LH only involves the ground-state energy. In fact, we can

[1] Equivalently, we can define the local Hamiltonian problem as $||H|| \leq 1$ and $1/\text{poly}(n)$ additive error. Our definition is the same as [32].

[2] In this paper the notation $O^*(\cdot)$ suppresses $\text{poly}(n)$ factors.

[3] We note that some previous works showed how to obtain larger ("super-Grover") speedups for certain applications in combinatorial optimization [26, 18] or for computing a coarser approximation of the ground-state energy [11].

[4] It is crucial here to consider relative error estimation. When we consider the problem to estimate the (normalized) QPF with additive error $1/\text{poly}(n)$, it is known to be DQC1-complete [8, 15] where DQC1 is the set of problems that can be solved efficiently with only "one-clean qubit" [35].

show that LH reduces to approximating the QPF, even with a constant relative error, and thus approximating QPF with constant-relative error is QMA-hard.[5] Furthermore, Bravyi et al. [9] show that approximating the QPF up to constant-relative error is computationally equivalent to approximately counting the number of witnesses accepted by a QMA verifier.

For algorithms, Poulin and Wocjan [48] demonstrated a quantum algorithm that approximates the partition function Z of an n-qubit system up to relative error δ in $O^*\left(\frac{1}{\delta}\sqrt{\frac{2^n}{Z}}\right)$ time. Later Bravyi, Chowdhury, Gosset, and Wocjan [9] proposed another algorithm achieving the same running time while improving the space complexity.

When the temperature is not too low (i.e., β is not too large), Z is not exponentially small (for example, $Z \approx O(1)$ when $\beta \approx 0.7n$) and the running time is then roughly $O^*(2^{n/2})$.

Our main question

In summary, existing classical and quantum algorithms for LH run in $O^*(2^n)$ and $O^*(2^{n/2})$ time, respectively, suggesting a fundamental barrier to further improvements. The situation for QPF is even more mysterious: no $O(2^n)$-time classical algorithm is currently known to the best of our knowledge, and existing quantum algorithms fail to achieve the expected quadratic speedup in the low-temperature regime. These observations naturally motivate the following questions:

Do there exist classical or quantum algorithms that are significantly faster than $O(2^n)$ or $O(2^{n/2})$ for computing the ground-state energy? What is the best possible algorithms for approximating quantum partition function even in the low-temperature regime?

Fine-grained complexity, SETH and QSETH

Our goal is to provide a $\Omega(2^n)$ classical (resp. $\Omega(2^{n/2})$ quantum) lower bound. We are going to use the language of fine-grained complexity [49, 50, 43], which is a relatively recent field of computational complexity theory that investigates complexity in a more fine-grained way than the conventional class like P and NP.

The k-local Hamiltonian problem is a generalization of k-SAT, the central problem in classical complexity theory and its fine-grained complexity has been well studied. The Strong Exponential Time Hypothesis (SETH) [29] states that k-SAT needs roughly 2^n time for large k (see Conjecture 14).[6] More recently, a quantum analog was proposed in [1, 12] as Quantum Strong Exponential-Time Hypothesis (QSETH). The conjecture is to claim that any quantum algorithm to solve k-SAT needs $\Omega(2^{n/2})$ time (see Conjecture 15). SETH and QSETH are regarded as plausible computational conjectures and serve as a useful tool for deriving conditional lower bounds for various fundamental problems. First, the SAT problem has been studied for decades, and despite this extensive effort, the $\Omega(2^n)$ (resp. $\Omega(2^{n/2})$) lower bound for classical (resp. quantum) algorithms has not been broken; designing quantum algorithms that refute these conjectures would likely require fundamentally new techniques and could lead to major breakthroughs in algorithm design. In addition, (Q)SETH has been used to establish conditional lower bounds for a wide range of problems, such as the

[5] To the best of our knowledge, [9] first mentioned that approximating QPF with a relative error is QMA-hard; we provide a self-contained formal proof in Section 5.1.

[6] We stress that (Q)SETH are about the complexity of k-SAT for large k. For small values of k, an algorithms better than $O(2^{n/2})$ are known. For example, there exists a 1.32793^n-time classical algorithm for 3-SAT [40].

orthogonal vectors problem [6, 1], the closest pair problem [19, 31, 1, 12], the edit distance problem [5, 12], lattice problems [13, 28]. These results highlight the value of (Q)SETH as a powerful framework for studying the hardness of computational problems.

Barriers to fine-grained complexity lower bounds for $O(1)$-local Hamiltonians

We aim to prove the fine-grained complexity lower bound for LH and QPF assuming SETH and QSETH. One straightforward approach is to map the clauses in the k-SAT instance Φ to projectors that act on local qubits. The sum of these local projectors is a k-local Hamiltonian H whose ground subspace encodes the solutions for Φ (see Appendix A for the construction). In this reduction, the number of qubits of H is equal to the number of variables in Φ and therefore directly gives a $\Omega(2^n)$ (resp. $\Omega(2^{n/2})$) lower bound assuming SETH (resp. QSETH). However, the locality of the induced Hamiltonian is precisely k, where k is the parameter of the k-SAT hard instance; this value depends on ε, as characterized by SETH and QSETH. Hence, this approach fails to establish $\Omega(2^n)$ (resp. $\Omega(2^{n/2})$) lower bounds for Hamiltonians with fixed locality (e.g., 3-local).

Another approach is to use known circuit-to-Hamiltonian reductions, in which a verification circuit $U = U_T \cdots U_1$ for a QMA problem (e.g., a verification circuit of size $T = \mathrm{poly}(n)$ for a k-SAT instance) is embedded into a local Hamiltonian H. In particular, for any verification circuit, there are reductions to make the induced Hamiltonian 5-local [34] or 2-local [32], which achieves our purpose of fixed locality. Following the idea by Feynman [21], the standard circuit-to-Hamiltonian construction [34, 32] introduces a clock register, which is used to represent $T+1$ orthogonal clock states. The ground state of the resulting Hamiltonian is the history state

$$\frac{1}{\sqrt{T+1}} \sum_{i=0}^{T} U_i \cdots U_0 \left|\psi\right\rangle_{\mathrm{circuit}} \otimes \left|i\right\rangle_{\mathrm{clock}},$$

where $\left|\psi\right\rangle_{\mathrm{circuit}}$ denotes the initial state of the circuit (including the witness) and $\left|i\right\rangle_{\mathrm{clock}}$ denotes the clock states. However, this standard reduction results in significant size overhead due to the number of additional qubits needed to implement the clock states: in the standard reduction, a unary clock [34, 32] is used, which requires $O(T)$ qubits. Note that the analysis to obtain fine-grained lower bounds is very sensitive to the size overhead between the circuit and the local Hamiltonian. In particular, in order to obtain the desired lower bounds based on (Q)SETH, the clock need to be implementable in $o(n)$ qubits. Although we could use the binary clock [34] to reduce the size of the clock register from $O(T)$ qubits to $O(\log T)$ qubits, the locality would then increase to $O(\log T)$.

In summary, we require a k-SAT-to-LH reduction that simultaneously satisfies the following two conditions – a combination that existing methods fail to achieve:

1. *Size-preserving*: the resulting Hamiltonian acts on $n + o(n)$ qubits, and
2. *Locality-independent*: its locality is a constant independent of k.

1.2 Our results

In this paper, to investigate the fine-grained complexity for QMA-hard problems, we show the first size-preserving circuit-to-Hamiltonian construction. We denote by $\lambda(H)$ the smallest eigenvalue (ground-state energy) of a local Hamiltonian H.

▶ **Theorem 1** (Informal version of Theorem 17). *Let $L = (L_{\mathrm{yes}}, L_{\mathrm{no}})$ be a promise language in* QMA*, and $x \in \{0,1\}^n$ be an input for the language L. Let U_x be a verification circuit acting on N qubits and T be the number of elementary gates of U_x. Then, for any integer $d \geq 1$, there exists a $(d+1)$-local Hamiltonian H_x acting on $N + O(T^{1/d})$ qubits satisfying the following properties;*

- $||H_x|| \leq \mathrm{poly}(n)$,
- *if* $x \in L_{\mathrm{yes}}$, $\lambda(H_x) \leq E_{\mathrm{yes}}$,
- *if* $x \in L_{\mathrm{no}}$, $\lambda(H_x) \geq E_{\mathrm{no}}$,
- $E_{\mathrm{no}} - E_{\mathrm{yes}} \geq \Omega(1)$.

Our construction is the first construction to achieve both the $O(1)$-locality and the $o(T)$-qubit size for the clock register. See Table 1 for a comparison between our construction and the standard constructions.

Table 1 Comparison between our result and standard circuit-to-Hamiltonian constructions. Locality and Size indicate the locality and size of the local Hamiltonian reduced from a quantum circuit of T gates. N denotes the qubit size of verification quantum circuits. Our result holds for any integer $d \geq 1$.

Clock	Locality	Size
Binary [34]	$O(\log T)$	$N + O(\log T)$
Unary [34, 32]	2	$N + O(T)$
Our work (Theorem 17)	$d + 1$	$N + O(T^{1/d})$

We then show k-SAT has a very efficient verification procedure as a unitary circuit (Lemma 23). By combining this with our size-preserving circuit-to-Hamiltonian construction, we obtain the SETH- and QSETH-hardness for the local Hamiltonian problem.

▶ **Theorem 2** (Informal version of Theorem 22). *Assuming SETH (resp. QSETH), estimating the ground-state energy of a 3-local Hamiltonian problem with any constant additive error cannot be solved by any classical algorithm in time* $O^*\left(2^{(1-\varepsilon)n}\right)$ *(resp. by any quantum algorithm in time* $O^*\left(2^{(1-\varepsilon)n/2}\right)$*) for any constant* $\varepsilon > 0$.

Our lower bound for the local Hamiltonian problem matches the performance of the best known classical and quantum algorithms.

▶ Remark 3. Recently, Buhrman et al. [11] constructed classical and quantum algorithms running in time $2^{n(1-\Omega(\delta/M))}$ and $2^{n(1-\Omega(\delta/M))/2}$, respectively, where $M := \sum_i \|H_i\|$, to estimate the ground-state energy of a $O(1)$-local Hamiltonian up to an additive error δ. These upper bounds do not contradict our conditional lower bounds since Theorem 2 is about estimating the ground-state energy within constant additive error, and the upper bounds given by [11] become $O(2^{n(1-1/\mathrm{poly}(n))})$ and $O(2^{n(1-1/\mathrm{poly}(n))/2})$ in this regime.

We further investigate fine-grained complexity of quantum partition functions. We first show the SETH- and QSETH-hardness to estimate QPF with constant relative error.

▶ **Theorem 4** (Informal version of Theorem 26). *Assuming SETH (resp. QSETH), approximating the quantum partition function for 3-local Hamiltonians to within any constant relative error cannot be done by any classical algorithm in time* $O\left(2^{n(1-\varepsilon)}\right)$ *(resp. quantum algorithm in time* $O\left(2^{n(1-\varepsilon)/2}\right)$*) for any constant* $\varepsilon > 0$.

We then show a new algorithm for QPF.

▶ **Theorem 5** (Informal version of Theorem 27). *There exists a quantum algorithm whose running time is* $O^*\left(2^{n/2}\right)$ *to approximate the quantum partition function for a constant-local Hamiltonian on* n *qubits up to a relative error* $1/\mathrm{poly}(n)$.

We now compare our algorithm with the one proposed in [48] and [9]. The running time of their algorithm is $O^*\left(\frac{1}{\delta}\sqrt{\frac{2^n}{Z}}\right)$, which depends on the value of the partition function Z. When the inverse temperature $\beta = O(n^c)$ for some large constant c (i.e., at low temperatures), the running time may exceed $O(2^{n/2})$. In contrast, the running time of our algorithm does not depend on Z: it runs in $O^*(2^{n/2})$ for all $\beta = \mathrm{poly}(n)$.

1.3 Our techniques

Compressed clock state and how to use it for the circuit-to-Hamiltonian construction

To ensure that our reduction works, we need a construction of the clock state associated with the induced Hamiltonian that satisfies the following three conditions:

- **(i)** The clock state uses $o(n)$ of qubits and can encode $\mathrm{poly}(n)$ of computation steps,
- **(ii)** the operation on the clock state needs to be constant-local, with locality is independent of the circuit, and
- **(iii)** the computation process of U is encoded in the ground state of the induced Hamiltonian H.

The unary clock satisfies conditions (ii) and (iii), but does not satisfy condition (i). In [14], a clock construction is proposed that satisfies conditions (i) and (ii). To represent $O(a^d)$ time on an a-qubit register for integers $a \geq d \geq 1$, they use a Johnson graph where the vertices of the Johnson graph (a, d) are the d-element subset of an a-element set and two vertices are adjacent when the intersection of the two vertices (subsets) contains $(d-1)$-elements (see Definition 20 for a formal definition[7]). However, unlike the circuit-to-Hamiltonian constructions [34, 32], the circuit-to-Hamiltonian reduction in [14] encodes the computation in the *time evolution* of H, rather than in its ground state. As a result, there is no guarantee regarding the structure or properties of the ground state in their construction.

We build on the clock-state construction from [14] and a 2-local Hamiltonian construction from Section 5 in [32].[8] By introducing a carefully designed penalty term into H, we enforce that the computation history is stored in the ground state. This yields a novel circuit-to-Hamiltonian reduction that simultaneously preserves the size of the Hamiltonian, maintains constant locality, and ensures that the circuit U is encoded in the ground state of the resulting Hamiltonian H. Our construction satisfies all three requirements (i), (ii) and (iii), overcoming the limitations that neither of the existing reductions can address individually.

Additional techniques to reduce the locality

From the approach above, we have a $d+2$-local Hamiltonian from a quantum circuit whose number of gates is $O(a^d)$. This is because we need a local term operating on $d+1$ qubits in the clock register and 1 qubit in the circuit register for the propagation Hamiltonian.[9]

To reduce the locality further, we introduce a new clock construction that combines the $(a, d-1)$-clock state from [14] and the unary clock on a qubits. Crucial properties of the two clocks are:

---

[7] Not to confuse readers, in the introduction we denote by a the register size of the (a, d)-clock state and by n the input size.

[8] Section 6 of [32] introduces another approach to obtain a 2-local Hamiltonian using the perturbation theory.

[9] We do not need to consider 2-local gates thanks to the technique in Section 5 of [32].

- for the $(a, d-1)$ clock, time can be identified by looking at $(d-1)$ position ($(d-1)$-locality), while increasing or decreasing time by one unit can be done by modifying d positions (d-locality);
- for the unary clock, increasing or decreasing time by one unit can be done by modifying one position (1-locality). Moreover, if time is 0 or a, it can be identified by looking at 1 position (1-locality) (see Table 2 in Section 3.1).

By making the unary clock move periodically from 0 to a and from a to 0, we increase and decrease most of the clock time by d-local operators. By carefully analyzing the locality of all local Hamiltonian, we have a $(d+1)$-local Hamiltonian from a circuit of $O(a^d)$ gates using the $2a$-qubit clock register.

The remaining task is to construct a verification circuit that uses $o(n)$ ancilla qubits for any k-SAT instance whose number of variables is n. To achieve this, we use a counter to record the number of clauses satisfied by a given assignment. The k-SAT formula is satisfied if and only if the value stored in the counter equals m. Since counting up to m requires only $\log m$ qubits, the number of ancilla qubits needed is $n_a = O(\log m) = o(n)$, as required. The number of gates for the verification circuit is $O(n \log^2 n)$ (Lemma 23) and thus we can use $d = 2$ and obtain the SETH- and QSETH-hardness for the 3-local Hamiltonian problem.

Fine-grained complexity and quantum algorithms for the QPF problem

The lower bound for QPF follows directly from the lower bound for LH. When β is a sufficiently large polynomial, the partition function is dominated by low-energy states. If the ground-state energy is at most E_{yes}, then by ignoring all other states except the ground state, we obtain $Z \geq e^{-\beta E_{\text{yes}}}$. Conversely, if the ground-state energy is at least E_{no}, then by treating all eigenstates as having energy E_{no}, we get $Z \leq 2^n e^{-\beta E_{\text{no}}}$. Therefore, by approximating Z, we can distinguish between the two cases and decide the ground-state energy. To ensure this distinction, it suffices that $(1-\delta)e^{-\beta E_{\text{yes}}} > (1+\delta)2^n e^{-\beta E_{\text{no}}}$. In our reduction, we have observed that for any $\delta > 0$, this inequality holds for all sufficiently large n.

To obtain an $O(\sqrt{2^n})$-time quantum algorithm for the QPF with $1/\text{poly}(n)$ relative error, we adopt a technique from the proof of the equivalence between approximating the number of witnesses for a QMA verifier and approximating the quantum partition function [9]. Briefly, the approach involves partitioning the energy spectrum into polynomially many intervals and selecting a representative energy value $\mathcal{E}_j$ for each interval j. By estimating the number of eigenstates M_j within each interval, the partition function Z can be approximated as:

$$\tilde{Z} = \sum_j M_j e^{-\beta \mathcal{E}_j}.$$

We now explain how to count M_j. First, we construct an energy estimation circuit U_{EE} that outputs the corresponding eigenvalue E_p for each eigenstate $|\psi_p\rangle$ of H. In other words, $U_{EE}|\psi_p\rangle = |E_p\rangle|\psi_p\rangle$. The circuit U_{EE} can be implemented using Hamiltonian simulation together with phase estimation. Then, we apply a circuit U_{dec} that decides whether E_p is in the interval j. Let $U_j := U_{dec}U_{EE}$. Together, we have $U_j|\psi_p\rangle = |1\rangle|\phi_p\rangle$ if E_p in the interval j and $U_j|\psi_p\rangle = |0\rangle|\phi_p\rangle$ otherwise, where $|\phi_p\rangle$ is some quantum state corresponding to $|\psi_p\rangle$.

Now, let us suppose that the following two "too-good-to-be-true" assumptions hold:

1. A uniform superposition of all eigenstates can be efficiently prepared, and
2. the energy estimation circuit, U_{EE}, can compute eigenvalues with zero error.

Then, by applying U_j on the state, we obtain

$$U_j \sum_p \frac{1}{\sqrt{N}} |\psi_p\rangle = \sqrt{\frac{M_j}{N}} |1\rangle|\xi_1\rangle + \sqrt{\frac{N-M_j}{N}} |0\rangle|\xi_0\rangle, \tag{1}$$

where $N = 2^n$ is the number of the eigenstates and $|\xi_1\rangle$, $|\xi_0\rangle$ are two quantum states orthogonal to each other. Finally, by using the well-known amplitude estimation algorithm on Equation (1), we can obtain M_j. This algorithms runs in $O(\sqrt{2^n})$ time.

We then remove the two "too-good-to-be-true" assumptions while keeping the $O(\sqrt{2^n})$ runtime. First, directly preparing the uniform superposition of all eigenstates is generally computationally hard since the eigenstates are unknown. To overcome this difficulty, a commonly used technique is to use EPR state (e.g., [48, 33]). We have that uniform superposition over a complete basis, tensored with its complex conjugate, is equivalent to an EPR state. Therefore, by applying U_j to the EPR state, we obtain

$$U_j \sum_p \frac{1}{\sqrt{N}} |\psi_p\rangle |\psi_p^*\rangle = \sqrt{\frac{M_j}{N}} |1\rangle |\xi_1'\rangle + \sqrt{\frac{N - M_j}{N}} |0\rangle |\xi_0'\rangle , \tag{2}$$

where $|\xi_1'\rangle$ and $|\xi_0'\rangle$ are orthogonal quantum states. By applying the amplitude estimation algorithm to the state in Equation (2), we can approximate M_j.

Second, the quantum energy estimation circuit described above introduces an additive error of $1/\mathrm{poly}(n)$, which can significantly impact the approximation of the quantum partition function. Specifically, if the eigenstates are densely concentrated near the boundaries of the intervals in the selected partition (within the precision of the energy estimation), we may "doubly count" these states. Such inaccuracies in state counting translate directly into a constant relative error in the final estimation of the partition function.

To reduce the constant relative error to $1/\mathrm{poly}(n)$, we observe that the "double-counting" issue only arises when the chosen partition unluckily has significant fraction of eigenvalues lie near the boundaries of intervals. Intuitively, by considering polynomially many partitions with mutually disjoint boundaries, any specific eigenstate can be miscounted in at most a constant number of these partitions. Consequently, if we run the algorithm over all such partitions, an averaging argument ensures that at least one set will have a number of double-counted eigenstates that is at most an inverse polynomial fraction of the total. This approach yields an algorithm achieving $1/\mathrm{poly}(n)$ relative error in $O(\sqrt{2^n})$ time.

1.4 Open questions

1. It is known that the 2-local Hamiltonian problem is QMA-hard [32]. **Can we show a size-preserving reduction to 2-local Hamiltonian and its (Q)SETH-hardness?** In our clock construction, we need to touch two clocks and simultaneously apply a 1-qubit gate, so we cannot improve the locality directly. Note that perturbation theory does not seem to work here either: the standard perturbation technique adds an ancilla qubit for each term, and thus does not give a size-preserving construction when applied directly to our 3-local Hamiltonian.
2. Even when we consider geometrically local [45] or specific 2-local terms such as Heisenberg interaction, antiferromagnetic XY interactions [17, 46], 2-local ZX and $ZZXX$ Hamiltonian [7], the problem is still QMA-hard. This leads to a natural question: **Does the (Q)SETH-hardness still hold for geometrically local Hamiltonians whose each local term is restricted to a specific form?**
3. Our relative threshold gap $(E_{\mathrm{no}} - E_{\mathrm{yes}})/M$ in the 3-local Hamiltonian problem is $1/p(n)$ where $M := \sum_{i=i}^{m} \|H_i\|$ and $p(n)$ is a polynomial with large degree. A natural question arises: **when the relative gap increases, for example, $(E_{\mathrm{no}} - E_{\mathrm{yes}})/M = \Omega(1)$, what conditional lower bound can we have under plausible conjectures?** Notably, the algorithm of [11] runs in time $O^*(2^{\frac{n}{2}(1-\Omega(1))})$ when $(E_{\mathrm{no}} - E_{\mathrm{yes}})/M = \Omega(1)$, and therefore having the same lower bound $\Omega(2^{\frac{n}{2}(1-o(1))})$ for the $(E_{\mathrm{no}} - E_{\mathrm{yes}})/M = \Omega(1)$ regime as we have for $(E_{\mathrm{no}} - E_{\mathrm{yes}})/M = 1/p(n)$ in our work is impossible.

4. In this paper, we show the SETH-hardness of the 3-local Hamiltonian problem, which is a quantum analog of MAX-3-SAT, and it is known that there exists a faster exponential time algorithm for MAX-2-SAT, say $O^*(1.732^n)$-time algorithm by [51]. **Can we show the fine-grained complexity of MAX-3-SAT?**
5. In this work, we show that LH can be reduced to QPF, and the reduction is fine-grained even for an arbitrary constant relative error; therefore, we can obtain the $\Omega(2^{n/2})$ lower bound assuming (Q)SETH. Notably, the lower bound holds for any constant relative error and locality at least 3. Along this line, we can ask **whether the lower bound continues to hold when considering 2-local Hamiltonian, geometrical restrictions, and specified local terms, or we can design faster algorithms under these constraints.**
6. Another interesting question is **whether the reverse reduction holds, namely, whether approximating QPF can be reduced to the LH problem.** At first glance, this seems unlikely, as QPF can be interpreted as estimating the dimension of the ground space in the low-temperature regime. However, since we are considering exponential lower bounds, such a reduction could, in principle, be allowed to run in super-polynomial time.
7. Our lower bound for the 3QPF problem holds for $\beta = O(n)$ and $\|H\|$ being a large degree polynomial. A natural question to ask is **what happens as the temperature increases.** In other words, when $\beta\|H\|$ becomes small, does the lower bound still hold in this regime? Or, at what point do we observe a *phase transition*?

1.5 Organization

In Section 2, we give preliminaries for the rest of the paper. In Section 3, we show the size-preserving circuit-to-Hamiltonian construction by introducing a new clock. In Section 4, we show the SETH-hardness of the local Hamiltonian problem. In Section 5, we show the fine-grained complexity to estimate the quantum partition function with relative error.

2 Preliminaries

We refer to [44, 20] for references in quantum computing, [23] for a reference and survey in quantum Hamiltonian complexity.

2.1 General notation

For $n \in \mathbb{N}$, we use $[n]$ to denote the set collecting all the positive integers smaller than or equal to n, that is, $\{1, 2, \ldots, n\}$.

For a bit string $x \in \{0,1\}^*$, we use $int(x)$ to denote the corresponding integer whose binary representation is x. For a nonnegative integer n, we use $bin(n)$ to denote the binary representation of n. For example, $int(01010111) = 87$ and $bin(87) = 01010111$.

For an n-bit string $x \in \{0,1\}^n$ and $i \in [n]$, we use $x[i]$ to denote the ith bit of x. If x is a binary representation of an integer, then $x[1]$ is the most significant bit and $x[n]$ is the least significant bit. Let $S \in [n]$, we use x_S to denote a new string that concatenates x's bits whose indices are in S. For example, if $x = 01010111$ and $S = \{1, 3, 7, 8\}$, then $x_S = 0011$. The Hamming weight of x is denoted by $wt(x)$, that is, $wt(x)$ is the number of 1's in x.

The identity operator is denoted by I. The Kronecker delta $\delta_{i,j}$ is defined by $\delta_{i,j} = 1$ if $i = j$ and $\delta_{jk} = 0$ if $i \neq j$. The indicator function is denoted by $\mathbb{1}_S(i)$, which is defined by $\mathbb{1}_S(i) = 1$ if $i \in S$ and $\mathbb{1}_S(i) = 0$ if $i \notin S$.

The big O star notation $O^*(\cdot)$ hides the polynomial factors in the standard big O notation. For example, $8 \cdot 2^{\frac{n}{2}} \cdot n^7 = O^*(2^{\frac{n}{2}})$. The negligible functions $\mathrm{negl}(\cdot)$ are functions that are smaller than any inverse polynomial: if $\mu(n) \in \mathrm{negl}(n)$ then for all $c \in \mathbb{N}$, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, it holds that $\mu(n) < \frac{1}{n^c}$.

2.2 Quantum computation

Quantum register

We use the sans-serif font to denote quantum registers, e.g., A, in, out, anc, clock. Throughout this paper, a register consists of qubits. A qubit may belong to different registers at the same time. For example, a qubit in the output register out of a quantum circuit is also in the ancilla register anc. By an abuse of notation, sometimes we treat the registers as sets. We say $\mathsf{A} \subseteq \mathsf{B}$ if any qubit in A is also a qubit in B. We use $\mathsf{A} \cup \mathsf{B}$ to denote the register that consists of the qubits that are in A or in B, $\mathsf{A} \cup \mathsf{B}$ to denote the portion that belongs to A and B at the same time, and $\mathsf{A} \setminus \mathsf{B}$ to denote the portion that belongs to A but not B. When we specify a system (that can be a quantum circuit or a physical qubit system), we use $\overline{\mathsf{A}}$ to denote the collection of qubits in the system that are not in A. We use $|\mathsf{A}|$ to denote the number of qubits in A. When we specify a register A, we use $\mathsf{A}[\mathrm{i}]$ to denote the ith qubit in A for $i \in [|\mathsf{A}|]$.

We use $|\psi\rangle_{\mathsf{A}}$ to denote the register A is in the state $|\psi\rangle$. For a Hermitian or a unitary operator X, we use X_{A} to denote X acting on the register A. When the registers are specified, the order of the tensor product of the operators is not sensitive. For example, $X_{\mathsf{A}} \otimes Y_{\mathsf{B}} = Y_{\mathsf{B}} \otimes X_{\mathsf{A}}$. We use $X^{\otimes k}$ to denote the kth tensor power of X for $k \in \mathbb{N}$. Namely, $X^{\otimes k} := \underbrace{X \otimes X \otimes \cdots \otimes X}_{k}$.

Quantum circuit

A quantum circuit is a unitary acting on the union of two disjoint register in and anc, which are called input register and ancilla register respectively. Also, there is an output register $\mathsf{out} \subseteq \mathsf{in} \cup \mathsf{anc}$. We require the ancilla register is $|0^{n_a}\rangle_{\mathsf{anc}}$ initially, where $n_a = |\mathsf{anc}|$. That is, if the circuit U takes a quantum state $|\psi\rangle$ as an input, then the final state of the circuit is $U|\psi\rangle_{\mathsf{in}}|0^{n_a}\rangle_{\mathsf{anc}}$. We use the notation $x \leftarrow U(|\psi\rangle)$ to denote the event that we obtain the measurement outcome $x \in \{0,1\}^{|\mathsf{out}|}$ when we measure on the output register out at the end of the circuit U that takes $|\psi\rangle$ as the input state. We have $\Pr[x \leftarrow U(|\psi\rangle)] = \langle\psi|_{\mathsf{in}}\langle 0^{n_a}|_{\mathsf{anc}} U^\dagger (|x\rangle\langle x|_{\mathsf{out}} \otimes I_{\overline{\mathsf{out}}}) U|\psi\rangle_{\mathsf{in}}|0^{n_a}\rangle_{\mathsf{anc}}$.

Universal gate set and elementary gates

A quantum circuit is implemented by a sequence of quantum gates. There is a finite set of one-qubit and two-qubit quantum gates called the universal gate set such that for any unitary U, we can use the gates in the universal gate sets to implement a quantum circuit arbitrarily close to U. We call a member in the universal gate set an elementary gate. We choose $\{\textsc{hadamard}, \pi/8\text{-}\textsc{gate}, \textsc{not}, \textsc{cnot}\}$ as the universal gate set [44]

The $\textsc{not}$ gate acts on a one-qubit register. The truth table of the $\textsc{not}$ gate is defined by $\textsc{not}|x\rangle = |(1+x) \mod 2\rangle$ where $x \in \{0,1\}$.

We also introduce a multi-control-NOT gate.

▶ **Definition 6** (Multi-control-NOT gate and Toffoli gate). *A multi-control gate, denoted by C^kNOT, is a quantum gate that acts on $k+1$ qubits. Let* C *be a k-qubit register and* T *be a one-qubit register. The truth table of the C^kCNOT gate is defined by*

$$C^k\textsc{cnot}|x_1,x_2,\ldots,x_k\rangle_{\mathsf{C}}|x_{k+1}\rangle_{\mathsf{T}} = |x_1,x_2,\ldots,x_k\rangle_{\mathsf{C}}|(x_1x_2\cdots x_k + x_{k+1}) \mod 2\rangle_{\mathsf{T}},$$

where $x_1,x_2,\ldots,x_{k+1}\in\{0,1\}$. We call C *the control qubits and* T *the target qubit.*

For $k=2$, we call C^2NOT a Toffoli gate.

According to the definition of C^kNOT, we have that CNOT is a special case of C^kNOT gate for $k=1$.

▶ Remark 7 (Decompose C^kNOT into Toffolis [44]). A C^kNOT gate can be constructed by $2k-3$ Toffoli gates associated with $k-2$ ancilla qubits. The ancilla qubits are in all zero state initially and stay unchanged at the end of circuit.

▶ Remark 8 (Decompose Toffoli into elementary gates [44]). A Toffoli gate is identical to a circuit that acts on three qubits and is composed of 17 elementary gates.

2.3 Local Hamiltonian problem

A Hamiltonian is an Hermitian operator acting on a quantum register. Let H be a Hamiltonian, we call the eigenvalues of H the energies of the Hamiltonian. We use $\lambda(H)$ to denote the smallest eigenvalue of H. We call $\lambda(H)$ the ground-state energy of H, and we call the corresponding eigenstate(s) the ground state(s) of H. We use $\|H\|$ to denote the operator norm of H, that is, the largest absolute value of the eigenvalues of H.

The k-local Hamiltonian is defined as follows.

▶ **Definition 9** (Local Hamiltonian). *We say a Hamiltonian H acting on n qubits is k-local if H can be written as $H=\sum_{i=1}^{m}H_i$ for $m=\mathrm{poly}(n)$ and for all i, the following holds.*

- *H_i is a Hamiltonian.*
- *H_i non-trivially acts on at most k qubits.*
- *$\|H_i\|\le\mathrm{poly}(n)$.*

We call k the locality of H, and define the local Hamiltonian problem formally.

▶ **Definition 10** (k-local Hamiltonian (k-LH) problem). *The local Hamiltonian problem is a decision problem that asks whether the ground-state energy of a k-local Hamiltonian is greater or less than given energy thresholds.*

- ***Inputs:*** *a k-local Hamiltonian H such that acting on n qubits where $k=O(1)$, and two energy thresholds $E_{\mathrm{yes}},E_{\mathrm{no}}$ satisfying $E_{\mathrm{no}}-E_{\mathrm{yes}}\ge\Omega(1)$.*
- ***Outputs:***
  - *YES, if there exists a quantum state $|\psi\rangle\in\mathbb{C}^{2^n}$ such that $\langle\psi|H|\psi\rangle\le E_{\mathrm{yes}}$.*
  - *NO, if $\langle\psi|H|\psi\rangle\ge E_{\mathrm{no}}$ for all $|\psi\rangle\in\mathbb{C}^{2^n}$.*

In other words, the k-LH problem asks to decide whether $\lambda(H)\le E_{\mathrm{yes}}$ or $\lambda(H)\ge E_{\mathrm{no}}$. We say an algorithm $\mathcal{A}_{\mathrm{LH}}$ solves k-LH$(H,E_{\mathrm{yes}},E_{\mathrm{no}})$ if $\mathcal{A}_{\mathrm{LH}}$ decides $(H,E_{\mathrm{yes}},E_{\mathrm{no}})$ correctly.

2.4 Quantum partition function

Let us introduce the quantum partition function formally.

▶ **Definition 11** (Approximating quantum partition function of k-local Hamiltonian (k-QPF) problem). *The quantum partition function problem is to approximate the partition function of a k-local Hamiltonian up to a multiplicative error under a certain temperature.*

- ***Inputs:*** *a k-local Hamiltonian H acting on n qubits where $k = O(1)$, an inverse temperature $\beta \leq \mathrm{poly}(n)$, and an error parameter $\delta \in (0,1)$.*
- ***Outputs:*** *$\widetilde{Z} \in \mathbb{R}$ such that*

$$(1-\delta)Z \leq \widetilde{Z} \leq (1+\delta)Z, \tag{3}$$

where $Z := \mathrm{tr}[e^{-\beta H}]$.

We say an algorithm $\mathcal{A}_{\mathrm{QPF}}$ solves k-QPF(H,β,δ) if $\mathcal{A}_{\mathrm{QPF}}$ outputs such $\widetilde{Z}$ on the inputs H,β,δ.

From now on, when we use the term QPF or k-QPF, it means to approximate the quantum partition function up to a relative error, instead of to compute the exact value.

2.5 Boolean formula, SAT, SETH and QSETH

A Boolean variable can be assigned to the value 0 or 1. We abbreviate the term Boolean variable as variable. A Boolean formula consists of variables associated with parentheses and logic connectives $\neg$ (NOT), $\vee$ (OR), and $\wedge$ (AND).

Let Φ be a Boolean formula with n variables $x_1, x_2, \ldots, x_n$. We use an an n-bit string $x \in \{0,1\}^n$ to denote an assignment to $x_1, x_2, \ldots, x_n$. The variable x_i for $i \in [n]$ is assigned to the ith bit of x. We use $\Phi(x)$ to denote the value of Φ when $x_1, x_2, \ldots, x_n$ are assigned to x. We say x satisfies Φ if $\Phi(x) = 1$.

For all variables x, it holds that $\neg(\neg x) = x$. Let $x_1, x_2, \ldots, x_n$ be variables and for each $i \in [n]$, let ℓ_i be either x_i or $\neg x_i$. It holds that $\neg(\ell_1 \wedge \ell_2 \wedge \cdots \wedge \ell_m) = \neg\ell_1 \vee \neg\ell_2 \vee \cdots \vee \neg\ell_m$.

We can use a quantum circuit to compute Boolean formulas. For $x \in \{0,1\}$, it holds that

$$\mathrm{NOT}\,|x\rangle = |\neg x\rangle\,. \tag{4}$$

For $x_1, x_2, \ldots, x_k \in \{0,1\}$, it holds that

$$C^k\mathrm{NOT}_{\mathsf{C}\cup\mathsf{T}}\,|x_1, x_2, \ldots, x_k\rangle_{\mathsf{C}}\,|0\rangle_{\mathsf{T}} = |x_1, x_2, \ldots, x_k\rangle_{\mathsf{C}}\,|x_1 \wedge x_2 \wedge \cdots \wedge x_k\rangle_{\mathsf{T}}\,. \tag{5}$$

We define the conjunctive normal form formula as follows.

▶ **Definition 12** (Conjunctive normal form (CNF) formula)**.** *Let Φ be a Boolean formula that consists of n variables $x_1, x_2, \ldots, x_k$. We say Φ is a kCNF formula if Φ is in the form of $m = \mathrm{poly}(n)$ smaller formulas connected by $\wedge$. Each smaller formula contains at most k variables, and the variables are connected by $\vee$.*

To be more precise, $\Phi = \varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_m$ where $m = \mathrm{poly}(n)$, and for all $i \in [m]$, the following constrains hold.

- *$\varphi_i = (\ell_{i,1} \vee \ell_{i_2} \vee \cdots \vee \ell_{i,k_i})$, where $\ell_{i,p}$ can be x_j or $\neg x_j$ for all $p \in [k_i]$ and $j \in [n]$.*
- *$k_i \leq k$.*
- *For each $j \in [n]$, x_j and $\neg x_j$ do not appear in φ_i at the same time; x_j, $\neg x_j$ appears in φ_i at most once.*

We say φ_i is a clause of Φ for all $i \in [m]$.

▶ **Definition 13** (Satisfiability for k-CNF (k-SAT) problem)**.** *The satisfiability problem is a decision problem that asks whether there is an assignment satisfying the given kCNF formula.*

- ***Inputs:*** *a k-CNF formula Φ that contains n variables. The number of clauses of Φ is $m = \mathrm{poly}(n)$.*
- ***Outputs:***
  - *YES, if there exists an assignment $x \in \{0,1\}^n$ such that $\Phi(x) = 1$.*
  - *No, if $\Phi(x) = 0$ for all $x \in \{0,1\}^n$.*

We say an algorithm $\mathcal{A}_{\mathrm{SAT}}$ solves k-SAT(Φ) if $\mathcal{A}_{\mathrm{SAT}}$ decides Φ correctly.

Though the exact lower bound for k-SAT problem is still unknown, it is widely believed that brute-force search is optimal for classical algorithm and Grover search is optimal for quantum algorithm. The Strong Exponential Time Hypothesis (SETH) [29] states that k-SAT needs roughly 2^n time for large k:

▶ **Conjecture 14** (SETH). *For all ε, there is some $k \geq 3$ such that k-SAT with n variables and $O(n)$ clauses cannot be solved classically in time $O(2^{(1-\varepsilon)n})$.*

A quantum analog was proposed in [1, 12] as Quantum Strong Exponential-Time Hypothesis (QSETH).

▶ **Conjecture 15** (Quantum strong exponential time hypothesis (QSETH) [1, 12]). *For all ε, there is some $k \geq 3$ such that k-SAT with n variables and $O(n)$ clauses cannot be solved quantumly in time $O(2^{(1-\varepsilon)n/2})$.*

In the two conjecture above, we can assume the number of clauses can be assumed to be $O(n)$ via the sparsification lemma [30].

2.6 Fine-grained reduction

Here we introduce the fine-grained reduction.

▶ **Definition 16** (Fine-grained reduction [49]). *Let P and Q be two problems and $\mathcal{A}_Q$ be an oracle that solves Q with probability greater than $\frac{2}{3}$. Let $p(\cdot)$ and $q(\cdot)$ be two non-decreasing functions. We say P is (p, q) reducible to Q if for all ε, there exist ξ, an algorithm $\mathcal{A}_P$, a constant d, and an integer $r(n)$ such that the algorithm $\mathcal{A}_P$ that can black-boxly assess to $\mathcal{A}_Q$ takes an instance of P with size n and satisfies the following.*

1. *$\mathcal{A}_P$ solves P with probability greater than $\frac{2}{3}$.*
2. *$\mathcal{A}_P$ runs in $d \cdot p(n)^{1-\xi}$ time.*
3. *$\mathcal{A}_P$ produces at most $r(n)$ instances of Q adaptively.*
4. *$\sum_{i=1}^{r(n)} (q(n_i))^{1-\varepsilon} \leq d \cdot (p(n))^{1-\xi}$, where n_i is the size of the ith instance of problem Q that is produced by $\mathcal{A}_P$.*

One can also use a more general reduction, *quantum fine-grained reduction* [1]. In a quantum fine-grained reduction, the reduction algorithm $\mathcal{A}_P$ is allowed to query the oracle $\mathcal{A}_Q$ in superposition. In this work, however, Definition 16 is sufficient to establish the lower bounds for both the Local Hamiltonian problem and the approximation of the Quantum Partition Function.

When P is (p, q) reducible to Q and every instance produced by $\mathcal{A}_P$ has size $n + o(n)$, we have that if P cannot be solved within time $O(p(n)^{1-\varepsilon})$ for any ε, then Q cannot be solved within time $O(q(n)^{1-\xi})$ for any ξ.

3 Size-preserving circuit-to-Hamiltonian construction

In this section, we prove the first size-preserving circuit-to-Hamiltonian construction as follows.

▶ **Theorem 17.** *Let $x \in \{0,1\}^n$ be an input and U_x be a* QMA *verification quantum circuit U_x acting on $N = \mathrm{poly}(n)$ qubits, and suppose U_x consists of $T = \mathrm{poly}(n)$ gates. Then, for any integer $d \geq 1$, there exists a $(d+1)$-local Hamiltonian H_x acting on $N + O(T^{1/d})$ qubits and $\|H_x\| \leq \mathrm{poly}(n)$ such that for all $\varepsilon_{\mathrm{err}} \in (0,1)$, the following two hold.*

- *If there exists a quantum proof* $|\psi\rangle$ *such that* $\Pr[U_x \text{ accepts } |\psi\rangle] \geq 1 - \varepsilon_{\text{err}}$*, then* $\lambda(H_x) \leq \varepsilon_{\text{err}}$.
- *If* $\Pr[U_x \text{ accepts } |\psi\rangle] \leq \varepsilon_{\text{err}}$ *for any quantum proof* $|\psi\rangle$*, then* $\lambda(H_x) \geq \frac{1}{2} - \varepsilon_{\text{err}}$.

To prove the theorem above, we first recall the strategy for the circuit-to-Hamiltonian construction in [32]. We then define our clock states and finally prove the statement.

3.1 The construction by Kitaev, Kempe and Regev [32]

In this section, we explain the 2-local construction from [32] with some adaptation.

We consider a promise problem $(L_{\text{yes}}, L_{\text{no}})$ in QMA. Let $U = U_T \cdots U_2 U_1$ be the quantum circuit of the verifier (which depends on the input), where $T = \text{poly}(n)$. We assume that U acts on $N = \text{poly}(n)$ qubits: the first m qubits contain the proof (the last $N - m$ qubits are initialized to $|0\rangle$). Here, each U_i is a 1-qubit gate or a 2-qubit gate. As in [32], we assume that the only 2-qubit gate used by the circuit is the controlled phase gate, each controlled phase gate is preceded and followed by two Z gates[10] Let $T_A \subseteq \{1, \ldots, T\}$ be the set of indices i such that U_i is a 1-qubit gate. Let $T_B \subseteq \{1, \ldots, T\}$ be the set of indices i such that U_i is a 2-qubit gate.

Meta-definition of the Hamiltonian [11]

The Hamiltonian is defined over the Hilbert space $\mathcal{H}_{\text{circuit}} \otimes \mathcal{H}_{\text{clock}}$, where $\mathcal{H}_{\text{circuit}}$ is a Hilbert space of dimension 2^N representing the computation and $\mathcal{H}_{\text{clock}}$ is a Hilbert space of dimension at least $T + 1$ representing the clock. There are $T + 1$ orthonormal unit vectors $|\gamma_0\rangle, \ldots, |\gamma_T\rangle$ in $\mathcal{H}_{\text{clock}}$ that are called the legal clock states. Let $\mathcal{H}_{\text{legal}}$ denote the subspace spanned by the states $|\gamma_0\rangle, \ldots, |\gamma_T\rangle$ and Π_{legal} be the projection into this subspace. Let $\mathcal{H}_{\text{illegal}}$ be its orthogonal subspace. We define the operators

$$\begin{aligned}
\mathcal{S}_t &= |\gamma_t\rangle\langle\gamma_t| \quad \text{for any } t \in \{0, \ldots, T\}, \\
\mathcal{A}_t &= |\gamma_{t+1}\rangle\langle\gamma_t| \quad \text{for any } t \in \{0, \ldots, T-1\}, \\
\mathcal{B}_t &= |\gamma_{t+2}\rangle\langle\gamma_t| \quad \text{for any } t \in \{0, \ldots, T-2\},
\end{aligned}$$

acting on $\mathcal{H}_{\text{clock}}$ that respectively stop the clock, increment the clock by one step and increment the clock by two steps.

Consider the following Hamiltonian:

$$H = \alpha_{\text{in}} H_{\text{in}} + \alpha_{\text{out}} H_{\text{out}} + \alpha_A \sum_{t \in T_A} H_{\text{prop},t} + \alpha_B \sum_{t \in T_B} (H_{\text{qubit},t} + H_{\text{time},t}) + \alpha_{\text{clock}} I \otimes H_{\text{clock}}.$$

The choice of the coefficients α_{in}, α_{out}, α_A, α_B, α_{clock} will be discussed later. The terms H_{in} and H_{out} are defined as follows:

$$H_{\text{in}} = \sum_{i=m+1}^{N} |1\rangle_i\langle 1|_i \otimes \mathcal{S}_0, \qquad H_{\text{out}} = |0\rangle_1\langle 0|_1 \otimes \mathcal{S}_T.$$

For any $t \in T_A$,

$$H_{\text{prop},t} = \frac{1}{2}\Big(I \otimes \mathcal{S}_t + I \otimes \mathcal{S}_{t-1} - U_t \otimes \mathcal{A}_{t-1} - U_t^\dagger \otimes \mathcal{A}_{t-1}^\dagger\Big).$$

[10] Any quantum circuit composed of the standard universal gate set can be compiled to the circuit described above with only linear blowup in circuit size because CNOT $= (I \otimes H)\text{CZ}(I \otimes H)$.

[11] This is a "meta-definition" since it does not define the concrete implementation of the clock states.

For any $t \in T_B$,

$$H_{\text{time},t} = \frac{1}{8} I \otimes \Big(\mathcal{S}_t + 6\mathcal{S}_{t+1} + \mathcal{S}_{t+2} + 2\mathcal{B}_t + 2\mathcal{B}_t^\dagger + \mathcal{A}_t + \mathcal{A}_t^\dagger + \mathcal{A}_{t+1} + \mathcal{A}_{t+1}^\dagger + \mathcal{S}_{t-3} + 6\mathcal{S}_{t-2} + \mathcal{S}_{t-1} + 2\mathcal{B}_{t-1} + 2\mathcal{B}_{t-1}^\dagger + \mathcal{A}_{t-3} + \mathcal{A}_{t-3}^\dagger + \mathcal{A}_{t-1} + \mathcal{A}_{t-1}^\dagger \Big),$$

$$H_{\text{qubit},t} = \frac{1}{2} \Big(-2 |0\rangle\langle 0|_{f_t} - 2 |0\rangle\langle 0|_{s_t} + |1\rangle\langle 1|_{f_t} + |1\rangle\langle 1|_{s_t} \Big) \otimes (\mathcal{A}_{t-1} + \mathcal{A}_{t-1}^\dagger),$$

where f_t and s_t denote the first and second qubits of the controlled phase gate U_t. The term H_{clock} will be chosen later so that the following two conditions are satisfied

$$\|H_{\text{clock}}\| \le \text{poly}(n), \tag{C1}$$

$$\langle \gamma_t | H_{\text{clock}} | \gamma_t \rangle = 0 \quad \text{for any} \quad t \in \{0, \dots, T\}, \tag{C2}$$

$$\langle \varphi | H_{\text{clock}} | \varphi \rangle \ge 1 \quad \text{for any} \quad |\varphi\rangle \in \mathcal{H}_{\text{illegal}}. \tag{C3}$$

Concrete implementation

A straightforward implementation uses a binary clock, as in the original construction by [34]. While requiring only $O(\log T)$ clock qubits (and no H_{clock} term), the locality of the Hamiltonian would become $O(\log n)$. We discuss below how to reduce the locality.

The basic idea is to replace each of the operator $\mathcal{S}_t, \mathcal{A}_t, \mathcal{B}_t$ by another operator $\widetilde{\mathcal{S}}_t, \widetilde{\mathcal{A}}_t, \widetilde{\mathcal{B}}_t$ that has better locality. In order for the implementation to be correct, the following three conditions need to be satisfied.

$$\text{For any} \quad t \in \{0, \dots, T\}, \quad \begin{cases} \widetilde{\mathcal{S}}_t |\gamma_t\rangle = |\gamma_t\rangle \\ \Pi_{\text{legal}} \widetilde{\mathcal{S}}_t |\varphi\rangle = 0 \quad \text{for any } |\varphi\rangle \in \mathcal{H}_{\text{clock}} \text{ orthogonal to } |\gamma_t\rangle. \end{cases} \tag{C4}$$

$$\text{For any} \quad t \in \{0, \dots, T-1\}, \quad \begin{cases} \widetilde{\mathcal{A}}_t |\gamma_t\rangle = |\gamma_{t+1}\rangle \\ \Pi_{\text{legal}} \widetilde{\mathcal{A}}_t |\varphi\rangle = 0 \quad \text{for any } |\varphi\rangle \in \mathcal{H}_{\text{clock}} \text{ orthogonal to } |\gamma_t\rangle. \end{cases} \tag{C5}$$

$$\text{For any} \quad t \in \{0, \dots, T-2\}, \quad \begin{cases} \widetilde{\mathcal{B}}_t |\gamma_t\rangle = |\gamma_{t+2}\rangle \\ \Pi_{\text{legal}} \widetilde{\mathcal{B}}_t |\varphi\rangle = 0 \quad \text{for any } |\varphi\rangle \in \mathcal{H}_{\text{clock}} \text{ orthogonal to } |\gamma_t\rangle. \end{cases} \tag{C6}$$

The first part of each condition says that each $\widetilde{\mathcal{S}}_t, \widetilde{\mathcal{A}}_t, \widetilde{\mathcal{B}}_t$ behaves as $\mathcal{S}_t, \mathcal{A}_t, \mathcal{B}_t$, respectively, on the clock state $|\gamma_t\rangle$. The second part says that while the behavior may differ from the behavior of $\mathcal{S}_t, \mathcal{A}_t, \mathcal{B}_t$ on other states, $\widetilde{\mathcal{S}}_t, \widetilde{\mathcal{A}}_t, \widetilde{\mathcal{B}}_t$ never maps an illegal state or even a legal state other than $|\gamma_t\rangle$ to a legal state.

The analysis based on the projection lemma (Lemma 1 in [32]) shows the following.

▶ **Proposition 18** (Adapted from Section 5 in [32]). *If Conditions* $(C_1) \sim (C_6)$ *are satisfied, coefficients* α_{in}, α_{out}, α_A, α_B, α_{clock} *can be chosen in the range* $[0, \text{poly}(n)]$ *so that the following holds:*

- *If there exists a quantum proof* $|\psi\rangle$ *such that* $\Pr[U_x \text{ accepts } |\psi\rangle] \ge 1 - \varepsilon$, *then* $\lambda(H_x) \le \varepsilon$.
- *If* $\Pr[U_x \text{ accepts } |\psi\rangle] \le \varepsilon$ *for any quantum proof* $|\psi\rangle$, *then* $\lambda(H_x) \ge \frac{1}{2} - \varepsilon$.

Implementation by Kempe-Kitaev-Regev

The implementation in [32] uses the unary encoding 0000, 1000, 1100, 1110, 1111$\cdots$ for the clock states, i.e., uses the T-qubit clock states

$$|\gamma_t\rangle = |1^t 0^{T-t}\rangle$$

for each $t \in \{0,\dots,T\}$, and the operators $\widetilde{\mathcal{S}}_t, \widetilde{\mathcal{A}}_t, \widetilde{\mathcal{B}}_t$ described in the following table:

Table 2 Concrete implementation of operators in [32].

Operator	t	Implementation	Locality
$\mathcal{S}_t$	$t=0$	$\widetilde{\mathcal{S}}_0 = \|0\rangle_1 \langle 0\|_1$	1
	$t \in \{1,\dots,T-1\}$	$\widetilde{\mathcal{S}}_t = \|10\rangle_{t,t+1} \langle 10\|_{t,t+1}$	2
	$t=T$	$\widetilde{\mathcal{S}}_T = \|1\rangle_T \langle 1\|_T$	1
$\mathcal{A}_t$	$t \in \{0,\dots,T-1\}$	$\widetilde{\mathcal{A}}_t = \|1\rangle_{t+1} \langle 0\|_{t+1}$	1
$\mathcal{B}_t$	$t \in \{0,\dots,T-2\}$	$\widetilde{\mathcal{B}}_t = \|11\rangle_{t+1,t+2} \langle 00\|_{t+1,t+2}$	2

The term H_{clock} used in [32] is defined as follows:

$$H_{\mathrm{clock}} = \sum_{1\le i<j\le T} |01\rangle_{ij} \langle 01|_{ij} \ .$$

This implementation satisfies Conditions $(\mathrm{C}_1) \sim (\mathrm{C}_6)$. With this implementation, each term H_{in}, H_{out}, $H_{\mathrm{prop},t}$, $H_{\mathrm{qubit},t}$, $H_{\mathrm{time},t}$, H_{clock} has locality 2. The whole Hamiltonian is thus 2-local.

3.2 (n,d)-clock state from [14] and associated operators to implement LH

In this subsection, we recall an (n,d)-clock state proposed in [14], and introduce and analyze operators to use it for circuit-to-Hamiltonian constructions. The (n,d)-clock state is based on a Johnson graph. The vertices of the Jonson graph are all the subsets of $[n]$ with size $k<n$. Two vertices are adjacent if and only if the subsets share exactly $d-1$ elements.

▶ **Definition 19** (Johnson graph). *Let $n,d \in \mathbb{N}$ and $d<n$. We say a Johnson graph $J(n,d)=(V,E)$ is a graph that satisfies the following requirement.*

- $V = \{S \subseteq [n] : |S| = d\}$.
- $E = \{(S,S') \in V^2 : |S \cap S'| = d-1\}$.

The number of vertices of $J(n,d)$ is $\binom{n}{d}$. In [2], it has been proved that for all n,d, there is a Hamiltonian path[12] in the Johnson graph $J(n,d)$. Also, the proof implicitly construct an algorithm that finds the Hamiltonian path in $O(n^d)$ time.

Now we explain how to construct the clock. A clock state is an n-qubit state. Each clock state $|\gamma_t\rangle$ corresponds to a vertex S_t on a Hamiltonian path in $J(n,d)$. All the clock states are basis states of the computational basis whose number of 1's equals to d. The ith qubit in $|\gamma_t\rangle$ is $|1\rangle$ if and only if i is chosen into S_t. In other words, when we treat the clock register as

[12] A Hamiltonian path is a path that visits every vertex in a graph exactly once. Please do not confuse with the physical quantity (local Hamiltonian) H.

the set $[n]$ and let $\mathsf{S_t}$ be the portion therein corresponding to S_t, the state in $\mathsf{S_t}$ is $|1^d\rangle$. The next clock $|\gamma_{t+1}\rangle$ corresponds to the vertex S_{t+1} that is adjacent to S_t in the Hamiltonian path. To update the $|\gamma_t\rangle$ to $|\gamma_{t+1}\rangle$, we make $d-1$ of 1's unchanged and flip two bits. Hence the update of the clock state can be done with constant local operation. We define the clock state formally as follows.

▶ **Definition 20** ((n,d)-clock state [14]). *A (n,d)-clock state is a collection of n-qubit quantum state $\{|\gamma_t\rangle\}_{t=0}^{T}$ defined by a Johnson graph $J(n,d)=(V,E)$ and $T=\binom{n}{d}-1$. Let $S_0,S_1,\ldots,S_T\in V$ and the sequence $S_0,S_1,\ldots,S_T\in V$ forms a Hamiltonian path in $J(n,d)$. For any $t\in\{0\}\cup[T]$ the state $|\gamma_t\rangle$ is defined by $|\gamma_t\rangle:=\bigotimes_{i\in[n]}|\mathbb{1}_{S_t}(i)\rangle$.*

To use the clock above for the circuit-to-Hamiltonian construction, we introduce several local operators to pause and move time on the clock. For all $t\in 0\cup[T]$, let $\mathsf{S_t}$ be the corresponding register of S_t. For all $t\in[T]$, we define the operator F_t as follows.

$$F_t:=|1\rangle\langle 0|_{\mathsf{S_{t-1}}\setminus\mathsf{S_t}}\otimes|0\rangle\langle 1|_{\mathsf{S_t}\setminus\mathsf{S_{t-1}}}\otimes|1^{d-1}\rangle\langle 1^{d-1}|_{\mathsf{S_t}\cap\mathsf{S_{t-1}}}. \tag{6}$$

Hence we have

$$F_t^\dagger:=|0\rangle\langle 1|_{\mathsf{S_{t-1}}\setminus\mathsf{S_t}}\otimes|1\rangle\langle 0|_{\mathsf{S_t}\setminus\mathsf{S_t}}\otimes|1^{d-1}\rangle\langle 1^{d-1}|_{\mathsf{S_t}\cap\mathsf{S_{t-1}}}. \tag{7}$$

It holds that

$$F_t\,|\gamma_{t'}\rangle=\delta_{t',t-1}\,|\gamma_{t'+1}\rangle\,, \tag{8}$$

and

$$F_t^\dagger\,|\gamma_{t'}\rangle=\delta_{t',t}\,|\gamma_{t'-1}\rangle\,. \tag{9}$$

That is, F_t "forwards" the clock $|\gamma_{t-1}\rangle$ one step, and eliminate all the other state $|\gamma_{t'}\rangle$ where $t'\neq t-1$. Likewise, $F_t^\dagger$ "backwards" the clock $|\gamma_t\rangle$ one step, and eliminate all the other state $|\gamma_{t'}\rangle$ where $t'\neq t$. We have that F_t is $(d+1)$-local. Note that F_t is neither unitary nor Hermitian.

For $t\in\{0\}\cup[T]$, we define the operator P_t as following.

$$P_t:=|1^d\rangle\langle 1^d|_{\mathsf{S_t}}. \tag{10}$$

We have $P_t^\dagger=P_t$. We also have

$$P_t\,|\gamma_{t'}\rangle=\delta_{t',t}\,|\gamma_{t'}\rangle\,. \tag{11}$$

That is, P_t "pauses" the clock $|\gamma_t\rangle$, and eliminate all the other state $|\gamma_{t'}\rangle$ where $t'\neq t$. The operator P_t is d-local.

We have seen that we can "forward", "backword" and "pause" the clock with local operators. To use the (n,d) clock for the circuit-to-Hamiltonian construction, we also want valid clock states $\{|\gamma_t\rangle\}_{t=0}^{T}$ to be the ground state for the Hamiltonian, and give some energy penalty to any other states orthogonal to valid clock states. In order to achieve this, we define

$$H_{\mathrm{stab}}:=\binom{n}{d}H_{>d}+H_{<d}-\left(\binom{n}{d}-1\right)I, \tag{12}$$

where

$$H_{>d}:=\sum_{\mathsf{S}:|\mathsf{S}|=d+1}|1^{d+1}\rangle\langle 1^{d+1}|_{\mathsf{S}}, \tag{13}$$

$$H_{<d}:=\sum_{\mathsf{S}:|\mathsf{S}|=d}\sum_{x\in\{0,1\}^d\setminus\{1^d\}}|x\rangle\langle x|_{\mathsf{S}}, \tag{14}$$

The term $H_{>d}$ gives penalty to the state that contains "too many" 1's, and the term $H_{<d}$ gives penalty to the state that contains "too few" 1's.

▶ **Lemma 21.** *The linear space for the ground states of H_{stab} is spanned by $\{|\gamma_t\rangle\}_{t=0}^{T}$, and for any state $|\phi\rangle$ orthogonal to all $\{|\gamma_t\rangle\}_{t=0}^{T}$, we have*

$$\langle\phi|H_{\mathrm{stab}}|\phi\rangle \geq 1.$$

Proof. Since H_{stab} is diagonalized in the computational basis i.e., each computational state is an eigenstate of H_{stab}. To verify the above statement, we can check $\langle x|H_{\mathrm{stab}}|x\rangle$ for all $x \in \{0,1\}^n$.

We divide the Hilbert space $\mathcal{H}_{\mathrm{clock}}$ into three subspaces $\mathcal{L}_{=d}$, $\mathcal{L}_{>d}$, and $\mathcal{L}_{<d}$ which are defined below.

$$\mathcal{L}_{<d} := \mathrm{Span}(\{w \in \{0,1\}^n : wt(w) < d\}), \tag{15}$$
$$\mathcal{L}_{=d} := \mathrm{Span}(\{y \in \{0,1\}^n : wt(y) = d\}), \tag{16}$$
$$\mathcal{L}_{>d} := \mathrm{Span}(\{z \in \{0,1\}^n : wt(z) > d\}) \tag{17}$$

Note that $\mathcal{L}_{=d}$ is the subspace spanned by the valid clock states $\{|\gamma_t\rangle\}_{t=0}^{T}$.

We first calculate $H_{>d}$ acting on the states in $\mathcal{L}_{=d}$, $\mathcal{L}_{>d}$, and $\mathcal{L}_{<d}$ respectively. For any S with size $d+1$, $|1^{d+1}\rangle\langle 1^{d+1}|_{\mathsf{S}}\,|x\rangle = |x\rangle$ if the subset-string $x_S = 1^{d+1}$. Hence, we have the following.

- For all $w \in \{0,1\}^n$ such that $wt(w) < d$, it holds that $\langle w|\,H_{>d}\,|w\rangle = 0$ because for all $S \subseteq [n]$ with size $d+1$, z_S has at most $d-1$ of 1's.
- For all $y \in \{0,1\}^n$ such that $wt(y) = d$, it holds that $\langle y|\,H_{>d}\,|y\rangle = 0$ because for all $S \subseteq [n]$ with size $d+1$, w_S has at most d of 1's.
- For all $z \in \{0,1\}^n$ such that $wt(z) > d$, it holds that $\langle z|\,H_{>d}\,|z\rangle = r$ where $r \geq 1$, because there is at least one $S \subseteq [n]$ with size $k+1$ such that $y_S = 1^{d+1}$.

Then we calculate $H_{<d}$ acting on the states in $\mathcal{L}_{=d}$, $\mathcal{L}_{>d}$, and $\mathcal{L}_{<d}$ respectively.

Let $w \in \{0,1\}^n$ such that $wt(w) < d$. For any S with size d, we have $\sum_{x\in\{0,1\}^d\setminus\{1^d\}}|x\rangle\langle x|_{\mathsf{S}}\,|w\rangle = |w\rangle$, because there exist exactly one $x \in \{0,1\}^d \setminus \{1^d\}$ such that $w_S = x$. Hence,

$$\sum_{\mathsf{S}:|\mathsf{S}|=d}\sum_{x\in\{0,1\}^d\setminus\{1^d\}}|x\rangle\langle x|_{\mathsf{S}}\,|z\rangle = \binom{n}{d}|z\rangle\,.$$

Let $y \in \{0,1\}^n$ such that $wt(y) = d$. Let $S' \subseteq [n]$ be the subset with size d such that $w_{S'} = 1^d$. For any S with size d such that $S \neq S'$, we have that $\sum_{x\in\{0,1\}^d\setminus\{1^d\}}|x\rangle\langle x|_{\mathsf{S}}\,|y\rangle = |y\rangle$, and for $S = S'$, we have $\sum_{x\in\{0,1\}^d\setminus\{1^d\}}|x\rangle\langle x|_{\mathsf{S}}\,|y\rangle = 0$; Hence,

$$\sum_{\mathsf{S}:|\mathsf{S}|=d}\sum_{x\in\{0,1\}^d\setminus\{1^d\}}|x\rangle\langle x|_{\mathsf{S}}\,|w\rangle = \left(\binom{n_{cl}}{d}-1\right)|w\rangle\,.$$

For $z \in \{0,1\}^n$ such that $wt(z) > d$, since the penalty has been given by $H_{>d}$ already, $H_{<d}\,|z\rangle = p\,|z\rangle$ where $p \geq 0$ is sufficient for us.

Therefore, we have the follows.

- For all $w \in \{0,1\}^n$ such that $wt(w) < d$, it holds that $H_{<d}\,|w\rangle = \binom{n}{d}|w\rangle$.
- For all $y \in \{0,1\}^n$ such that $wt(y) = d$, it holds that $H_{<d}\,|y\rangle = \left(\binom{n}{d}-1\right)|y\rangle$.
- For all $z \in \{0,1\}^n$ such that $wt(z) > d$, it holds that $H_{<d}\,|z\rangle = p\,|z\rangle$ where $p \geq 0$.

As a result, we have the follows.

- For all $w \in \{0,1\}^n$ such that $wt(w) < d$, it holds that $\langle w| H_{\text{stab}} |w\rangle = 1$.
- For all $y \in \{0,1\}^n$ such that $wt(y) = d$, it holds that $\langle y| H_{\text{stab}} |y\rangle = 0$.
- For all $z \in \{0,1\}^n$ such that $wt(z) > d$, it holds that $\langle z| H_{\text{stab}} |z\rangle \geq 1$.

Consequently, if $|\phi\rangle \in \mathcal{L}_{=d}$, then $\langle\phi|H_{\text{stab}}|\phi\rangle = 0$, and if $|\phi\rangle \in \mathcal{L}_{=d}^{\perp} = \mathcal{L}_{<d} \oplus \mathcal{L}_{>d}$, then $\langle\phi|H_{\text{stab}}|\phi\rangle \geq 1$. ◀

3.3 Our new clock states and proof of Theorem 17

We show how to modify the construction from Section 3.1 to obtain a length-preserving construction. The construction is the same as the construction in Section 3.1, except that we are using a new clock, new clock operators, and a new term H_{clock}.

Proof of Theorem 17. If $d = 1$, the statement follows from the construction of [32] in Section 3.1. For the rest of the proof, we assume $d \geq 2$.

To further reduce the locality, we introduce a new clock to count from 0 to T. We assume (without loss of generality) that T is of the form $T = \binom{a}{d-1}(a+1) - 1$ for some a. We use two clocks, each using a qubits. The first clock is the $(a, d-1)$-clock following Definition 20, and the second clock is the unary clock on a qubits. Parameters are slightly different from the previous section, and not to confuse readers, we clarify the operator of the $(a, d-1)$-clock states.

$$F'_t := |1\rangle\langle 0|_{\mathsf{S}_{t-1}\setminus\mathsf{S}_t} \otimes |0\rangle\langle 1|_{\mathsf{S}_t\setminus\mathsf{S}_{t-1}} \otimes |1^{d-2}\rangle\langle 1^{d-2}|_{\mathsf{S}_t\cap\mathsf{S}_{t-1}}. \tag{18}$$

$$P'_t := |1^{d-1}\rangle\langle 1^{d-1}|_{\mathsf{S}_t}. \tag{19}$$

$$H'_{\text{stab}} := \binom{a}{d-1} H'_{>d-1} + H'_{<d-1} - \left(\binom{a}{d-1} - 1\right) I, \tag{20}$$

where

$$H'_{>d-1} := \sum_{\mathsf{S}:|\mathsf{S}|=d} |1^d\rangle\langle 1^d|_{\mathsf{S}}, \tag{21}$$

$$H'_{<d-1} := \sum_{\mathsf{S}:|\mathsf{S}|=d-1} \sum_{x\in\{0,1\}^{d-1}\setminus\{1^{d-1}\}} |x\rangle\langle x|_{\mathsf{S}}, \tag{22}$$

The whole clock thus requires only $2a \leq 2T^{1/d}$ qubits. To represent the order of the $T+1$ representations of the clock, we define a map

$$g\colon \{0,\dots,T\} \to \{1,\dots,\binom{a}{d-1}\} \times \{0,\dots,a\}$$

as $g(t) = (1 + \lfloor t/(a+1)\rfloor, \min(m, 2a+1-m))$ where $m = t \bmod (2a+2)$. We implicitly use g to interpret time steps $t \in \{0,\dots,T\}$ as pairs $(t_1,t_2) \in \{1,\dots,\binom{a}{d}\} \times \{0,\dots,a\}$. We explain the construction of g for $a = 3, d = 1$ in Table 3.

Our implementation of the clock operators is described in Table 4.

The term H_{clock}. The term H_{clock} is defined as follows:

$$H_{\text{clock}} = H'_{\text{stab}} \otimes I + \sum_{0\leq i<j\leq a} I \otimes |01\rangle_{ij}\langle 01|_{ij},$$

where in term, the left part acts on the register corresponding to the first clock and the right part acts on the qubits corresponding to the second clock.

Table 3 Our clock for $T = 11$ (i.e., $a = 3$). The part in red represents the first clock, while the part in blue represents the second clock. The first clock uses the encoding 100, 010, 001. The second clock uses the encoding 000, 100, 110, 111.

t	$g(t) = (t_1, t_2)$	clock state
$t = 0$	$(1, 0)$	$\|100\rangle \otimes \|000\rangle$
$t = 1$	$(1, 1)$	$\|100\rangle \otimes \|100\rangle$
$t = 2$	$(1, 2)$	$\|100\rangle \otimes \|110\rangle$
$t = 3$	$(1, 3)$	$\|100\rangle \otimes \|111\rangle$
$t = 4$	$(2, 3)$	$\|010\rangle \otimes \|111\rangle$
$t = 5$	$(2, 2)$	$\|010\rangle \otimes \|110\rangle$
$t = 6$	$(2, 1)$	$\|010\rangle \otimes \|100\rangle$
$t = 7$	$(2, 0)$	$\|010\rangle \otimes \|000\rangle$
$t = 8$	$(3, 0)$	$\|001\rangle \otimes \|000\rangle$
$t = 9$	$(3, 1)$	$\|001\rangle \otimes \|100\rangle$
$t = 10$	$(3, 2)$	$\|001\rangle \otimes \|110\rangle$
$t = 11$	$(3, 3)$	$\|001\rangle \otimes \|111\rangle$

Table 4 The implementation of clock operations. The part in red acts on the qubits corresponding to the first clock and the part in blue acts on the qubits corresponding to the second clock.

Operator	t_1	t_2	Implementation	Locality
$\mathcal{S}_t$	any	$t_2 = 0$	$P'_{t_1} \otimes \|0\rangle_1 \langle 0\|_1$	d
		$t_2 \in \{1, \ldots, a-1\}$	$P'_{t_1} \otimes \|10\rangle_{t_2,t_2+1} \langle 10\|_{t_2,t_2+1}$	$d+1$
		$t_2 = a$	$P'_{t_1} \otimes \|1\rangle_a \langle 1\|_a$	d
$\mathcal{A}_t$	odd	$t_2 \in \{0, \ldots, a-1\}$	$P'_{t_1} \otimes \|1\rangle_{t_2+1} \langle 0\|_{t_2+1}$	d
		$t_2 = a$ (if $t_1 \neq \binom{a}{d-1}$)	$F'_{t_1+1} \otimes \|1\rangle_a \langle 1\|_a$	$d+1$
	even	$t_2 \in \{1, \ldots, a\}$	$P'_{t_1} \otimes \|0\rangle_{t_2} \langle 1\|_{t_2}$	d
		$t_2 = 0$	$F'_{t_1+1} \otimes \|0\rangle_1 \langle 0\|_1$	$d+1$
$\mathcal{B}_t$	odd	$t_2 \in \{0, \ldots, a-2\}$	$P'_{t_1} \otimes \|11\rangle_{t_2+1,t_2+2} \langle 00\|_{t_2+1,t_2+2}$	$d+1$
		$t_2 = a-1$ (if $t_1 \neq a$)	$F'_{t_1+1} \otimes \|0\rangle_a \langle 1\|_a$	$d+1$
		$t_2 = a$ (if $t_1 \neq a$)	$F'_{t_1+1} \otimes \|1\rangle_a \langle 0\|_a$	$d+1$
	even	$t_2 \in \{2, \ldots, a-2\}$	$P'_{t_1} \otimes \|00\rangle_{t_2-1,t_2} \langle 11\|_{t_2-1,t_2}$	$d+1$
		$t_2 = 1$	$F'_{t_1+1} \otimes \|0\rangle_1 \langle 1\|_1$	$d+1$
		$t_2 = 0$	$F'_{t_1+1} \otimes \|1\rangle_1 \langle 0\|_1$	$d+1$

Analysis of the whole Hamiltonian. Without loss of generality, we assume that

$$U_t = I \text{ for any } t \text{ of the form } t = (t_1, 0) \text{ with } t_1 \text{ even or } t = (t_1, a) \text{ with } t_1 \text{ odd.} \qquad (\star)$$

The assumption can be fulfilled by adding the dummy gates I at such time steps. We add those time steps such that $U_t = I$ to T_A, i.e., we consider the gates $U_t = I$ as 1-qubit gates. The locality of each term of the whole Hamiltonian is represented in the following table (note that we crucially use Assumption $(\star)$ to guarantee that the locality of $H_{\text{prop},t}$ and $H_{\text{qubit},t}$ is $d+1$):

Term	Locality
H_{in}	$1+d = d+1$
H_{out}	$1+d = d+1$
$H_{\text{prop},t}$	$\max\{d+1, 1+d\} = d+1$
$H_{\text{qubit},t}$	$1+d = d+1$
$H_{\text{time},t}$	$\max\{d, d+1\} = d+1$
H_{clock}	d

The whole Hamiltonian is thus $(d+1)$-local. We can easily check that Conditions $(\text{C}_1) \sim (\text{C}_6)$ are satisfied. From Proposition 18, we conclude that the circuit-to-Hamiltonian construction is correct. From the condition $(\star)$, T is $\binom{a}{d-1}a$. Therefore, we can take $a = O(T^{1/d})$ and the number of qubits H_x acting on is $N + O(T^{1/d})$. ◀

4 Fine-grained hardness for local Hamiltonian problem

In this section, we are going to prove the lower bound for the 3-LH problem.

▶ **Theorem 22** (Lower bound for 3-LH). *Assume that SETH (resp. QSETH) holds. Then, for any constant $\xi > 0$, for any classical (resp. quantum) algorithm $\mathcal{A}_{LH}$, there exists a 3-local Hamiltonian H acting on n_H qubits associated with $E_{\text{yes}}, E_{\text{no}}$ satisfying $E_{\text{no}} - E_{\text{yes}} \geq \Omega(1)$ such that $\mathcal{A}_{\text{LH}}$ cannot decide $\lambda(H) \geq E_{\text{no}}$ or $\lambda(H) \leq E_{\text{yes}}$ with probability greater than $2/3$ in $O(2^{n_H(1-\xi)})$ (resp. $O(2^{n_H(1-\xi)/2})$) time.*

We will prove Theorem 22 in Section 4.1. To prove it, we will use the size-preserving circuit-to-Hamiltonian construction (Theorem 17) associated with the following lemma.

▶ **Lemma 23** (A quantum circuit can compute k-CNF). *For any positive integer k and $c > 0$, for all $n \in \mathbb{N}$, for any k-CNF formula Φ that contains n variables and $m = O(n^c)$ clauses, there exists a quantum circuit U_Φ that acts on n input qubits and at most $2c\log n + 2$ ancilla qubits, and U_Φ consists of at most $34c^2n^c\log^2 n + (70k+2)n^c + 35c\log n$ elementary gates such that $U_\Phi |x\rangle_{\mathsf{in}} \otimes |0\rangle_{\mathsf{anc}} = |\Phi(x)\rangle_{\mathsf{out}} \otimes |\psi_x\rangle_{\overline{\mathsf{out}}}$ for all $x \in \{0,1\}^n$, where $|\psi_x\rangle$ is some quantum state depending on x. The construction of U_Φ can be done in $\text{poly}(n)$ time.*[13]

We defer the proof of Lemma 23 to Section 4.2.

[13] It is worth noting that an alternative quantum circuit for computing kCNF appears in [27], Corollary 10. That construction has width $k+2\big(\lceil \log n\rceil + \lceil c\log n\rceil\big)$ and size at most $4\times 3^{\lceil \log_2 n\rceil + \lceil c\log_2 n\rceil - 1}$. However, its circuit size is $O(n^{c+1})$, whereas ours is $O(n^c \log^2 n)$. This gap impacts the locality of the resulting Hamiltonian, which is why Lemma 23 is necessary for our results.

4.1 SETH- and QSETH-hardness of the 3-local Hamiltonian problem

Proof of Theorem 22. Let $\Phi = \varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_m$ be a kCNF formula defined on n variables $x_1, x_2, \ldots, x_n$, where $m = O(n^c)$ and $c \in [1, 2)$. We construct an algorithm $\mathcal{A}_{\mathrm{SAT}}$ that solves k-SAT(Φ) by using $\mathcal{A}_{\mathrm{LH}}$ as a subroutine.

1. Upon receiving a k-SAT instance Φ, construct a quantum circuit U_Φ from Lemma 23.
2. Construct a Hamiltonian H_{U_Φ} acting on $n_H = n + o(n)$ qubits from U_Φ, and obtain two energy thresholds $E_{\mathrm{yes}}, E_{\mathrm{no}}$ where $E_{\mathrm{no}} - E_{\mathrm{yes}} = \Omega(1)$ by Theorem 17.
3. Run $\mathcal{A}_{\mathrm{LH}}$ on the input H_{U_Φ}. If $\lambda(H_{U_\Phi}) \leq E_{\mathrm{yes}}$, then return YES. If $\lambda(H_{U_\Phi}) \geq E_{\mathrm{no}}$, then return NO.

The running time of $\mathcal{A}_{\mathrm{SAT}}$ is poly(n).

We now show the correctness of $\mathcal{A}_{\mathrm{SAT}}$. By Lemma 23, we have the following:

- If there exists an assignment $x \in \{0,1\}^n$ such that $\Phi(x) = 1$, then $\Pr[1 \leftarrow U_\Phi(|x\rangle)] = 1$.
- If $\Phi(x) \neq 1$ for all assignments $x \in \{0,1\}^n$, then $\Pr[1 \leftarrow U_\Phi(|x\rangle)] = 0$ for all $x \in \{0,1\}^n$.

Combining Theorem 17 and choosing $\varepsilon_{\mathrm{err}} < \frac{1}{4}$, we obtain:

- $||H_{U_\Phi}|| \leq \mathrm{poly}(n_H)$.
- If there exists $x \in \{0,1\}^n$ such that $\Phi(x) = 1$, then $\lambda(H_{U_\Phi}) \leq E_{\mathrm{yes}}$.
- If $\Phi(x) \neq 1$ for all $x \in \{0,1\}^n$, then $\lambda(H_{U_\Phi}) \geq E_{\mathrm{no}}$.
- $E_{\mathrm{no}} - E_{\mathrm{yes}} = \Omega(1)$

Therefore, $\mathcal{A}_{\mathrm{SAT}}$ has the same success probability as $\mathcal{A}_{\mathrm{LH}}$.

Next, we show that $n_H = n + o(n)$ and that H_{U_Φ} is 3-local. Let n_a and g denote the number of ancilla qubits and elementary gates in U_Φ, respectively. By Lemma 23, $n_a = o(n)$, and $g \leq 34c^2 n^c \log^2 n + (70k + 2)n^c + 35c \log n$. For all sufficiently large n, the gate number g is upper-bounded by $35c^2 n^{c'}$, where $c' \in (c, 2)$ is a constant.

By Theorem 17, we can apply $d = 2$ and the number of qubits of H_{U_Φ} is $n_H = n + O(\log n) + o(n) = n + o(n)$ and H_{U_Φ} is 3-local. Furthermore, we have that the threshold gap $E_{\mathrm{no}} - E_{\mathrm{yes}} = \Omega(1)$.

Because $n_H = n + o(n)$, we have $n_H \leq (1 + \eta)n$ where $\eta > 0$ is an arbitrary small constant for all sufficiently large n. For any constant $\xi > 0$, we can take $\xi > \eta > 0$ and a constant $\varepsilon = \xi - \eta + \xi\eta$ such that $\varepsilon > 0$, and it holds that $O(2^{n(1+\eta)(1-\xi)}) \leq O(2^{n(1-\varepsilon)})$ (and $O(2^{\frac{n}{2}(1+\eta)(1-\xi)}) \leq O(2^{\frac{n}{2}(1-\varepsilon)})$).

Assume there exists a classical (resp. quantum) algorithm $\mathcal{A}_{\mathrm{LH}}$ that solves 3-LH($H, E_{\mathrm{yes}}, E_{\mathrm{no}}$) for $E_{\mathrm{no}} - E_{\mathrm{yes}} \geq \Omega(1)$ in $O(2^{n_H(1-\xi)})$ (resp. $O(2^{\frac{n_H}{2}(1-\xi)})$) time for a constant $\xi > 0$. Then we have the classical (resp. quantum) algorithm $\mathcal{A}_{\mathrm{SAT}}$ solves k-SAT defined on n variables in $O(2^{n(1-\varepsilon)})$ (resp. $O(2^{\frac{n}{2}(1-\varepsilon)})$) time for a constant $\varepsilon > 0$. Besides, the reduction is polynomial time. As a result, it violates SETH which is defined in Conjecture 14 (resp. QSETH which is defined in Conjecture 15). By taking a contraposition, we have the statement. ◀

4.2 Efficient verification circuit for SAT

In this section, we show the construction of the quantum circuit U_Φ that computes the formula $\Phi = \varphi_1 \wedge \cdots \wedge \varphi_m$ defined on n variables $x_1, x_2, \ldots x_n$, and $m = O(n^c)$. For each $i \in [m]$, the clause $\varphi_i = \ell_{i,1} \vee \ell_{i,2} \vee \cdots \vee \ell_{i,k_i}$ where for each $p \in [k_i]$, $\ell_{i,p}$ can be either x_j or $\neg x_j$ where $j \in [n]$, and $k_i \leq k$. Let $r := \lceil \log m \rceil$.

Proof of Lemma 23. The ideal is to compute φ_i for each $i \in [m]$ sequentially. We initially set a counter, and increment the counter by one if $\varphi_i(x) = 1$. After compute all φ_i, we check whether the counter is equal to m. Let k_i be the number of variables (x_j or $\neg x_j$) appearing

in φ_i. We have $\Phi(x) = 1$ if and only if the counter equals m. We introduce the ancilla register $\mathsf{anc} = \mathsf{cls} \cup \mathsf{cnt} \cup \mathsf{out}$ where cls is a one-qubit register that temporarily stores the value of $\varphi_i(x)$ for each i, cnt is an r-qubit ($r = \lceil \log m \rceil$) that serves as the counter, and out is a one-qubit register that outputs $\Phi(x)$.

To compute each φ_i, we construct W_i acting on $\mathsf{in} \cup \mathsf{cls}$ such that $W_i |x\rangle_{\mathsf{in}} |0\rangle_{\mathsf{cls}} = |\psi_x\rangle_{\mathsf{in}} |\varphi_i\rangle_{\mathsf{cls}}$, where $|\psi_x\rangle$ is some state depending on x.

To increment the counter by one, we construct a unitary ADDONE acting on cnt such that $\text{ADDONE} |y\rangle_{\mathsf{cnt}} = |bin(int(y) + 1)\rangle_{\mathsf{cnt}}$ for all $y \in \{0,1\}^r \setminus \{1^r\}$. We apply ADDONE only when $\varphi_i(x) = 1$. This is done by letting ADDONE be controlled by the resister cls. Denote the control-ADDONE operator by $C\text{ADDONE}$. Then $C\text{ADDONE}_{\mathsf{cls}\cup\mathsf{cnt}} |0\rangle_{\mathsf{cls}} |y\rangle_{\mathsf{cnt}} = |0\rangle_{\mathsf{cls}} |y\rangle_{\mathsf{cnt}}$ and $C\text{ADDONE}_{\mathsf{cls}\cup\mathsf{cnt}} |1\rangle_{\mathsf{cls}} |y\rangle_{\mathsf{cnt}} = |0\rangle_{\mathsf{cls}} |bin(int(y) + 1)\rangle_{\mathsf{cnt}}$.

After calculating $\varphi_i(x)$ and applying $C\text{ADDONE}$, we apply $W_i^\dagger$ on $\mathsf{in} \cup \mathsf{cls}$ to restore the state to $|x\rangle_{\mathsf{in}} |0\rangle_{\mathsf{cls}}$.

To check whether the counter is equal to m in the final step, we construct a compare operator, denoted by COMPARE, that acts on $\mathsf{cnt} \cup \mathsf{out}$. The operator COMPARE satisfies $\text{COMPARE} |y\rangle_{\mathsf{cnt}} |0\rangle_{\mathsf{out}} = |y\rangle_{\mathsf{cnt}} |\delta_{m,int(y)}\rangle_{\mathsf{out}}$ for all $y \in \{0,1\}^r$.

To sum up, we construct the quantum circuit U_Φ as follows:

$$U_\Phi := \text{COMPARE}(W_m^\dagger C\text{ADDONE} W_m)(W_{m-1}^\dagger C\text{ADDONE} W_{m-1}) \cdots (W_1^\dagger C\text{ADDONE} W_1). \quad (23)$$

Next, we explain how to implement W_i, ADDONE, and COMPARE gates.

Because $\ell_{i_1} \vee \ell_{i_2} \vee \cdots \vee \ell_{i_{k_i}} = \neg(\neg\ell_{i_1} \wedge \neg\ell_{i_2} \wedge \cdots \wedge \neg\ell_{i_{k_i}})$, we can use $C^k\text{NOT}$ together with NOT gates to implement W_i. Let $\mathsf{S_i} \subseteq \mathsf{in}$ be the register defined by $\mathsf{S_i} := \{j \in [n] : x_j \in \varphi_i \text{ or } \neg x_j \in \varphi_i\}$, i.e., $\mathsf{S_i}$ consists of the jth qubits in in such that x_j or $\neg x_j$ in φ_i; and let $\mathsf{R_i} \subseteq \mathsf{in}$ be the register defined by $\mathsf{R_i} := \{j \in [n] : x_j \in \varphi_i\}$, i.e., $\mathsf{R_i}$ consists of the jth qubits in in such that x_j in φ_i, but $\neg x_j$ does not. We construct W_i by $W_i := \text{NOT} \cdot_{\mathsf{cls}} C^{k_i}\text{NOT}_{\mathsf{S_i}\cup\mathsf{cls}} \cdot \text{NOT}_{\mathsf{R_i}}^{\otimes|\mathsf{R_i}|}$.

To implement the ADDONE, observe that for $y \in \{0,1\}^r$, when $int(y)$ is incremented by one, a bit $y[p]$ will be flipped if and only if all the bits with order lower than p are 1's. That is, $y[p]$ will be flipped if and only if for all $p' > p$, $y[p'] = 1$. Hence, ADDONE can be composed of a sequence of multi-control-NOT gates. The qth layer of ADDONE is a $C^{r-q}\text{NOT}$ gate whose control qubits are $q, q+1, \ldots, r$ and whose target is the $q$the qubit. The last layer of ADDONE is a NOT gate acting on the last qubit. To implement the control-ADDONE operation $C\text{ADDONE}$, we let every $C^{r-q}\text{NOT}$ be controlled by register cls. In other words, the qth layer of $C\text{ADDONE}$ is a $C^{r+1-q}\text{NOT}$ gates whose control qubits are $q, q+1, \ldots, r \cup \mathsf{cls}$ and whose target is the qth qubit in cnt.

The COMPARE operator checks whether the value stored in the counter equal to m. That is, COMPARE operator compares each bit in the counter with $bin(m)$, which can be implemented by a $C^r\text{NOT}$ gate. Let $\mathsf{P} \subseteq \mathsf{cnt}$ be defined by $\mathsf{P} := \{j \in [r] : bin(m)[j] = 0\}$. We construct COMPARE by $\text{COMPARE} := C^r\text{NOT}_{\mathsf{cnt}\cup\mathsf{out}} \cdot \text{NOT}_{\mathsf{P}}^{\otimes|P|}$.

We decompose each multi-control-NOT gate into Toffoli gates. For W_i (and $W_i^\dagger$), there is a $C^{k_i}\text{NOT}$ inside, and it can be decomposed into at most $2k$ of Toffoli gates using at most k ancilla bits.

The largest gate in the $C\text{ADDONE}$ is a control $C^r\text{NOT}$ gate, and there are r layers. Hence, $C\text{ADDONE}$ can be decomposed into at most $2r^2$ Toffoli gates. Because the ancillas for the multi-control-NOT gates can be reused, the ancillas required for $C\text{ADDONE}$ is at most r.

The COMPARE operation contains a $C^r\text{NOT}$ gate, which can be decomposed into at most $2r$ Toffoli gates using at most r ancilla qubits.

The ancilla qubits can be reused for the multi-control-NOT gates. Hence, to decomposed U_Φ into Toffoli gates, the number of ancilla qubits required is at most r. Therefore, the total number of ancilla qubits for U_Φ is $|\mathsf{anc}| + r = |\mathsf{cls}| + |\mathsf{cnt}| + |\mathsf{out}| + r = 2\log m + 2 = 2c \log n + 2$.

The total number of Toffoli gates in U_Φ is at most $(4k+2r^2)\cdot m + 2r = 2c^2n^c\log^2 n + 4kn^c + 2c\log n$. We further decompose the Toffoli gates into elementary gates, yielding $34c^2n^c\log^2 n + 68kn^c + 34c\log n$ elementary gates in total. In addition to Toffoli gates, there are at most $2(k+1)\cdot m + r = (2k+2)n^c + c\log n$ of NOT gates. Therefore, the total number of gates is at most $34c^2n^cc^2\log^2 n + (70k+2)n^c + 35c\log n$. ◀

5 Fine-grained complexity of quantum partition function problem

The fine-grain lower bound for the local Hamiltonian problem directly gives us the fine-grain lower bound for the quantum partition function.

5.1 SETH- and QSETH-hardness of quantum partition functions

▶ **Lemma 24** (Fine-grained reduction from k-LH to k-QPF). *For any $T(n) \in \omega(\mathrm{poly}(n))$, k-LH$(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ is $(T(n), T(n))$ reducible to k-QPF(H, β, δ) in which H acts on n qubits, $\beta \geq \frac{n}{E_{\mathrm{no}}-E_{\mathrm{yes}}}$ and δ satisfies that $\frac{1-\delta}{1+\delta} \geq e^{-0.3n}$.*

We emphasize that in Lemma 24, H in the k-QPF problem and the k-LH problem is the same Hamiltonian.

To better understand Lemma 24, we unpack the underlying fine-grained reduction as follows. Suppose there exist $\xi > 0$ and an algorithm $\mathcal{A}_{\mathrm{QPF}}$ that approximate $\mathrm{tr}[e^{-\beta H}]$ up to relative error δ satisfying $\frac{1-\delta}{1+\delta} \geq e^{-0.3n}$ with probability greater than $2/3$ in $O(T(n)^{1-\xi})$ time, then for any $\varepsilon > 0$, we can use $\mathcal{A}_{\mathrm{QPF}}$ to construct an algorithm $\mathcal{A}_{\mathrm{LH}}$ such that given thresholds $E_{\mathrm{yes}}, E_{\mathrm{no}}$ satisfying $E_{\mathrm{no}} - E_{\mathrm{yes}} \geq n/\beta$, the algorithm $\mathcal{A}_{\mathrm{LH}}$ decides weather $\lambda(H) \geq E_{\mathrm{no}}$ or $\lambda(H) \leq E_{\mathrm{yes}}$ with probability greater than $2/3$ in $O(T(n)^{1-\varepsilon})$ time.

On the other hand, suppose for all $\varepsilon > 0$, for any algorithm $\mathcal{A}_{\mathrm{LH}}$, for infinitely many n there exists a k-local Hamiltonian H_n acting on n qubits and two energy thresholds $E_{\mathrm{yes}}, E_{\mathrm{no}}$ such that $\mathcal{A}_{\mathrm{LH}}$ cannot solve k-LH$(H_n, E_{\mathrm{yes}}, E_{\mathrm{no}})$ with probability greater than $\frac{2}{3}$ in $O(T(n)^{1-\varepsilon})$ time, then for all $\xi > 0$, for all $\delta > 0$, for infinitely many $n \geq n_1$, any algorithm $\mathcal{A}_{\mathrm{QPF}}$ cannot solve k-QPF(H_n, β, δ), where $\beta \geq \frac{n}{E_{\mathrm{no}}-E_{\mathrm{yes}}}$, with probability greater than $2/3$ in $O(T(n)^{1-\xi})$ time. The integer n_1 satisfies that $\frac{1-\delta}{1+\delta} = e^{-0.3n_1}$.

Proof of Lemma 24. Let $(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ be the k-LH instance, where H is a k-local Hamiltonian acts on n qubits . We are going to construct an algorithm $\mathcal{A}_{\mathrm{LH}}$ that solves k-LH$(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ by using $\mathcal{A}_{\mathrm{QPF}}$ as a subroutine.

If $(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ is YES case, then there is at least one eigenstate of H whose energy is lower than or equal to E_{yes}. Hence $Z(\beta) \geq e^{-\beta E_{\mathrm{yes}}}$.

If $(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ is NO case, then all 2^n number of eigenstates of H have energy higher than or equal to E_{no}. Hence, $Z(\beta) \leq 2^n e^{-\beta E_{\mathrm{no}}} < e^{-\beta E_{\mathrm{no}}+0.7n}$.

When $\beta \geq \frac{n}{E_{\mathrm{no}}-E_{\mathrm{yes}}} \in \mathrm{poly}(n)$, and δ satisfies $\frac{1-\delta}{1+\delta} \geq e^{-0.3n}$, it holds that

$$\frac{1-\delta}{1+\delta} \geq e^{-0.3n} \geq e^{-\beta(E_{\mathrm{no}}-E_{\mathrm{yes}})+0.7n}.$$

Hence we have $(1-\delta)e^{-\beta E_{\mathrm{yes}}} \geq (1+\delta)e^{-\beta E_{\mathrm{no}}+0.7n}$.

Now we present the algorithm $\mathcal{A}_{\mathrm{LH}}$ that solves k-LH$(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ by using $\mathcal{A}_{\mathrm{QPF}}$.

1. When receiving $H, E_{\mathrm{yes}}, E_{\mathrm{no}}$, calculate δ_0 such that $\frac{1-\delta_0}{1+\delta_0} = e^{-0.3n}$ and $\beta_0 = \frac{n}{E_{\mathrm{no}}-E_{\mathrm{yes}}}$. Set $\beta \geq \beta_0$ and set $\delta \leq \delta_0$.
2. Run $\mathcal{A}_{\mathrm{QPF}}$ on the input H, β, δ, and get the output $\widetilde{Z}$.
3. If $\widetilde{Z} \geq (1-\delta)e^{-\beta E_{\mathrm{yes}}}$, then output YES. If $\widetilde{Z} \leq (1+\delta)e^{-\beta E_{\mathrm{no}}+0.7n}$, then output NO.

When $\mathcal{A}_{\mathrm{QPF}}$ solves k-QPF(H,β,δ) successfully, it is guaranteed that $(1-\delta)Z(\beta) \leq \widetilde{Z} \leq (1+\delta)Z(\beta)$. When the inputs of k-LH is YES case, $\widetilde{Z} \geq (1-\delta)Z(\beta) \geq (1-\delta)e^{-\beta E_{\mathrm{yes}}}$; and when NO case, $\widetilde{Z} \leq (1+\delta)Z(\beta) < (1+\delta)e^{-\beta E_{\mathrm{no}}+0.7n}$.

By the choice of β and δ, it holds that $(1-\delta)e^{-\beta E_{\mathrm{yes}}} \geq (1+\delta)e^{-\beta E_{\mathrm{no}}+0.7n}$. Therefore, when $\mathcal{A}_{\mathrm{QPF}}$ solves k-QPF(H,β,δ) successfully, $\mathcal{A}_{\mathrm{LH}}$ decides k-LH$(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ correctly. The success probability of $\mathcal{A}_{\mathrm{LH}}$ is the same as $\mathcal{A}_{\mathrm{QPF}}$. Therefore, the requirement 1 in Definition 16 is satisfied.

The algorithm $\mathcal{A}_{\mathrm{LH}}$ queries $\mathcal{A}_{\mathrm{QPF}}$ once in Step 2., and the instance of k-QPF is exactly the input of k-LH. For any ε, we choose $\xi = \varepsilon$. We have $T(n)^{1-\varepsilon} \leq d \cdot T(n)^{1-\xi}$ for any constant $d > 1$. Therefore, the requirement 3 and 4 in Definition 16 are satisfied.

Finally, Step 1. and Step 3. in the algorithm $\mathcal{A}_{\mathrm{LH}}$ run in poly(n) time. Hence, $\mathcal{A}_{\mathrm{LH}}$ runs in poly(n) times, which is less than $d \cdot T(n)^{1-\xi}$ for some constant d because $T(n)$ is super-polynomial. consequently, the requirement 2 is satisfied. This finishes the proof. ◀

▶ Remark 25. The reduction in Lemma 24 is polynomial-time. Since k-LH is QMA-hard, we have that k-QPF is also QMA-hard.

The fine-grain reduction above implies the lower bound for the quantum partition function problem.

▶ **Theorem 26.** *Assume SETH (resp. QSETH) holds. For any $\xi > 0$ and any $\delta \in (0,1)$, for any quantum algorithm $\mathcal{A}_{\mathrm{QPF}}$, for infinitely many n, there exists a 3-local Hamiltonian H acting on n qubits, associated with an inverse temperature $\beta_0 = O(n)$, such that for all $\beta \geq \beta_0$, the algorithm $\mathcal{A}_{\mathrm{QPF}}$ cannot solve* QPF(H,β,δ) *in $O(2^{n(1-\xi)})$ classical time (resp. $O(2^{\frac{n}{2}(1-\xi)})$ quantum time) with probability greater than $2/3$.*

Proof. We prove the theorem by contradiction. Assume that there exists an algorithm $\mathcal{A}_{\mathrm{QPF}}$ that solves the QPF problem. Then, by Lemma 24, we can construct an algorithm $\mathcal{A}_{\mathrm{LH}}$ that solves 3-local Hamiltonian problem, which contradicts Theorem 22.

Suppose there exist $\xi > 0$ and $\delta \in (0,1)$ such that there exist an algorithm $\mathcal{A}_{\mathrm{QPF}}$ and $n_1 \in \mathbb{N}$ satisfying the following: for all $n > n_1$, for all H acting on n qubits, and for arbitrarily large $\beta \geq O(n)$, the algorithm $\mathcal{A}_{\mathrm{QPF}}$ solves QPF(H,β,δ) with probability greater than $2/3$ in $O(2^{n(1-\xi)})$ classical time.

Let $(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ be an LH instance, where H is a 3-local Hamiltonian acting on n qubits with sufficiently large n (we will specify how large n shall be later), and $E_{\mathrm{no}} - E_{\mathrm{yes}} = \Omega(1)$. Choose β such that $\beta \geq O(n) > \frac{n}{E_{\mathrm{no}} - E_{\mathrm{yes}}}$. By Lemma 24, we construct $\mathcal{A}_{\mathrm{LH}}$, in which we query $\mathcal{A}_{\mathrm{QPF}}$ on the instance (H,β,δ). The algorithm runs in $O(2^{n(1-\xi)})$ time.

Let n_2 be the smallest integer such that $\frac{1-\delta}{1+\delta} \geq e^{-0.3n_2}$. By the hypothesis at the beginning of the proof and by Lemma 24, when $n \geq \max\{n_1, n_2\}$, the algorithm $\mathcal{A}_{\mathrm{LH}}$ decides $(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ successfully with probability greater than $2/3$.

Note that the reduction works for all H acting on n qubits and $E_{\mathrm{yes}}, E_{\mathrm{no}}$ satisfying $E_{\mathrm{no}} - E_{\mathrm{yes}} = \Omega(1)$ as long as $n \geq \max\{n_1, n_2\}$. We thus obtain a contradiction with Theorem 22. The lower bound of quantum algorithms assuming QSETH follows in the same proof strategy. ◀

5.2 $O^*(2^{\frac{n}{2}})$-time quantum algorithm for k-QPF problem

In this section, we propose an algorithm that estimates the quantum partition function under a polynomial inverse temperature up to an arbitrary inverse polynomial error in $O^*(2^{\frac{n}{2}})$ time. The running time matches our conditional lower bound. The input Hamiltonian H is n-qubit, constant-local, semi-positive and satisfies $\|H\| < 1$.

▶ **Theorem 27** ($O^*(2^{\frac{n}{2}})$ time algorithm for k-QPF). *For all sufficiently large $n \in \mathbb{N}$, for all n-qubit constant-local, semi-definite Hamiltonian H satisfying $\|H\| < 1$, $\beta < \mathrm{poly}(n)$, and $\delta \geq 1/n^c$ for arbitrary positive constant c, there exists a quantum algorithm that solves k-QPF(H, β, δ) in $O^*(2^{\frac{n}{2}})$ time with successful probability $1 - \mathrm{negl}(n)$.*

Tools that are used in the proof of QPF algorithm

Before proving Theorem 27, we first list some tools that will be used in the proof.

The main idea of the algorithm is dividing the energy range $[0, 1)$ into polynomial many disjoint intervals $I_0, I_1, \ldots, I_L$. For each interval I_ℓ, we designate a value $\mathcal{E}_\ell \in I_\ell$ to represent the characteristic energy of I_ℓ. Let M_ℓ be the number of the Hamiltonian's eigenstates whose corresponding energy is in the interval I_ℓ. The approximation of the quantum partition function is

$$\sum_{\ell=0}^{L} M_\ell e^{-\beta \mathcal{E}_\ell}.$$

To estimate the number of eigenstates M_ℓ, there are two key ingredients: (i) energy estimation, and (ii) quantum counting.

Given the description of a local Hamiltonian H and one of H's eigenstate $|\psi_j\rangle$, the energy estimation algorithm outputs its corresponding energy E_j. The energy estimation algorithm can be implemented by phase estimation associated with Hamiltonian simulation.

▶ **Lemma 28** (Phase estimation [44]). *Let U be a n-qubit unitary and $|u_\theta\rangle$ be U's eigenstate satisfying that $U|u_\theta\rangle = e^{i\theta}|u_\theta\rangle$. There exists a quantum circuit $P_{U,r}$, which takes $|u_\theta\rangle$ as the input state and executes control-U operator $O(2^r)$ times with additional $\mathrm{poly}(n)$ number of gates, such that the output state is*

$$|u_\theta\rangle \otimes \sum_{x \in \{0,1\}^r} \alpha_x |x\rangle, \tag{24}$$

where $\alpha \in \mathbb{C}$, satisfying that

$$\sum_{x : |\frac{int(x)\cdot 2\pi}{2^r} - \theta| \leq \frac{2\pi}{2^b}} |\alpha_x|^2 \geq 1 - \frac{1}{2^{r-b}} \tag{25}$$

for any $b < r - 1$.

When we choose $b = r - 2$, we have that the estimation $\widetilde{\theta} := \frac{int(x)\cdot 2\pi}{2^r}$ is in the interval $\theta \pm \frac{2\pi}{2^b}$ with probability greater than $\frac{3}{4}$. We can amplify the probability to $1 - \mathrm{negl}(n)$ by repeating the procedure $\mathrm{poly}(n)$ times and using median of means technique [4, 10].

▶ **Lemma 29** (Confidence amplification of phase estimation). *Following Lemma 28, there exists a quantum circuit $A_{U,r,m}$, which executes $P_{U,r}$ $m = \mathrm{poly}(n)$ times, with additional $\mathrm{poly}(n)$ gates, such that the output state is*

$$|u_\theta\rangle \otimes \sum_{x \in \{0,1\}^r} \alpha'_x |x\rangle |g_x\rangle, \tag{26}$$

where $\alpha' \in \mathbb{C}$, and $|g_x\rangle$ is some state depending on x, satisfying

$$\sum_{x : |\frac{int(x)\cdot 2\pi}{2^r} - \theta| \leq \frac{2\pi}{2^{(r-2)}}} |\alpha'_x|^2 \geq 1 - e^{-\Theta(m)}. \tag{27}$$

Proof. Because after the execution of the phase estimation $P_{U,r}$, the input state $|u_\theta\rangle$ remains unchanged and separable from $\sum_x \alpha_x |x\rangle$, we can reuse the input state. Assume we measure the register that contains $\sum_x \alpha_x |x\rangle$ after each execution of $P_{U,r}$. Let the outcome of each execution be $x_1, x_2, \ldots, x_m$. By Lemma 28, when we choose $b = r-2$, it holds that for each $i \in [m]$ the probability that $\widetilde{\theta_i} := \frac{int(x_i)\cdot 2\pi}{2^r}$ is in the interval $\theta \pm \frac{2\pi}{2^{r-2}}$ is greater than $\frac{3}{4}$.

Because the register that stores $\sum_x \alpha_x |x\rangle$ for each execution are separable, we have that the measurement outcomes of all the execution are independent. Then, by Chernoff bound, the probability that there are more than half of $\widetilde{\theta_1}, \widetilde{\theta_2}, \ldots, \widetilde{\theta_m}$ outside the interval $\theta \pm \frac{2\pi}{2^{r-2}}$ is less than $e^{-\Theta(m)}$. Let $\widetilde{x}$ be the the median of $int(x_1), int(x_2), \ldots, int(x_m)$ and let $\widetilde{\theta} := \frac{\widetilde{x}\cdot 2\pi}{2^r}$. We have the event that $\widetilde{\theta}$ is outside the interval $\theta \pm \frac{2\pi}{2^{r-2}}$ has probability less than $e^{-\Theta(m)}$.

Now we describe the quantum circuit $A_{U,r,m}$. The quantum circuit $A_{U,r,m}$ has m ancilla registers $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_m$ each of them is r-qubit, and an input register U takes the input state $|u_\theta\rangle$. The quantum circuit $A_{U,r,m}$ executes $P_{U,r}$ m times sequentially on the input $|u_\theta\rangle$ and in the ith execution of $P_{U,r}$, it stores $\sum_x \alpha_x |x\rangle$ in the register X_i. After m times execution, a circuit that computes the median of $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_m$ is applied. Since $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_m$ are separable and computing the median is a classical circuit, the distribution of $\widetilde{x}$ is the same regardless of the measurement taking place before or after the computing of median. As a result, Equation Equation (27) holds. ◀

To implement the energy estimation of the local Hamiltonian H, we run phase estimation for $U = e^{-iH}$, which can be implemented by Hamiltonian simulation.

▶ **Lemma 30** (Hamiltonian simulation [41]). *For all sufficiently large $n \in \mathbb{N}$, for all constant-local Hamiltonian $H \in \mathbb{C}^{2^c \times 2^n}$, evolution time $t \in \mathbb{R}^+$, and tolerated error $\varepsilon \in (0,1)$, there exists a quantum circuit $U_{\mathrm{sim}}(H,t,\varepsilon)$ running in $\mathrm{poly}(n,t,\log(1/\varepsilon))$ such that for any quantum state $|\psi\rangle \in \mathbb{C}^{2^n}$, it holds that*

$$\|(U_{\mathrm{sim}}(H,t,\varepsilon) - e^{-iHt} \otimes I)\,|\psi\rangle \otimes |0^{a_{\mathrm{sim}}}\rangle\,\|_2 \leq \varepsilon, \tag{28}$$

where a_{sim} is the number of ancilla qubits of $U_{\mathrm{sim}}(H,t,\varepsilon)$.

Now we formally define the energy estimation.

▶ **Definition 31** (Energy estimation). *Let $H \in \mathbb{C}^{2^n \times 2^n}$ be a constant-local Hamiltonian satisfying that all the eigenvalues of H are in the interval $[0,1)$. We first define an ideal unitary $\widehat{U}_{\mathrm{EE}}(H,\delta E,\eta)$ that estimates the corresponding energy of H's eigenstate $|\psi_j\rangle$. We say $\widehat{U}_{\mathrm{EE}}(H,\delta E,\eta)$ is an ideal energy estimation of the Hamiltonian H if for any H's eigenstate $|\psi_j\rangle$, it holds that*

$$\widehat{U}_{\mathrm{EE}}(H,\delta E,\eta)\,|\psi_j\rangle\,|0^{a_{\mathrm{EE}}}\rangle = |\psi_j\rangle \otimes \sum_{E'} \alpha_{E'}\,|E'\rangle\,|g_{E'}\rangle\,, \tag{29}$$

where a_{EE} is the number of ancilla qubits, $\alpha_{E'} \in \mathbb{C}$, and $|g_{E'}\rangle$ is some state depending on E', satisfying

$$\sum_{E' \in [E_j - \delta E, E_j + \delta E]} |\alpha_{E'}|^2 \geq 1 - \eta, \tag{30}$$

where E_j is the corresponding energy of $|\psi_j\rangle$.

Then, we say a quantum circuit $U_{\mathrm{EE}}(H,\delta E,\eta,\varepsilon)$ is an implementation of energy estimation of H if

$$\|U_{\mathrm{EE}}(H,\delta E,\eta,\varepsilon) - \widehat{U}_{\mathrm{EE}}(H,\delta E,\eta)\| \leq \varepsilon. \tag{31}$$

▶ **Lemma 32** (Efficient energy estimation algorithm). *For all sufficiently large $n \in \mathbb{N}$, For all constant-local, semi-definite Hamiltonian $H \in \mathbb{C}^{2^n \times 2^n}$ satisfying $\|H\| < 1$, and all tolerated error $\delta E = 1/\mathrm{poly}(n)$, there exists a energy estimation implementation $U_{\mathrm{EE}}(H, \delta E, \eta = e^{-n}, \varepsilon = 2^{-n})$ that runs in $\mathrm{poly}(n)$ time.*

Proof. It holds that $e^{-iH}|\psi_j\rangle = e^{-iE_j}|\psi_j\rangle$. We can use phase estimation to obtain E_j. We first consider the ideal unitary $\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta)$. By Lemma 29, we run $A_{U,r,m}$ with $U = e^{-iH}$, $r := \log(\frac{2\pi}{\delta E}) + 2$, and $m = Cn$ where C is a large enough constant, on the input $|\psi_j\rangle$. Let $E' := int(x) \cdot 2\pi/2^r$, we have that Equation Equation (29) and Equation (30) holds with $\eta = e^{-n}$.

Then, we consider the efficient implementation $U_{\mathrm{EE}}(H, \delta E, \eta = e^{-n}, \varepsilon = 2^{-n})$. We replace U in $A_{U,r,m}$ with the implementation of Hamiltonian simulation $U_{\mathrm{sim}}(H, t = 1, \varepsilon/(m \cdot 2^r))$ described in Lemma 30. (To be more precise, we need to implement control-U in $A_{U,r,m}$. Hence, we need to replace the gates in $U_{\mathrm{sim}}(H, t = 1, \varepsilon/(m \cdot 2^r))$ with their controlled version.) There are $m \cdot 2^r$ many of U in $A_{U,r,m}$. By triangular inequality, Equation Equation (31) holds.

For the running time, by Lemma 30, the implementation of $U_{\mathrm{sim}}(H, t = 1, \varepsilon/(m \cdot 2^r))$ takes $\mathrm{poly}(n, 1, \log(/m \cdot 2^r \varepsilon)) = \mathrm{poly}(n, 1, (\log(n) + \delta E + \log(1/\varepsilon)))$ time. We have $\delta E = 1/\mathrm{poly}(n)$ and $\varepsilon = 2^{-n}$, the running time of each implementation of U is $\mathrm{poly}(n)$. There are $m \cdot 2^r = \mathrm{poly}(n)$ many of U in $A_{U,r,m}$. As a result the overall running time of $U_{\mathrm{EE}}(H, \delta E, \eta = e^{-n}, \varepsilon = 2^{-n})$ is $\mathrm{poly}(n)$. ◀

We use quantum counting to estimate the number of eigenstates in a given interval.

▶ **Lemma 33** (Quantum counting). *Let a quantum circuit U taking a n-qubit input state $|\psi\rangle$ satisfy $U|\psi\rangle|0^a\rangle = \sqrt{\frac{M}{N}}|1\rangle|\xi^1\rangle + \sqrt{\frac{N-M}{N}}|0\rangle|\xi^0\rangle$, where $N = 2^n$, $M \leq N$[14], and a is the number of ancilla qubits. For any $c \in \mathbb{N}$, there is a quantum circuit that executes U $O(\mathrm{poly}(n) \cdot 2^{\frac{n}{2}})$ times with additional $O(\mathrm{poly}(n) \cdot 2^{\frac{n}{2}})$ many gates and outputs $\widetilde{M}$ such that $(1 - \Theta(\frac{1}{n^c}))M \leq \widetilde{M} \leq (1 + \Theta(\frac{1}{n^c}))M$ with probability $1 - \mathrm{negl}(n)$.*

In [44], there has already been a proof for the amplitude estimation algorithm that outputs $\widetilde{M}$ satisfying $(1 - \frac{1}{\sqrt{M}})M \leq \widetilde{M} \leq (1 + \frac{1}{\sqrt{M}})M$. To achieve the relative error as small as $\Theta(1/n^c)$, we increase the number of Grover iteration in [44] from $n/2 + 3$ to $n/2 + \Theta(c \log n)$.

To use quantum counting to estimate the number of eigenstates in a given interval, we need to prepare the uniform superposition of all the eigenstates as the input state. However, in general, the eigenstates are unknown. We use the property that EPR state is identical to the uniform superposition of the states and its complex conjugate in any orthogonal basis to overcome this issue.

▶ **Lemma 34.** *For any complete orthogonal basis of n-qubit system $\{|\psi_i\rangle\}_{j=1}^N$,i.e., $\sum_{j=1}^N |\psi_j\rangle\langle\psi_j| = I$ and $\langle\psi_j|\psi_{j'}\rangle = \delta_{j,j'}$ where $N = 2^n$, it holds that*

$$\sum_{j=1}^{N} \sqrt{\frac{1}{N}}|\psi_j\rangle|\psi_j^*\rangle = \sum_{k=1}^{N} \sqrt{\frac{1}{N}}|k\rangle|k\rangle. \tag{32}$$

where $|\psi_j^*\rangle$ is the complex conjugate of $|\psi_j\rangle$.

[14] Actually it requires that $M < N/2$. We can overcome this issue by adding an additional qubit.

Proof. Let $V = \sum_{j,k=1}^{N} V_{k,j}|k\rangle\langle j|$ be a unitary such that $|\psi_j\rangle = V|j\rangle$ for all $j \in [N]$. We have

$$\sum_{j=1}^{N} |\psi_j\rangle \otimes |\psi_j^*\rangle = \sum_{j=1}^{N} \left(\left(\sum_{k=1}^{N} V_{k,j} |k\rangle \right) \otimes \left(\sum_{k'=1}^{N} V_{k',j}^* |k'\rangle \right) \right)$$
$$= \sum_{j=1}^{N} \sum_{k=1}^{N} \sum_{k'=1}^{N} V_{k,j} V_{k',j}^* |k\rangle \otimes |k'\rangle . \tag{33}$$

Because V is a unitary, we have $\sum_{j=1}^{N} V_{k,j} V_{k',j}^* = \delta_{k,k'}$. Equation (33) becomes $\sum_{j=1}^{N} |k\rangle \otimes |k\rangle$. ◀

Proof of Theorem 27

Proof of Theorem 27. Let $N := 2^n$.

For any interval I and $E \in [0,1)$, we define the operator $U_{\mathrm{dec}}(I)|E\rangle|0^a_{\mathrm{dec}}\rangle = |\mathbb{1}_I(E)\rangle|*\rangle$, where a_{dec} is the number of ancilla qubits and $|*\rangle$ is some state. That is, $U_{\mathrm{dec}}(I)|E\rangle|0^a_{\mathrm{dec}}\rangle = |1\rangle|\xi_{E,1}\rangle$ if $E \in I$ and $U_{\mathrm{dec}}(I)|E\rangle|0^a_{\mathrm{dec}}\rangle = |0\rangle|\xi_{E,0}\rangle$ if $E \notin I$, where $|\xi_{E,1}\rangle$ and $|\xi_{E,0}\rangle$ are some states.

We write down the algorithm.

1. Let $L = 4n^c\beta$.
2. For $k \in \{0,1,\ldots,L-1\}$:
   - 2.1 For $\ell \in \{0,1,\ldots,L\}$(If $k=0$, then $\ell \in \{1,2,\ldots,L\}$):
     - 2.1.1 Define the characteristic energy $\mathcal{E}_{k,\ell} := \frac{\ell-1}{L} + \frac{k}{L^2}$, the interval $I_{k,\ell} := [\mathcal{E}_{k,l}, \mathcal{E}_{k,\ell} + \frac{1}{L})$, and $I'_{k,\ell} := [\mathcal{E}_{k,\ell} - 1/2L^2, \mathcal{E}_{k,\ell} + 1/L + 1/2L^2)$.
     - 2.1.2 Define $U_{k,\ell} := U_{\mathrm{dec}}(I'_{k,\ell}) U_{\mathrm{EE}}(H, \delta E = \frac{1}{2L^2}, \eta = e^{-n}, \varepsilon = 2^{-n})$, where U_{EE} is defined in Definition 31.
     - 2.1.3 Prepare the EPR state $|\Psi\rangle := \sum_{k=1}^{N} \sqrt{\frac{1}{N}} |k\rangle|k\rangle$.
     - 2.1.4 Execute quantum counting (Lemma 33) on $U_{k,\ell}|\Psi\rangle|0^{a_U}\rangle$ up to a relative error $\delta_{\mathrm{C}} = \frac{1}{4n^c}$, where $U_{k,\ell}$ acts on the first half of the EPR state and a_U is the number of ancilla qubits. Let $\widetilde{M}_{k,\ell}$ be the output of the quantum counting.
   - 2.2 Let $\widetilde{Z}_k := \sum_{\ell=0}^{L} \widetilde{M}_{k,\ell} e^{-\beta \mathcal{E}_{k,\ell}}$.
3. Let $\widetilde{Z} := \min\{\widetilde{Z}_k\}_{k=0}^{L-1}$.
4. Output $\widetilde{Z}$.

Now we claim the correctness of the algorithm. We define the following notations. Let $I^-_{k,\ell} := [\mathcal{E}_{k,\ell} - 1/L^2, \mathcal{E}_{k,\ell})$, $I^+_{k,\ell} := [\mathcal{E}_{k,\ell} + 1/L, \mathcal{E}_{k,\ell} + 1/L + 1/L^2)$, and $I''_{k,\ell} := I^-_{k,\ell} \cup I_{k,\ell} \cup I^+_{k,\ell}$. Let $Z_{k,\ell} := \sum_{j:E_j \in I_{k,\ell}} e^{-\beta E_j}$ be the partition function that are contributed by the states whose energies are in the interval $I_{k,\ell}$. Also, we define $Z^-_{k,\ell} := \sum_{j:E_j \in I^-_{k,\ell}} e^{-\beta E_j}$ and $Z^+_{k,\ell} := \sum_{j:E_j \in I^+_{k,\ell}} e^{-\beta E_j}$. Let $M_{k,\ell}$ be the number of states whose energies are in the interval $I_{k,\ell}$, $M^-_{k,\ell}$ be the number of states whose energies are in the interval $I^-_{k,\ell}$, and $M^+_{k,\ell}$ be the number of states whose energies are in the interval $I^+_{k,\ell}$.

The quantum partition function is dominated by low energy. The characteristic energy $\mathcal{E}_{k,\ell}$ is lower than all the energy in $I_{k,\ell}$. Hence, $M_{k,\ell} e^{-\mathcal{E}_{k,\ell}}$ gives a upper bound for $Z_{k,\ell}$, i.e.,

$$Z_{k,\ell} \leq M_{k,\ell} e^{-\mathcal{E}_{k,\ell}} \tag{34}$$

for any k, ℓ.

On the other hand, $\mathcal{E}_{k.\ell}$ is greater than all the energy decreased by $1/L$ in $I_{k,\ell}$. Hence, $M_{k,\ell}e^{-\mathcal{E}_{k,\ell}}$ gives a lower bound for $Z_{k,\ell}e^{1/L}$, i.e.,

$$M_{k,\ell}e^{-\mathcal{E}_{k,\ell}} \leq Z_{k,\ell}e^{1/L}. \tag{35}$$

for any k, ℓ.

Similarly, we have

$$M^{-}_{k,\ell}e^{-\beta\mathcal{E}_{k,\ell}} \leq Z^{-}_{k,\ell}, \tag{36}$$

and

$$M^{+}_{k,\ell}e^{-\beta\mathcal{E}_{k,\ell}} \leq Z^{+}_{k,\ell}e^{1/L+1/L^2}. \tag{37}$$

First we consider the ideal case. That is, using $\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = 0)$ (Equation Equation (29)) instead of U_{EE} in Step 2.1.2. According to Definition 31, if $E_j \in I_{k,l}$, it holds that

$$\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = 0) \left|\psi_j\right\rangle \left|0^{a_{\mathrm{EE}}}\right\rangle = \left|\psi_j\right\rangle \otimes \sum_{E' \in I'_{k,\ell}} \alpha_{E'} \left|E'\right\rangle \left|g_{E'}\right\rangle;$$

and if $E_j \notin I''_{k,\ell}$, it holds that

$$\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = 0) \left|\psi_j\right\rangle \left|0^{a_{\mathrm{EE}}}\right\rangle = \left|\psi_j\right\rangle \otimes \sum_{E' \notin I'_{k,\ell}} \alpha_{E'} \left|E'\right\rangle \left|g_{E'}\right\rangle.$$

In other words, letting $\widehat{U}_{k,\ell} := U_{\mathrm{dec}}(I'_{k,\ell})\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = 0)$, we have

$$\widehat{U}_{k,\ell} \left|\psi_j\right\rangle \left|0^{a_U}\right\rangle = \begin{cases} \left|1\right\rangle \left|\xi^1_j\right\rangle & , \text{ if } E_j \in I_{k,\ell}, \\ \alpha_j \left|1\right\rangle \left|\xi^1_j\right\rangle + \beta_j \left|0\right\rangle \left|\xi^0_j\right\rangle & , \text{ if } E_j \in I^{-}_{k,\ell} \cup I^{+}_{k,\ell}, \\ \left|0\right\rangle \left|\xi^0_{\ell,j}\right\rangle & , \text{ if } E_j \notin I''_{\ell}, \end{cases} \tag{38}$$

where $\alpha_j, \beta_j \in \mathbb{C}$ satisfy $|\alpha_{\ell,j}| + |\beta_{\ell,j}| = 1$, $|\xi^1_j\rangle$, $|\xi^0_j\rangle$ are quantum states depending on j. The coefficients α_j, β_j and the states $\left|\xi^1_j\right\rangle, \left|\xi^0_j\right\rangle$ may depend on k, ℓ.

When $\widehat{U}_{k,\ell}$ acts on the first half of $|\Psi\rangle$, combining Equation (38) and Lemma 34, we have

$$\widehat{U}_{k,\ell} \left|\Psi\right\rangle \left|0^{a_U}\right\rangle = \sqrt{\frac{\widehat{M}_{k,\ell}}{N}} \left|1\right\rangle \left|\xi^1\right\rangle + \sqrt{\frac{N - \widehat{M}_{k,\ell}}{N}} \left|0\right\rangle \left|\xi^0\right\rangle, \tag{39}$$

where $\widehat{M}_{k,\ell} \in [N]$ satisfies

$$M_{k,\ell} \leq \widehat{M}_{k,\ell} \leq M^{-}_{k,\ell} + M_{k,\ell} + M^{+}_{k,\ell}, \tag{40}$$

and $|\xi^1\rangle, |\xi^0\rangle$ are some quantum states.

Let $\widetilde{M}^{ideal}_{k,\ell}$ be the output in Step 2.1.4. when replacing $U_{k,\ell}$ with $\widehat{U}_{k,\ell}$. By the guarantee of quantum counting (Lemma 33), we have that

$$(1 - \delta_{\mathrm{C}})\widehat{M}_{k,\ell} \leq \widetilde{M}^{ideal}_{k,\ell} \leq (1 + \delta_{\mathrm{C}})(\widehat{M}_{k,\ell}). \tag{41}$$

By union bound, the event that Equation Equation (41) holds for all k, ℓ has probability $1 - \mathrm{negl}(n)$.

Now we lower bound $\widetilde{Z}$. Because $M_{k,\ell} \leq \widehat{M}_{k,\ell}$, from Equation Equation (41), we have

$(1-\delta_{\mathrm{C}})M_{k,\ell} \leq \widetilde{M}_{k,\ell}^{ideal}$.

Multiplying by $e^{-\beta\mathcal{E}_{k,\ell}}$ and combining with Equation Equation (34), we have

$$(1-\delta_{\mathrm{C}})Z_{k,\ell} \leq \widetilde{M}_{k,\ell}^{ideal} e^{-\beta\mathcal{E}_{k,\ell}}. \tag{42}$$

For any k, when summing ℓ over $\{0,1,\ldots,L\}$ in Equation Equation (42), we have

$$(1-\delta_{\mathrm{C}})Z \leq \sum_{\ell} \widetilde{M}_{k,\ell}^{ideal} e^{-\beta\mathcal{E}_{k,\ell}}. \tag{43}$$

For the upper bound, because $\widehat{M}_{k,\ell} \leq M_{k,\ell}^{-} + M_{k,\ell} + M_{k,\ell}^{+}$, combining with the guarantee of quantum counting (Equation Equation (41)), we have

$\widetilde{M}_{k,\ell}^{ideal} \leq (1+\delta_{\mathrm{C}})(M_{k,\ell}^{-} + M_{k,\ell} + M_{k,\ell}^{+})$.

Multiplying by $e^{-\beta\mathcal{E}_{k,\ell}}$ and combing with Equation Equation (36) and Equation (37), we have

$$\widetilde{M}_{k,\ell}^{ideal} e^{-\beta\mathcal{E}_{k,\ell}} \leq (1+\delta_{\mathrm{C}})(Z_{k,\ell}^{-} + Z_{k,\ell}e^{-\beta/L} + Z_{k,\ell}^{+}e^{\beta/L+\beta/L^2}). \tag{44}$$

Fixing a k and summing ℓ over $\{0,1,\ldots,L\}$ in Equation Equation (44), we have

$$\sum_{\ell} \widetilde{M}_{k,\ell}^{ideal} e^{-\beta\mathcal{E}_{k,\ell}} \leq (1+\delta_{\mathrm{C}})\Big((\sum_{\ell} Z_{k,\ell}^{-}) + Ze^{\beta/L} + (\sum_{\ell} Z^{+}e^{\beta/L+\beta/L^2})\Big). \tag{45}$$

Consider the term $\sum_{\ell} Z_{k,\ell}^{-} + \sum_{\ell} Z^{+}e^{\beta/L+\beta/L^2}$ and sum over k. That is

$$\sum_{k=0}^{L-1}\Big(\sum_{\ell} Z_{k,\ell}^{-} + \sum_{\ell} Z^{+}e^{\beta/L+\beta/L^2}\Big). \tag{46}$$

Since

$$\bigcup_{k=0}^{L-1}\bigcup_{\ell} I_{k,\ell}^{-} = [-1/L, 0) \cup [0, 1-1/L^2),$$

and all the eigenstates are in the interval $[0,1)$, we have that $\sum_{k=0}^{L-1}\sum_{\ell} Z_{k,\ell}^{-}$ is contributed by the eigenstates in the interval $[0, 1-1/L^2)$. Therefore,

$$\sum_{k=0}^{L-1}\sum_{\ell} Z_{k,\ell}^{-} \leq Z. \tag{47}$$

Similarly, we have

$$\sum_{k=0}^{L-1}\sum_{\ell} Z_{k,\ell}^{+} \leq Z. \tag{48}$$

Plugging Equation Equation (47) and Equation (48) into Equation Equation (46), we have

$$\sum_{k=0}^{L-1}\Big(\sum_{\ell} Z_{k,\ell}^{-} + \sum_{\ell} Z^{+}e^{\beta/L+\beta/L^2}\Big) \leq Z(1+e^{\beta/L+\beta/L^2}). \tag{49}$$

By average argument, there is a k such that

$$\Big(\sum_{\ell} Z^{-}_{k,\ell} + \sum_{\ell} Z^{+} e^{\beta/L+\beta/L^2}\Big) \leq \frac{1}{L} Z(1 + e^{\beta/L+\beta/L^2}). \tag{50}$$

Let $\widetilde{Z}^{ideal}$ be the output of the algorithm when replacing $U_{k,\ell}$ with $\widetilde{U}_{k,\ell}$. By Equation Equation (45) and Equation (50), we have

$$\widetilde{Z}^{ideal} \leq (1+\delta_{\mathrm{C}})\big(\frac{1}{L} + e^{\beta/L} + \frac{1}{L} e^{\beta/L+\beta/L^2}\big) Z. \tag{51}$$

Recall that $\delta_{\mathrm{C}} = \frac{1}{4n^c}$, $L = 4n^c\beta$, and $\beta = \mathrm{poly}(n)$. We have

$$\frac{1}{L} + e^{\beta/L} + \frac{1}{L} e^{\beta/L+\beta/L^2} = e^{\frac{1}{4n^c}} + \frac{e^{\frac{1}{4n^c}(1+\frac{1}{4n^c\beta})}}{4n^c\beta} + \frac{1}{4n^c\beta} \leq 1 + \frac{1}{2n^c}$$

for sufficiently large n. As a result, Equation Equation (51) becomes

$$\widetilde{Z}^{ideal} \leq \big(1 + \frac{1}{4n^c}\big)\big(1 + \frac{1}{2n^c}\big) Z \leq \big(1 + \frac{1}{n^c}\big) Z \tag{52}$$

for sufficiently large n.

Finally, combing Equation Equation (43) and Equation (52), we have

$$(1 - 1/n^c) Z \leq \widetilde{Z}^{ideal} \leq (1 + 1/n^c) Z. \tag{53}$$

Then, we consider the real implementation. There are two steps to change the ideal algorithm to the real implementation. First, we define a intermediate algorithm by replacing $\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = 0)$ in the ideal algorithm with $\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = e^{-n})$. By Lemma 33, there are $O(\mathrm{poly}(n) \cdot 2^{\frac{n}{2}})$ many $\widehat{U}_{EE}$ in the algorithm. Also, we have $\|\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = e^{-n}) - \widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = 0)\| \leq e^{-n}$. By triangular inequity, the distance between the ideal algorithm and the intermediate algorithm is $\mathrm{negl}(n)$.

For the second step, we get the real implementation by replacing $\widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = e^{-n})$ with $U_{\mathrm{EE}}(H, \delta E, \eta = e^{-n}, \varepsilon = 2^{-n})$. We have $\|U_{\mathrm{EE}}(H, \delta E, \eta = e^{-n}, \varepsilon = 2^{-n}) - \widehat{U}_{\mathrm{EE}}(H, \delta E, \eta = e^{-n})\| \leq 2^{-n}$. Again, by triangular inequity, the distance between the intermediate algorithm and the real implementation is $\mathrm{negl}(n)$. As a result, the distance between ideal algorithm and the real implementation is $\mathrm{negl}(n)$. Constituently, the real implementation outputs the correct estimation of quantum partition function with probability $1 - \mathrm{negl}(n)$.

Finally, we analyze the running time. There are $\mathrm{poly}(n)$ of iterations over k and ℓ and each iteration runs in $O(\mathrm{poly}(n) \cdot 2^{\frac{n}{2}})$ time. Therefore the overall running time is $O^*(2^{\frac{n}{2}})$. ◀

References

1. Scott Aaronson, Nai-Hui Chia, Han-Hsuan Lin, Chunhao Wang, and Ruizhe Zhang. On the quantum complexity of closest pair and related problems. In *Proceedings of the 35th Computational Complexity Conference (CCC 2020)*, pages 16:1–16:43, 2020. doi:10.4230/LIPIcs.CCC.2020.16.
2. Brian Alspach. Johnson graphs are Hamilton-connected. *Ars Mathematica Contemporanea*, 6(1):21–23, 2012. doi:10.26493/1855-3974.291.574.
3. Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum SDP-Solvers: Better upper and lower bounds. *Quantum*, 4:230, 2020. doi:10.22331/q-2020-02-14-230.
4. Yosi Atia and Dorit Aharonov. Fast-forwarding of Hamiltonians and exponentially precise measurements. *Nature Communications*, 8(1):1572, 2017. doi:10.1038/s41467-017-01637-7.

5 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing (STOC 2015)*, pages 51–58, 2015. doi:10.1145/2746539.2746612.

6 Marshall Ball, Alon Rosen, Manuel Sabin, and Prashant Nalini Vasudevan. Average-case fine-grained hardness. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*, pages 483–496, 2017. doi:10.1145/3055399.3055466.

7 Jacob D. Biamonte and Peter J. Love. Realizable hamiltonians for universal adiabatic quantum computers. *Physical Review A*, 78:012352, 2008. doi:10.1103/PhysRevA.78.012352.

8 Fernando GSL Brandão. Entanglement theory and the quantum simulation of many-body physics. *arXiv preprint*, 2008. arXiv:0810.0026.

9 Sergey Bravyi, Anirban Chowdhury, David Gosset, and Pawel Wocjan. Quantum Hamiltonian complexity in thermal equilibrium. *Nature Physics*, 18(11):1367–1370, 2022.

10 Sergey Bravyi, David Gosset, Robert König, and Kristan Temme. Approximation algorithms for quantum many-body problems. *Journal of Mathematical Physics*, 60(3):032203, March 2019. doi:10.1063/1.5085428.

11 Harry Buhrman, Sevag Gharibian, Zeph Landau, François Le Gall, Norbert Schuch, and Suguru Tamaki. Beating the natural Grover bound for low-energy estimation and state preparation. *Physical Review Letters*, 135:030601, 2025. doi:10.1103/29qw-bssx.

12 Harry Buhrman, Subhasree Patro, and Florian Speelman. A framework of quantum strong exponential-time hypotheses. In *Proceedings of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, pages 19:1–19:19, 2021. doi:10.4230/LIPIcs.STACS.2021.19.

13 Yanlin Chen, Yilei Chen, Rajendra Kumar, Subhasree Patro, and Florian Speelman. QSETH strikes again: Finer quantum lower bounds for lattice problem, strong simulation, hitting set problem, and more. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025)*, pages 6:1–6:24, 2025. doi:10.4230/LIPIcs.APPROX/RANDOM.2025.6.

14 Nai-Hui Chia, Kai-Min Chung, Yao-Ching Hsieh, Han-Hsuan Lin, Yao-Ting Lin, and Yu-Ching Shen. On the impossibility of general parallel fast-forwarding of Hamiltonian simulation. In *Proceeding of the 38th Computational Complexity Conference (CCC 2023)*, pages 33:1–33:45, 2023. doi:10.4230/LIPIcs.CCC.2023.33.

15 Anirban N Chowdhury, Rolando D Somma, and Yiğit Subaşı. Computing partition functions in the one-clean-qubit model. *Physical Review A*, 103(3):032422, 2021.

16 Stephen Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing (STOC 1971)*, pages 151–158, 1971. doi:10.1145/800157.805047.

17 Toby Cubitt and Ashley Montanaro. Complexity classification of local Hamiltonian problems. *SIAM Journal on Computing*, 45(2):268–316, 2016. doi:10.1137/140998287.

18 Alexander M. Dalzell, Nicola Pancotti, Earl T. Campbell, and Fernando G. S. L. Brandão. Mind the Gap: Achieving a Super-Grover Quantum Speedup by Jumping to the End. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC 2023)*, pages 1131–1144, 2023. doi:10.1145/3564246.3585203.

19 Roee David, Karthik C. S., and Bundit Laekhanukit. On the complexity of closest pair via polar-pair of point-sets. *SIAM Journal on Discrete Mathematics*, 33(1):509–527, 2019. doi:10.1137/17M1128733.

20 Ronald de Wolf. Quantum computing: Lecture notes. *arXiv preprint arXiv:1907.09415*, 2019. arXiv:1907.09415.

21 Richard P Feynman. Quantum mechanical computers. *Foundations of physics*, 16(6):507–531, 1986.

22 Yimin Ge, Jordi Tura, and J. Ignacio Cirac. Faster ground state preparation and high-precision ground energy estimation with fewer qubits. *Journal of Mathematical Physics*, 60(2):022202, February 2019. doi:10.1063/1.5027484.

23 Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. Quantum Hamiltonian complexity. *Foundations and Trends in Theoretical Computer Science*, 10(3):159–282, 2015. doi:10.1561/0400000066.

24 András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*, pages 193–204, 2019. doi:10.1145/3313276.3316366.

25 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 212–219, 1996. doi:10.1145/237814.237866.

26 M. B. Hastings. A Short Path Quantum Algorithm for Exact Optimization. *Quantum*, 2:78, 2018. doi:10.22331/q-2018-07-26-78.

27 Cupjin Huang, Michael Newman, and Mario Szegedy. Explicit lower bounds on strong quantum simulation. *IEEE Transactions on Information Theory*, 66(9):5585–5600, 2020. doi:10.1109/TIT.2020.3004427.

28 Jeremy Ahrens Huang, Young Kun Ko, and Chunhao Wang. On the (classical and quantum) fine-grained complexity of log-approximate CVP and Max-Cut. *arXiv preprint arXiv:2411.04124*, 2024. doi:10.48550/arXiv.2411.04124.

29 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

30 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/JCSS.2001.1774.

31 C. S. Karthik and Pasin Manurangsi. On closest pair in euclidean metric: Monochromatic is as hard as bichromatic. *Combinatorica*, 40(4):539–573, 2020. doi:10.1007/s00493-019-4113-1.

32 Julia Kempe, Alexei Y. Kitaev, and Oded Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006. doi:10.1137/S0097539704445226.

33 Alex Kerzner, Vlad Gheorghiu, Michele Mosca, Thomas Guilbaud, Federico Carminati, Fabio Fracas, and Luca Dellantonio. A square-root speedup for finding the smallest eigenvalue. *Quantum Science and Technology*, 6(4):045025, 2024. doi:10.1088/2058-9565/ad6a36.

34 Alexei Yu. Kitaev, Alexander H. Shen, and Mikhail N. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.

35 Emanuel Knill and Raymond Laflamme. Power of one bit of quantum information. *Physical Review Letters*, 81(25):5672, 1998.

36 Jacek Kuczyński and Henryk Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992. doi:10.1137/0613066.

37 Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, 1950. doi:10.6028/jres.045.026.

38 Leonid Levin. Universal search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

39 Lin Lin and Yu Tong. Near-optimal ground state preparation. *Quantum*, 4:372, 2020. doi:10.22331/q-2020-12-14-372.

40 Sixue Liu. Chain, generalization of covering code, and deterministic algorithm for k-SAT. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, pages 88:1–88:13, 2018. doi:10.4230/LIPIcs.ICALP.2018.88.

41 Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian simulation by quantum signal processing. *Phys. Rev. Lett.*, 118:010501, 2017. doi:10.1103/PhysRevLett.118.010501.

42 John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2:040203, 2021. doi:10.1103/PRXQuantum.2.040203.

43 Jesper Nederlof. An invitation to "fine-grained complexity of NP-complete problems". *arXiv preprint arXiv:2601.05044*, 2026. doi:10.48550/arXiv.2601.05044.

44 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

45 Roberto Oliveira and Barbara M. Terhal. The complexity of quantum spin systems on a two-dimensional square lattice. *Quantum Information and Computation*, 8(10):900–924, 2008. doi:10.26421/QIC8.10-2.

46 Stephen Piddock and Ashley Montanaro. The complexity of antiferromagnetic interactions and 2D lattices. *Quantum Information and Computation*, 17(7-8):636–672, 2017. doi:10.26421/QIC17.7-8-6.

47 David Poulin and Pawel Wocjan. Preparing ground states of quantum many-body systems on a quantum computer. *Physical Review Letters*, 102(13):130503, 2009. doi:10.1103/PHYSREVLETT.102.130503.

48 David Poulin and Pawel Wocjan. Sampling from the Thermal Quantum Gibbs State and Evaluating Partition Functions with a Quantum Computer. *Phys. Rev. Lett.*, 103:220502, November 2009. doi:10.1103/PhysRevLett.103.220502.

49 Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis. In *Proceedings of the 10th International Symposium on Parameterized and Exact Computation (IPEC 2015)*, pages 17–29, 2015. doi:10.4230/LIPIcs.IPEC.2015.17.

50 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3447–3487. World Scientific, 2018.

51 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2):357–365, 2005. doi:10.1016/j.tcs.2005.09.023.

A A trivial fine-grained reduction to $k(\varepsilon)$LH from $k(\varepsilon)$SAT

▶ **Theorem 35** (Lower bound of $k(\varepsilon)$LH). *Assuming SETH (resp. QSETH), for any $\varepsilon > 0$, there is k (depending on ε) such that for any algorithm, there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, there is a Hamiltonian H acting on n qubits, associated $E_{\mathrm{yes}}, E_{\mathrm{no}}$ satisfying $E_{\mathrm{no}} - E_{\mathrm{yes}} \geq \Omega(1)$ such that $\mathrm{LH}(H, E_{\mathrm{yes}}, E_{\mathrm{no}})$ cannot be solved in $O(2^{n(1-\varepsilon)})$ classical time (resp. $O(2^{\frac{n}{2}(1-\varepsilon)})$ quantum time).*

Proof. We construct a Hamiltonian H from a k-SAT instance $\Phi = \varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_m$.

Let $S_i \subseteq [n]$ be the set collecting the index j such that x_j or $\neg x_j$ appears in φ_i, and let $\mathsf{S_i}$ be the corresponding register. Let $y_i \in \{0,1\}^{|S_i|}$ be the assignment to the variables appearing in φ_i such that $\varphi_i(y_i) = 0$. Because φ_i is in the disjunctive form, y_i is unique. Let $H := \sum_{i=1}^{m} H_i$, where $H_i := I - |y_i\rangle\langle y_i|_{\mathsf{S_i}}$ for all $i \in [m]$. It holds that $H_i |y_i\rangle = 0$ if $\varphi_i(y_i) = 1$ and $H_i |y_i\rangle = |y_i\rangle$ if $\varphi_i(y_i) = 0$. Each H_i acts non-trivially on at most k qubits. Hence, H is k-local.

If there exists an assignment $x \in \{0,1\}^n$ satisfying Φ, then $H|x\rangle = 0$. Otherwise, $\lambda(H) \geq 1$. Hence, solving $\mathrm{LH}(H, E_{\mathrm{yes}} = 0, E_{\mathrm{no}} = 1)$ can decide k-SAT. The construction of H takes $\mathrm{poly}(n)$ time. Hence, if $\mathrm{LH}(H, E_{\mathrm{yes}} = 0, E_{\mathrm{no}} = 1)$ is solved in $O(2^n(1-\varepsilon))$ time, then k-SAT(Φ) is also solved in $O(2^{n(1-\varepsilon)})$ classical time, which violates SETH. The argument assuming QSETH follows in the same proof strategy. ◀

Resolution Width Lifts to Near-Quadratic-Depth Res(⊕) Size

Dmitry Itsykson
Ben-Gurion University of the Negev, Beer-Sheva, Israel

Vladimir Podolskii
Tufts University, Medford, MA, USA

Alexander Shekhovtsov
EPFL, Lausanne, Switzerland

Abstract

We show that for any unsatisfiable CNF formula φ that requires resolution refutation width at least w, and for any 1-stifling gadget g (for example, $g = \mathrm{MAJ}_3$), (1) every resolution-over-parities (Res(⊕)) refutation of the lifted formula $\varphi \circ g$ of size at most S has depth at least $\Omega(w^2/\log S)$; (2) every Res(⊕) refutation of the lifted formula $\varphi \circ g$ has size $\Omega(w^2)$.

The first result substantially extends and simplifies all previously known lifting theorems for bounded-depth Res(⊕). The lifting result of Itsykson and Knop [29] requires gadgets of logarithmic size and applies only to refutations of depth at most $O(n \log n)$, whereas our result applies to nearly quadratic depth. The liftings of Bhattacharya and Chattopadhyay [13] and of Byramji and Impagliazzo [16] apply to nearly quadratic depth as well, but rely on a much stronger assumption of $(\Omega(n), \Omega(n))$-DT-hardness, which is far less standard than large resolution width.

Our proof combines the random-walk-with-restarts method of Alekseev and Itsykson [4] with a new idea: the random walk is defined relative to the structure of the refutation graph, rather than by a distribution on inputs induced by the formula.

Using this technique, we substantially strengthen the supercritical size-depth tradeoff of Itsykson and Knop [29], both by improving the depth lower bound and by reducing the size of the separating formulas to polynomial in the number of variables, with the latter resolving an open question posed in [29]. In particular, we construct a family of polynomial-size formulas that admit polynomial-size resolution refutations, while any Res(⊕) refutation of depth $o(n^2/\log^4 n)$ necessarily has superpolynomial size.

Our second result yields a pure quadratic lower bound on the size of Res(⊕) refutations, improving upon the previously known near-quadratic lower bound of [16].

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases lifting theorems, resolution depth, resolution over parities, resolution width, supercritical tradeoff, random walk with restarts

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.13

Related Version *ECCC Version*: https://eccc.weizmann.ac.il/report/2026/018/

Funding *Dmitry Itsykson*: Supported by European Research Council Grant No. 949707.

1 Introduction

Propositional proof complexity studies propositional proof systems used to certify the unsatisfiability of CNF formulas. One of the central research directions in proof complexity, known as Cook's program, seeks to establish superpolynomial lower bounds on the proof size required by specific proof systems. The ultimate goal of this program is to prove NP ≠ coNP, which is equivalent to proving superpolynomial lower bounds for every propositional proof system.

© Dmitry Itsykson, Vladimir Podolskii, and Alexander Shekhovtsov;
licensed under Creative Commons License CC-BY 4.0

In this paper, we consider two proof systems. The first is *resolution*, one of the most extensively studied propositional proof systems. Resolution proves the unsatisfiability of a formula by deriving new clauses using a single inference rule, which allows one to derive a clause $A \vee B$ from the clauses $A \vee x$ and $B \vee \neg x$. A refutation of a CNF formula φ is a derivation of the empty clause from the clauses of φ. Important complexity measures are the width of a resolution refutation, defined as the maximum number of literals in any clause of the refutation, and the depth, defined as the length of the longest path from an initial clause of the formula to the empty clause.

While many exponential-size lower bounds are known for weak proof systems such as resolution, we lack superpolynomial lower bounds for Frege systems, which include standard propositional proof systems from logic textbooks. A Frege derivation is a sequence of Boolean formulas; each of them is either an axiom or is obtained from the previous by a set of sound and implicationally complete inference rules. Proving Frege lower bounds is often compared to proving Boolean formula/circuit lower bounds for explicit Boolean functions, and both seem intractable. However, progress has been made in restricted settings. A superpolynomial lower bound for constant-depth circuits computing parity was proven in the 1980s [24, 1]. Ajtai later used a similar approach to prove a superpolynomial lower bound for bounded-depth Frege systems [2]. Razborov and Smolenski proved lower bounds for constant-depth circuits with $\neg$, $\vee$, $\wedge$, and MOD_p gates in 1987 [40, 36]. The analogous problem of proving a lower bound for constant-depth Frege systems using $\neg$, $\vee$, $\wedge$ and MOD_p gates (denoted $\mathrm{AC}^0[p]$-Frege) is open for all $p > 1$.

The second proof system considered in this paper is *resolution over parities* (Res(⊕)), an extension of resolution that permits reasoning modulo 2. In contrast to standard resolution, Res(⊕) operates with linear clauses, which are disjunctions of linear equations over $\mathbb{F}_2$. Its resolution rule allows one to infer $C \vee D$ from the premises $C \vee (f = 0)$ and $D \vee (f = 1)$, where f is a linear form. In addition, Res(⊕) includes a weakening rule, which allows one to derive a linear clause D from a clause C whenever C semantically implies D. The depth of a Res(⊕) refutation is the maximum number of resolution steps along a path from an initial clause to the empty clause.

Res(⊕) is the weakest known subsystem of $\mathrm{AC}^0[2]$-Frege for which superpolynomial lower bounds are still open. This makes the study of lower bounds for Res(⊕) a particularly important direction.

1.1 Superpolynomial Lower Bounds for Subsystems of Res(⊕)

The first lower bounds for tree-like Res(⊕) were proved by Itsykson and Sokolov [31, 32]. Since then, a number of works have established exponential lower bounds for tree-like Res(⊕) on classical combinatorial formulas [26, 28, 30, 34, 16].

Chattopadhyay, Mande, Sanyal, and Sherif [18], and independently Beame and Kroth [7], introduced a lifting approach for proving lower bounds in tree-like Res(⊕).

Given a CNF formula $\varphi(y_1, y_2, \ldots, y_n)$ and a Boolean function $g : \{0,1\}^\ell \to \{0,1\}$, called a *gadget*, the *lifted formula* $\varphi \circ g$ is defined as the CNF encoding of

$$\varphi\big(g(x_{1,1}, \ldots, x_{1,\ell}), \ldots, g(x_{n,1}, \ldots, x_{n,\ell})\big),$$

where each variable y_i in φ is replaced by $g(x_{i,1}, \ldots, x_{i,\ell})$ over fresh variables $x_{i,1}, \ldots, x_{i,\ell}$.

Chattopadhyay, Mande, Sanyal, and Sherif [18] also introduced the notion of *k-stifling gadgets*. A Boolean function $g : \{0,1\}^\ell \to \{0,1\}$ is called a k-stifling gadget if, for every $a \in \{0,1\}$ and every choice of $\ell - k$ input variables, there exists an assignment to these $\ell - k$ variables such that the value of g is fixed to a, regardless of the values of the remaining k variables.

They [18] showed that if every resolution refutation of a formula φ has depth at least h, and g is a k-stifling gadget, then any tree-like Res($\oplus$) refutation of the lifted formula $\varphi \circ g$ must have size at least 2^{kh}.

Efremenko, Garlik, and Itsykson [20] made the first progress beyond tree-like Res($\oplus$) by proving an exponential lower bound for (bottom-)regular Res($\oplus$). Building on this result, Alekseev and Itsykson [4] established an exponential lower bound for a stronger model – namely, Res($\oplus$) refutations of depth at most $n \log\log n$, where n is the number of variables. Their proof introduced a technique based on random walks with restarts, which has since become standard in this area. Subsequently, Efremenko and Itsykson [21] presented an alternative analysis of the same approach, improving the depth lower bound to $n \log n$. In all these results, the hard instance is a Tseitin formula over an $O(\log n)$-degree expander, lifted by an arbitrary 2-stifling gadget.

Itsykson and Knop [29] proved the following lifting theorem.

▶ **Theorem 1.1** ([29]). *Suppose that every resolution refutation of φ has either width at least w or depth at least h. Let s be an integer such that $h \geq s^2 w$. Then any* Res($\oplus$) *refutation of $\varphi \circ \oplus_s \circ \mathrm{MAJ}_5$ has either size at least 2^w or depth $\Omega(s^2 w)$.*

Theorem 1.1 implies that any CNF formula with resolution width $\Omega(n)$ can be lifted by an $O(\log n)$-size gadget to a formula for which every Res($\oplus$) refutation has either exponential size or depth $\Omega(n \log n)$.

All of the aforementioned papers [4, 21, 29] apply the random-walk-with-restarts method under the uniform distribution on inputs. This choice leads to an exponentially small success probability, which in turn forces the random walk to terminate after only a small number of rounds, thereby limiting the achievable depth lower bounds.

Bhattacharya and Chattopadhyay [13] were the first to apply the random-walk-with-restarts technique under a non-uniform input distribution. This innovation yields a 2^{n^ε} lower bound for the size of Res($\oplus$) refutations of nearly quadratic depth $n^{2-\varepsilon}$. Their hard instance is obtained by lifting a formula with the IP gadget of size $O(\log n)$. The base formula is required to satisfy an $(\Omega(n), \Omega(n))$-DT-hardness property. Although the $(\Omega(n), \Omega(n))$-DT-hardness property is somewhat technical[1], it is known to hold for Tseitin formulas over constant-degree expanders.

In recent work, Byramji and Impagliazzo [16], building on ideas from lifting theorems for parity decision trees [18, 7, 35, 15], obtained several additional results. Among them is a 2^{n^ε} lower bound on the size of Res($\oplus$) refutations of the binary pigeonhole principle BPHP_n^{n+1} of depth $n^{2-\varepsilon}$. Their main lifting contribution is an improvement in the gadget size: it suffices to lift an $(\Omega(n), \Omega(n))$-DT-hard formula using an arbitrary 1-stifling gadget. When applied to Tseitin formulas, this yields lower bounds for formulas of size $O(n)$, meaning that the refutation depth is almost quadratic in the formula size, rather than in the number of variables as in previous works.

[1] Although we do not use the notion of (p, q)-DT-hard in this paper, for the sake of completeness we provide the definition here. A CNF formula φ is called (p, q)-DT-hard if there exists a non-empty set of partial assignments $\mathcal{P}$ that satisfies the following properties: 1) No partial assignment in $\mathcal{P}$ falsifies any clause of φ; 2) $\mathcal{P}$ is closed under taking subassignments (i.e., if $\rho \in \mathcal{P}$ and ρ' is a subassignment of ρ, then $\rho' \in \mathcal{P}$); 3) For any partial assignment $\rho \in \mathcal{P}$ that fixes at most p variables, there is a distribution ν_ρ over the remaining unassigned variables such that for any decision tree T of depth at most q querying these variables, $\Pr[\rho \cup \sigma_{T,\nu_\rho} \in \mathcal{P}] \geq \frac{1}{3}$, where σ_{T,ν_ρ} is the random leaf assignment reached by running T on the distribution ν_ρ.

In a recent paper, Alekseev and Gaevoy [3] introduced a new family of formulas, the *Constrained Bit Pigeonhole Principle*, and proved a lower bound of $2^{n^{\Omega(\varepsilon)}}$ for Res(⊕) refutations of depth at most $n^{2-\varepsilon}$. Using these formulas, the authors were able to apply the random-walk-with-restarts framework with the uniform distribution over inputs.

Recently, Efremenko and Itsykson [22] established that for every $\varepsilon > 0$, a lower bound of $2^{n(1-\varepsilon)}$ holds for Depth-n Res(⊕) refutations, where n denotes the number of variables in the refuted formula. Consequently, this lower bound is essentially as strong as what is predicted by the Strong Exponential Time Hypothesis (SETH). Their proof relies on a refinement of the random-walk-with-restarts method, combined with a lifting argument from formulas of extremely large resolution width.

Supercritical tradeoff

We say a formula φ exhibits a *supercritical tradeoff* between two proof complexity measures, μ and ν, if the following holds: while φ has some proofs with small μ and others with small ν, any proof that restricts μ below a certain threshold simultaneously forces ν to exceed the maximum worst-case upper bound known for arbitrary formulas.

In recent years, a surge of work has established supercritical tradeoffs across various proof systems [37, 6, 11, 8, 39, 38, 12, 23, 14, 19, 27, 17]. Specifically, supercritical tradeoffs between width and depth for resolution have been achieved via distinct constructions by Fleming et al. [23] and Buss and Thapen [14]. Expanding on these results, de Rezende et al. [19] and Göös et al. [27] demonstrated that such tradeoffs can be made truly supercritical, meaning the lower bounds are quantified in terms of formula size rather than merely the number of variables.

Another consequence of Theorem 1.1 is that, when combined with the supercritical width–depth tradeoff for Resolution due to Buss and Thapen [14], it yields a *supercritical tradeoff* between depth and Res(⊕) size [29]. The advantage of using the particular tradeoff from [14] in [29], rather than the related results of [23, 19, 27], is that it applies to larger width parameters relative to the number of variables in the formula; in particular, it works for widths $w = \Omega(n/\log n)$. Specifically, this allows [29] to show that there exists a family of formulas ψ_n on n variables, of size $n^{O(\log^2 n)}$, that admit resolution refutations of size $n^{O(\log^2 n)}$, yet for which any Res(⊕) refutation of size at most $2^{n/\log^4 n}$ must have depth $\Omega(n \log n)$. This depth lower bound exceeds the trivial worst-case upper bound of n. Moreover, applying Theorem 1.1 with other values of parameters shows that Depth-$n^{4/3-\varepsilon}$ Res(⊕) does not polynomially simulate resolution for any $\varepsilon > 0$.

1.2 Polynomial Lower Bounds for Unrestricted Res(⊕)

Khaniki [33] proved an almost quadratic lower bound $\Omega\left(\frac{n^2}{\log^{\log\log n} n}\right)$ for a dag-like version of Res(⊕) with *syntactic weakening rules*. It remains unclear whether this lower bound persists for the semantic weakening rules that have since become standard. Although these variants of Res(⊕) are polynomially related [32], this relationship does not yield polynomial lower bounds.

Prior to our work, the only known superlinear lower bound for the standard version of Res(⊕) (i.e., with semantic weakening) followed from the results of Byramji and Impagliazzo [16]. Specifically, their size–depth tradeoff implies an $\Omega\left(\frac{n^2}{\log n}\right)$ lower bound on the size of Res(⊕) refutations of Tseitin formulas over constant-degree expanders lifted by the MAJ_3 gadget.

1.3 Our Goals

The main goal of this paper is to identify a standard complexity measure of CNF formulas that enables lifting to size lower bounds for Res($\oplus$) refutations of depth up to $n^{2-\varepsilon}$. Existing lifting results by Bhattacharya and Chattopadhyay [13] and by Byramji and Impagliazzo [16] rely on a rather strong and nonstandard assumption of $(\Omega(n), \Omega(n))$-DT-hardness. Our objective is to replace this assumption with a more natural and well-studied measure.

Motivated by the lifting result of Itsykson and Knop [29], which applies to smaller depths, we aim to show that resolution width – defined as the minimum possible width of a resolution refutation – can serve as an appropriate source of hardness.

Resolution width is a standard and extensively studied measure in proof complexity. Since the seminal work of Ben-Sasson and Wigderson [10], which showed that resolution width lower bounds of $n^{1/2+\varepsilon}$ imply exponential lower bounds on resolution size, proving width lower bounds has become the standard approach for establishing size lower bounds in resolution. Atserias and Dalmau [5] provided a game characterization of resolution width that offers a general framework for proving such lower bounds and, moreover, implies that clause space in resolution is also lower bounded by resolution width. This characterization readily implies that any $(\Omega(n), \Omega(n))$-DT-hard formula has resolution width at least $\Omega(n)$.

Alekhnovich and Razborov (as attributed in the paper by Ben-Sasson [9]) demonstrated that resolution width can be lifted via the parity gadget $\oplus_2$ to resolution size: if a formula φ requires resolution width w, then the composed formula $\varphi \circ \oplus_2$ requires resolution size $2^{\Omega(w)}$.

Finally, Garg, Göös, Kamath, and Sokolov [25] showed that resolution width lifts via an indexing gadget to size lower bounds in cutting planes and in DAG-like communication protocols. The latter result, in particular, implies lower bounds for a wide range of proof systems as well as for the size of monotone Boolean circuits.

Another goal of the paper is to address the near-quadratic lower bound for Res($\oplus$) established via the size–depth tradeoff of [16]. We aim to obtain a direct proof of such a lower bound, or possibly of a stronger one.

1.4 Our Contributions

Recall that a Boolean function $g : \{0,1\}^n \to \{0,1\}$ is called *1-stifling* if, for every $a \in \{0,1\}$ and every choice of $n-1$ input variables, there exists an assignment to these variables that fixes the value of g to a, independently of the value of the remaining variable. The simplest example of a 1-stifling gadget is the 3-bit Majority function MAJ_3.

Our main result is the following lifting theorem.

▶ **Theorem 1.2** (Theorem 6.1). *Assume that every resolution refutation of an unsatisfiable CNF formula φ has either width at least w, or depth at least h. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget, where ℓ is a constant. Then any* Res($\oplus$) *refutation of $\varphi \circ g$ of size at most S has depth at least $\Omega\left(\min\{h, w^2/\log S\}\right)$.*

For $h = \infty$, Theorem 1.2 yields a lifting theorem from resolution width.

▶ **Corollary 1.3.** *Assume that every resolution refutation of an unsatisfiable CNF formula φ has width at least w. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget, where ℓ is a constant. Then any* Res($\oplus$) *refutation of $\varphi \circ g$ of size at most S has depth at least $\Omega\left(w^2/\log S\right)$.*

Corollary 1.3 implies that, starting from formulas requiring resolution width $\Omega(n)$, one obtains 2^{n^ε} lower bounds for Depth-$n^{2-\varepsilon}$ Res($\oplus$) refutations, as in [13, 16]. Moreover, if the starting formula is an $O(1)$-CNF and has $O(n)$ clauses, then, as in [16], one can obtain

formulas for which the size–depth tradeoff yields nearly quadratic depth in terms of the *formula size*, rather than the number of variables. The advantage of our result compared to [13, 16] is that it relies on a standard complexity measure and applies to a wide range of formulas, thanks to the numerous known lower bounds on resolution width.

Let us compare Theorem 1.2 with the main result of [29], namely Theorem 1.1. First, Theorem 1.2 relies on a significantly simpler gadget: any 1-stifling function, as opposed to a composition of parity with a 2-stifling gadget (namely MAJ_5). Second, Theorem 1.2 yields larger depth lower bounds without a substantial increase in the size of the formula.

Combining Theorem 1.2 with the supercritical width–depth tradeoff for resolution due to Buss and Thapen [14], we obtain the following theorem.

▶ **Theorem 1.4** (Theorem 6.4). *There is a family of formulas ψ_n from n variables of size* $\mathrm{poly}(n)$ *such that ψ_n has polynomial size resolution refutation for any $S > 0$, any* $\mathrm{Res}(\oplus)$ *refutation of size S has depth at least $\Omega(n^2/(\log^2 n \log S))$.*

Theorem 1.4 improves the result of [29] in two respects. First, the formulas ψ_n have polynomial size and admit polynomial-size resolution refutations, whereas in [29] the corresponding formulas were only of quasipolynomial size. This resolves an open question posed in [29]. Second, the theorem applies to a wider range of depths, which can be used to obtain stronger separation between resolution and bounded depth $\mathrm{Res}(\oplus)$.

▶ **Corollary 1.5** (Corollary 6.5). *If $d(n) = o(n^2/\log^4 n)$, then Depth-$d(n)$* $\mathrm{Res}(\oplus)$ *does not polynomially simulate resolution.*

Size lower bound

Applying Corollary 1.3 to an $O(1)$-CNF formula φ with n variables and $O(n)$ clauses that requires resolution width $\Omega(n)$, and choosing $S = n^2$ and $g = \mathrm{MAJ}_3$, we obtain that every $\mathrm{Res}(\oplus)$ refutation of $\varphi \circ \mathrm{MAJ}_3$ (which has size $O(n)$) with size at most n^2 must have depth at least $\Omega\left(\frac{n^2}{\log n}\right)$. The same lower bound can also be derived from the results of [16]. We strengthen this conclusion by obtaining a *pure* quadratic lower bound. This improvement holds for the same class of formulas as above and is established by the following theorem.

▶ **Theorem 1.6** (Theorem 7.4). *Let φ be an unsatisfiable CNF formula such that every resolution refutation of φ has width at least w. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget. Then any* $\mathrm{Res}(\oplus)$ *refutation of $\varphi \circ g$ has size at least $\frac{w(w+1)}{2}$.*

1.5 Our Techniques

We always use the top-down definition of $\mathrm{Res}(\oplus)$ as a parity decision DAG [20]. A *parity decision DAG* refuting an unsatisfiable CNF formula φ is a directed acyclic graph with a single source and several sinks, satisfying the following properties:

- Each node v of the DAG is labeled with an $\mathbb{F}_2$-linear system Φ_v over the variables of φ.
- The source is labeled with the empty system (i.e., identically true).
- For every sink v, there exists a clause C of φ such that Φ_v is inconsistent with C.
- Every non-sink node v is additionally labeled with a linear form f_v and has two children v_0 and v_1. The edge (v, v_0) is labeled by the equation $f_v = 0$, and the edge (v, v_1) is labeled by $f_v = 1$. Moreover, for each $\alpha \in \{0,1\}$, the system $\Phi_v \wedge (f_v = \alpha)$ semantically implies Φ_{v_α}.

Quadratic size lower bound

It is easier to start with the proof idea of Theorem 1.6.

Let $\varphi(y_1, \dots, y_m)$ be an unsatisfiable CNF formula whose resolution width is at least w, and let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget. Recall that the lifted formula $\varphi \circ g$ is over the variable set $\{x_{i,j} \mid i \in [m], j \in [\ell]\}$. Suppose that G is a parity decision DAG refuting $\varphi \circ g$. Our goal is to show that the size of G is at least $\frac{w(w+1)}{2}$.

We prove this by induction on w. For the induction step, it suffices to construct a restriction ρ such that:

- the restricted DAG $G|_\rho$ refutes $\varphi' \circ g$, where φ' is an unsatisfiable formula of resolution width at least $w-1$; and
- the restriction ρ falsifies at least w linear systems labeling nodes of G. Consequently, the size of $G|_\rho$ is smaller than the size of G by at least w.

Once such a restriction is obtained, the induction follows immediately.

Fix an index $i \in [m]$ and choose $a \in \{0,1\}$ such that the restricted formula $\varphi|_{y_i=a}$ has resolution width at least $w-1$. Since g is 1-stifling, there exists a partial assignment σ that assigns Boolean values to all variables $\{x_{i,j} \mid j \in [\ell]\}$ except for a single variable x_{i,j_0}, in such a way that the value of $g(x_{i,1}, \dots, x_{i,\ell})$ is fixed to a, independently of the value of x_{i,j_0}.

Observe that $G|_\sigma$ is a parity decision DAG refuting $\varphi|_{y_i=a} \circ g$. Hence, we may assume that the variable x_{i,j_0} does not appear in the clauses falsifying at the sinks of $G|_\sigma$. Therefore, we may substitute an arbitrary affine form h for x_{i,j_0} – that is, define

$$\rho := \sigma \cup \{x_{i,j_0} := h\},$$

and the resulting DAG $G|_\rho$ still refutes $\varphi|_{y_i=a} \circ g$.

Hence, it suffices to find an affine equation that is semantically implied by at least w linear systems labeling nodes of G. Once such an equation is identified, we choose the variable x_{i,j_0} appearing in it and substitute for it an affine form that falsifies the equation, thereby eliminating all these nodes.

By the result of Alekseev and Itsykson [4], any parity decision DAG refuting $\varphi \circ g$ contains a node v whose associated linear system Φ_v has rank at least w. Using standard properties of parity decision DAGs, one can show that along the path from the source to v there exists an affine equation that is implied by at least w of the linear systems labeling nodes on this path. This completes the induction step.

Lower bound for bounded-depth $\mathbf{Res}(\oplus)$

Next, we describe our proof strategy for our main lifting result. For simplicity, we concentrate on the proof of Corollary 1.3.

Again, consider a parity decision DAG G of size S refuting $\varphi \circ g$.

The proof goes by induction on the resolution width w of the CNF φ. The key step of the proof is to find a node v in G on depth at least $w/6$ and to substitute some of the variables of $\varphi \circ g$ by affine functions of the remaining variables in such a way that the subgraph of G with the root in v refutes $\varphi' \circ g$ and the resolution width of φ' is at least $w - O(\log S)$. Once we show how to find such a v, the result follows easily.

To find the vertex v, similarly to the previous papers, we adopt the random-walk-with-restart technique. We follow a random path in G starting from the root for $O(w)$ steps. We show that if the rank of the linear system Φ_v in the final vertex of the path is d, then this vertex can be reached by the random path only with probability $2^{-\Omega(d)}$ (here, for simplicity we assume that the size ℓ of the gadget g is constant). Thus, if all final vertices have linear

systems of rank larger than $O(\log S)$, then there are more than S of them, which contradicts the restriction on the size of G. As a result, there is a vertex v that has a system of rank at most $O(\log S)$. We show that in this case, we can fix $O(\log S)$ input variables of φ to obtain a refutation of $\varphi' \circ g$ with the desired bound on the resolution width of φ'.

To achieve that, as in the previous case (that is, the quadratic size lower bound), we fix in each block of variables of $\varphi \circ g$ all but one variable to 0/1 constants. These values are chosen – again as in the previous case – so as to reduce the width of φ by at most one per block. The remaining unfixed variables are then substituted by affine functions chosen to satisfy the linear system Φ_v. A sufficient condition ensuring that such a restriction satisfies the linear system is formulated in terms of *closure* (see Lemma 4.5 for details). By the properties of closure, it suffices to verify this condition for the linear system written along the edges of the path. This is the standard use of closure, dating back to its introduction in [20].

The key novelty of our argument is the way we pick a random path. Previous papers picked a random input and followed the path corresponding to this input. Instead, we gradually fix the input during the random walk: on each step of the walk, we fix some bits in order to be able to move to the next vertex. For this, in each intermediate vertex v of the path, we consider the linear form f_v queried in v. We pick some i such that $\{x_{i,j} \mid j \in [\ell]\}$ contains a variable x_{ij} of f. We consider the variable y_i of φ and pick the value $a \in \{0,1\}$ for it such that, informally speaking, fixing $y_i = a$ decreases the resolution width of φ by at most 1. Then we flip a coin and do one of the following with probabilities 1/2:

1. Assign the values to variables in $\{x_{i,j} \mid j \in [\ell]\}$ uniformly at random among $g^{-1}(a)$.
2. Assign all variables of $\{x_{i,j} \mid j \in [\ell]\}$ except x_{ij} in such a way that g evaluates to a regardless of the value of x_{ij} (this is possible since g is 1-stifling). Then we fix x_{ij} to be the linear combination of other variables of f in such a way that f evaluates to 0 and 1 with probabilities 1/2.

Next, we repeat the procedure, in the second case in the next vertex of G, and in the first case potentially in the same vertex of G.

The combination of these two actions allows us to show that the probability that the path reaches any given vertex labeled by a linear system with large rank is small. Here we use properties of 1-stifling gadget again (see Lemma 2.5). Additionally, the second action ensures that the random path makes many steps in G.

Organization

The rest of the paper is organized as follows. In Section 2, we give the necessary definitions and prove the basic properties of 1-stifled gadgets. In Section 3, we study proof-complexity properties of CNF formulas via families of games and winning strategies, which are typically represented as sets of partial assignments. In Section 4, we analyze the application of affine restrictions to parity decision DAGs computing lifted formulas, and we present a sufficient condition under which a linear system can be satisfied by an affine restriction of small size. In Section 5, we discuss the construction of the random paths and their properties. In Section 6, we combine everything together to obtain our lifting result and its corollaries. Finally, in Section 7, we prove a quadratic size lower bound.

2 Preliminaries

Throughout this paper, all scalars are from the field $\mathbb{F}_2$. Let X be a set of variables taking values in $\mathbb{F}_2$. A linear form in variables from X is a homogeneous linear polynomial over $\mathbb{F}_2$ in variables from X or, in other words, a polynomial $\sum_i^n x_i a_i$, where $x_i \in X$ is a variable

and $a_i \in \mathbb{F}_2$ for all $i \in [n]$. An affine form is an arbitrary linear polynomial over $\mathbb{F}_2$, i.e. $\sum_i^n x_i a_i + a_0$, where $x_i \in X$ is a variable and $a_i \in \mathbb{F}_2$ for all $i \in \{0, 1, \ldots, n\}$. A linear equation is an equality $f = a$, where f is a linear form and $a \in \mathbb{F}_2$.

2.1 CNF Formulas

A *literal* is either a propositional variable or its negation. For a propositional variable x, we write $x^0 := \neg x$ and $x^1 := x$.

A *clause* is a disjunction of literals. The empty clause is identified with the constant false. A *CNF formula* is a conjunction of clauses, and the empty conjunction is identified with the constant true.

Observe that a conjunction of CNF formulas is again a CNF formula. Let $\phi_1, \phi_2, \ldots, \phi_k$ be CNF formulas, and assume that for each $i \in [k]$, $\phi_i = \bigwedge_{j \in K_i} C_{i,j}$, where each $C_{i,j}$ is a clause. We define the *standard CNF representation* of the disjunction $\bigvee_{i=1}^k \phi_i$ to be the CNF formula

$$\bigwedge_{(j_1,\ldots,j_k) \in K_1 \times \cdots \times K_k} (C_{1,j_1} \vee C_{2,j_2} \vee \cdots \vee C_{k,j_k}) . \tag{1}$$

2.2 Partial Assignments and Affine Restrictions

Affine restrictions

By an *affine restriction* we mean a set of assignments of the form $\rho = \{x_1 := f_1, x_2 := f_2, \ldots, x_k := f_k\}$, where $x_1, \ldots, x_k$ are distinct variables, each $f_1, \ldots, f_k$ is an $\mathbb{F}_2$-affine form, and none of the variables $x_1, \ldots, x_k$ occurs in any of the forms $f_1, \ldots, f_k$. The set $\{x_1, \ldots, x_k\}$ is called the *support* of ρ.

Partial assignments

An affine restriction ρ is called *plain*, or a *partial assignment*, if all affine forms f_i from the right hand sides are 0/1-constants.

We say that a partial assignment ρ *satisfies* a literal x^a if $x := a \in \rho$, and *falsifies* x^a if $x := 1 - a \in \rho$.

Let C be a clause and let ρ be a partial assignment. The restricted clause $C|_\rho$ is defined as follows. If ρ satisfies at least one literal of C, then $C|_\rho$ is the constant 1, and we say that ρ satisfies C. Otherwise, $C|_\rho$ is obtained from C by deleting all literals falsified by ρ. We say that ρ *falsifies* C if $C|_\rho$ is the empty clause.

For a CNF formula ϕ and a partial assignment ρ, we define the restricted CNF formula $\phi|_\rho$ as follows. The formula $\phi|_\rho$ is the constant 0 if ρ falsifies any clause of ϕ; otherwise,

$$\phi|_\rho = \bigwedge_{C \in \phi \,:\, \rho \text{ does not satisfy } C} C|_\rho .$$

2.3 Resolution

Let φ be an unsatisfiable CNF formula. A resolution refutation of φ is a sequence of clauses $C_1, C_2, \ldots, C_s$ such that C_s is the empty clause (i.e., identically false) and for every $i \in [s]$ the clause C_i is either a clause of φ or is obtained from previous clauses by the *resolution rule* that allows us to derive a clause $C \vee D$ from clauses $C \vee x$ and $D \vee \neg x$.

The *size* of a resolution refutation is the number of clauses in it. The *depth* of a resolution refutation is the length of the longest path between the empty clause and the clause of the original formula. The *width* of a resolution refutation is the maximal size of a clause from the refutation, where the size of a clause is the number of literals it contains. The resolution width of an unsatisfiable CNF formula φ is the minimal possible width over all resolution refutations of φ.

2.4 Resolution Over Parities

A *linear clause* is a disjunction of $\mathbb{F}_2$-linear equations: $\bigvee_{i=1}^{t}(f_i = a_i)$. Note that over $\mathbb{F}_2$ a linear clause $\bigvee_{i=1}^{t}(f_i = a_i)$ may be represented as the negation of a linear system: $\neg\bigwedge_{i=1}^{t}(f_i = a_i + 1)$.

Now we define the proof system resolution over parities (Res(⊕)) [32].

Let φ be an unsatisfiable CNF formula. A Res(⊕) refutation of φ is a sequence of linear clauses $C_1, C_2, \ldots, C_s$ such that C_s is the empty clause (i.e., identically false) and for every $i \in [s]$ the clause C_i is either a clause of φ or is obtained from previous clauses by one of the following inference rules:

- *Resolution rule* allows us to derive a linear clause $C \vee D$ from linear clauses $C \vee (f = a)$ and $D \vee (f = a + 1)$.
- *Weakening rule* allows us to derive from a linear clause C any linear clause D in the variables of φ that semantically follows from C (i.e., any assignment satisfying C also satisfies D).

The *size* of a Res(⊕) refutation is the number of linear clauses in it. The *depth* of a Res(⊕) refutation is the maximal number of resolution rules applied on a path between a clause of the initial formula and the empty clause. Note that weakening rules are not counted in the definition of the depth. The *width* of a Res(⊕) refutation is the maximal rank of the negation of a linear clause from the refutation.

▶ Remark 2.1. A resolution refutation of a formula φ is a special case of a Res(⊕) refutation, where all linear clauses are plain (i.e., disjunctions of literals).

For any function $f(n)$, we denote by Depth-$f(n)$ Res(⊕) the subsystem of Res(⊕) consisting of refutations with depth at most $f(n)$, where n is the number of variables in the formula being refuted.

For a linear clause C we denote by $L(C)$ the set of linear forms that appear in C; i.e. $L\left(\bigvee_{i=1}^{t}(f_i = a_i)\right) = \{f_1, f_2, \ldots, f_t\}$. The same notation we use for linear systems: if Ψ is a $\mathbb{F}_2$-linear system, $L(\Psi)$ denotes the set of all linear forms from Ψ.

2.5 Parity Decision DAG

We say that an $\mathbb{F}_2$-linear system Φ_1 *semantically implies* an $\mathbb{F}_2$-linear system Φ_2 if every assignment satisfying Φ_1 also satisfies Φ_2. We denote semantic implication by $\Phi_1 \models \Phi_2$.

A *parity decision DAG* (also known as an affine DAG) is a directed acyclic graph with a single source and several sinks, satisfying the following properties:

- Each node v of the DAG is labeled with an $\mathbb{F}_2$-linear system Φ_v.
- Every non-sink node v is additionally labeled with a linear form f_v and has two children v_0 and v_1. The edge (v, v_0) is labeled by the equation $f_v = 0$, and the edge (v, v_1) is labeled by $f_v = 1$. Moreover, for each $\alpha \in \{0, 1\}$, the system $\Phi_v \wedge (f_v = \alpha)$ semantically implies Φ_{v_α}.

We say that a parity decision DAG refutes an unsatisfiable CNF formula φ if all linear systems and linear forms used as labels are over $\mathrm{Vars}(\varphi)$ and

- The source is labeled with the empty system (i.e., identically true).
- For every sink v, there exists a clause C of φ such that Φ_v is inconsistent with C.

The size of a parity decision DAG is the number of nodes, the depth is the length of the longest source-to-sink path, and the width is the maximal rank of the linear systems labeling nodes of the DAG.

Similarly to the case of resolution, it is known [20] that every Res($\oplus$) refutation of φ can be efficiently transformed into a parity decision DAG refuting φ without increasing its size, width, and depth.

▶ **Lemma 2.2** ([20]). *Let G be a parity decision DAG, and let u and v be two nodes such that there exists a path p from u to v. Let Ψ_p denote the linear system consisting of the conjunction of all equations labeling the edges along p. Then $\Phi_u \wedge \Psi_p \models \Phi_v$.*

2.6 Lifting Settings

We assume that $Y = \{y_1, y_2, \dots, y_m\}$ is the set of variables of an unlifted formula φ.

Let $g : \{0,1\}^\ell \to \{0,1\}$ be a Boolean function that we refer to as a gadget. We now define a lifted version of the formula φ by a gadget g. The set of variables X of the lifted formula is partitioned into m blocks, where each block corresponds to a variable from Y. The variables in X are indexed by two indices, where the first index indicates the block: $X = \{x_{i,j} \mid i \in [m]\ j \in [\ell]\}$.

We define the lifted formula $\varphi \circ g$ as follows. For every $i \in [m]$, we substitute the variable y_i by $g(x_{i,1}, x_{i,2}, \dots, x_{i,\ell})$. We then convert the resulting formula into CNF in a clause-wise manner. That is, for each clause C of φ, we perform the substitution to obtain a formula $C \circ g$, convert it to CNF, and finally take the conjunction of all resulting clauses over all C.

We now explain how to convert $C \circ g$ into CNF. Let $C = \bigvee_{s=1}^{t} y_{i_s}^{a_s}$ be a clause. Then $C \circ g = \bigvee_{s=1}^{t} g^{a_s}(x_{i_s,1}, x_{i_s,2}, \dots, x_{i_s,\ell})$. Assume that for every $a \in \{0,1\}$, we fix a CNF representation of the Boolean function g^a. Under this assumption, $C \circ g$ is represented in CNF using the standard encoding of a disjunction of CNF formulas, as specified in (1).

An affine restriction ρ over lifted variables is called *block-respecting* if, for every block $i \in [m]$, ρ is either undefined on all variables of block i or defined on all variables of block i.

The *block-size* of ρ is the number of blocks on which ρ is defined; we denote it by $\mathrm{bsize}(\rho)$.

Let $g : \{0,1\}^\ell \to \{0,1\}$ be a gadget. For every block-respecting partial assignment τ to the lifted variables X, we define the *induced assignment* $\mathrm{induced}_g(\tau)$ as a partial assignment σ to the unlifted variables in the same set of blocks where τ is defined, such that for each $i \in [m]$ and $j \in [\ell]$, $\sigma(y_{i,j}) = g\big(\tau(x_{i,1}), \tau(x_{i,2}) \dots \tau(x_{i,\ell})\big)$. We will slightly abuse notation and use the term *induced assignment* also for partial assignments τ that are not block-respecting, provided that in every block containing at least one variable assigned by τ, the assigned values uniquely determine the value of g, independently of the values of the unassigned variables in that block. Formally, in this case we consider a minimal extension τ' of τ to a block-respecting assignment and define $\mathrm{induced}_g(\tau)$ as $\mathrm{induced}_g(\tau')$.

2.7 Stifling and Subspace-Resilient Gadgets

In this section, we introduce and discuss the properties of gadgets that are needed for the lifting construction.

▶ **Definition 2.3** (1-stifling function [18]). *A function $g : \{0,1\}^\ell \to \{0,1\}$ is called* 1-stifling *if for every index $i \in [\ell]$ and every output value $a \in \{0,1\}$, there exists an input $x \in \{0,1\}^\ell$ such that for all inputs $y \in \{0,1\}^\ell$ that agree with x on all coordinates in $[\ell] \setminus \{i\}$, we have $g(y) = a$.*

Our main example of 1-stifling gadget is the 3-bit majority function $\mathrm{MAJ}_3 : \{0,1\}^3 \to \{0,1\}$. It is straightforward to verify that this function is 1-stifling.

We also require a property ensuring that the preimages of g are not fully correlated with any affine subspace of codimension 1.

▶ **Definition 2.4** (1-subspace-resilient function). *A function $g : \{0,1\}^\ell \to \{0,1\}$ is called* 1-subspace-resilient *if for every affine subspace $W \subseteq \{0,1\}^\ell$ of codimension 1 and every $y \in \{0,1\}$, the preimage $g^{-1}(y)$ is not contained in W.*

Next we observe a simple quantitative version of subspace resilience.

▶ **Lemma 2.5.** *Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-subspace-resilient function, $y \in \{0,1\}$ and $W \subseteq \{0,1\}^\ell$ be an affine subspace of codimension 1. Pick element x of $g^{-1}(y)$ uniformly at random. Then $\Pr[x \notin W] \geq 2^{-l}$.*

Proof. Since g is 1-subspace-resilient, we have $g^{-1}(y) \not\subseteq W$, and hence $|g^{-1}(y) \cap W| \leq |g^{-1}(y)| - 1$. Therefore, $\Pr[x \in W] = \frac{|g^{-1}(y) \cap W|}{|g^{-1}(y)|} \leq 1 - \frac{1}{|g^{-1}(y)|} \leq 1 - 2^{-\ell}$, where the last inequality follows from the trivial bound $|g^{-1}(y)| \leq 2^\ell$. ◀

The following lemma establishes that 1-stifling functions are necessarily 1-subspace-resilient:

▶ **Lemma 2.6.** *Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling function. Then g is 1-subspace-resilient.*

Proof. Any affine subspace $W \subseteq \{0,1\}^\ell$ of codimension 1 can be described by a linear equation $a_1 x_1 + \cdots + a_\ell x_\ell = b \pmod 2$, where not all coefficients a_i are zero. Suppose, towards a contradiction, that there exists $y \in \{0,1\}$ such that the entire preimage $g^{-1}(y)$ is contained in W. Let $i \in [\ell]$ be an index with $a_i = 1$. By 1-stiflingness, there exists an input $x \in \{0,1\}^\ell$ such that $g(z) = y$ for all inputs z agreeing with x on all coordinates except possibly i. In particular, both x and $x^{\oplus i}$ belong to $g^{-1}(y)$, where $x^{\oplus i}$ denotes x with the i-th bit flipped. However, exactly one of x and $x^{\oplus i}$ satisfies the defining equation of W, since flipping the i-th bit toggles the left-hand side. This contradicts the assumption that $g^{-1}(y) \subseteq W$. Hence, g is 1-subspace-resilient. ◀

▶ **Corollary 2.7.** MAJ_3 *is 1-subspace-resilient function.*

2.8 Safe Sets and Closure

In our proofs, we use the notion of *closure*, introduced in [20], which we review and formalize in this section.

Let $X = \{x_{i,j} \mid i \in [m],\ j \in [\ell]\}$ be the set of lifted variables.

Let F be a finite set of $\mathbb{F}_2$-linear forms over a set of variables X. By $\langle F \rangle$ we denote the linear span of the set F. We say that F is *safe* if there exists a basis $f_1, f_2, \ldots, f_k$ of $\langle F \rangle$ and variables $x_{i_1,j_1},\ x_{i_2,j_2},\ \ldots,\ x_{i_k,j_k}$ such that $i_1, i_2, \ldots, i_k$ are pairwise distinct elements of $[m]$ and, for every $t \in [k]$, the variable x_{i_t,j_t} appears in f_t with a nonzero coefficient and does not appear in f_s for $s \neq t$.

For a set of blocks $S \subseteq [m]$ and a linear form f, let $f[\setminus S]$ denote the linear form obtained from f by setting to zero (equivalently, removing) all variables whose blocks belong to S. For a set of linear forms F, define $F[\setminus S] = \{f[\setminus S] \mid f \in F\}$.

A *closure* of a set of linear forms F is an inclusion-minimal set $S \subseteq [m]$ such that $F[\setminus S]$ is safe.

▶ **Lemma 2.8** ([20]). *For any F, its closure is unique.*

Since the closure of F is unique, we denote it by $\mathrm{Cl}(F)$.

We also need the following properties of closure.

▶ **Lemma 2.9** ([20]). *Closure satisfies the following properties:*

1. *(Monotonicity) If $F \subseteq G$, then $\mathrm{Cl}(F) \subseteq \mathrm{Cl}(G)$;*
2. *(Span invariant) If $\langle F \rangle = \langle G \rangle$, then $\mathrm{Cl}(F) = \mathrm{Cl}(G)$;*
3. *(Size bound) $|\mathrm{Cl}(F)| + \dim\langle F[\setminus \mathrm{Cl}(F)]\rangle \leq \dim\langle F\rangle$.*

3 Sets of Good Assignments

In this section, we define certain sets of partial assignments for unlifted formulas that capture their essential properties. We first give a general, abstract definition; in Subsection 3.1, we give a specialization that characterizes the width-depth tradeoff in resolution.

Let φ be an unsatisfiable CNF formula over the set of variables $Y = \{y_i \mid i \in [m]\}$.

▶ **Definition 3.1** (Set of w-good assignments). *For a natural number w, a set $\mathcal{H}$ is called a* set of w-good assignments *for φ if $\mathcal{H}$ consists of partial assignments of size at most w and satisfies the following conditions:*

- *The empty assignment ε belongs to $\mathcal{H}$.*
- *For every $\rho \in \mathcal{H}$, the assignment ρ does not falsify any clause of φ.*
- *If $\rho \in \mathcal{H}$ and $|\rho| < w$, then for every variable y_i on which ρ is undefined, there exists $\rho' \in \mathcal{H}$ such that $\rho \subseteq \rho'$, ρ' is defined on the variable y_i and $|\rho'| = |\rho| + 1$.*

Atserias and Dalmau [5] showed that an unsatisfiable CNF formula φ admits a set of w-good assignments that is additionally *closed under taking subassignments* if and only if every resolution refutation of φ has width greater than w. For our purposes, we require a slightly more refined characterization capturing the absence of resolution refutations that are simultaneously of small width and small depth.

3.1 Characterization of Depth-Width Tradeoff

Consider a game between two players, Alice and Bob, defined with respect to an unsatisfiable CNF formula φ and two integer parameters w and h. Throughout the game, the players maintain a partial assignment to the variables of φ. Initially, the assignment is empty, and at all times its size is required to be at most w.

On each turn, Alice has two possible moves:

- If the size of the current assignment is less than w, Alice may ask Bob for the value of some variable x of φ. Bob then chooses and assigns a value to x.
- Alice may erase one variable from the domain of the current assignment.

Alice wins if the current assignment falsifies (contradicts) at least one clause of φ. Bob wins if he can answer at least h of Alice's questions without allowing Alice to win.

Now we define a winning strategy for Bob in this game.

Let $\mathcal{B}$ be a set of pairs (ρ, i) of a partial assignment ρ and an integer number i. We say that $\mathcal{B}$ is a (w, h)-winning strategy for φ if the following conditions hold:

- $(\varepsilon, 0) \in \mathcal{B}$, where ε is an empty assignment.
- If $(\rho, i) \in \mathcal{B}$, then $|\rho| \leq w$, $i \leq h$ and ρ doesn't falsify any clause of φ.
- If $(\rho, i) \in \mathcal{B}$ and $\rho' \subseteq \rho$, then $(\rho', i) \in \mathcal{B}$.
- If $(\rho, i) \in \mathcal{B}$, $|\rho| < w$, $i < h$, and $x \in \mathrm{Vars}(\varphi) \setminus \mathrm{Dom}(\rho)$, then there exists $a \in \{0, 1\}$ such that $(\rho \cup \{x := a\}, i + 1) \in \mathcal{B}$.

It is easy to see that if there exists a (w, h)-winning strategy $\mathcal{B}$ for φ, then Bob wins the game. Indeed, for every $i \in \{0, 1, \ldots, h\}$, Bob can guarantee that after his i-th response the current partial assignment ρ satisfies $(\rho, i) \in \mathcal{B}$. Consequently, ρ does not falsify any clause of φ, and Alice cannot win within h rounds.

▶ **Lemma 3.2** ([29]). *Let $w \geq 0$ and $h \geq 0$ be some integers; and let φ be an unsatisfiable CNF formula such that φ doesn't have a resolution refutation of width at most w and simultaneously with depth at most h. Then there exists a (w, h)-winning strategy for φ.*

▶ **Lemma 3.3.** *Let $\mathcal{B}$ be a (w, h)-winning strategy for φ, and define $\mathcal{H} := \{\tau \mid (\tau, |\tau|) \in \mathcal{B}\}$. Then $\mathcal{H}$ is a $\min\{w, h\}$-good set of assignments for φ.*

Proof. Since $(\varepsilon, 0) \in \mathcal{B}$, it follows that $\varepsilon \in \mathcal{H}$.

For any $\rho \in \mathcal{H}$, we have $(\rho, |\rho|) \in \mathcal{B}$, and hence ρ does not falsify any clause of φ.

Now let $\rho \in \mathcal{H}$ satisfy $|\rho| < \min\{w, h\}$, and let y_i be a variable of φ on which ρ is not defined. Since $(\rho, |\rho|) \in \mathcal{B}$ and $|\rho| < \min\{w, h\}$, by the definition of a (w, h)-winning strategy there exists a value $a \in \{0, 1\}$ such that $(\rho \cup \{y_i := a\}, |\rho| + 1) \in \mathcal{B}$. Consequently, $\rho \cup \{y_i := a\} \in \mathcal{H}$. ◀

4 Applying Affine Restrictions to Parity Decision DAGs

In this section, we study applications of affine restrictions to parity decision DAGs and introduce the formal definition of applying an affine restriction. In the next subsection, we consider *structured affine restrictions* that can be applied to parity decision DAGs refuting lifted formulas $\varphi \circ g$. We show that, after such an application, the resulting graph is again a parity decision DAG refuting a lifted formula $\varphi' \circ g$, where φ' is obtained from φ by applying a partial assignment. Finally, in Subsection 4.2, we present a sufficient condition under which a linear system can be satisfied by a structured affine restriction of small size.

Let G be a parity decision DAG and ρ be an affine restriction. We define a parity decision DAG $G|_\rho$ as follows. First, we apply ρ to all linear systems labeling the nodes of G and to all linear equations labeling its edges. We then remove all edges whose labels become unsatisfiable under this substitution. Next, we delete all nodes whose associated linear systems become unsatisfiable, and subsequently remove all nodes that are no longer reachable from the source.

After these steps, some nodes may have an out-degree one. In this case, we merge such a node with its unique child and label the resulting node with the linear system of the child.

4.1 Affine Restrictions for Parity Decision DAGs Refuting Lifted Formulas

▶ **Proposition 4.1.** *Suppose a parity decision DAG G refutes a CNF formula φ, and let ρ be an affine restriction that decomposes as a disjoint union $\rho = \rho_1 \cup \rho_2$, where ρ_1 is a partial assignment (i.e., an assignment that sets certain variables to 0 or 1). Assume that there exists a CNF formula ψ such that every clause of $\varphi|_{\rho_1}$ contains some clause of ψ, and that ψ does not involve any variables in the support of ρ_2. Then the restricted DAG $G|_\rho$ refutes ψ.*

Proof. It is straightforward that $G|_{\rho_1}$ refutes $\varphi|_{\rho_1}$. Since every clause of $\varphi|_{\rho_1}$ contains some clause of ψ, it follows that $G|_{\rho_1}$ also refutes ψ. Finally, we have $G|_\rho = (G|_{\rho_1})|_{\rho_2}$, which refutes $\psi|_{\rho_2}$. But $\psi|_{\rho_2} = \psi$, since ψ does not involve any variables from the support of ρ_2. ◀

▶ **Proposition 4.2.** *Consider the lifted formula $\varphi \circ g$ and $i \in [m]$. Assume that a partial assignment ρ is supported on the subset of the ith block of lifted variables and assigns values to variables in such a way that, regardless of the values of the remaining unassigned lifted variables, ρ induces a partial assignment ρ' on the variable y_i. Then the formula $(\varphi \circ g)|_\rho$ is semantically equivalent to the formula $\varphi|_{\rho'} \circ g$ and, moreover, every clause of the former formula contains a clause of the latter formula.*

Proof. Since lifting is defined clause-wise, it suffices to prove the statement for an arbitrary clause C of φ. If the clause C does not contain y_i, then neither ρ affects $C \circ g$ nor ρ' affects C.

So, we can assume that $C = y_i^a \vee D$. Then $C \circ g$ is the standard CNF representation of the disjunction $g^a(x_{i,1}, x_{i,2}, \dots, x_{i,\ell}) \vee (D \circ g)$.

We consider two cases.

Case 1. ρ' satisfies y_i^a. Then, the Boolean function $g^a(x_{i,1}, x_{i,2}, \dots, x_{i,\ell})|_\rho$ is identically true, and hence every clause of the CNF formula $g^a(x_{i,1}, x_{i,2}, \dots, x_{i,\ell})$ is satisfied by ρ. Therefore, $C \circ g$ is satisfied by ρ. In this case, both formulas $(C \circ g)|_\rho$ and $C|_{\rho'} \circ g$ are identically true and thus contain no clauses in their CNF representations.

Case 2. ρ' falsifies y_i^a. Hence, the formula $g^a(x_{i,1}, x_{i,2}, \dots, x_{i,\ell})|_\rho$ is unsatisfiable.

Observe that $(C \circ g)|_\rho$ is the standard CNF representation of the disjunction

$$g^a(x_{i,1}, x_{i,2}, \dots, x_{i,\ell})|_\rho \vee (D \circ g)|_\rho.$$

Since ρ assigns variables only from the ith block, we have $(D \circ g)|_\rho = D \circ g$. So, $(C \circ g)|_\rho$ coincides with $g^a(x_{i,1}, x_{i,2}, \dots, x_{i,\ell})|_\rho \vee (D \circ g)$ and on the other hand, $C|_{\rho'} \circ g$ coincides with $D \circ g$.

As $g^a(x_{i,1}, x_{i,2}, \dots, x_{i,\ell})|_\rho$ is unsatisfiable, it follows that $(C \circ g)|_\rho$ and $C|_{\rho'} \circ g$ are semantically equivalent. Moreover, every clause of the former formula contains a clause of the latter formula. ◀

▶ **Definition 4.3** ($\mathcal{H}$-good affine restriction)**.** *Let $\mathcal{H}$ be a set of w-good assignments for an unsatisfiable CNF formula φ. An affine restriction σ on variables of $\varphi \circ g$ is called* $\mathcal{H}$-good *if it satisfies the following properties:*

- *σ is block-respecting and has block-size at most w.*
- *For every block, σ either fixes all variables to constants, or fixes all but one variable. In the latter case, the assignment to the fixed variables induces an assignment to the unlifted variable corresponding to that block, independent of the value of the remaining variable. The last variable in the block is restricted by σ to an affine function of the lifted variables from the blocks not touched by σ.*
- *The assignment induced by σ on the unlifted variables belongs to $\mathcal{H}$.*

Propositions 4.2 and 4.1 imply the following lemma.

▶ **Lemma 4.4.** *Let $\mathcal{H}$ be a set of w-good assignments for an unsatisfiable CNF φ. Let σ be an $\mathcal{H}$-good affine restriction that induces a partial assignment $\rho \in \mathcal{H}$ on the unlifted variables. Let G be a parity decision DAG refuting $\varphi \circ g$, and let v be a node of G labeled with a linear system Φ_v such that σ satisfies Φ_v. Denote by G_v the subgraph of G consisting of all nodes reachable from v. Then the restricted DAG $G_v|_\sigma$ refutes $\varphi|_\rho \circ g$.*

Proof. By definition, G_v is a parity decision DAG. Before applying σ, we cannot claim that G_v refutes $\varphi \circ g$, because the linear system at its source may be non-empty. However, after applying σ, the source of $G_v|_\sigma$ contains an empty (constant-true) linear system.

Let $\sigma = \sigma_1 \cup \sigma_2$ be a partition of σ, where σ_1 denotes the constant assignment (mapping variables to $\{0,1\}$) and σ_2 denotes the affine restriction defined on the rest of the domain. By the conditions of the lemma, σ_1 induces a partial assignment ρ on unlifted variables, independent of the values of the remaining variables in these blocks.

By applying Proposition 4.2 to all blocks on which σ_1 is defined, we obtain that the formula $(\varphi \circ g)|_{\sigma_1}$ is semantically equivalent to $\varphi|_\rho \circ g$ and every clause of the former formula contains a clause of the latter formula. Then by Proposition 4.1, the restricted DAG $G_v|_\sigma$ refutes $\varphi|_\rho \circ g$. ◀

4.2 Satisfying a Linear System by an $\mathcal{H}$-Good restriction

Since we consider only 1-stifling gadgets, for each lifted block it is possible to fix all but one of the lifted variables, thereby uniquely determining corresponding value of the unlifted variable. We exploit this property in the following lemma.

▶ **Lemma 4.5.** *Let g be a 1-stifling gadget and $\mathcal{H}$ be a set of w-good assignments for an unsatisfiable CNF φ. Suppose that a linear system Φ over lifted variables has a solution ρ such that the induced assignment* $\mathrm{induced}_g(\rho)$*, when restricted to the variables with indices in* $\mathrm{Cl}(L(\Phi))$*, belongs to $\mathcal{H}$. Assume further that* $\dim(L(\Phi)) \leq w$*. Then Φ can be satisfied by an $\mathcal{H}$-good restriction of block-size at most* $\dim(L(\Phi))$*.*

Proof. Let σ_1 be the restriction of ρ to the lifted variables from the blocks in $\mathrm{Cl}(L(\Phi))$. By assumption, σ_1 induces an assignment $\tau \in \mathcal{H}$ on the corresponding unlifted variables.

The restricted system $\Phi|_{\sigma_1}$ has a safe set of linear forms and is satisfiable. Hence, $\Phi|_{\sigma_1}$ can be equivalently rewritten in the form $\bigwedge_{s=1}^{r} x_{i_s,j_s} = a_s$, where the indices $i_1, i_2, \ldots, i_r$ are pairwise distinct, each a_s is an affine function that does not depend on the variables $x_{i_1,j_1}, x_{i_2,j_2}, \ldots, x_{i_r,j_r}$, and r is the rank of the system $\Phi|_{\sigma_1}$.

Observe that $r = \dim\langle L(\Phi)[\setminus \mathrm{Cl}(L(\Phi))]\rangle$. By Lemma 2.9, $\dim(L(\Phi)) \geq |\mathrm{Cl}(L(\Phi))| + \dim\langle L(\Phi)[\setminus \mathrm{Cl}(L(\Phi))]\rangle$, and therefore $r + |\mathrm{Cl}(L(\Phi))| \leq w$.

We now construct an $\mathcal{H}$-good restriction σ satisfying Φ. Extend σ_1 as follows. For each $s \in [r]$, assign $x_{i_s,j_s} := a_s$. Next, extend the induced assignment τ to the unlifted variable y_{i_s} so that τ remains in $\mathcal{H}$, and assign values to all remaining lifted variables in block i_s so as to force the corresponding unlifted block variables to agree with this extension of τ. Such an assignment is possible because g is 1-stifling.

The resulting restriction σ satisfies Φ, and its block-size is $|\mathrm{Cl}(L(\Phi))| + \dim\langle L(\Phi)[\setminus \mathrm{Cl}(L(\Phi))]\rangle$, which is at most $\dim(L(\Phi))$. ◀

5 Distribution on Paths

In this section, we present our main technical contribution: the construction of a random path.

Throughout this section, let φ be an unsatisfiable CNF formula over variables $Y = \{y_i \mid i \in [m]\}$, and let $g : \{0,1\}^\ell \to \{0,1\}$ be a gadget. We write $X = \{x_{i,j} \mid i \in [m], , j \in [\ell]\}$ for the set of lifted variables, that is, the variables of the lifted formula $\varphi \circ g$.

Let G be a parity decision DAG refuting $\varphi \circ g$. Consider the set of paths $\mathcal{P}(G)$ in G that start at the source. For a path $p \in \mathcal{P}(G)$, denote by Ψ_p the linear system consisting of linear equations written on the edges of p.

▶ **Theorem 5.1.** *Assume that $g : \{0,1\}^\ell \to \{0,1\}$ is a 1-stifling gadget. Let $\mathcal{H}$ be a set of w-good partial assignments for φ, where w is a positive integer.*

Consider a parity decision DAG G refuting $\varphi \circ g$. There exists a distribution $\mathcal{D}$ supported on the paths, $\mathcal{P}(G)$, from the source of G with the following properties:

1. *With probability at least $1/4$, the length of a path $p \sim \mathcal{D}$ is at least $(w-1)/6$.*
2. *For every p from the support of $\mathcal{D}$, a linear system Ψ_p has a solution τ and a set $J \subseteq [m]$ such that the restriction of $\mathrm{induced}_g(\tau)$ to variables from blocks J belongs to $\mathcal{H}$ and $\mathrm{Cl}(L(\Psi_p)) \subseteq J$.*
3. *For every linear system Φ over variables X, $\Pr_{p\sim\mathcal{D}}[\Psi_p \models \Phi] \le \exp\left(-\frac{1}{2^{\ell+1}\ell} \cdot \dim(L(\Phi))\right)$.*

The rest of this section is devoted to the proof of Theorem 5.1.

5.1 Construction of the Distribution $\mathcal{D}$

We define the distribution $\mathcal{D}$ by specifying a randomized process that generates a path $p \sim \mathcal{D}$. The path p is constructed iteratively, where each step extends the path built so far. Initially, p has length 0 and consists of a single node – the source of G.

Along with the path, we maintain an $\mathcal{H}$-good affine restriction ρ, which is initially empty. Throughout the process, I denotes the set of blocks on which ρ is defined, and σ denotes the partial assignment to the unlifted variables induced by ρ. Since ρ is $\mathcal{H}$-good, we have $\sigma \in \mathcal{H}$.

We will also view affine restrictions as systems of linear equations. Specifically, an affine restriction $\{x_1 := h_1, x_2 := h_2, \ldots, x_k := h_k\}$ corresponds to the linear system $\bigwedge_{i=1}^{k}(x_i + h_i = 0)$.

Invariants

We maintain the following invariants throughout the construction:

- ρ is an $\mathcal{H}$-good affine restriction defined on the set of blocks I, and ρ induces the partial assignment σ to the unlifted variables. Recall that this means:
  - ρ is defined on the lifted variables belonging to the blocks in I;
  - for every $i \in I$, ρ either assigns 0/1-constants to all lifted variables of the ith block, or assigns constants to all but one lifted variable. In the latter case, regardless of the value of the remaining variable, the value of the unlifted variable is already determined by the gadget g;
  - ρ induces σ on the unlifted variables from the blocks in I, and $\sigma \in \mathcal{H}$.
- The length of p is at most $w/2$, and $|I| \le w/2$;
- ρ, viewed as a linear system, semantically implies Ψ_p.

Variables in the domain of ρ that are not assigned constants by ρ are called *principal variables*. Since ρ is $\mathcal{H}$-good, each block in I contains at most one principal variable.

It is immediate that these invariants hold initially. In Claim 5.3 below, we will establish that as long as the invariants are preserved, the endpoint of the path p cannot be a sink of G.

Sampling algorithm

While $|p| < \lfloor w/2 \rfloor$ and $|I| \le \lfloor w/2 \rfloor$, we attempt to extend the path p. Once either $|p|$ or $|I|$ reaches $\lfloor w/2 \rfloor$, the process terminates and outputs the path p.

Path prolongation

Let f be the linear form queried at the endpoint of the current path p, and consider the restriction $f|_\rho$.

- While $f|_\rho$ is not a constant and $|I| < \lfloor w/2 \rfloor$, perform the following steps:
  1. Choose a variable $x_{i,j}$ appearing in $f|_\rho$.
  2. Let $\alpha \in \{0,1\}$ be such that $\sigma \cup \{y_i := \alpha\} \in \mathcal{H}$.
  3. Toss a fair coin $c \in \{0,1\}$.
  4. **Branch A: sample the ith block.** If $c = 1$:
     a. Sample β uniformly at random from $g^{-1}(\alpha)$; recall that $g^{-1}(\alpha) \subseteq \{0,1\}^\ell$.
     b. Extend ρ on block i as follows:
        - Let π denote the partial assignment $\{x_{i,1} := \beta_1, \ldots, x_{i,\ell} := \beta_\ell\}$;
        - update $\rho := \rho|_\pi$, which substitutes the values of π into the right-hand sides of the assignments in ρ;
        - set $\rho := \rho \cup \pi$.
  5. **Branch B: assign a principal variable.** If $c = 0$:
     a. Consider a partial assignment π that sets all lifted variables of the ith block except $x_{i,j}$ so that the value of the gadget g on block i equals α, regardless of the value of $x_{i,j}$. Such an assignment exists since g is 1-stifling.
     b. Toss a fair coin $b \in \{0,1\}$.
     c. Extend ρ on block i as follows:
        - update $\rho := \rho|_\pi$, substituting the values of π into the right-hand sides of the assignments in ρ;
        - update $\rho := \rho|_{x_{i,j} := f|_{\rho\cup\pi} + x_{i,j} + b}$, substituting the new value of $x_{i,j}$ into the right-hand sides of the assignments in ρ;
        - set $\rho := \rho \cup \pi \cup \{x_{i,j} := f|_{\rho\cup\pi} + x_{i,j} + b\}$.
- If $f|_\rho$ is a constant, extend the path p along the edge labeled by the equation $f = f|_\rho$.
- If $|I| = \lfloor w/2 \rfloor$ or $|p| = \lfloor w/2 \rfloor$, terminate the process and return p.

Invariant verification

Observe that in both branches A and B, the restriction ρ remains $\mathcal{H}$-good. It is also immediate that $|p| \le w/2$ and $|I| \le w/2$ is preserved throughout the process.

Whenever the path p is extended by an edge labeled with an equation $f = a$, this equation is semantically implied by the current restriction ρ. Since each update to ρ only strengthens the linear system – so that the updated system semantically implies the previous one, it follows that at every stage the current restriction ρ semantically implies Ψ_p. Hence, all invariants are preserved.

Algorithm correctness

▷ Claim 5.2. Let π be a block-respecting partial assignment defined on the lifted variables of the blocks in $[m] \setminus I$. Then the linear system $\rho \wedge \pi$ is satisfiable.

Proof. Without loss of generality, assume that π is defined on all lifted variables from the blocks in $[m] \setminus I$. The system $\rho \wedge \pi$ is satisfiable because it suffices to assign values to the principal variables consistently with the values of the linear forms to which they are equated by ρ, as determined by π. ◁

▷ Claim 5.3. If the invariants are satisfied, then the endpoint of the path p cannot be a sink of G.

Proof. It suffices to show that for every clause of $\varphi \circ g$, the system Ψ_p admits a solution that satisfies this clause. For this it is enough to prove that for any clause C of φ, there exists a solution to the linear system ρ that satisfies $C \circ g$.

Since $\sigma \in \mathcal{H}$, the partial assignment σ does not falsify C. Therefore, there exists a full assignment σ' to the unlifted variables that satisfies C. Now consider a block-respecting partial assignment π defined on the lifted variables of the blocks in $[m] \setminus I$ such that, on the unlifted variables, π induces the values prescribed by σ' on the corresponding blocks.

By Claim 5.2, the linear system $\rho \wedge \pi$ is satisfiable. By construction of the lifting, any solution to $\rho \wedge \pi$ satisfies $C \circ g$. This concludes the proof. ◁

Proof of Condition (1) in Theorem 5.1

▷ Claim 5.4. With probability at least $1/4$, the returned path has length at least $(w-1)/6$.

Proof. The process terminates once either $|p| = \lfloor w/2 \rfloor$ or $|I| = \lfloor w/2 \rfloor$.

In the first case, we immediately have $|p| \geq \lfloor w/2 \rfloor \geq (w-1)/6$, and there is nothing to prove. We therefore focus on the second case, where $|I| = \lfloor w/2 \rfloor$ at termination.

Observe that every time Branch B is executed (that is, when the coin c lands on 0), the restriction $f|_\rho$ becomes a constant, and hence the path p is extended by one edge. Consequently, the length of p is at least the number of times the coin c takes the value 0.

Each iteration of the loop increases $|I|$ by one, so the total number of coin tosses is exactly $\lfloor w/2 \rfloor$. Since c is uniformly distributed in $\{0,1\}$, the expected number of times $c = 1$ is $\frac{1}{2}\lfloor w/2 \rfloor$.

By Markov's inequality, the probability that the number of ones is at least $\frac{4}{3} \cdot \frac{1}{2}\lfloor w/2 \rfloor = \frac{2}{3}\lfloor w/2 \rfloor$ is at most $3/4$. Therefore, with probability at least $1/4$, the number of ones is less than $\frac{2}{3}\lfloor w/2 \rfloor$, which implies that the number of zeros is greater than $\frac{1}{3}\lfloor w/2 \rfloor$.

Hence, with probability at least $1/4$, we have $|p| \geq \frac{1}{3}\lfloor w/2 \rfloor \geq (w-1)/6$ as claimed. ◁

Proof of Condition (2) in Theorem 5.1

▷ Claim 5.5. If the invariants are satisfied, then Ψ_p admits a solution τ and there exists a set $J \subseteq [m]$ such that the restriction of $\mathrm{induced}_g(\tau)$ to the variables from blocks in J belongs to $\mathcal{H}$ and $\mathrm{Cl}(L(\Psi_p)) \subseteq J$.

Proof. Since ρ semantically implies Ψ_p, it suffices to exhibit a solution of the linear system ρ with the stated properties.

Note that $|I \cup \mathrm{Cl}(L(\Psi_p))| \leq |I| + |p| \leq w$. Therefore, the partial assignment $\sigma \in \mathcal{H}$ can be extended to an assignment $\sigma' \in \mathcal{H}$ that is defined on the unlifted variables from the blocks in $I \cup \mathrm{Cl}(L(\Psi_p))$. Let π be a block-respecting partial assignment defined on the lifted variables of the blocks in $\mathrm{Cl}(L(\Psi_p)) \setminus I$ such that $\mathrm{induced}_g(\pi)$ agrees with σ' on the unlifted variables from these blocks. Such an assignment π exists because g is 1-stifling.

By Claim 5.2, the linear system $\rho \wedge \pi$ is satisfiable. Any solution τ of this system satisfies Ψ_p and has the required property for the set $J := I \cup \mathrm{Cl}(L(\Psi_p))$. ◁

In the next subsection, we establish Condition (3) in Theorem 5.1.

5.2 Estimating the Probability of Semantic Implication

In this subsection, we prove the following lemma.

▶ **Lemma 5.6.** *For any linear system Φ over variables X, it holds that*

$$\Pr[\rho \models \Phi] \leq \left(1 - 2^{-(\ell+1)}\right)^{\dim(L(\Phi))/\ell},$$

where ρ is the final value of ρ and the probability is taken over random bits used in the algorithm generating the distribution D.

Since by the invariants, $\rho \models \Psi_p$ during the execution of the algorithm generating D, this is also true for their final values. Hence, Lemma 5.6 implies that $\Pr_{p\sim D}[\Psi_p \models \Phi] \leq \Pr[\rho \models \Phi] \leq (1 - 2^{-(\ell+1)})^{\dim(L(\Phi))/\ell} \leq \exp\left(-\frac{1}{2^{\ell+1}\ell} \cdot \dim(L(\Phi)\right)$, thus, establishing Condition (3) in Theorem 5.1.

To prove Lemma 5.6, we carefully analyze what happens during the generation algorithm. Let us denote by V_ρ the linear space generated by linear forms $L(\rho)$. Denote by U the linear space generated by linear forms $L(\Phi)$. If $\rho \models \Phi$, then any equation of Φ is a linear combination of several equations of ρ, hence $U \subseteq V_\rho$. We examine how $V_\rho \cap U$ is changing during the generation algorithm. Initially ρ is an empty system, thus $V_\rho \cap U = \varnothing$.

Notice that ρ updates in the manner such that the updated ρ semantically implies the previous values. Thus, once we receive ρ that is not consistent with Φ, it will be inconsistent with Φ up to the end and $\rho \not\models \Phi$.

There are two different ways in which ρ can be modified. In Branch A, we substitute constants to all variables from the ith block, so $L(\rho)$ is increased by $\{x_{i,k} \mid k \in [\ell]\}$. In Branch B, we substitute constants to all variables from the ith block but $x_{i,j}$ and $L(\rho)$ is increased by $\{x_{i,k} \mid k \in [\ell] \setminus \{j\}\} \cup \{f\}$.

▶ **Lemma 5.7.** *Consider Step (2) of the algorithm (i.e., we have not yet tossed coin c). Assume that either $\dim(\langle V_\rho \cup \{x_{i,k} \mid k \in [\ell]\}\rangle \cap U)$ or $\dim(\langle V_\rho \cup \{x_{i,k} \mid k \in [\ell] \setminus \{j\}\} \cup \{f\}\rangle \cap U)$ is greater than $\dim(V_\rho \cap U)$. Then, with probability at least $2^{-(\ell+1)}$ after execution of the current iteration of the cycle ρ will be inconsistent with Φ.*

Proof.

Case 1. Adding linear forms $x_{i,k}$ for $k \in [\ell]$ to V_ρ increases its intersection with U:

Since the intersection $\langle V_\rho \cup \{x_{i,k} \mid k \in [\ell]\}\rangle \cap U$ is strictly greater than $V_\rho \cap U$, there is a subset $S \subseteq [\ell]$ and $v \in V$, such that $v + x_{i,S} \in U \setminus V$, where $x_{i,S}$ denotes $\sum_{j\in S} x_{i,j}$. Since $v + x_{i,S} \in U$, Φ semantically implies some fixed value $\gamma \in \{0, 1\}$ for linear form $v + x_{i,S} \in U$. Since $v \in V$, ρ implies some fixed value $\delta \in \{0, 1\}$ for linear form v.

With probability $\frac{1}{2}$, we toss $c = 1$, and the algorithm will go to Branch A. And to the lifted variables from the block i we substitute the random value from $g^{-1}(\alpha)$. Since g is 1-stifling, by Lemma 2.5, with probability at least $2^{-\ell}$ the value of the linear form $x_{i,S}$ will be different from $\gamma + \delta$. Hence, after the Branch A, ρ and Φ will be inconsistent since they imply different values to the linear form $v + x_{i,S}$. And this happens with probability at least $\frac{1}{2} \cdot 2^{-\ell} = 2^{-(\ell+1)}$.

Case 2. Assume the condition of case 1 does not hold but adding linear forms $\{x_{i,k} \mid k \in [\ell] \setminus \{j\}\} \cup \{f\}$ to V_ρ increases its intersection with U.

Then there is a subset $S \subseteq [\ell] \setminus \{j\}$ and $v \in V$, such that $v + f + x_{i,S} \in U \setminus V$. Since $v + f + x_{i,S} \in U$, Φ semantically implies some fixed value $\gamma \in \{0, 1\}$ for linear form $v + f + x_{i,S} \in U$. Since $v \in V$, ρ implies some fixed value $\delta \in \{0, 1\}$ for linear form v.

With probability $\frac{1}{2}$, we toss $c = 0$, and the algorithm will go to Branch B. Then ρ will fix $x_{i,S}$ by some way and fix f uniformly at random. Hence, with the probability $\frac{1}{2}$, ρ and Φ will imply different values of the linear form $v + f + x_{i,S}$. This series of outcomes happens with probability $\frac{1}{2} \cdot \frac{1}{2} \geq 2^{-(l+1)}$, and ρ and Φ are inconsistent in it. ◀

Now we are ready to prove Lemma 5.6.

Proof of Lemma 5.6. Observe that after each iteration of the main loop (that is, after executing either Branch A or Branch B), the quantity $\dim(V_\rho \cap U)$ increases by at most ℓ. If the final value of Ψ_p semantically implies Φ, then so does the final value of ρ. In particular, this means that $\dim(V_\rho \cap U) = \dim(L(\Phi))$.

Consequently, there must be at least $\dim(L(\Phi))/\ell$ iterations of the main loop during which the intersection $V_\rho \cap U$ increases.

By Lemma 5.7, in each such iteration, with probability at least $2^{-(\ell+1)}$, the restriction ρ becomes inconsistent with Φ. Therefore, the probability that the final restriction ρ semantically implies Φ is at most $\left(1 - 2^{-(\ell+1)}\right)^{\dim(L(\Phi))/\ell}$. ◀

6 The Main Lifting Theorem

In this section, we prove our main lifting theorem and its corollaries.

▶ **Theorem 6.1.** *Assume that every resolution refutation of an unsatisfiable CNF formula ϕ has either width at least w, or depth at least h. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget, where ℓ is a constant. Then any* $\mathrm{Res}(\oplus)$ *refutation of $\phi \circ g$ of size S has depth at least $\Omega\left(\min\{h, w^2/\log S\}\right)$.*

As outlined in the introduction, Theorem 6.1 is proved by induction on the width w. We begin with the lemma that forms the core of the induction step.

Let $\mathcal{H}$ be a set of w-good assignments for φ. We say that $\mathcal{H}$ has one-time rollback property if for every partial assignment τ that is subassignment of some assignment from $\mathcal{H}$, there is a set of w-good assignments $\mathcal{H}_\tau$ for φ such that $\tau \in \mathcal{H}_\tau$.

For example, if $\mathcal{H}$ is closed under taking subassignments, then $\mathcal{H}$ has one-time rollback property and $\mathcal{H}_\tau = \mathcal{H}$.

▶ **Lemma 6.2.** *Let φ be an unsatisfiable CNF formula, and let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget, where ℓ is a constant. Let $\mathcal{H}$ be a set of w-good partial assignments for φ. Suppose that G is a parity decision DAG refuting $\varphi \circ g$ of size S. Then there exists an absolute constant $c > 0$ and a node v of G such that:*

- *there is a path from the source of G to v of length at least $(w-1)/6$;*
- *if $\mathcal{H}$ has the one-time rollback property and $w > c \log S$, then there exists an affine restriction ρ such that $G_v|_\rho$ is a parity decision DAG refuting $\varphi|_\sigma \circ g$, where:*
  - *G_v denotes the subgraph of G consisting of all nodes reachable from v;*
  - *σ is a partial assignment to the unlifted variables of size at most $c \log S$, and $\sigma \in \mathcal{H}_\tau$ for some subassignment τ of an assignment from $\mathcal{H}$.*

Proof. By Theorem 5.1, there exists a distribution D supported on paths from the source of G such that:

1. With probability at least $1/4$, a path $p \sim D$ has length at least $(w-1)/6$.
2. For every p from the support of $\mathcal{D}$, a linear system Ψ_p has a solution π and a set $J \subseteq [m]$ such that the restriction of $\mathrm{induced}_g(\pi)$ to variables from blocks J belongs to $\mathcal{H}$ and $\mathrm{Cl}(L(\Psi_p)) \subseteq J$.
3. For every linear system Φ over variables X, $\Pr_{p \sim D}[\Psi_p \models \Phi] \leq 2^{-\Omega(\dim(L(\Phi)))}$.

For a path p starting at the source of G, let v_p denote its endpoint and let Φ_{v_p} be the linear system labeling v_p. Let E be the set of all systems Φ_{v_p} corresponding to paths p in the support of D whose length is at least $(w-1)/6$. By construction, $\Pr_{p\sim D}[\Phi_{v_p} \in E] \geq \frac{1}{4}$.

On the other hand, for every $\Phi \in E$, $\Pr_{p\sim D}[\Phi_{v_p} = \Phi] \leq \Pr_{p\sim D}[\Psi_p \models \Phi] \leq 2^{-\Omega(\dim(L(\Phi)))}$, where the first inequality follows from Lemma 2.2. Since $|E| \leq S$, there exists a path p_0 of length at least $(w-1)/6$ such that $\dim(\Phi_{v_{p_0}}) \leq O(\log(4S)) = O(\log S)$. Let $c > 0$ be an absolute constant such that $\dim(\Phi_{v_{p_0}}) \leq c \log S$.

By Lemma 2.2, we have $\Psi_{p_0} \models \Phi_{v_{p_0}}$. Since Ψ_{p_0} is satisfiable, it follows that $L(\Phi_{v_{p_0}}) \subseteq \langle L(\Psi_{p_0})\rangle$, and by the properties of the closure (Lemma 2.9), $\mathrm{Cl}(L(\Phi_{v_{p_0}})) \subseteq \mathrm{Cl}(L(\Psi_{p_0}))$.

Let π be a solution to Ψ_{p_0} witnessing the second statement of Theorem 5.1. Then π also satisfies $\Phi_{v_{p_0}}$. Let τ denote the restriction of $\mathrm{induced}_g(\pi)$ to the variables from blocks in $\mathrm{Cl}(L(\Phi_{v_{p_0}}))$. Then τ is a subassignment of some assignment in $\mathcal{H}$. Since $\mathcal{H}$ has the one-time rollback property, there exists a family $\mathcal{H}_\tau$ of w-good partial assignments for φ that contains τ.

By Lemma 4.5, the system $\Phi_{v_{p_0}}$ can be satisfied by an $\mathcal{H}_\tau$-good affine restriction ρ of block-size at most $O(\log(S))$. Let $\sigma \in \mathcal{H}_\tau$ be an assignment induced by ρ on the unlifted variables; the size of σ is at most $O(\log S)$. Consider the subgraph $G_{v_{p_0}}$ of G consisting of all nodes reachable from v_{p_0}. By Lemma 4.4, applying ρ to $G_{v_{p_0}}$ yields a parity decision DAG refuting $\varphi|_\sigma \circ g$. ◀

Proof of Theorem 6.1. By Lemma 3.2, there exists a (w,h)-winning strategy for ϕ.

Define $d(w,h,S)$ as the minimum depth of a parity decision DAG of size at most S refuting $\varphi \circ g$, where the minimum ranges over all CNF formulas φ possessing a (w,h)-winning strategy. Our goal is to show that $d(w,h,S) \geq \Omega\left(\min\{h, w^2/\log S\}\right)$. Assume that this minimum is attained by a formula φ, and fix a (w,h)-winning strategy $\mathcal{B}$ for φ.

Consider $\mathcal{H} := \{\tau \mid (\tau, |\tau|) \in \mathcal{B}\}$. By Lemma 3.3, $\mathcal{H}$ is a $\min\{w,h\}$-good set of assignments for φ.

If $h < 2w$, then the theorem follows from Lemma 6.2. So we assume that $h \geq 2w$. In this case, $\mathcal{H}$ has the one-time rollback property. Indeed, consider some $\tau \in \mathcal{H}$, we know that $(\tau, |\tau|) \in \mathcal{B}$. If $\sigma \subseteq \tau$, then $(\sigma, |\tau|) \in \mathcal{B}$. Let us define $\mathcal{H}_\sigma := \{\pi \mid \exists i \leq |\tau| + |\pi| : (\pi, i) \in \mathcal{B}\}$. It is easy to see that $\mathcal{H}_\sigma$ is a set of w-good assignments for φ and that $\sigma \in \mathcal{H}_\sigma$.

Consider a parity decision DAG G refuting $\varphi \circ g$, let S be an upper bound on the size of G. By Lemma 6.2 applied to $\mathcal{H}$ there is a node v of G such that there is an absolute constant $c > 0$ and a path from the source to v of length at least $(w-1)/6$ (we denote by G_v a subgraph of G that contains all nodes that are reachable from v) and there is an affine restriction ρ of block-size at most $c \log S$ and a partial assignment σ from $\mathcal{H}_\tau$ for some $\tau \in \mathcal{H}$ of size at most $c \log S$ such that $G_v|_\rho$ is a parity decision DAG refuting $(\varphi|_\sigma) \circ g$.

Since the block-size of σ is at most $c \log S$, there exists a $(h - w - \lceil c \log S \rceil, w - \lceil c \log S \rceil)$-winning strategy $\mathcal{B}'$ for $\varphi|_\sigma$; for example if $(\sigma, k) \in \mathcal{B}$, we can choose $\mathcal{B}' = \{(\pi \setminus \sigma, j) \mid (\pi, j+k) \in \mathcal{B}, \sigma \subseteq \pi\}$.

Therefore, $d(w,h,S) \geq \frac{w-1}{6} + d(w - \lceil c \log S \rceil, h - w - \lceil c \log S \rceil, S)$. Unrolling the recurrence yields $d(h,w,S) \geq \Omega(\min\{h, w^2/\log S\})$. ◀

We apply this theorem to the supercritical size-depth tradeoff for resolution by Buss and Thapen [14]. It is convenient to use the corollary of this tradeoff presented in [29].

▶ **Theorem 6.3** ([14, 29]). *There exists a family of unsatisfiable CNF formulas $\{\Psi_n\}_{n=1}^{\infty}$ such that*

- *Ψ_n contains n variables;*

- *the width of* Ψ_n *is* $O(\log n)$ *and, moreover,* Ψ_n *has a resolution refutation of size* $\mathrm{poly}(n)$ *and of width* $O(\log n)$*;*
- *any resolution refutation of* Ψ_n *of width at most* $n/40 \log n$ *has depth greater than* $n^2/400 \log^2 n$*.*

▶ **Theorem 6.4.** *There is a family of formulas* ψ_n *from* n *variables of size* $\mathrm{poly}(n)$ *such that* ψ_n *has polynomial size resolution refutation, and for any* $S > 0$*, any* $\mathrm{Res}(\oplus)$ *refutation of size* S *has depth at least* $\Omega(n^2/(\log^2 n \log S))$*.*

Proof. Let Ψ_n be a formula from Theorem 6.3. Let $g = \mathrm{MAJ}_3$; it is a 1-stifling gadget. Then $\Psi_n \circ g$ is a formula in $O(\log n)$-CNF of size $\mathrm{poly}(n)$ and it has a resolution refutation of size $\mathrm{poly}(n)$. By Theorem 6.1, any $\mathrm{Res}(\oplus)$ refutation of $\Psi_n \circ g$ of size at most S has depth at least $\Omega(n^2/(\log^2 n \log S))$. ◀

▶ **Corollary 6.5.** *If* $d(n) = o(n^2/\log^4 n)$*, then Depth-*$d(n)$ $\mathrm{Res}(\oplus)$ *does not polynomially simulates resolution.*

Proof. Consider the formula ψ_n from Theorem 6.4. Applying Theorem 6.4 with $S = 2^{\log^2 n}$, we obtain that any $\mathrm{Res}(\oplus)$ refutation of ψ_n of depth $d(n)$ must have size at least $2^{\log^2 n}$, which is superpolynomial in n. ◀

7 A Quadratic Size Lower Bound for Unrestricted $\mathrm{Res}(\oplus)$

In this section, we present a quadratic size lower bound for $\mathrm{Res}(\oplus)$. We begin with a linear-algebraic lemma, next recall the necessary previous results, and then conclude with the proof of the main theorem.

▶ **Lemma 7.1.** *Let* $V_1, V_2, \ldots, V_s$ *be linear subspaces of* $\mathbb{F}^n$*, where* $\mathbb{F}$ *is a field. Assume that* $V_1 = \{0\}$ *and that for every* $i \in [s-1]$ *there exists a vector* $x_i \in \mathbb{F}^n$ *such that* $V_{i+1} \subseteq \langle V_i \cup \{x_i\}\rangle$*. Then there exists a nonzero vector* x *such that* $\big|\{i \in [s] \mid x \in V_i\}\big| \geq \dim V_s$*, that is,* x *belongs to at least* $\dim V_s$ *of the subspaces* V_i*.*

Proof. We prove the lemma by induction on $k := \dim V_s$. The base cases $k = 0$ and $k = 1$ are immediate. Assume that $k > 1$.

For each $i \in [s]$, define $V'_i := V_i \cap V_s$. Note that $V'_{i+1} = V_{i+1} \cap V_s \subseteq \langle V_i \cup \{x_i\}\rangle \cap V_s$. Notice that, $\dim\big(\langle V_i \cup \{x_i\}\rangle \cap V_s\big) \leq \dim(V_i \cap V_s) + 1 = \dim(V'_i) + 1$. Therefore, there exists a vector $x'_i \in \mathbb{F}^n$ such that $\langle V_i \cup \{x_i\}\rangle \cap V_s \subseteq \langle (V_i \cap V_s) \cup \{x'_i\}\rangle = \langle V'_i \cup \{x'_i\}\rangle$, and hence $V'_{i+1} \subseteq \langle V'_i \cup \{x'_i\}\rangle$ for every $i \in [s-1]$.

Since $V'_1 = \{0\}$, $\dim(V'_s) = k$, and for every $i \in [s-1]$ we have $\dim(V'_{i+1}) \leq \dim(V'_i) + 1$, there exists $j \in [s-1]$ such that $\dim(V'_j) = k - 1$.

Applying the induction hypothesis to the sequence $V'_1, V'_2, \ldots, V'_j$, we obtain a nonzero vector x that belongs to at least $k-1$ of these subspaces. Since V_s contains every V'_i, the vector x also belongs to V_s. Hence x belongs to at least k of the original subspaces $V_1, \ldots, V_s$ (namely, to at least $k-1$ among $V_1, \ldots, V_j$, and also to V_s), completing the induction step. ◀

We will use the following result of Alekseev and Itsykson [4], which shows that resolution width lifts to width in $\mathrm{Res}(\oplus)$.

▶ **Theorem 7.2** ([4])**.** *Let* φ *be an unsatisfiable CNF formula such that every resolution refutation of* φ *has width at least* w*. Let* g *be a* 1*-stifling gadget. Then every* $\mathrm{Res}(\oplus)$ *refutation of* $\varphi \circ g$ *has width at least* w*.*

We will also need the following basic fact about resolution width.

▶ **Lemma 7.3** ([10]). *Let φ be an unsatisfiable CNF formula of resolution width w, and let x be a variable of φ. Then there exists a value $a \in \{0,1\}$ such that the restricted formula $\varphi|_{x:=a}$ has resolution width at least $w-1$.*

▶ **Theorem 7.4.** *Let $\varphi(y_1, y_2, \ldots, y_m)$ be an unsatisfiable CNF formula such that every resolution refutation of φ has width at least w. Let $g : \{0,1\}^\ell \to \{0,1\}$ be a 1-stifling gadget. Then any* Res(⊕) *refutation of $\varphi \circ g$ has size at least $\frac{w(w+1)}{2}$.*

Proof. We prove the theorem by induction on w. The base case $w = 1$ is trivial.

Let G be a parity decision DAG refuting $\varphi \circ g$ of minimum size. By Theorem 7.2, G contains a node v such that the linear system Φ_v labeling this node has rank at least w. Fix a path $v_1, \ldots, v_s = v$ from the source v_1 to v. Since G has minimum size, we may assume that for all $i \in [s]$ the system Φ_{v_i} is satisfiable, and that for all $i \in [s-1]$ the equation labeling the edge (v_i, v_{i+1}) is consistent with Φ_{v_i}.

For each $i \in [s]$, let V_i denote the vector space of linear equations spanned by the equations of Φ_{v_i}. We have $V_1 = \{0 = 0\}$ and $\dim(V_s) = \mathrm{rk}(\Phi_v) \geq w$. Moreover, for every $i \in [s-1]$, the space V_{i+1} is contained in the span of V_i and the equation labeling the edge (v_i, v_{i+1}). By Lemma 7.1, there exists a nontrivial equation $f = a$ that belongs to at least w of the spaces V_i, for $i \in [s]$.

Since all systems Φ_{v_i} are satisfiable, $f = a$ is different from $0 = 1$. In particular, the linear form f contains at least one variable x_{i_0,j_0}.

By Lemma 7.3, there exists $a \in \{0,1\}$ such that the restricted formula $\varphi|_{y_{i_0}:=a}$ has resolution width at least $w-1$. Let ρ_1 be a partial assignment that assigns Boolean values to all lifted variables from the i_0th block except x_{i_0,j_0}, in such a way that these assignments force $y_{i_0} = a$ independently of the value of x_{i_0,j_0}. Such an assignment ρ_1 exists since g is 1-stifling. By Proposition 4.2, the restricted formula $(\varphi \circ g)|_{\rho_1}$ is semantically equivalent to $\varphi|_{y_{i_0}:=a} \circ g$, and every clause of the former contains a clause of the latter.

Next, define an affine restriction ρ_2 supported on the one variable $x_{i_0,j_0} := f|_{\rho_1} + x_{i_0,j_0} + a + 1$, and let $\rho := \rho_1 \cup \rho_2$. By Proposition 4.1, the restricted DAG $G|_\rho$ refutates $\varphi|_{y_{i_0}:=a} \circ g$.

Since the resolution width of $\varphi|_{y_{i_0}:=a}$ is at least $w-1$, the induction hypothesis implies that the size of $G|_\rho$ is at least $\frac{(w-1)w}{2}$. On the other hand, G contains at least w nodes whose linear systems are falsified by ρ, namely those whose associated spaces contain the equation $f = a$. Therefore, the size of $G|_\rho$ is smaller than the size of G by at least w nodes. It follows that the size of G is at least $\frac{w(w+1)}{2}$, completing the proof. ◀

References

1 M. Ajtai. $\sum^1_1$-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983. doi:10.1016/0168-0072(83)90038-6.

2 Miklós Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994. doi:10.1007/BF01302964.

3 Yaroslav Alekseev and Nikita Gaevoy. New polynomial-depth Res(+) lower bounds. *Electron. Colloquium Comput. Complex.*, TR26-007, 2026. URL: https://eccc.weizmann.ac.il/report/2026/007.

4 Yaroslav Alekseev and Dmitry Itsykson. Lifting to bounded-depth and regular resolutions over parities via games. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025*, pages 584–595. ACM, 2025. doi:10.1145/3717823.3718150.

5 Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008. doi:10.1016/J.JCSS.2007.06.025.

6 Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space trade-offs in resolution: Superpolynomial lower bounds for superlinear space. *SIAM J. Comput.*, 45(4):1612–1645, 2016. doi:10.1137/130914085.

7 Paul Beame and Sajin Koroth. On Disperser/Lifting Properties of the Index and Inner-Product Functions. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *LIPIcs*, pages 14:1–14:17, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2023.14.

8 Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus: extended abstract. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 813–822. ACM, 2013. doi:10.1145/2488608.2488711.

9 Eli Ben-Sasson. Size space tradeoffs for resolution. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing*, pages 457–464. ACM, 2002. doi:10.1145/509907.509975.

10 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001. doi:10.1145/375827.375835.

11 Christoph Berkholz. On the complexity of finding narrow proofs. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 351–360. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.48.

12 Christoph Berkholz and Jakob Nordström. Supercritical space-width trade-offs for resolution. *SIAM J. Comput.*, 49(1):98–118, 2020. doi:10.1137/16M1109072.

13 Sreejata Kishor Bhattacharya and Arkadev Chattopadhyay. Exponential lower bounds on the size of reslin proofs of nearly quadratic depth. *Electron. Colloquium Comput. Complex.*, TR25-106, 2025. URL: https://eccc.weizmann.ac.il/report/2025/106/.

14 Sam Buss and Neil Thapen. A simple supercritical tradeoff between size and height in resolution. *Inf. Process. Lett.*, 191:106589, 2026. doi:10.1016/J.IPL.2025.106589.

15 Farzan Byramji and Russell Impagliazzo. Lifting to randomized parity decision trees. In Alina Ene and Eshan Chattopadhyay, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2025, Berkeley, CA, USA, August 11-13, 2025*, volume 353 of *LIPIcs*, pages 55:1–55:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.APPROX/RANDOM.2025.55.

16 Farzan Byramji and Russell Impagliazzo. Lower bounds for bit pigeonhole principles in bounded-depth resolution over parities. *CoRR*, abs/2511.20023, 2025. doi:10.48550/arXiv.2511.20023.

17 Arkadev Chattopadhyay and Pavel Dvorák. Super-critical trade-offs in resolution over parities via lifting. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference, CCC 2025, August 5-8, 2025, Toronto, Canada*, volume 339 of *LIPIcs*, pages 24:1–24:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CCC.2025.24.

18 Arkadev Chattopadhyay, Nikhil S. Mande, Swagato Sanyal, and Suhail Sherif. Lifting to parity decision trees via stifling. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023*, volume 251 of *LIPIcs*, pages 33:1–33:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.33.

19 Susanna F. de Rezende, Noah Fleming, Duri Andrea Janett, Jakob Nordström, and Shuo Pang. Truly supercritical trade-offs for resolution, cutting planes, monotone circuits, and weisfeiler–leman. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 1371–1382, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3717823.3718271.

20 Klim Efremenko, Michal Garlík, and Dmitry Itsykson. Lower bounds for regular resolution over parities. *SIAM J. Comput.*, 54(4):887–915, 2025. doi:10.1137/24M1696640.

21 Klim Efremenko and Dmitry Itsykson. Amortized Closure and Its Applications in Lifting for Resolution over Parities. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference (CCC 2025)*, volume 339 of *LIPIcs*, pages 8:1–8:24, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2025.8.

22 Klim Efremenko and Dmitry Itsykson. Strong ETH holds for bounded-depth resolution over parities. *Electron. Colloquium Comput. Complex.*, TR25-188, 2025. To appear in Proceedings of STOC 2026. URL: https://eccc.weizmann.ac.il/report/2025/188.

23 Noah Fleming, Toniann Pitassi, and Robert Robere. Extremely Deep Proofs. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *LIPIcs*, pages 70:1–70:23, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2022.70.

24 Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*, pages 260–270, 1981. doi:10.1109/SFCS.1981.35.

25 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory Comput.*, 16:1–30, 2020. doi:10.4086/TOC.2020.V016A013.

26 Michal Garlík and Leszek Aleksander Kolodziejczyk. Some subsystems of constant-depth Frege with parity. *ACM Trans. Comput. Log.*, 19(4):29:1–29:34, 2018. doi:10.1145/3243126.

27 Mika Göös, Gilbert Maystre, Kilian Risse, and Dmitry Sokolov. Supercritical tradeoffs for monotone circuits. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025*, pages 1359–1370. ACM, 2025. doi:10.1145/3717823.3718229.

28 Svyatoslav Gryaznov, Sergei Ovcharov, and Artur Riazanov. Resolution over linear equations: Combinatorial games for tree-like size and space. *ACM Trans. Comput. Theory*, 16(3):15:1–15:15, 2024. doi:10.1145/3675415.

29 Dmitry Itsykson and Alexander Knop. Supercritical tradeoff between size and depth for resolution over parities. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference, ITCS 2026*, volume 362 of *LIPIcs*, pages 81:1–81:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi:10.4230/LIPIcs.ITCS.2026.81.

30 Dmitry Itsykson and Artur Riazanov. Proof complexity of natural formulas via communication arguments. In Valentine Kabanets, editor, *36th Computational Complexity Conference*, volume 200 of *LIPIcs*, pages 3:1–3:34. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.3.

31 Dmitry Itsykson and Dmitry Sokolov. Lower bounds for splittings by linear combinations. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014*, pages 372–383, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg. doi:10.1007/978-3-662-44465-8_32.

32 Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Ann. Pure Appl. Log.*, 171(1), 2020. doi:10.1016/J.APAL.2019.102722.

33 Erfan Khaniki. On proof complexity of resolution over polynomial calculus. *ACM Trans. Comput. Log.*, 23(3):16:1–16:24, 2022. doi:10.1145/3506702.

34 Jan Krajíček. Randomized feasible interpolation and monotone circuits with a local oracle. *J. Math. Log.*, 18(2):1850012:1–1850012:27, 2018. doi:10.1142/S0219061318500125.

35 Vladimir Podolskii and Alexander Shekhovtsov. Randomized lifting to semi-structured communication complexity via linear diversity. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, volume 325 of *LIPIcs*, pages 78:1–78:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ITCS.2025.78.

36 A. A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mat. Zametki*, 41:598–607, 1987.

37 Alexander A. Razborov. A new kind of tradeoffs in propositional proof complexity. *J. ACM*, 63(2):16:1–16:14, 2016. doi:10.1145/2858790.

38 Alexander A. Razborov. On the width of semialgebraic proofs and algorithms. *Math. Oper. Res.*, 42(4):1106–1134, 2017. doi:10.1287/MOOR.2016.0840.

39 Alexander A. Razborov. On space and depth in resolution. *Comput. Complex.*, 27(3):511–559, 2018. doi:10.1007/S00037-017-0163-1.

40 Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82, 1987. doi:10.1145/28395.28404.

A Weak Regularity Lemma for Polynomials

Guy Moshkovitz
City University of New York, NY, USA

Dora Woodruff
Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

A regularity lemma for polynomials provides a decomposition in terms of a bounded number of approximately independent polynomials. Such regularity lemmas play an important role in numerous results, yet suffer from the familiar shortcoming of having tower-type bounds or worse. In this paper we design a new, weaker regularity lemma with strong bounds. The new regularity lemma in particular provides means for quantitatively studying the curves contained in the image of a polynomial map, which is beyond the reach of standard methods.

The weak regularity lemma turns out to be powerful enough to yield results on arithmetic circuits and polynomial ranks that may be of independent interest:

- A general upper bound on the arithmetic circuit size of low-degree polynomials based solely on their image.
- An upper bound on the top fan-in of depth-4 arithmetic formulas under similar conditions.
- A quantitative bound for the Green-Tao notion of rank for polynomials, significantly improving on a result of Karam.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases weak regularity lemma, finite-field polynomials, polynomial maps, structure-versus-randomness, arithmetic circuits

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.14

Related Version *Full Version*: https://arxiv.org/abs/2509.21536

Funding *Guy Moshkovitz*: NSF Award DMS-2302988 and a PSC-CUNY award. The project began while D.W. was a summer graduate research assistant supported by this NSF award.

Acknowledgements We are grateful to the anonymous referees for multiple useful comments.

1 Introduction

Regularity lemmas for polynomials have served as a key component in a variety of important results, as in the proof of the finite-field Gowers inverse conjecture in additive combinatorics and higher-order Fourier analysis (e.g. [15, 38]), Stillman's conjecture in commutative algebra [1], and the Reed-Muller weight distribution problem in coding theory [3, 23], to name but a few. They serve as a key component in higher-order arithmetic regularity lemmas (e.g., [13, 14, 15, 17, 22]).[1] As is usually the case with regularity lemmas, they unfortunately only have weak bounds, which are typically inherited by applications. In this paper we introduce a weaker regularity lemma for polynomials, with far better bounds, that is nevertheless powerful enough for some applications. For example, we obtain strong bounds for a result of Karam [21] on the relation between the generalized rank of a polynomial and its range. Our results also yield a method to prove *upper* bounds for arithmetic circuits.

[1] Roughly speaking, arithmetic regularity lemmas approximate general functions in terms of low-degree polynomials that are approximately independent. (This approximate independence is indispensable; to quote Green [14]: second-order arithmetic regularity lemmas "reduce the study of general functions... to the study of projections... onto quadratic factors. This, however, is of little consequence unless we can study those supposedly simple objects.")

© Guy Moshkovitz and Dora Woodruff;
licensed under Creative Commons License CC-BY 4.0

Roughly speaking, the usual regularity lemmas for polynomials (see, e.g., Proposition 8.1 in [9], Lemma 2.4 in [15], or Lemma 3.11 in [14]) can be described as results about polynomial decompositions. Namely, they express any n-variate polynomial $P\colon \mathbb{F}^n \to \mathbb{F}$ – and more generally, any polynomial map $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$ – in terms of a bounded number (independent of n) of polynomials $\mathbf{X} = (X_1, \ldots, X_k)$, that is, $P = F(\mathbf{X})$, with the property that the polynomials in $\mathbf{X}$ "behave" approximately like k (jointly independent) variables. Although the precise definition of "behave" can differ depending on the context, the intuition is as follows: for many purposes, any polynomial can be analyzed as though it had only k variables. The number k of *parts* $X_1, \ldots, X_k$ in the decomposition depends only on the degree d and the number m of the given polynomials $\mathbf{P} = (P_1, \ldots, P_m)$, and as mentioned above, this dependency – befitting a regularity lemma – can be quite astronomical, often having worse than a tower-type growth (see, e.g., the Remark following Lemma 2.4 in [15]).[2] A main reason for these weak bounds is that in order to guarantee approximate independence, the polynomial parts $X_1, \ldots, X_k$, as well as all their linear combinations, are required to have *rank* (see Section 2) that is large as a function of k, their number. Thus, the larger the number of parts, the more difficult it is to satisfy the rank requirement, which translates to an extremely large number of steps in the process to construct them.[3]

In the setting of graphs, Frieze and Kannan [11] famously obtained a weak regularity lemma that has both strong bounds and multiple interesting applications. For comparison, the usual graph regularity lemma of Szemerédi decomposes any graph – by partitioning its vertices – into a bounded number of parts (independent of the number of vertices) with the guarantee that almost *all* graph parts are "approximately" random. The weak graph regularity lemma decomposes any graph in the same sense, but comes with only one, global guarantee. Our weak regularity lemma for polynomials is partly inspired by this approach.

1.1 Weak regularity for polynomials

The basic idea of our weak regularity lemma is to relax the requirement on the k polynomials $X_1, \ldots, X_k$, and only require that at least one of them, of maximal degree, behaves like a free variable with respect to all the others. That is, conditioned on $X_2, \ldots, X_k$ taking *any* given value $y \in \mathbb{F}^{k-1}$, we require that X_1 takes any value in $\mathbb{F}$ with approximately the same frequency. This imitates the case of a genuinely free variable; or more concretely, the case where $X_1 = x_1$, and $X_i = X_i(x_2, \ldots, x_n)$ for $i > 1$, where $x_1, \ldots, x_n$ are variables.

Intuitively, the benefit of having a weak regular decomposition for a polynomial map $\mathbf{P}$ is that it reveals some information about the univariate polynomials "induced" by $\mathbf{P}$. In particular, it enables us to argue about the image of $\mathbf{P}$ (for which standard rank-based tools are unsuitable, as they typically give information only about equidistribution); indeed, a corollary of the weak regularity lemma will be that lines that $\mathbf{X}$ intersects somewhat often must in fact be fully contained in the image of $\mathbf{X}$. This translates to a polynomial curve (univariate polynomial map) contained in the image of $\mathbf{P}$. See below for more about applications.

---

[2] Interestingly, recent regularization results of Lampert and Ziegler [29, 27] give strong bounds; however, unlike regularity lemmas, these do not provide a decomposition of $\mathbf{P}$ as a polynomial in $X_1, \ldots, X_k$, but rather place $\mathbf{P}$ in the ideal generated by them.

[3] See, e.g., Proposition 8.1 in [9] for a clean example of this phenomenon: the reason that the process terminates at all is a decrease in the lexicographic order of the tuple of degrees of $\mathbf{X}$ in each step.

We note that throughout the paper we work with finite fields $\mathbb{F}$, for which the notion of weak regularity can be most elementarily defined. Moreover, rank-based methods, which we rely on extensively in the proof (as do proofs of other regularity lemmas for polynomials), tend to be most challenging in the finite-field setting.[4]

Formally, we define weak regularity as follows. We refer the reader to Section 2 for a more formal discussion, and the proofs. For tuples of polynomials $\mathbf{P} = (P_1, \ldots, P_m)$ and $\mathbf{X} = (X_1, \ldots, X_k)$ over $\mathbb{F}$, we write $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$ if every component P_i of $\mathbf{P}$ can be written as $F_i(X_1, \ldots, X_k)$ for some polynomial F_i. Henceforth, for any k-tuple $v = (v_1, \ldots, v_k)$, we use the notation v' for the $(k-1)$-tuple $v' = (v_2, \ldots, v_k)$.

▶ **Definition** (weak regularity)**.** A tuple of polynomials $\mathbf{P}$ over $\mathbb{F} = \mathbb{F}_q$ has a *weak ϵ-regular decomposition* of size k if $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$, for some k homogeneous polynomials $\mathbf{X} = (X_1, \ldots, X_k)$, such that:

- for every $y \in \mathbb{F}^k$, $|\Pr[\mathbf{X} = y] - q^{-1}\Pr[\mathbf{X}' = y']| \leq \epsilon q^{-k}$,
- $\mathbf{P} \not\subseteq \mathbb{F}[\mathbf{X}']$, and
- $\deg(X_1) = \max_i \deg(X_i)$.

The probabilities here are with respect to a uniformly random input; that is, $\Pr[\mathbf{X} = y]$ is a shorthand for $\Pr_{x \in \mathbb{F}^n}[\mathbf{X}(x) = y]$, with x chosen uniformly at random from $\mathbb{F}^n$. Intuitively, the probabilistic condition can be motivated as follows. If X_1 were truly an independent variable x_1, with $X_i = X_i(x_2, \ldots, x_n)$ for $i > 1$, we would certainly have both $\Pr[X_1 = y_1] = \frac{1}{q}$ and $\Pr[\mathbf{X} = y] = \Pr[X_1 = y_1]\Pr[\mathbf{X}' = y']$. Hence, the probabilistic condition mimics the behavior we would expect for an independent variable x_1.

Note that the two additional conditions at the end of the definition are necessary to avoid trivialities. This is clearly the case for the first, which guarantees that the decomposition of $\mathbf{P}$ depends on X_1. As for the other additional condition, that X_1 is of maximal degree – note that we can decompose any polynomial $P = P(x_1, \ldots, x_n)$, of degree d in x_1 say, as $\sum_{i=0}^{d} x_1^i C_i(x_2, \ldots, x_n)$; that is, we always have a small decomposition $P = F(x_1, C_0, \ldots, C_d)$ (where $F(z, y_0, y_1, \ldots, y_d) = \sum_{i=0}^{d} z^i y_i$). Although x_1 is genuinely independent of the other parts of the decomposition $C_0, \ldots, C_d$, its degree $\deg(x_1) = 1$ is generally far from the largest degree among them.

Let us also mention two extreme cases in the definition of weak regularity. First, if $\mathbf{P}$ is a tuple of forms that, viewed as a map $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$, is already equidistributed, that is, $|\Pr[\mathbf{P} = y] - q^{-m}| \leq \epsilon q^{-m}$ for every $y \in \mathbb{F}^m$, then $\mathbf{P}$ trivially has a small weak ϵ-regular decomposition. Second, $\mathbf{P}$ always has the trivial decomposition $\mathbf{P} = \mathbf{P}(\mathbf{X})$ with $\mathbf{X} = (x_1, \ldots, x_n)$ its variables, which is a weak 0-regular (i.e., a perfectly weak regular decomposition; assuming without loss of generality that $\mathbf{P}$ depends on x_1), though of a trivially large size n.

Our main result is that a rather small weak regular decomposition always exists (see Theorem 8).

▶ **Theorem 1** (weak regularity lemma)**.** *Let $\epsilon = q^{-c}$ with $c > 1$. Every polynomial m-tuple $\mathbf{P}$ over $\mathbb{F} = \mathbb{F}_q$, of degrees at most d with $d < \mathrm{char}(\mathbb{F})$, has a weak ϵ-regular decomposition of size at most $(mc)^{2^{d(1+o(1))}}$.*[5]

[4] To get an analogue of our weak regularity lemma for infinite fields would require replacing the probabilistic notion of approximate independence in the definition of weak regularity below, for example by a geometric or by a commutative algebra analogue (see, e.g., [28, 9]), and adapting the proof accordingly. Technically, this would require replacing the notion of bias of a polynomial (see Section 2.2) by standard analogues for infinite fields. For the structure-vs-randomness theorem we rely on (Theorem 17), a geometric analogue over infinite fields has long been known [25, 36] with, in fact, a slightly better, linear bound.

[5] Throughout, the $o(1)$ term in $d(1 + o(1))$ goes to 0 as d goes to infinity.

For comparison, the usual regularity lemmas for polynomials only give tower-type bounds or worse; namely, the bounds are at the very least a tower of exponents, of height depending on d, with m at the top. Our weak regularity lemma, in particular, is polynomial in m (which is especially relevant in the important case of low-degree polynomials $d = O(1)$).

The proof of our weak regularity lemma relies on several components, including a combinatorial regularization process over "punctured" spaces of polynomials, and a Fourier-analytic quasirandomness argument over such spaces. See Section 1.4 for more information.

1.2 Applications for arithmetic circuits

Using the weak regularity lemma, one can prove results that go beyond standard notions of structure for polynomials such as rank or partition rank. These standard notions only give "shallow" decompositions – their building blocks are generally just one degree lower – whereas we will be interested in decompositions that go "deeper". Concretely, in the next two sections we present applications that give novel structural results for polynomials based solely on their image.

Inspired by [21], we make the following definition for polynomial maps (that is, tuples of polynomials). Define the *univariate degree* of a polynomial map $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$ to be the smallest degree of a non-constant curve contained in its image:

$$\operatorname{udeg}(\mathbf{P}) = \min\{\deg(\mathbf{U}) \mid \mathbf{U}\colon \mathbb{F} \to \mathbb{F}^m \text{ non-constant}\colon \operatorname{Im}\mathbf{U} \subseteq \operatorname{Im}\mathbf{P}\}.$$

By convention, if $\mathbf{P}$ is constant then $\operatorname{udeg}(\mathbf{P}) = \infty$. Note that $1 \le \operatorname{udeg}(\mathbf{P}) \le \deg(\mathbf{P})$ $(:= \max_i \deg(P_i))$ for any non-constant $\mathbf{P}$;[6] see [21] and Section 3.1 for more about this.

▶ Remark 2. Univariate degree is related to the concepts of subspace-evasive sets and variety-evasive sets over finite fields, which are sets that have small intersections with, in particular, any line and any curve of low degree, respectively. (Explicit constructions were given by Dvir, Kollár, and Lovett [7, 6].) In particular, if $\operatorname{Im}(\mathbf{P})$ evades curves in the sense that $|\operatorname{Im}(\mathbf{P}) \cap C| < |\mathbb{F}|/u$ for every curve $C \subseteq \mathbb{F}^m$ of degree u, then $\operatorname{udeg}(\mathbf{P}) > u$. (Indeed, otherwise $C \subseteq \operatorname{Im}(\mathbf{P})$ with $\deg(C) = u$, and $|C| \ge |\mathbb{F}|/u$.) However, the converse does not hold – large $\operatorname{udeg}(\mathbf{P})$ is about $\operatorname{Im}(\mathbf{P})$ avoiding containment of entire curves, and so is a weaker property than $\operatorname{Im}(\mathbf{P})$ avoiding any significant intersections with curves. Thus, applications requiring large $\operatorname{udeg}(\mathbf{P})$, as we give in this paper, require an even weaker condition on $\operatorname{Im}(\mathbf{P})$ than being variety evasive.

Let $L(P)$ denote the arithmetic formula size of a polynomial P. It is known that for most degree-d polynomials P in n variables, over any field, $L(P) \ge \binom{n+d}{d}^{1-o(1)}$. Indeed, a nice argument of Hrubeš and Yehudayoff (Theorem 3.7 in [19]) shows that this holds even when restricted to polynomials with coefficients in $\{0, 1\}$. In fact, a similar conclusion holds more generally for arithmetic circuits.[7]

We use the weak regularity lemma to show that for low-degree polynomials, high univariate degree implies the existence of formulas that are considerably smaller than the typical $n^{d-o(1)}$, with a power-saving bound.

▶ **Theorem 3.** *Let P be an n-variate polynomial of degree at most d and univariate degree at least u, over any finite field $\mathbb{F}$ with $\operatorname{char}(\mathbb{F}) > d$. If $d \le \frac{1}{2}\log_2\log_2(n)$ then*

$$L(P) \le n^{\lfloor d/u \rfloor + o(1)}.$$

[6] Indeed, if $\mathbf{P}(u) \ne \mathbf{P}(v)$ then $\mathbf{U}(t) = \mathbf{P}(u + t(v-u))$ is non-constant and $\deg(\mathbf{U}) \le \deg(\mathbf{P})$.

[7] This traces back to Stoß [37], at least in the univariate case. See also Lovett [30].

We note that any improvement to the upper bound in the weak regularity lemma would translate to a less stringent condition on d in Theorem 3 (e.g., a polynomial upper bound d^C in Theorem 8 would translate to the condition $d \leq n^{o(1)}$).

Our machinery also has a direct implication for the study of depth-4 arithmetic formulas. A $\sum^{[r]}\prod^{[a]}\sum\prod^{[b]}$-formula (e.g., [24, 26]) for a polynomial P is a decomposition of the form $P = \sum_{i=1}^{r}\prod_{j=1}^{a_i} Q_{i,j}$ with $a_i \leq a$ and $\deg(Q_{i,j}) \leq b$. It is an important challenge to understand the interplay between any of the different "fan-in" parameters (r, a, b). Trivially, if the bottom fan-in b is allowed to be as large as $\deg(P)$, then P has a $\sum^{[r]}\prod^{[a]}\sum\prod^{[b]}$-formula with $r = a = 1$. The question becomes interesting when the bottom fan-in b is bounded away from $\deg(P)$. Using the weak regularity lemma, we deduce an *upper bound* on the top fan-in r from a lower bound on the univariate degree. (It is worth noting that the model of depth-4 homogeneous arithmetic formulas with top fan-in r was shown by Kumar and Saraf [26] to be already powerful for $r = \Theta(\log n)$.)

▶ **Theorem 4.** *Let $P\colon \mathbb{F}^n \to \mathbb{F}$ with $\deg(P) \leq d$, over any finite field $\mathbb{F}$ with $d < \mathrm{char}(\mathbb{F})$. Suppose $P \in \mathbb{F}[\mathbf{P}]$ with $\mathbf{P} = (P_1, \ldots, P_m)$, $\deg(\mathbf{P}) \leq d$, such that $\mathrm{udeg}(\mathbf{P}) = u$. Then there is a $\sum^{[r]}\prod^{[2u]}\sum\prod^{[d/u]}$-formula for P with top fan-in $r \leq (2m)^{2^{d(1+o(1))}}$. Moreover, the formula is homogeneous[8] if P is.*

Contrapositively, the above results can also be interpreted as implying a barrier for lower bound results: roughly, any strong enough lower bound method must somehow avoid applying to polynomial maps of high univariate degree. It therefore seems that the notion of univariate degree, and more generally a qualitative understanding of the curves contained in the image of a polynomial map, could be an important data point to consider when studying questions about arithmetic circuits.[9]

There are also connections with polynomial identity testing. For example, a line of work by Oliveira, Sengupta, and others (e.g., [34, 12]) uses a regularity lemma for polynomials to obtain efficient identity testing algorithms for depth-4 circuits with bounded top and bottom fan-ins. We leave it for future work to determine whether our weak regularity lemma could suffice here.

1.3 Applications for generalized ranks of polynomials

Is there an algebraic explanation for non-surjectivity of (multivariate) polynomials? An important class of non-surjective polynomials are those of the form $P = F(Q)$ with $1 < \deg(Q) < \deg(P)$ – sometimes called *uni-multivariate decomposable* [16] – for some non-surjective univariate F (for example, $P = (Q(x_1, \ldots, x_n))^t$ with $\gcd(t, |\mathbb{F}| - 1) > 1$). For such a polynomial of degree d, it must be that $\deg(Q) \leq d/2$ (as $\deg(F) \geq 2$). We show that this is always the case, in the sense that every non-surjective polynomial can be written in terms of a rather small number of polynomials all of degree at most $d/2$ (despite most polynomials not being uni-multivariate decomposable [39]). In fact, we also show a significant generalization of this to polynomial maps (note that the image of a polynomial map holds information about the dependencies between its various polynomials).

[8] That is, each $Q_{i,j}$ is homogeneous, and all products $\prod_j Q_{i,j}$ are of the same degree.

[9] Univariate degree should not be confused with the concept of an elusive function of Raz [35], which is a polynomial map $\mathbf{P}$ satisfying $\mathrm{Im}\,\mathbf{P} \not\subseteq \mathrm{Im}\,\mathbf{\Gamma}$, for every polynomial map $\mathbf{\Gamma}$ with given parameters. Indeed, univariate degree is about containment in the other direction: a lower bound on $\mathrm{udeg}(\mathbf{P})$ means that $\mathrm{Im}\,\mathbf{U} \not\subseteq \mathrm{Im}\,\mathbf{P}$, for every polynomial curve $\mathbf{U}$ of a given degree.

Following Green and Tao ([15], Definition 1.6), let $\mathrm{rank}_t(P)$, for a polynomial P and $t \geq 0$, denote the least number of polynomials of degree at most t such that P can be expressed as a function of only them. Note that this extends naturally to polynomial maps $\mathbf{P}$. One easy-to-state application of the weak regularity lemma is that any polynomial that is not surjective, and more generally any polynomial map $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$ whose image $\operatorname{Im}\mathbf{P}$ ($\subseteq \mathbb{F}^m$) does not contain a line, must have low $\mathrm{rank}_t(\mathbf{P})$, for some t bounded away from $\deg(\mathbf{P})$.

▶ **Theorem** (see Theorem 21). Let $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$ be a polynomial map with $\deg(\mathbf{P}) \leq d$, over any finite field $\mathbb{F}$ with $d < \mathrm{char}(\mathbb{F})$. If $\operatorname{Im}\mathbf{P}$ does not contain a line then

$$\mathrm{rank}_{d/2}(\mathbf{P}) \leq (2m)^{2^{d(1+o(1))}}.$$

▶ Remark 5. Just for the special case of $\mathbb{F}_3$, there is a vast literature on subsets of $\mathbb{F}_3^m$ avoiding a line, also known as cap sets. The most basic example of a cap set is $\{0,1\}^m$; this cap set leads to the following easy observation: for any (low-degree) polynomial map $\mathbf{P}\colon \mathbb{F}_3^n \to \mathbb{F}_3^m$, in any number of variables n, its "square" $\mathbf{P}^2 := (P_1^2, \ldots, P_m^2)\colon \mathbb{F}_3^n \to \mathbb{F}_3^m$ is a (low-degree) polynomial map with a line-free image. More generally for $\mathbb{F}_p$, with any prime $p \geq 3$, consider for example the analogue of Behrend's construction (see, e.g., Lemma 17 in [10]); it corresponds to the following observation: if $\mathrm{Im}(P_i) \subseteq \{0, 1, \ldots, (p-1)/2\}$ for every i, and the real number $\sum_{i=1}^m P_i^2$ is constant, then the image of $\mathbf{P} = (P_1, \ldots, P_m)$ contains no three points on a line, and in particular is line-free. See, e.g., [8] and the references within for more constructions of cap sets.

One can significantly extend this result, by considering the univariate degree udeg, which we recall is the smallest degree of a curve contained in the image of $\mathbf{P}$. We note that $\mathrm{udeg}(\mathbf{P}) = 1$ if and only if $\operatorname{Im}\mathbf{P}$ contains a line (and for $m = 1$ this means surjectivity).

It is not hard to show that, as n tends to infinity, we cannot hope to bound $\mathrm{rank}_t(P)$ for $t < \deg(P)/\mathrm{udeg}(P)$ (see, e.g., Section 3.1). Karam proved (see Theorem 1.3[10] in [21]) that we indeed can for $t = \deg(P)/\mathrm{udeg}(P)$.

▶ **Theorem 6** ([21]). *Let $P\colon \mathbb{F}^n \to \mathbb{F}$ be a polynomial with $\deg(P) \leq d$, over any prime finite field $\mathbb{F} = \mathbb{F}_p$ with $d < p$. For $t = \deg(P)/\mathrm{udeg}(P)$, we have $\mathrm{rank}_t(P) \leq \gamma(p,d)$, for some function γ.*

The proof uses a variant of the usual regularity lemma for polynomials, and thus yields an extremely weak bound $\gamma(p,d)$ on $\mathrm{rank}_t(P)$.

We show that the weak regularity lemma is sufficiently powerful to bound rank_t for any polynomial, and in fact also for any polynomial map and over non-prime fields too, yielding a correspondingly strong bound.

▶ **Theorem** (see Theorem 24). Let $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$ be a polynomial map with $\deg(\mathbf{P}) \leq d$, over any finite field $\mathbb{F}$ with $d < \mathrm{char}(\mathbb{F})$. For $t = \deg(\mathbf{P})/\mathrm{udeg}(\mathbf{P})$,

$$\mathrm{rank}_t(\mathbf{P}) \leq (2m)^{2^{d(1+o(1))}}.$$

Roughly speaking, our proof follows by showing that for any weak ϵ-regular decomposition $\mathbf{P} = \mathbf{F}(\mathbf{X})$, for ϵ bounded away from 1, we have $\mathrm{udeg}(\mathbf{P}) \leq \deg_1(\mathbf{F})$, the degree of $\mathbf{F}$ in its first variable (so $\operatorname{Im}\mathbf{U} \subseteq \operatorname{Im}\mathbf{P}$ for some non-constant curve $\mathbf{U}$ with $\deg(\mathbf{U}) = \deg_1(\mathbf{F})$). Using the guarantees of the homogeneity of $\mathbf{X}$ and of the maximality of $\deg(X_1)$ ($= \deg(\mathbf{X})$), and removing "redundant" monomials from $\mathbf{F}$, we will deduce $\deg_1(\mathbf{F}) \leq \deg(\mathbf{P})/\deg(\mathbf{X})$, that is, $\deg(\mathbf{X}) \leq \deg(\mathbf{P})/\mathrm{udeg}(\mathbf{P}) = t$, which would allow us to bound rank_t.

[10] More precisely, Karam's result is stated for the closely related notion of rk_t; see Section 3.

1.4 The proof

Towards the proof of the weak regularity lemma, we obtain results on linear spaces of polynomials and forms (homogeneous polynomials) that may be of independent interest. In particular, we analyze subspaces of high-rank polynomials (where the rank is allowed to grow with the number of spanning polynomials, i.e., the dimension of the space). More particularly, we analyze *punctured* spaces of high-rank polynomials. By a punctured space we mean the complement of a linear subspace: $V \setminus U$ where V is a linear space (of polynomials) and U is a strict subspace.[11] For comparison, the classical regularity lemmas for polynomials work only with the "trivial" punctured space $V \setminus \{0\}$; or more concretely, *all* nonzero linear combinations of the parts $\mathbf{X}$ are required to have high rank. Obtaining a high-rank punctured space $V \setminus U$, rather than a high-rank *trivial* punctured space $V \setminus \{0\}$, turns out to be achievable with far better bounds. We call this result a "rank-regularity lemma", and note that its strong bounds eventually translate to strong bounds for our weak regularity lemma. Using it, the rest of our proof has, very roughly, the following logical structure (see Section 2 for all definitions):

$$\begin{aligned}\mathbf{X}\ t\text{-rank-regular} &\Leftrightarrow \mathrm{rk}(X_1 + \mathrm{Sp}\,\mathbf{X}') > t|\mathbf{X}| \\ &\Rightarrow |\,\mathrm{bias}(X_1 + \mathrm{Sp}\,\mathbf{X}')| < \epsilon q^{-|\mathbf{X}|} \\ &\Rightarrow \mathbf{X}\ \text{weak}\ \epsilon\text{-regular}.\end{aligned}$$

The proof moreover guarantees the additional required conditions, e.g., that X_1 has maximal degree. To translate from high rank to low bias, our proof relies on recent results on the relationship between structure and randomness for polynomials (see e.g. [20, 31, 5, 32]). These results are proved by going through the corresponding multilinear forms – which requires the characteristic of the field to be larger than the degree of the parts – by relating analytic rank and partition rank (defined by Gowers and Wolf [13] and by Naslund [33], respectively).

1.5 Organization

In Section 2 we state and prove our weak regularity lemma. The proof has several components. In Section 2.1 we prove a rank-regularity lemma, which finds a small decomposition where every linear combination outside of a strict subspace has high rank. In Section 2.2 we show that a similar condition in terms of bias (rather than rank) suffices to get weak regularity, using Fourier-analytic techniques and known structure-vs-randomness results. Finally, in Section 2.3, we combine the aforementioned pieces to prove the weak regularity lemma. In Section 3 we obtain the applications of the weak regularity lemma outlined above.

1.6 Preliminaries

An important aspect of regularity lemmas for polynomials is that they are independent of the number of variables n (similarly to how, for example, the graph regularity lemma is independent of the number of vertices of the graph). As such, it will be convenient to use the notation $\mathrm{Poly}(\mathbb{F})$ for the set of all polynomials over $\mathbb{F}$ (meaning with coefficients in $\mathbb{F}$) in any (finite) number of variables $x_1, \ldots, x_n$ ($n \in \mathbb{N}$). While we work with (formal) polynomials in this paper, note that any n-variate polynomial over $\mathbb{F}$ determines a function $\mathbb{F}^n \to \mathbb{F}$

[11] For example, a punctured space of codimension 1 ($V \setminus U$ with $\mathrm{codim}_V(U) = 1$) is of the form $\{cw+u \,|\, c \neq 0, u \in U\}$ for some $w \in V$.

(uniquely so if every variable has degree smaller than $|\mathbb{F}|$). We write $\mathrm{Poly}_d(\mathbb{F})$ for the set of polynomials of (total) degree at most d. We henceforth omit $\mathbb{F}$ from our notation when it is clear from context. Naturally, $\mathrm{Poly}^m = \{(P_1, \ldots, P_m) \,|\, \forall i\colon P_i \in \mathrm{Poly}\}$ stands for the set of m-tuples of polynomials, or equivalently, polynomial maps into $\mathbb{F}^m$; and analogously for Poly_d^m. Similarly, we denote by Form the set of all homogeneous polynomials (or forms for short) of *positive* degree, in any (finite) number of variables; we denote by Form_d the set of forms of degree d; and by Form_d^m the set of m-tuples of forms of degree d. For polynomials $X_1, \ldots, X_r \in \mathrm{Poly}(\mathbb{F})$, we denote by $\mathbb{F}[X_1, \ldots, X_r]$ the algebra they generate, that is, $\{F(X_1, \ldots, X_r) \,|\, r\text{-variate } F \in \mathrm{Poly}\}$. Of course, if $x_1, \ldots, x_n$ are independent variables then $\mathbb{F}[x_1, \ldots, x_n]$ is the usual ring of n-variate polynomials in $x_1, \ldots, x_n$. We denote by $\deg_i(P)$ the degree of P in the variable x_i (that is, its degree when viewed as a univariate polynomial in x_i).

We will frequently use boldface letters to emphasize that an object is a tuple; so for example, we will use $\mathbf{P}$, instead of P, when dealing with a tuple of polynomials (or a polynomial map). We will use the convention that for a tuple $\mathbf{X} = (X_1, \ldots, X_k)$ and a set A, the notation $\mathbf{X} \subseteq A$ is a shorthand for $X_i \in A$ for all i. For $\mathbf{P} \subseteq \mathrm{Poly}$ we denote $\deg(\mathbf{P}) = \max_i \deg(P_i)$. More generally, for a set $V \subseteq \mathrm{Poly}$ we denote $\deg(V) = \max_{P \in V} \deg(P)$.

Probabilities, unless otherwise specified, are to be understood with respect to a uniformly random input; that is, for $\mathbf{X}\colon \mathbb{F}^n \to \mathbb{F}^k$, $\Pr[\mathbf{X} = y]$ is a shorthand for $\Pr_{x \in \mathbb{F}^n}[\mathbf{X}(x) = y]$ with x chosen uniformly at random from $\mathbb{F}^n$. The image (or range) of a polynomial map $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$ is denoted $\operatorname{Im}\mathbf{P} = \{\mathbf{P}(x) | x \in \mathbb{F}^n\}$ ($\subseteq \mathbb{F}^m$). For a subset $S \subseteq \mathbb{F}^n$, we write $\mathbf{P}(S) = \{\mathbf{P}(x) | x \in S\}$.

Note that $\mathrm{Poly}_d(\mathbb{F})$ is a linear space over $\mathbb{F}$. We write $U \leq V$ when U is a subspace of V. We denote the linear span of $\mathbf{X} \in \mathrm{Poly}_d^k(\mathbb{F})$ by $\operatorname{Sp}\mathbf{X} = \operatorname{Sp}\{X_1, \ldots, X_k\} = \{a \cdot \mathbf{X} | a \in \mathbb{F}^k\}$, which is of course a subspace of $\mathrm{Poly}_d(\mathbb{F})$.

For $\mathbf{P} \in \mathrm{Poly}^m(\mathbb{F})$, we refer to an expression of the form $\mathbf{P} = \mathbf{F}(\mathbf{X})$ (or $\mathbf{P} = \mathbf{F} \circ \mathbf{X}$), with $\mathbf{F} \in \mathrm{Poly}^m(\mathbb{F})$ and $\mathbf{X} \in \mathrm{Poly}^k(\mathbb{F})$, as a *decomposition* of $\mathbf{P}$ (so that for every component P_i of $\mathbf{P}$, we have the decomposition $P_i = F_i(\mathbf{X})$). Note that this is a composition of (formal) polynomials. When the map $\mathbf{F}$ is immaterial, we denote the decomposition by $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$.

As mentioned above, all fields considered in this paper are finite; we denote by $\mathbb{F}_q$ the finite field of size q (so q is a power of the characteristic $\mathrm{char}(\mathbb{F})$).

2 Weak Regularity

For convenience, we repeat here the definitions of weak regularity and the weak regularity lemma. Recall that for a k-tuple $\mathbf{y} = (y_1, \ldots, y_k)$, we denote by $\mathbf{y}'$ the $(k-1)$-tuple $\mathbf{y}' = (y_2, \ldots, y_k)$.

▶ **Definition 7** (weak ϵ-regularity). *Let $\mathbb{F} = \mathbb{F}_q$. We say that $\mathbf{X} \in \mathrm{Form}^k(\mathbb{F})$ is* weak ϵ-regular *if $\deg(X_1) = \deg(\mathbf{X})$, and for every $y \in \mathbb{F}^k$,*

$$\left| \Pr[\mathbf{X} = \mathbf{y}] - q^{-1} \Pr[\mathbf{X}' = \mathbf{y}'] \right| \leq \epsilon q^{-k}.$$

For $\mathbf{P} \subseteq \mathrm{Poly}(\mathbb{F})$, we say that $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$ is a weak ϵ-regular decomposition *if $\mathbf{X} \subseteq \mathrm{Form}$ is weak ϵ-regular and $\mathbf{P} \not\subseteq \mathbb{F}[\mathbf{X}']$.*[12]

[12] For completeness, the empty decomposition ($k = 0$) is trivially weak ϵ-regular for any ϵ. Thus, constant polynomial maps $\mathbf{P}$ trivially have a weak ϵ-regular decomposition.

Roughly speaking, a set of forms $\mathbf{X} \subseteq \mathrm{Form}$ is weak ϵ-regular if at least one of its members, of largest degree, is approximately "unbiased" and "independent" of the rest; moreover, a polynomial map $\mathbf{P}$ has a weak ϵ-regular decomposition if it can be expressed in terms of a weak ϵ-regular $\mathbf{X}$, and depends on X_1. Intuitively, P can be thought of as a polynomial in only k variables, and although these may not be independent, free variables, at least one of them approximately is.

Let us also note that the probabilistic condition in Definition 7 means that $\Pr[\mathbf{X} = \mathbf{y}]/q^{-k} = \alpha \pm \epsilon$ for every $\mathbf{y} \in \mathbb{F}^k$ with $\Pr[\mathbf{X}' = \mathbf{y}']/q^{-(k-1)} = \alpha$ (recall $\mathbf{X}'$ is a $(k-1)$-tuple). Intuitively, this means $\mathbf{X}$ is distributed like $\mathbf{X}'$ up to an additive error of ϵ. We also mention that Definition 7 implies $\deg(\mathbf{X}) \leq \deg(\mathbf{P})$ (see Corollary 29).

▶ **Theorem 8** (weak regularity lemma). *Let $c \geq 1$. Every $\mathbf{P} \in \mathrm{Poly}_d^m(\mathbb{F}_q)$, with $d < \mathrm{char}(\mathbb{F}_q)$, has a weak ϵ-regular decomposition of size*

$$k \leq (2mc)^{2^{d(1+o(1))}},$$

where we write $\epsilon q^{-k} = q^{-ck}$ for the error term.

Note that the decomposition size is polynomial in m. We remark that Theorem 8 is even stronger than the statement in the Introduction (Theorem 1), since here ϵ is a function of the decomposition size k (namely, $\epsilon = q^{-(c-1)k}$).

2.1 Rank-regularity

The proof of the weak regularity lemma has several components. At the core of the proof is what we call a *rank-regularity lemma*[13] (see Definition 10). Roughly, the requirement for the parts $\mathbf{X}$ in a rank-regular decomposition is that they are homogeneous (not necessarily all of the same degree), and that every linear combination of $\mathbf{X}$ outside of a strict subspace has high rank (see Definition 9). We will later convert a rank-regular decomposition into a weak regular decomposition. We begin by setting up the basic definitions we need, and state the results that form the steps of our overall strategy.

An important definition in the study of polynomials is that of *rank*, sometimes also called strength [1] or Schmidt rank [36]. Recall that a polynomial is *reducible* if it can be factorized as the product of two non-constant polynomials (all polynomials over the same field).

▶ **Definition 9** (rank). *The* rank *of a form $P \in \mathrm{Form}_d(\mathbb{F})$ is:*

- *For $d \geq 2$: the least number of reducible forms that sum up to P:*

$$\mathrm{rk}(P) = \min\left\{r \,\middle|\, \exists R_1, \ldots, R_r \in \mathrm{Form}_d(\mathbb{F}) \text{ reducible}: P = R_1 + \cdots + R_r\right\}.$$

- *For $d = 1$, $\mathrm{rk}(P) = \infty$.*

More generally, the rank of a polynomial $P \in \mathrm{Poly}(\mathbb{F})$ is the rank of its highest-degree homogeneous part,[14] *or 0 if P is a constant.*

Note that for a homogeneous $P \in \mathrm{Form}_d$ with $\mathrm{rk}(P) \leq r$ $(< \infty)$, we can write $P = Q(Y_1, \ldots, Y_k)$ with $k \leq 2r$, for some homogeneous $Q \in \mathrm{Form}_2$ and $Y_i \in \mathrm{Form}$ with $\deg(Y_i) < d$. This uses the fact that a reducible form R factorizes as the product of two forms (of lower degree); indeed, the product of the lowest-degree homogeneous parts of the factors is a homogeneous part of R.

[13] Or "rankularity" lemma for short.

[14] If $\deg(P) = d$ and we write $P = \sum_{i=0}^{d} P_i$ with P_i a form of degree i, then $\mathrm{rk}(P) = \mathrm{rk}(P_d)$.

The core of the proof of our weak regularity lemma is a weak regularity lemma for rk. For $\mathbf{P} \subseteq \mathrm{Poly}$, call a decomposition $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$ *minimal* if $\mathbf{P} \not\subseteq \mathbb{F}[\mathbf{X}^*]$ for every $\mathbf{X}^* \subseteq \mathrm{Form}$ with $\mathrm{Sp}\,\mathbf{X}^* \subsetneq \mathrm{Sp}\,\mathbf{X}$. Put differently, no strict subspace $V' \lneq \mathrm{Sp}\,\mathbf{X}$ generates $\mathbf{P}$ (where V' has a basis of forms $\mathbf{X}^*$).

▶ **Definition 10** (rank-regularity). *For any $t \geq 0$:*

- $\mathbf{X} \in \mathrm{Form}^r$ *is* t-rank-regular *if* $\mathrm{rk}(X) \geq tr$ *for all* $X \in \mathrm{Sp}\,\mathbf{X}$ *outside of a strict subspace.*
- $\mathbf{P} \in \mathrm{Poly}^m(\mathbb{F})$ *has a* t-rank-regular decomposition of size r *if* $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$ *is a minimal decomposition for some* t-rank-regular $\mathbf{X} \in \mathrm{Form}^r(\mathbb{F})$.

Note that "all $X \in \mathrm{Sp}\,\mathbf{X}$ outside of a strict subspace" refers, explicitly, to all $X \in (\mathrm{Sp}\,\mathbf{X}) \setminus U$ for some strict subspace $U \lneq \mathrm{Sp}\,\mathbf{X}$.[15] In other words, $\mathbf{X}$ is t-rank-regular if $\mathrm{Sp}\{X \in \mathrm{Sp}\,\mathbf{X} \mid \mathrm{rk}(X) < tr\} \neq \mathrm{Sp}\,\mathbf{X}$. This aspect[16] of the definition is one reason that we are able to obtain strong bounds for this regularity lemma. We also mention that Definition 10 implies $\deg(\mathbf{X}) \leq \deg(\mathbf{P})$ (see Corollary 29).

▶ **Theorem 11** (rank-regularity lemma). *Every $\mathbf{P} \in \mathrm{Poly}_d^m$ has a t-rank-regular decomposition of size at most $((2t+1)dm)^{2^d}$.*

2.1.1 Linear spaces of polynomials

Here we prove several useful properties of linear spaces of polynomials, which will be used in the proof of the rank-regularity lemma and the weak regularity lemma. We begin with some useful notation.

Henceforth, for a linear space $V \leq \mathrm{Poly}$ and $R > 0$, we denote the span of its low-rank polynomials by

$$U_R(V) = \mathrm{Sp}\{X \in V \mid \mathrm{rk}(X) < R\}.$$

Note that $\mathbf{X} \in \mathrm{Form}^k$ is t-rank-regular if and only if $U_R(V) \neq V$ for $V = \mathrm{Sp}\,\mathbf{X}$ and $R = tk$.

We say that a finite-dimensional linear space $V \leq \mathrm{Poly}$ is *homogeneous* if V is spanned by forms.[17] Equivalently, V is the sum – and thus the direct sum – of its homogeneous parts: $V = \sum_{i \geq 0} V_i$ where $V_i := V \cap \mathrm{Form}_i$. We abbreviate $V^{\uparrow} = V_{\deg(V)}$, that is,

$$V^{\uparrow} = V \cap \mathrm{Form}_{\deg(V)},$$

where we recall the notation $\deg(V) = \max_{X \in V} \deg(X)$.

For the proof of the rank-regularity lemma (Theorem 11) we will need the following lemma on rank-regularity of forms of top degree.

▶ **Lemma 12.** *Let $\mathbf{Y} \in \mathrm{Form}^m$, and let $\mathbf{Y}^* \in \mathrm{Form}^m$ have the same underlying set as $\mathbf{Y}^{\uparrow}$. For any t, if $\mathbf{Y}^*$ is t-rank-regular then $\mathbf{Y}$ is t-rank-regular.*

Proof. Put $V = \mathrm{Sp}(\mathbf{Y})$, $V^* = \mathrm{Sp}(\mathbf{Y}^*)$, and note $V^* = \mathrm{Sp}(\mathbf{Y}^{\uparrow}) = (\mathrm{Sp}\,\mathbf{Y})^{\uparrow} = V^{\uparrow}$. Let $R = tm$, and suppose $U_R(V) = V$. We need to show $U_R(V^*) = V^*$ (with the same R, as the tuples $\mathbf{Y}$ and $\mathbf{Y}^*$ have the same number m of polynomials), or equivalently, $U_R(V^{\uparrow}) = V^{\uparrow}$.

[15] For example, U might be $\mathrm{Sp}(\mathbf{X}')$, provided $\mathbf{X}$ is linearly independent.

[16] Equivalently, this is about high-rank punctured spaces: if $V \leq \mathrm{Poly}$ has a punctured space W ($= V \setminus U$, $U \lneq V$) with $\mathrm{rk}(P) \geq t \dim(V)$ for every $P \in W$, then any basis of forms for V is t-rank-regular (and vice versa).

[17] By convention, the zero space $\{0\}$ is homogeneous.

Let $P \in V^{\uparrow}$, that is, $P \in V$ is a form with $\deg(P) = \deg(V)$. Since $U_R(V) = V$, P is a linear combination of $P_i \in V$ with $\mathrm{rk}(P_i) < R$. Then P is a linear combination of their homogeneous parts of degree $\deg(V)$, which are:

- also in V (as V is homogeneous), and thus in $V^{\uparrow}$, and
- of rank $< R$ (by definition of rank for non-homogeneous polynomials).

Thus, $P \in U_R(V^{\uparrow})$. We deduce that $V^{\uparrow} \subseteq U_R(V^{\uparrow})$, and since the containment in the other direction trivially holds, we are done. ◀

For the rest of the proof of the weak regularity lemma (Theorem 8), we will need Lemma 14 and Lemma 15 below.

We begin with the following closure property of the subspaces $U_R(V)$.

▷ Claim 13. Let $V \leq \mathrm{Poly}$, $R > 0$. Every $X \in V$ with $\deg(X) < \deg(U_R(V))$ satisfies $X \in U_R(V)$.

Proof. Abbreviate $U = U_R(V)$. Let $X_0 \in U$ with $\mathrm{rk}(X_0) < R$ of maximal degree, $\deg(X_0) = \deg(U)$. We claim that $X_0 + X \in U$. Indeed, since $\deg(X) < \deg(U) = \deg(X_0)$, we have $\mathrm{rk}(X_0 + X) = \mathrm{rk}((X_0 + X)^{\uparrow}) = \mathrm{rk}(X_0^{\uparrow}) = \mathrm{rk}(X_0) < R$. Thus, as desired, $X \in U$, being the difference $X = (X_0 + X) - X_0$ of two members of the linear space U. ◁

▶ **Lemma 14.** *Let $V \leq \mathrm{Poly}$, $R > 0$. If V is homogeneous then so is $U_R(V)$.*

Proof. Put $U = U_R(V)$. For $X \in \mathrm{Poly}$, denote by $X^{(i)} \in \mathrm{Form}_i$ the degree-i part of X (so that $X = \sum_i X^{(i)}$). First, we prove another closure property of U (in a homogeneous V): for every $X \in V$, if $\mathrm{rk}(X) < R$ then $X^{(i)} \in U$ for every i. Let us prove this. If $i = \deg(X)$ (that is, $X^{(i)} = X^{\uparrow}$) then $\mathrm{rk}(X^{\uparrow}) = \mathrm{rk}(X) < R$; since V is homogeneous, $X^{\uparrow} \in V$, so $X^{\uparrow} \in U$, as needed. On the other hand, if $i < \deg(X)$ then $\deg(X^{(i)}) < \deg(X) \leq \deg(U)$, since $X \in U$, and since $X^{(i)} \in V$ by the homogeneity of V, we can apply Claim 13 on $X^{(i)}$ to deduce $X^{(i)} \in U$. This proves our claim.

Let $\mathbf{X} = \{X \in V \,|\, \mathrm{rk}(X) < R\}$, so that $U = \mathrm{Sp}(\mathbf{X})$ by definition. By the claim above, $\mathbf{X}^* := \{X^{(i)} \,|\, X \in \mathbf{X}, i \leq \deg(X)\}$ satisfies $\mathbf{X}^* \subseteq U$. Since $U = \mathrm{Sp}(\mathbf{X}) \subseteq \mathrm{Sp}(\mathbf{X}^*) \subseteq U$, we deduce that U has a spanning set of forms, which completes the proof. ◀

▶ **Lemma 15.** *Let $V \leq \mathrm{Poly}$, $R > 0$. If V is homogeneous, then $U_R(V) = V$ if and only if $U_R(V^{\uparrow}) = V^{\uparrow}$.*

Proof. The forward direction is equivalent to Lemma 12: if $U_R(V) = V$, then $\deg(U_R(V)) = \deg(V)$, which implies $V^{\uparrow} \subseteq U_R(V)^{\uparrow} \subseteq U_R(V^{\uparrow})$, as needed.

For the other direction, suppose $U_R(V^{\uparrow}) = V^{\uparrow}$. Then $V^{\uparrow} \subseteq U_R(V)$, as $U_R(V^{\uparrow}) \subseteq U_R(V)$, and in particular, $\deg(U_R(V)) = \deg(V)$. Let us denote $V' = \bigoplus_{i<\deg(V)} V_i$. Claim 13 says $V' \subseteq U_R(V)$. Then $V = V^{\uparrow} + V' \subseteq U_R(V) + U_R(V) = U_R(V) \subseteq V$, that is, $U_R(V) = V$, as needed. ◀

2.1.2 Proof of the rank-regularity lemma

Theorem 11 will be proved using a repeated application of the following decomposition lemma. It shows that if a tuple of forms $\mathbf{X}$, all of the same degree d, is not t-rank-regular, then we can decompose it into not-too-many forms of varying degrees, all strictly smaller than d. Crucially, the lemma considers not just individual low-rank forms, but the whole subspace they span.

▶ **Lemma 16.** *If* $\mathbf{X} \in \mathrm{Form}_d^r(\mathbb{F})$ *is not* t*-rank-regular, then* $\mathbf{X} \subseteq \mathbb{F}[\mathbf{Y}]$ *for some* $\mathbf{Y} \in \mathrm{Form}^R$ *with* $\deg(\mathbf{Y}) < d$ *and* $R \leq 2tr^2$*.*

Proof. Assume $d \geq 2$, as otherwise the hypothesis is vacuous. Put $V = \mathrm{Sp}\,\mathbf{X}$ $(\leq \mathrm{Form}_d)$ and $k = \dim V$ $(\leq r)$. Let

$$\mathbf{X}^- = \{X \in V \mid \mathrm{rk}(X) < tr\}.$$

Suppose $\mathbf{X}$ is not t-rank-regular, so that $\mathrm{Sp}(\mathbf{X}^-) = V$. Thus, $\mathbf{X}^-$ contains a basis $\mathbf{B} = (B_1, \ldots, B_k) \in \mathrm{Form}_d^k$ for V. This has the following implications:

- Since $\mathbf{X} \subseteq \mathrm{Sp}\,\mathbf{B}$ $(= V)$, there is a linear map $\mathbf{L}\colon \mathbb{F}^k \to \mathbb{F}^r$ such that $\mathbf{X} = \mathbf{L}(\mathbf{B})$.
- Since $\mathbf{B} \subseteq \mathbf{X}^- \cap \mathrm{Form}$, and by the definition of rk (recall Definition 9 and the subsequent remark), we have $\mathbf{B} = \mathbf{Q}(\mathbf{Y})$ for some quadratic $\mathbf{Q} \in \mathrm{Form}_2^k$ and some $\mathbf{Y} \in \mathrm{Form}^R$ with $\deg(\mathbf{Y}) < d$, where $R \leq \sum_{i\in[k]} 2\,\mathrm{rk}(B_i) < 2tr \cdot k$.

Combining the above, we deduce that $\mathbf{X} = \mathbf{L}(\mathbf{Q}(\mathbf{Y})) = \mathbf{F}(\mathbf{Y})$, where $\mathbf{F} := \mathbf{L}(\mathbf{Q}) \in \mathrm{Form}_2^r$ and $\mathbf{Y} \in \mathrm{Form}^R$ with $\deg(\mathbf{Y}) < d$ and $R < 2tr \cdot k \leq 2tr^2$, which completes the proof. ◀

We prove the rank-regularity lemma by an iterative construction with at most d steps, maintaining at all times a tuple of forms of various degrees. At each step, we move to a minimal decomposition, and if it is not a t-rank-regular decomposition then, we show, its subset of forms of maximal degree is also not t-rank-regular. This last deduction requires care – as the rank requirement in Definition 10 ($\mathrm{rk}(X) \geq tr$) depends on the number of polynomials in the decomposition – and follows using Lemma 12. Having done that, we decompose the maximum-degree forms in terms of lower degree forms using Lemma 16, which strictly decreases the maximum degree of the forms in the decomposition, as needed.

Proof of Theorem 11. Assume $\mathbf{P}$ is non-constant, as otherwise there is nothing to prove. Let $\mathbf{X^0} \in \mathrm{Form}^{r_0}$ be obtained from $\mathbf{P} \in \mathrm{Poly}^m$ by replacing each component of $\mathbf{P}$ with its nonzero homogeneous parts of degree $1, 2, \ldots, d$. Note that $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X^0}]$ and $r_0 \leq dm$.

We claim that for some $s \geq 0$, there are s homogeneous polynomial maps $\mathbf{X^1}, \ldots, \mathbf{X^s}$, with $\mathbf{X^i} \in \mathrm{Form}^{r_i}$ for $i = 1, \ldots, s$, such that $\mathbf{P} \in \mathbb{F}[\mathbf{X^s}]$ is a t-rank-regular decomposition, and for every $i \geq 0$:

- $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X^i}]$,
- $\deg(\mathbf{X^i}) \leq d - i$, and
- $r_i \leq (2t+1)^{2^i-1}(dm)^{2^i}$.

The process terminates (that is, s is well defined); indeed, any $\mathbf{X} \subseteq \mathrm{Form}$ with $\deg(\mathbf{X}) = 1$ is t-rank-regular for any t, since $\mathrm{rk}(X) = \infty$ for every $X \in \mathrm{Sp}(\mathbf{X}) \setminus \{0\}$, so a minimal decomposition $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$ with $\deg(\mathbf{X}) = 1$ is a t-rank-regular decomposition for any t. Proving the above claim would complete the proof, since $1 \leq \deg(\mathbf{X^s}) \leq d - s$ implies $s < d$, and thus $\mathbf{X^s} \in \mathrm{Form}^{r_s}$ with $r_s \leq ((2t+1)dm)^{2^d}$, as desired.

For each i starting with $i = 0$, given $\mathbf{X^i}$ we proceed as follows:

- Consider a homogeneous subspace $W \leq \mathrm{Sp}(\mathbf{X^i})$ of minimal dimension satisfying $\mathbf{P} \subseteq \mathbb{F}[W]$, and let $\mathbf{Y} \in \mathrm{Form}^{r'}$ be a basis of W. Then $\mathbf{P} \subseteq \mathbb{F}[\mathbf{Y}]$ is a minimal decomposition, and moreover, $\deg(\mathbf{Y}) \leq \deg(W) \leq \deg(\mathbf{X^i})$, and $r' = \dim W \leq \dim \mathrm{Sp}(\mathbf{X^i}) \leq r_i$.
- Now, if $\mathbf{P} \subseteq \mathbb{F}[\mathbf{Y}]$ is t-rank-regular then set $s = i$, $\mathbf{X^s} = \mathbf{Y}$ and $r_s = r'$, finishing the construction. Otherwise, $\mathbf{Y}$ must not be t-rank-regular. We will use $\mathbf{Y}$ to construct $\mathbf{X^{i+1}}$ with $\deg(\mathbf{X^{i+1}}) < \deg(\mathbf{X^i})$. In order to apply Lemma 16, we need a tuple of forms satisfying two conditions: they are all of the same degree, and the tuple is not t-rank-regular.

Put $\ell = \deg(\mathbf{Y})$, and obtain $\mathbf{Y}^* \in \mathrm{Form}^{r'}$ from $\mathbf{Y}$ by replacing, in each component of $\mathbf{Y}$, a polynomial of degree smaller than ℓ by another polynomial from $\mathbf{Y}$ of degree ℓ.[18] By Lemma 12, $\mathbf{Y}^*$ is not t-rank-regular.
Apply Lemma 16 on $\mathbf{Y}^*$. Merge the resulting tuple of forms with the forms in $\mathbf{Y}$ of degree smaller than ℓ to obtain $\mathbf{X^{i+1}} \in \mathrm{Form}^{r_{i+1}}$ with $\mathbf{Y} \subseteq \mathbb{F}[\mathbf{X^{i+1}}]$, $\deg(\mathbf{X^{i+1}}) < \deg(\mathbf{X^i})$, and $r_{i+1} \leq 2tr_i^2 + r_i \leq (2t+1)r_i^2$.

To prove that this construction gives the claim above, we proceed by induction on $i \geq 0$. The induction base $i = 0$ follows trivially from the construction of $\mathbf{X^0}$, as $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X^0}]$ with $\deg(\mathbf{X^0}) \leq d$, and $r_0 \leq dm = (2t+1)^0(dm)^1$. For the induction step we have, by the construction of $\mathbf{X^{i+1}}$ and the induction hypothesis, that $\mathbf{P} \subseteq \mathbb{F}[\mathbf{Y}] \subseteq \mathbb{F}[\mathbf{X^{i+1}}]$, that $\deg(\mathbf{X^{i+1}}) \leq \deg(\mathbf{X^i}) - 1 \leq d - (i+1)$, and that $r_{i+1} \leq (2t+1)r_i^2 \leq (2t+1)((2t+1)^{2^i-1}(dm)^{2^i})^2 = (2t+1)^{2^{i+1}-1}(dm)^{2^{i+1}}$, as needed. This completes the proof. ◀

2.2 Bias and weak regularity

We will use the notion of bias of a polynomial, which is a standard analytic measure related to deviation from equidistribution. Henceforth, fix any nontrivial additive character[19] $\chi\colon \mathbb{F} \to \mathbb{C}$ (for concreteness, if $\mathbb{F} = \mathbb{F}_p$ is a prime field, take $\chi(y) = \omega^y$, where ω is the principal root of unity, $\omega = e^{2\pi i/p}$, and y on the right hand side is viewed as an element of $\{0, \ldots, p-1\}$.) The *bias* of $P \in \mathrm{Poly}(\mathbb{F})$ is the complex number

$$\mathrm{bias}(P) = \mathop{\mathbb{E}}_{x \in \mathbb{F}^n} \chi(P(x)).$$

As we will see below, simple analysis relates bias and equidistribution as follows (see also e.g. [2], or Claim 33 in [4]):

$$\forall y \in \mathbb{F}\colon \left| \Pr[P = y] - \frac{1}{|\mathbb{F}|} \right| \leq \max_{a \in \mathbb{F}^\times} |\,\mathrm{bias}(aP)|.$$

Towards proving that rank-regularity implies weak regularity, we will use the best bound known for the structure versus randomness problem for polynomials, which is almost linear.

▶ **Theorem 17** ([32], Corollary 8.5 and Remark 8.3). *For any $P \in \mathrm{Poly}_d(\mathbb{F})$ with $\mathrm{char}(\mathbb{F}) > d$,*

$$\mathrm{rk}(P) \geq r \quad \Rightarrow \quad |\,\mathrm{bias}(P)| \leq |\mathbb{F}|^{-c_d r/\log_2(r+1)}$$

and $c_d = \Omega_d(1)$, namely $c_d \geq 2^{-d^{1+o(1)}}$.

We remark that [32] even gives that the logarithmic term in Theorem 17 goes down with $|\mathbb{F}|$ (namely, denominator $\log_{|\mathbb{F}|}(r+1)+1$), but we do not need this. We also remark that one can replace the use of Theorem 17 in our proof by earlier bounds of Milićević [31] or Janzer [20], and still obtain reasonable, albeit somewhat worse bounds in Theorem 8.

Using bias, a Fourier-analytic argument will help us to deduce weak regularity.

▶ **Lemma 18.** *Let $U \lneq V \leq \mathrm{Poly}(\mathbb{F}_q)$ with $\dim(V) = k$. Suppose $|\,\mathrm{bias}(X)| \leq \epsilon q^{-k}$ for all $X \in V \setminus U$. Then any basis $\mathbf{X} \in \mathrm{Form}^k$ of V that contains a basis of U, with $X_1 \notin U$ and $\deg(X_1) = \deg(\mathbf{X})$, is weak ϵ-regular.*

[18] That is, $\mathbf{Y}^*$ is obtained from $\mathbf{Y}^\uparrow$ by arbitrarily duplicating components to achieve size r'.
[19] A nontrivial additive character of $\mathbb{F}$ is a homomorphism $\chi\colon \mathbb{F} \to \mathbb{C}^\times$ ($\chi(a+b) = \chi(a)\chi(b)$) with $\chi \not\equiv 1$. Equivalently, $\chi(y) = \omega^{\mathrm{Tr}(cy)}$, with $\omega = e^{2\pi i/p}$, $p = \mathrm{char}(\mathbb{F})$, $c \in \mathbb{F}^\times$, and Tr is the trace of $\mathbb{F}$ on $\mathbb{F}_p$.

That is, Lemma 18 requires of $\mathbf{X}$ that: $\mathbf{X}$ is a basis of V, $U \subseteq \mathrm{Sp}(\mathbf{X}')$, $X_1 \notin U$, and $\deg(X_1) = \deg(\mathbf{X})$.

Recall that $\mathbf{X}$ being weak ϵ-regular implies that for every $y \in \mathbb{F}^k$,

$$\left| \Pr[\mathbf{X} = \mathbf{y}] - |\mathbb{F}|^{-1} \Pr[\mathbf{X}' = \mathbf{y}'] \right| \leq \epsilon |\mathbb{F}|^{-k}.$$

To prove Lemma 18, we will bound the left hand side by $\max_{a_1 \neq 0} |\operatorname{bias}(a \cdot \mathbf{X})|$.

It will be convenient to use a more geometric formulation of weak ϵ-regularity: for every line $\ell \subseteq \mathbb{F}^k$ in the direction of e_1 (so $\ell = \{te_1 + \mathbf{z} | t \in \mathbb{F}\}$ for some $\mathbf{z} \in \mathbb{F}^k$),[20]

$$\forall y \in \ell \colon |\Pr[\mathbf{X} = \mathbf{y}] - |\mathbb{F}|^{-1} \Pr[\mathbf{X} \in \ell]| \leq \epsilon |\mathbb{F}|^{-k}. \tag{1}$$

Proof of Lemma 18. We will use the standard fact that the distribution $(\Pr[\mathbf{X} = y])_{y \in \mathbb{F}^k}$ can be expressed in terms of biases, by using simple Fourier analysis. Indeed, since[21]

$$\forall v \in \mathbb{F}^m \colon \quad \mathop{\mathbb{E}}_{a \in \mathbb{F}^m} \chi(a \cdot v) = \delta[v = 0] := \begin{cases} 1 & v = 0 \\ 0 & v \neq 0 \end{cases}, \tag{2}$$

we have that for every $y \in \mathbb{F}^k$:

$$\Pr_x[\mathbf{X}(x) = y] = \mathop{\mathbb{E}}_x \delta[\mathbf{X}(x) - y = 0] = \mathop{\mathbb{E}}_x \mathop{\mathbb{E}}_{a \in \mathbb{F}^k} \chi(a \cdot (\mathbf{X}(x) - y)) = \mathop{\mathbb{E}}_{a \in \mathbb{F}^k} \operatorname{bias}(a \cdot (\mathbf{X} - y)). \tag{3}$$

Recall that $\mathbf{X}$ contains a basis for U, and $X_1 \notin U$. We claim that for every $a \in \mathbb{F}^k$ with $a_1 \neq 0$, we have that $a \cdot \mathbf{X} \notin U$. Indeed, write $a \cdot \mathbf{X} = a_1 X_1 + X'$ with $X' \in \mathrm{Sp}(\mathbf{X}')$. Our assumption on $\mathbf{X}$ implies that $U \subseteq \mathrm{Sp}(\mathbf{X}')$, so if $a \cdot \mathbf{X} \in U$ then $a_1 X_1 \in U - X' \subseteq \mathrm{Sp}(\mathbf{X}')$, a contradiction as $a_1 \neq 0$. Using our assumption on U, we deduce that

$$\forall a \in \mathbb{F}^k \colon \quad a_1 \neq 0 \quad \Rightarrow \quad |\operatorname{bias}(a \cdot \mathbf{X})| \leq \epsilon |\mathbb{F}|^{-k}. \tag{4}$$

Put $v = e_1 \in \mathbb{F}^k$. Let $\ell \subseteq \mathbb{F}^k$ be a line in the direction of v, so that $\ell = \{tv + z | t \in \mathbb{F}\}$ for some $z \in \ell$. We claim that the following equality holds:

$$\Pr_x[\mathbf{X}(x) \in \ell] = \mathop{\mathbb{E}}_{\substack{a \in \mathbb{F}^k : \\ a \cdot v = 0}} \operatorname{bias}(a \cdot (\mathbf{X} - z)). \tag{5}$$

Indeed,

$$\begin{aligned} \Pr_x[\mathbf{X}(x) \in \ell] &= \sum_{t \in \mathbb{F}} \Pr_x[\mathbf{X}(x) = tv + z] = \sum_{t \in \mathbb{F}} \mathop{\mathbb{E}}_{a \in \mathbb{F}^k} \mathop{\mathbb{E}}_x \chi(a \cdot (\mathbf{X}(x) - (tv + z))) \\ &= \mathop{\mathbb{E}}_{a \in \mathbb{F}^k} \Big(\sum_{t \in \mathbb{F}} \chi(t(a \cdot v)) \Big) \mathop{\mathbb{E}}_x \chi(a \cdot (\mathbf{X}(x) - z)) \\ &= \mathop{\mathbb{E}}_{a \in \mathbb{F}^k} \big(|\mathbb{F}| \delta[a \cdot v = 0] \big) \operatorname{bias}(a \cdot (\mathbf{X} - z)) = \frac{1}{|\mathbb{F}|^{k-1}} \sum_{\substack{a \in \mathbb{F}^k : \\ a \cdot v = 0}} \operatorname{bias}(a \cdot (\mathbf{X} - z)) \end{aligned}$$

where (3) is used in the second equality, and (2) is used in the penultimate equality with $m = 1$. Combining the point equality (3) and the line equality (5), we deduce (for $y = z$)

$$\Pr_x[\mathbf{X}(x) = z] = |\mathbb{F}|^{-1} \Pr_x[\mathbf{X}(x) \in \ell] + |\mathbb{F}|^{-k} \sum_{\substack{a \in \mathbb{F}^k : \\ a \cdot v \neq 0}} \operatorname{bias}(a \cdot (\mathbf{X} - z)).$$

[20] Indeed, any $\mathbf{y} \in \ell$ is of the form $\mathbf{y} = (t, \mathbf{z}')$ for some $t \in \mathbb{F}$, so $\Pr[\mathbf{X} \in \ell] = \Pr[\mathbf{X}' = \mathbf{z}']$ and $\mathbf{z}' = \mathbf{y}'$.

[21] $\sum_{a \in \mathbb{F}^m} \chi(a \cdot v) = \sum_{a \in \mathbb{F}^m} \prod_{i=1}^m \chi(a_i v_i) = \prod_{i=1}^m \sum_{a_i \in \mathbb{F}} \chi(a_i v_i) = \prod_{i=1}^m \mathbf{1}(v_i = 0) |\mathbb{F}| = \mathbf{1}(v = 0) |\mathbb{F}|^m$.

Now, we bound the magnitude of the rightmost term as follows:

$$|\mathbb{F}|^{-k}\Big|\sum_{\substack{a\in\mathbb{F}^k:\\ a\cdot v\neq 0}}\operatorname{bias}(a\cdot(\mathbf{X}-z))\Big| \le \max_{\substack{a\in\mathbb{F}^k:\\ a\cdot v\neq 0}}|\operatorname{bias}(a\cdot(\mathbf{X}-z))| = \max_{\substack{a\in\mathbb{F}^k:\\ a\cdot v\neq 0}}|\operatorname{bias}(a\cdot\mathbf{X})|.$$

Therefore,

$$\Big|\Pr_x[\mathbf{X}(x)=y]-|\mathbb{F}|^{-1}\Pr_x[\mathbf{X}(x)\in\ell]\Big| \le \max_{\substack{a\in\mathbb{F}^k:\\ a\cdot v\neq 0}}|\operatorname{bias}(a\cdot\mathbf{X})| \le \epsilon|\mathbb{F}|^{-k},$$

using (4) in the last inequality. Thus, $\mathbf{X}$ is weak ϵ-regular, as desired. ◀

2.3 Putting everything together

The proof of the weak regularity lemma will rely on Theorem 11, Theorem 17, and Lemma 18. Careful work will be required to convert rank-regularity, which gives information only about the highest-degree part (recall Definition 9) of the polynomials in our linear space, to weak regularity, which takes into account polynomials of all degrees, while still retaining the homogeneity of the parts. Moreover, rank-regularity depends only on the linear space spanned by the parts, whereas weak regularity depends on the particular parts (and so does not allow the first part to be repeated, for example).

Proof of Theorem 8. Put $g(t,d,m)=((2t+1)dm)^{2^d}$. We claim that the following holds for every polynomial $P\in\operatorname{Poly}_d(\mathbb{F})$ and every $1\le k\le g(t,d,m)$:

$$|\operatorname{bias}(P)|\ge q^{-ck} \quad\Rightarrow\quad \operatorname{rk}(P)<tk \tag{6}$$

for some $t=2^{d^{1+o(1)}}(c\log(2m))^{1+o(1)}$. Indeed, by Theorem 17, $\operatorname{rk}(P)\ge tk$ implies that $|\operatorname{bias}(P)|\le q^{-c_d tk/\log_2(tk+1)}$ with $c_d\ge 2^{-d^{1+o(1)}}$, that is,

$$\begin{aligned}(-\log_q|\operatorname{bias}(P)|)/k &\ge \frac{c_d t}{\log_2(tk+1)} \ge \frac{c_d t}{\log_2(tg(t,d,m)+1)}\\ &\ge \frac{c_d t}{(2^d+1)\log_2((2t+1)dm)} > c,\end{aligned}$$

that is, $|\operatorname{bias}(P)|<q^{-ck}$.

With t as above, apply Theorem 11 to find a t-rank-regular decomposition $\mathbf{Y}\subseteq\text{Form of }\mathbf{P}$ of size at most $g(t,d,m)$ so that $\mathbf{P}\subseteq\mathbb{F}[\mathbf{Y}]$ is a minimal decomposition and $\mathbf{Y}$ is t-rank-regular. For simplicity, assume without loss of generality that $\mathbf{Y}$ is linearly independent[22] (and nonempty, as otherwise there is nothing to prove). Put $V=\operatorname{Sp}\mathbf{Y}$ and $k=\dim V$ (>0). To finish, we will use $\mathbf{Y}$ to obtain a weak ϵ-regular decomposition of $\mathbf{P}$ of size $k\le g(t,d,m)\le(2mc)^{2^{d(1+o(1))}}$.

Consider the subspace of V given by

$$U=\operatorname{Sp}\{X\in V\mid \operatorname{rk}(X)<tk\},$$

and the subset of V given by

$$U_\epsilon=\{X\in V\mid |\operatorname{bias}(X)|\ge q^{-ck}\}.$$

Then:

[22] This preserves $\mathbb{F}[\mathbf{Y}]$ and minimality, and $\operatorname{rk}(X)\ge t\cdot\dim\operatorname{Sp}(\mathbf{Y})$ outside of a strict subspace.

1. U is homogeneous. This is by Lemma 14 and since V is (it is spanned by $\mathbf{Y} \subseteq \mathrm{Form}$).
2. $V^{\uparrow} \setminus U \neq \emptyset$. Indeed, by Lemma 15, the t-rank-regularity of $\mathbf{Y}$ implies $U_{tk}(V^{\uparrow}) \neq V^{\uparrow}$; thus, $V^{\uparrow} \setminus U = V^{\uparrow} \setminus (U \cap V^{\uparrow}) = V^{\uparrow} \setminus U_{tk}(V^{\uparrow}) \neq \emptyset$.
3. $U_{\epsilon} \subseteq U$. Indeed, for $X \in U_{\epsilon}$ we have $\mathrm{rk}(X) < tk$ by (6), using that $k \leq g(t,d,m)$ and that $\deg(X) \leq \deg(\mathbf{Y}) \leq d$ (see Corollary 29).

Let $X_1 \in V^{\uparrow} \setminus U$ be nonzero, using 2 (as $V^{\uparrow} \setminus U \neq \emptyset, \{0\}$), add to it a homogeneous basis of U, using 1, and extend this independent set into a homogeneous basis $\mathbf{X} \in \mathrm{Form}^k$ of V. Since U satisfies $(V \setminus U) \cap U_{\epsilon} = \emptyset$ by 3, we have $|\operatorname{bias}(X)| \leq q^{-ck}$ for every $X \in V \setminus U$. Therefore, we can apply Lemma 18 on $U \lneq V$, and $\mathbf{X}$ (with $\epsilon = q^{-(c-1)k}$), and deduce that $\mathbf{X}$ is weak ϵ-regular.

Finally, since $\operatorname{Sp}\mathbf{Y} = V = \operatorname{Sp}\mathbf{X}$, we have that $\mathbf{P} \subseteq \mathbb{F}[\mathbf{Y}] = \mathbb{F}[\mathbf{X}]$; moreover, by definition, since $\mathbf{P} \subseteq \mathbb{F}[\mathbf{Y}]$ is a minimal decomposition, so is $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$. In particular, $\mathbf{P} \nsubseteq \mathbb{F}[\mathbf{X}']$, since $\mathbf{X}' \subseteq \mathrm{Form}$ satisfies $\operatorname{Sp}\mathbf{X}' \subsetneq \operatorname{Sp}\mathbf{X}$. Thus, $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$ is a weak ϵ-regular decomposition. The size of this decomposition is k, which, as discussed above, completes the proof. ◀

3 Applications

In this section we obtain several results along the theme that one can deduce strong bounds on the algebraic structure of polynomials, and polynomial maps, based solely on their image. The proofs use the weak regularity lemma.

We begin with the following standard quantification of algebraic structure, which constrains the degrees of the polynomials in the decomposition.

▶ **Definition 19** (t-rank). *Call a polynomial* t-reducible *if it is the product of polynomials of degree at most* t. *Denote, for* $P \in \mathrm{Poly}(\mathbb{F})$ *with* $\deg(P) = d$,

$$\mathrm{rk}_t(P) = \min\left\{r \,\middle|\, \exists R_1, \ldots, R_r \in \mathrm{Poly}_d(\mathbb{F}) \text{ } t\text{-reducible}\colon P = R_1 + \cdots + R_r\right\}.$$

For completeness, set $\mathrm{rk}_0(P) = 0$ *for constant* P.

This definition was used by Karam [21], and is closely related to an earlier definition of Green and Tao [15], $\mathrm{rank}_t(P)$, which expresses P over a finite field as any function of $\mathrm{rank}_t(P)$ polynomials of degree at most t. (Note that $\mathrm{rank}_t(P) \leq O(\deg(P)/t)\,\mathrm{rk}_t(P)$, as each t-reducible polynomial can be expressed in terms of $O(\deg(P)/t)$ polynomials of degree at most t; thus, proving an upper bound on $\mathrm{rk}_t(P)$, as we will do below, would imply a similar upper bound on $\mathrm{rank}_t(P)$.)

▶ Remark 20. If $\mathrm{rk}_t(P)$ is small, for t bounded away from $\deg(P)$, then P is "correlated" with a polynomial of considerably smaller degree (unlike most polynomials of a given degree). More precisely, by a standard Fourier-analytic argument (see e.g. [20]), $P \in \mathrm{Poly}(\mathbb{F}_q)$ has correlation $|\operatorname{bias}(P-Q)| \geq q^{-\mathrm{rank}_t(P)}$ $(\geq q^{-\deg(P)\,\mathrm{rk}_t(P)})$ with some polynomial Q of degree at most t.[23]

One can extend rk_t to polynomial tuples as follows. Define $\mathrm{rk}_t(\mathbf{P})$ for $\mathbf{P} \in \mathrm{Poly}^m$ to be the least number r of t-reducible polynomials $R_1, \ldots, R_r$ such that $\mathbf{P} \subseteq \operatorname{Sp}\{R_1, \ldots, R_r\}$. Note that for $m = 1$, this indeed reduces to Definition 19.

An application of the weak regularity lemma that is quite simple to state involves polynomial maps whose image is a line-free set. Recall that a line in $\mathbb{F}_q^m$ is a set of q elements of the form $\{ta + b \,|\, t \in \mathbb{F}_q\}$ for some $a, b \in \mathbb{F}_q^m$ $(a \neq 0)$. Henceforth, we denote

$$f(m,d) = (2m)^{2^{d(1+o(1))}}.$$

[23] In the context of the Gowers inverse theorem, this is denoted $\left\|\omega^P\right\|_{u^{t+1}} \geq q^{-\mathrm{rank}_t(P)}$.

▶ **Theorem 21.** *Let* $\mathbf{P} \in \mathrm{Poly}_d^m(\mathbb{F})$ *with* $d < \mathrm{char}(\mathbb{F})$*. If* $\mathrm{Im}\,\mathbf{P} \subseteq \mathbb{F}^m$ *does not contain a line then* $\mathrm{rk}_{d/2}(\mathbf{P}) \le f(m,d)$*.*

In particular, the $m = 1$ case of Theorem 21 is about non-surjective polynomials.

Theorem 21 is a special case of a theorem about the degree of curves contained in the image of a polynomial map. We next recall the definition of the *univariate degree* of a polynomial map $\mathbf{P}$, which, roughly, measures the sparsity of the image of P, or how far P is from being surjective – the larger it is, the further P is from being surjective.

▶ **Definition 22** (Univariate degree)**.** *For* $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$*,*

$$\mathrm{udeg}(\mathbf{P}) = \min\{\deg(\mathbf{U}) \mid \mathbf{U}\colon \mathbb{F} \to \mathbb{F}^m \text{ non-constant}\colon \mathrm{Im}\,\mathbf{U} \subseteq \mathrm{Im}\,\mathbf{P}\}.$$

By convention, if $\mathbf{P}$ *is constant then* $\mathrm{udeg}(\mathbf{P}) = \infty$*.*[24]

We include a brief list of properties of univariate degree in Section 3.1 below.

▶ Remark 23. The special case of Definition 22 where $(n, m, \mathrm{udeg}(P)) = (1,1,1)$ is already very interesting. Indeed, note that for any finite field $\mathbb{F}$, a *univariate* polynomial $P\colon \mathbb{F} \to \mathbb{F}$ satisfies $\mathrm{udeg}(P) = 1$ if and only if P is a permutation of $\mathbb{F}$; such polynomials are naturally called *permutation polynomials*, and have attracted much attention (see e.g. [18] and the references within).

Our main result in this section is as follows.

▶ **Theorem 24.** *For every* $\mathbf{P} \in \mathrm{Poly}_d^m(\mathbb{F})$ *with* $d < \mathrm{char}(\mathbb{F})$ *and with* $u = \mathrm{udeg}(\mathbf{P})$*, we have* $\mathrm{rk}_{d/u}(\mathbf{P}) \le f(m,d)$*.*

We highlight two extreme special cases. One is where $\mathrm{udeg}(\mathbf{P})$ is very large, namely $\mathrm{udeg}(\mathbf{P}) > \deg(\mathbf{P})/2$, in which case we directly obtain a small depth-3 (homogeneous) arithmetic formula for $\mathbf{P}$: each component P_i is expressed as a linear combination of (the same) few products of degree-1 polynomials.

▶ **Corollary 25.** *For* $\mathbf{P} \in \mathrm{Poly}_d^m(\mathbb{F})$ *with* $d < \mathrm{char}(\mathbb{F})$*, if* $\mathrm{udeg}(\mathbf{P}) > d/2$ *then* $\mathrm{rk}_1(\mathbf{P}) \le f(m,d)$*.*

The other extreme special case is Theorem 21, which is the special case where $\mathrm{udeg}(\mathbf{P}) > 1$. Indeed, $\mathrm{udeg}(\mathbf{P}) = 1$ means $\mathrm{Im}\,\mathbf{U} \subseteq \mathrm{Im}\,\mathbf{P}$ for some univariate $\mathbf{U}\colon \mathbb{F} \to \mathbb{F}^m$ with $\deg(\mathbf{U}) = 1$, and observe that $\deg(\mathbf{U}) = 1$ if and only if $\mathrm{Im}\,\mathbf{U}$ is a line. Indeed, $\deg(\mathbf{U}) = 1$ means $U_i(t) = ta_i + b_i$ with $a_i, b_i \in \mathbb{F}$, or equivalently, $\mathbf{U}(t) = ta + b$ with $a, b \in \mathbb{F}^m$, where $a \neq 0$. Thus, if $\mathrm{Im}\,\mathbf{P}$ does not contain a line then $\mathrm{udeg}(\mathbf{P}) \ge 2$, so Theorem 24 implies $\mathrm{rk}_{d/2}(\mathbf{P}) \le f(m,d)$ (that is, Theorem 21).

3.1 Properties of univariate degree

Recall the definition of univariate degree for a polynomial map $\mathbf{P}\colon \mathbb{F}^n \to \mathbb{F}^m$:

$$\mathrm{udeg}(\mathbf{P}) = \min\{\deg(\mathbf{U}) \mid \mathbf{U}\colon \mathbb{F} \to \mathbb{F}^m \text{ non-constant}\colon \mathrm{Im}\,\mathbf{U} \subseteq \mathrm{Im}\,\mathbf{P}\}.$$

We have $\mathrm{udeg}(Q^t) \ge \mathrm{udeg}(x^t)$ for any (multivariate) polynomial Q, with equality for any surjective Q (see Proposition 26 (3)), and moreover, it is not hard to show that $\mathrm{udeg}(x^t) = \gcd(t, p-1)$ assuming $\mathbb{F} = \mathbb{F}_p$ prime (as $\mathrm{Im}(x^t) = \mathrm{Im}(x^g)$ with $g = \gcd(t, p-1)$). Thus, for

[24] We require $\mathbf{U}$ to be non-constant as a polynomial. This is in fact equivalent to requiring $\mathbf{U}$ to be non-constant as a function, since we always have $\mathrm{udeg}(\mathbf{P}) < |\mathbb{F}|$ (for non-constant $\mathbf{P}$): if $a \neq b \in \mathrm{Im}\,\mathbf{P}$ then $\mathbf{U}(t) = a + (b-a)t^{|\mathbb{F}|-1}$ has $\mathrm{Im}\,\mathbf{U} \subseteq \mathrm{Im}\,\mathbf{P}$ and is non-constant.

$P = Q^t$ we have $\operatorname{udeg}(P) \geq t$ for any t dividing $p-1$. In this case, $\deg(P)/\operatorname{udeg}(P) \leq \deg(Q)$, with equality if Q is surjective. Now, if we take Q to be a random polynomial $Q\colon \mathbb{F}^n \to \mathbb{F}$, then we have that $\operatorname{rk}_{\deg(Q)-1}(P)$ is unbounded (as $n \to \infty$; see Example 1.4 in [21]). We conclude that one cannot hope to bound $\operatorname{rk}_t(P)$ for $t < \deg(P)/\operatorname{udeg}(P)$.

Below is a brief list of observations regarding the notion of univariate degree, which we include for the sake of completeness. For simplicity, we state them for the case of a single polynomial ($m = 1$). We will use the notation $\overline{P}$ for the unique *reduced* polynomial equal to P as a function, where reduced means that each variable has degree smaller than $|\mathbb{F}|$ (since $x^{|\mathbb{F}|} \equiv x$ as functions over $\mathbb{F}$).[25] Observe that $\operatorname{udeg}(P) = \operatorname{udeg}(\overline{P})$, since $\operatorname{Im}(P) = \operatorname{Im}(\overline{P})$.

▶ **Proposition 26.** *For non-constant $P \in \operatorname{Poly}(\mathbb{F})$, the univariate degree $\operatorname{udeg}(P)$ satisfies the following properties:*

1. $\operatorname{udeg}(P) \leq \min_i \deg_i(P)$ *($\leq \deg(P)$), over all i with $\deg_i(\overline{P}) > 0$.*
2. $\operatorname{udeg}(P) < |\mathbb{F}|$.
3. $\operatorname{udeg}(F(\mathbf{X})) \geq \operatorname{udeg}(F)$*, for any $F \in \operatorname{Poly}$, with equality if $\mathbf{X}$ is surjective.*
4. $\operatorname{udeg}(aP + b) = \operatorname{udeg}(P)$ *for any $a, b \in \mathbb{F}$ with $a \neq 0$.*
5. $\operatorname{udeg}(P^2) \geq 2$ *if $\operatorname{char}(\mathbb{F}) > 2$, with equality if P is surjective.*

Proof. In order:

1. It suffices to assume P is reduced, as $\operatorname{udeg}(P) = \operatorname{udeg}(\overline{P})$ and $\deg_i(\overline{P}) \leq \deg_i(P)$. Let i with $k := \deg_i(P) > 0$, and for concreteness assume $i = 1$. Write P as a univariate polynomial in x_1: $P = \sum_{j=0}^{k} x_1^j C_j(x_2, \ldots, x_n)$. Since C_k is both a nonzero polynomial and reduced (as P is), it is nonzero as a function as well: $C_k(c) \neq 0$ for some $c \in \mathbb{F}^{n-1}$. Thus, the univariate polynomial $U(t) = P(t, c) = \sum_{j=0}^{k} t^j C_j(c)$ is of degree k. Since $\operatorname{Im} U = \{P(t, c) \mid t \in \mathbb{F}\} \subseteq \operatorname{Im} P$ and $k > 0$, we have $\operatorname{udeg}(P) \leq k$.
2. $\operatorname{udeg}(P) = \operatorname{udeg}(\overline{P}) \leq \min_i \deg_i(P) < |\mathbb{F}|$ by the previous item.
3. We may assume F is non-constant. For U witnessing $\operatorname{udeg}(F(\mathbf{X}))$, we have $\operatorname{Im}(U) \subseteq \operatorname{Im}(F(\mathbf{X})) \subseteq \operatorname{Im}(F)$, so $\operatorname{udeg}(F) \leq \deg(U) = \operatorname{udeg}(F(\mathbf{X}))$. Moreover, if $\mathbf{X}$ is surjective then $\operatorname{Im}(F) = \operatorname{Im}(F(\mathbf{X}))$, and generally, $\operatorname{udeg}(P)$ depends only on $\operatorname{Im}(P)$.
4. If a univariate U satisfies $\operatorname{Im}(U) \subseteq \operatorname{Im}(P)$, then $\operatorname{Im}(aU + b) \subseteq \operatorname{Im}(aP + b)$, so $\operatorname{udeg}(aP + b) \leq \deg(aU + b) = \deg(U)$, which shows $\operatorname{udeg}(aP + b) \leq \operatorname{udeg}(P)$ if we choose U that witnesses $\operatorname{udeg}(P)$. The equality follows by symmetry.
5. $\operatorname{udeg}(P^2) \geq \operatorname{udeg}(x^2)$, and we claim that $\operatorname{udeg}(x^2) = 2$. It is not surjective as $\gcd(2, |\mathbb{F}| - 1) = 2$, using that $|\mathbb{F}|$ is odd. Thus, $2 \leq \operatorname{udeg}(x^2) \leq \deg(x^2) = 2$. Moreover, if P is surjective then $\operatorname{udeg}(P^2) = \operatorname{udeg}(x^2)$. ◀

3.2 Polynomial decompositions

First, we will need the following immediate property of weak ϵ-regular tuples. Note that for an arbitrary map $\mathbf{X}\colon \mathbb{F}^n \to \mathbb{F}^r$, if a line $\ell \subseteq \mathbb{F}^r$ in a given direction is chosen uniformly at random, then $\mathbb{E}_\ell \Pr[\mathbf{X} \in \ell] = q^{-(r-1)}$. We show that if $\mathbf{X}$ is weak regular, then lines that are somewhat popular (that is, close to being as popular as a random line) must be fully contained in the image of $\mathbf{X}$.

▶ **Lemma 27.** *Let $\mathbf{X} \subseteq \operatorname{Form}^r(\mathbb{F}_q)$ be weak ϵ-regular. For every line $\ell \subseteq \mathbb{F}^r$ in direction e_1, if $\Pr[\mathbf{X} \in \ell] > \epsilon q^{-(r-1)}$ then $\ell \subseteq \operatorname{Im} \mathbf{X}$.*

[25] That is, $\overline{P} = P \pmod{x_1^{|\mathbb{F}|} - x_1, \ldots, x_n^{|\mathbb{F}|} - x_n}$.

Proof. By the definition of weak ϵ-regularity (recall (1)), for every $y \in \ell$ we have

$$\Pr[\mathbf{X} = y] \geq \Pr[\mathbf{X} \in \ell]/q - \epsilon q^{-r} > 0.$$

Thus $y \in \operatorname{Im} \mathbf{X}$, which completes the proof. ◀

Motivated by Lemma 27, we proceed to prove results about popular lines in the image of polynomial decompositions (equivalently, of polynomials in not necessarily independent variables). These results do not assume weak regularity.

We say that a polynomial decomposition $P = F(\mathbf{X})$ is *irredundant* if there is no polynomial F' with $\operatorname{Supp} F' \subsetneq \operatorname{Supp} F$ such that $F'(\mathbf{X}) = F(\mathbf{X})$.[26] Note that the definition of a weak ϵ-regular decomposition (Definition 7) is independent of the choice of F, so there is no loss of generality in assuming it is irredundant, as we will later do.

First, we show that a decomposition of a polynomial P into forms need not involve terms of higher degree than that of P.

▶ **Lemma 28** (irredundant degree). *For $P \in \mathrm{Poly}$, let $P = F(\mathbf{X})$, with $\mathbf{X} \subseteq \mathrm{Form}$, be an irredundant decomposition. For every monomial M in the support of F, $\deg M(\mathbf{X}) \leq \deg P$.*

Proof. Since every X_i is homogeneous, for every monomial $M \in \operatorname{Supp}(F)$, all the monomials that result from expanding $M(\mathbf{X})$ are of the same degree. Thus, for each k, the degree-k part of $F(\mathbf{X})$ is equal to $\sum M(\mathbf{X})$ with the sum over all monomials $M \in \operatorname{Supp}(F)$ with $\deg M(\mathbf{X}) = k$. Therefore, if F' is obtained from F by removing from its support all monomials M with $\deg M(\mathbf{X}) = k$, for $k > \deg F(\mathbf{X})$, then we still have $F'(\mathbf{X}) = F(\mathbf{X})$. We deduce that if there exists a monomial $M \in \operatorname{Supp}(F)$ with $\deg M(\mathbf{X}) > \deg F(\mathbf{X})$, then $P = F(\mathbf{X})$ is not an irredundant decomposition, completing the proof. ◀

▶ **Corollary 29.** *For any decomposition $P \subseteq \mathbb{F}[\mathbf{X}]$, with $P \in \mathrm{Poly}$ and $\mathbf{X} \subseteq \mathrm{Form}$, if $P \nsubseteq \mathbb{F}[\mathbf{X} \setminus \{X_i\}]$ then $\deg(X_i) \leq \deg(P)$.*

Proof. Without loss of generality, $P = F(\mathbf{X})$ is an irredundant decomposition. We have $\deg_i(F) \geq 1$, since $P \nsubseteq \mathbb{F}[\mathbf{X} \setminus \{X_i\}]$. Apply Lemma 28 on any monomial M in the support of F with $\deg_i(M) \geq 1$. Then $\deg(X_i) \leq \deg M(\mathbf{X}) \leq \deg(P)$, as needed. ◀

We next prove that in the image of any polynomial map $\mathbf{X}$ one can find a popular line that avoids a given low-degree zero set.

▶ **Lemma 30** (Schwartz-Zippel for decompositions). *Let $P \in \mathrm{Poly}_d(\mathbb{F}_q)$, with $d < q$, be nonzero. If $P = F(\mathbf{X})$ with $\mathbf{X} \in \mathrm{Poly}^k$ (and $F \in \mathrm{Poly}$), then there exists $y \in \mathbb{F}_q^k$ such that $F(y) \neq 0$ and*

$$\Pr[F \neq 0] \cdot \Pr[\mathbf{X} = y] \geq (1 - d/q) q^{-k}.$$

In particular, $\Pr[\mathbf{X} = y] \geq (1 - d/q) q^{-k}$. Note that if $\mathbf{X}$ is perfectly equidistributed (e.g., independent variables), we retrieve the usual Schwartz-Zippel lemma: $\Pr[F \neq 0] \geq 1 - d/q$.

Proof. Consider $\{y \in \mathbb{F}^k \mid F(y) \neq 0\}$, the complement of the zero set $\mathrm{Z}(F)$ of F. We have

$$\Pr_x[P(x) \neq 0] = \Pr_x[\mathbf{X} \notin \mathrm{Z}(F)] = \sum_{y \notin \mathrm{Z}(F)} \Pr[\mathbf{X} = y] \leq q^k \Pr[F \neq 0] \max_{y \notin \mathrm{Z}(F)} \Pr[\mathbf{X} = y].$$

On the other hand, as P is nonzero, the Schwartz-Zippel lemma gives $\Pr[P \neq 0] \geq 1 - d/q$. Combining the above, we get $\Pr[F \neq 0] \max_{y \notin \mathrm{Z}(F)} \Pr[\mathbf{X} = y] \geq (1 - d/q) q^{-k}$, as desired. ◀

[26] As usual, this is an equality of polynomials. Recall that $\operatorname{Supp} F$ denotes the support of F, meaning the set of monomials appearing with a nonzero coefficient in F.

We combine the above lemmas to find a popular line that the decomposition nontrivially depends on. Denote by $\deg_1(F)$ the degree of $F \in \mathbb{F}[x_1, \ldots, x_r]$ in its first variable (that is, the largest exponent of x_1 in the monomials in the support of F).

▶ **Corollary 31.** *Let $P \in \mathrm{Poly}_d(\mathbb{F}_q)$ with $d < q$. If $P = F(\mathbf{X})$ with $\mathbf{X} \in \mathrm{Form}^r$ is an irredundant decomposition with $\deg_1 F > 0$, then there exists a line $\ell \subseteq \mathbb{F}_q^r$ in direction e_1 such that $F|_\ell$ is non-constant and $\Pr[\mathbf{X} \in \ell] > (1 - d/q)q^{-(r-1)}$.*

Proof. Put $\mathbb{F} = \mathbb{F}_q$. Write $F(t, \mathbf{y}) = \sum_{i \geq 0} t^i C_i(\mathbf{y})$ with $C_i \in \mathrm{Poly}$, the unique representation of F as a linear combination of the monomials in its support. We claim that the following holds for some $j \geq 1$:

- $C_j(\mathbf{X}')$ is a nonzero polynomial; indeed, otherwise we have $P = F(\mathbf{X}) = F(X_1, \mathbf{X}') = C_0(\mathbf{X}')$; moreover, $\operatorname{Supp} C_0 \subseteq \operatorname{Supp} F$, and $\operatorname{Supp} C_0 \neq \operatorname{Supp} F$ by the assumption $\deg_1(F) > 0$. This contradicts the assumption that $P = F(\mathbf{X})$ is an irredundant decomposition.
- $C_j(\mathbf{X}')$ is of degree smaller than d; indeed, since $\mathbf{X} \subseteq \mathrm{Form}$ and the decomposition $P = F(\mathbf{X})$ is irredundant, we have that $\deg(C_j(\mathbf{X}')) < \deg(X_1^j C_j(\mathbf{X}')) \leq \deg(P) \leq d$, using Lemma 28 in the second inequality.

Apply Lemma 30 on the nonzero $C_j(\mathbf{X}') \in \mathrm{Poly}_{d-1}$ to obtain $y_0 \in \mathbb{F}_q^{r-1}$ such that $C_j(y_0) \neq 0$ and $\Pr[\mathbf{X}' = y_0] > (1 - d/q)q^{-(r-1)}$. Observe that $\deg(F(t, y_0)) \geq j$ (≥ 1), so in particular, $F(t, y_0)$ is a non-constant univariate polynomial. Thus, the restriction $F|_\ell$ of F to the line $\ell = \{(t, y_0) \mid t \in \mathbb{F}\} \subseteq \mathbb{F}^r$ is non-constant. Moreover, observe that $\Pr[\mathbf{X} \in \ell] = \Pr[\mathbf{X}' = y_0]$. This completes the proof. ◀

Finally, we use the following easy (and tight) binomial coefficient bound.

▷ Claim 32. $\binom{n+t}{t} \leq \frac{3}{2} n^t$ for every $n \geq 2$, $t \geq 1$.

Proof. By induction on t, with base case $\binom{n+1}{1} = n+1 \leq \frac{3}{2} n$ as $n \geq 2$. For the induction step, $\binom{n+t+1}{t+1} = \binom{n+t}{t} \frac{n+t+1}{t+1} \leq \frac{3}{2} n^t \cdot n$, using the bound $\frac{n+t+1}{t+1} \leq n$, or equivalently, $1 \leq t(n-1)$. ◁

3.3 Proofs of the applications

Proof of Theorem 24. Put $\mathbb{F} = \mathbb{F}_q$. Assume $\mathbf{P}$ is non-constant, as otherwise $\mathrm{rk}_t(\mathbf{P}) = 0$ by convention and we are done. Put $\epsilon = 1 - d/q$. By Theorem 8 (with $c = 2$, so $q^{-(c-1)k} \leq 1/q \leq \epsilon$), $\mathbf{P}$ has a weak ϵ-regular decomposition $\mathbf{P} = \mathbf{F}(\mathbf{X})$, with $\mathbf{X} \in \mathrm{Form}^r$, of size $r \leq (2m)^{2^{d(1+o(1))}}$. We will prove that $\deg(\mathbf{X}) \leq \deg(\mathbf{P})/\operatorname{udeg}(\mathbf{P})$.

We assume without loss of generality that each decomposition $P_i = F_i(\mathbf{X})$ is irredundant (by removing monomials from the support of F_i if necessary). Since $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$ but $\mathbf{P} \not\subseteq \mathbb{F}[\mathbf{X}']$, there is a component P_j of $\mathbf{P}$ such that $P_j \in \mathbb{F}[\mathbf{X}] \setminus \mathbb{F}[\mathbf{X}']$, and therefore $\deg_1(F_j) > 0$. Apply Corollary 31 on $P_j \in \mathrm{Poly}_d(\mathbb{F})$ to obtain a line $\ell \subseteq \mathbb{F}^r$ in direction e_1 such that $F_j|_\ell$ is non-constant and $\Pr[\mathbf{X} \in \ell] > (1 - d/q)q^{-(r-1)}$. Apply Lemma 27 on ℓ, using the weak ϵ-regularity of $\mathbf{X}$ and our choice of ϵ, to deduce that

$$\ell \subseteq \operatorname{Im} \mathbf{X} \text{ and } \mathbf{F}|_\ell \text{ is non-constant.} \tag{7}$$

Let $\mathbf{U} \colon \mathbb{F} \to \mathbb{F}^m$ be the curve $\mathbf{U} = \mathbf{F}(\mathbf{L})$, where $\mathbf{L} \colon \mathbb{F} \to \mathbb{F}^r$ satisfies $\operatorname{Im}(\mathbf{L}) = \ell$. (Explicitly, if $\ell = \{te_1 + y \mid t \in \mathbb{F}\}$, then $\mathbf{L}(t) = te_1 + y$, so $\mathbf{U}(t) = \mathbf{F}(\mathbf{L}(t)) = \mathbf{F}(t + y_1, y_2, \ldots, y_r)$.) Then $\operatorname{Im} \mathbf{U} = \mathbf{F}(\operatorname{Im}(\mathbf{L})) = \mathbf{F}(\ell)$, which implies that

$$\operatorname{Im} \mathbf{U} \subseteq \operatorname{Im} \mathbf{P} \text{ and } \mathbf{U} \text{ is non-constant,}$$

since $\mathbf{F}(\ell) \subseteq \mathbf{F}(\operatorname{Im} \mathbf{X}) = \operatorname{Im} \mathbf{P}$ by (7). Thus, by definition, $\operatorname{udeg}(\mathbf{P}) \leq \deg(\mathbf{U})$.

Let $i \in [m]$; abbreviate $P = P_i$, $F = F_i$, and $U = U_i$, so that $P = F(\mathbf{X})$ and $U = F(\mathbf{L})$. We claim that the following chain of inequalities holds:

$$\deg(U) \le \deg_1(F) \le \deg(P)/\deg(X_1) = \deg(P)/\deg(\mathbf{X}).$$

The first inequality is immediate from the construction of U. The second inequality follows using Lemma 28: as $P = F(\mathbf{X})$, for every monomial $M = \prod_{j=1}^{r} x_j^{m_j(M)}$ in the support of F, we have $m_1(M)\deg(X_1) \le \deg(M(\mathbf{X})) \le \deg(P)$; since $\deg_1(F) = \max_M m_1(M)$ over $M \in \operatorname{Supp} F$, we indeed obtain that $\deg_1(F)\deg(X_1) \le \deg(P)$. The final equality follows from the construction of $\mathbf{X}$. We obtain the following degree bound for $\mathbf{X}$, using i satisfying $\deg(U_i) = \deg(\mathbf{U})$:

$$\deg(\mathbf{X}) \le \deg(P)/\deg(U) \le \deg(\mathbf{P})/\deg(\mathbf{U}) \le \deg(\mathbf{P})/\operatorname{udeg}(\mathbf{P}).$$

Finally, as every component P_i of $\mathbf{P}$ satisfies $P_i = F_i(\mathbf{X})$ with $F_i \colon \mathbb{F}^r \to \mathbb{F}$ and $\deg(F_i) \le \deg(P_i) \le d$, by Lemma 28 (and as $\deg(X_i) \ge 1$), we deduce that $P_1, \ldots, P_m$ lie in the span of all products of at most d members of $\mathbf{X}$. We are therefore done (using Claim 32, assuming $r \ge 2$ without loss of generality):

$$\operatorname{rk}_{d/u}(\mathbf{P}) \le \binom{r+d}{d} \le \frac{3}{2} r^d \le (2m)^{2^{d(1+o(1))}}. \qquad \blacktriangleleft$$

We are now ready to deduce Theorems 3 and 4.

Proof of Theorem 3. Let P be an n-variate polynomial with $\deg(P) \le d$ and $\operatorname{udeg}(P) \ge u$. Assume P is non-constant, as otherwise there is nothing to prove. Put $t = \lfloor d/u \rfloor$ (≥ 1). Then $P = R_1 + \cdots + R_{\operatorname{rk}_t(P)}$ where each $R_i \in \operatorname{Poly}_d$ is a product of polynomials of degree at most t, that is, $R_i = \prod_{j \le d_i} F_{i,j}$ with $\deg(F_{i,j}) \le t$ and $d_i \le d$, by Definition 19. Since, trivially, $L(F_{i,j}) \le O(\binom{n+t}{t}) \le O(n^t)$ (Claim 32, assuming $n \ge 2$), we deduce that

$$L(P) \le O\Big(\sum_{i,j} L(F_{i,j})\Big) \le O(d \cdot \operatorname{rk}_t(P) \cdot n^t) \le n^{t+o(1)},$$

where the last inequality uses Theorem 24, and the statement's assumption on d, to bound

$$\operatorname{rk}_t(P) \le f(1,d) = 2^{2^{d(1+o(1))}} \le 2^{(\log_2(n))^{\frac{1}{2}+o(1)}} = n^{(\log_2(n))^{-(\frac{1}{2}+o(1))}} \le n^{o(1)}. \qquad \blacktriangleleft$$

Proof of Theorem 4. Let $P \subseteq \mathbb{F}[\mathbf{P}]$ with $\operatorname{udeg}(\mathbf{P}) = u$. As in the proof of Theorem 24, using the weak regularity lemma we obtain a decomposition $\mathbf{P} \subseteq \mathbb{F}[\mathbf{X}]$ with $\mathbf{X} \in \operatorname{Form}^s$ of size $s \le (2m)^{2^{d(1+o(1))}}$ such that $\deg(\mathbf{X}) \le d/u$. Observe that $P \in \mathbb{F}[\mathbf{P}] \subseteq \mathbb{F}[\mathbf{X}]$. Using Lemma 28 with an irredundant representation (by removing monomials if necessary), it follows that we can write

$$P = \sum_{i=1}^{r} c_i \prod_j Q_{i,j} \tag{8}$$

with $Q_{i,j} \in \mathbf{X}$, $c_i \in \mathbb{F}$, $\deg(\prod_j Q_{i,j}) \le d$, and $r \le \sum_{i=0}^{d} s^i \le (2m)^{2^{d(1+o(1))}}$. Note that $\deg(Q_{i,j}) \le \deg(\mathbf{X}) \le d/u$.

We claim that we can guarantee that the number of multiplicands in each product $\prod_j Q_{i,j}$ is at most $2u$. Note that we can express each product as a product of homogeneous polynomials $\prod_{j=1}^{a_i} Q'_{i,j}$ all of whose degrees, except at most one of them, satisfy $\frac{1}{2}(d/u) < \deg(Q'_{i,j}) \le d/u$ (note that $Q'_{i,j}$ are not necessarily members of $\mathbf{X}$); indeed, this follows by repeatedly

using the fact that if $\deg(Q_{i,j_1}), \deg(Q_{i,j_2}) \leq \frac{1}{2}(d/u)$ then $\deg(Q_{i,j_1}Q_{i,j_2}) \leq d/u$. So, $d \geq \sum_{j=1}^{a_i} \deg(Q'_{i,j}) > (a_i - 1)\frac{1}{2}(d/u)$, hence $a_i \leq 2u$ as claimed. Thus, (8) gives a $\sum^{[r]} \prod^{[2u]} \sum \prod^{[d/u]}$-formula for P, as needed.

Finally, if P is homogeneous then, since every member of $\mathbf{X}$ is homogeneous, without loss of generality the products in the formula satisfy $\deg(\prod_j Q_{i,j}) = \deg(P)$ for every i, and therefore the formula is homogeneous, as desired. ◀

References

1 Tigran Ananyan and Melvin Hochster. Small subalgebras of polynomial rings and Stillman's Conjecture. *Journal of the American Mathematical Society*, 33(1):291–309, 2020. doi:10.1090/jams/932.

2 Paul Beame, Shayan Oveis Gharan, and Xin Yang. On the bias of Reed–Muller codes over odd prime fields. *SIAM Journal on Discrete Mathematics*, 34(2):1232–1247, 2020. doi:10.1137/18M1215104.

3 Abhishek Bhowmick and Shachar Lovett. Bias vs structure of polynomials in large fields, and applications in information theory. *IEEE Transactions on Information Theory*, 69(2):963–977, 2023. doi:10.1109/TIT.2022.3214372.

4 Andrej Bogdanov and Emanuele Viola. Pseudorandom bits for polynomials. *SIAM Journal on Computing*, 39(6):2464–2486, 2010. doi:10.1137/070712109.

5 Alex Cohen and Guy Moshkovitz. Partition and analytic rank are equivalent over large fields. *Duke Mathematical Journal*, 172(12):2433–2470, 2023. doi:10.1215/00127094-2022-0086.

6 Zeev Dvir, János Kollár, and Shachar Lovett. Variety evasive sets. *Computational Complexity*, 23(4):509–529, 2014. doi:10.1007/s00037-013-0073-9.

7 Zeev Dvir and Shachar Lovett. Subspace evasive sets. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, STOC 2012, pages 351–358. ACM, 2012. doi:10.1145/2213977.2214010.

8 Christian Elsholtz, Zach Hunter, Laura Proske, and Lisa Sauermann. Improving Behrend's construction: Sets without arithmetic progressions in integers and over finite fields, 2024. arXiv:2406.12290.

9 Daniel Erman, Steven V. Sam, and Andrew Snowden. Cubics in 10 variables vs. cubics in 1000 variables: uniformity phenomena for bounded degree polynomials. *Bulletin of the American Mathematical Society*, 56(1):87–114, 2019. doi:10.1090/bull/1652.

10 Jacob Fox and Huy Tuan Pham. Popular progression differences in vector spaces II. *Discrete Analysis*, 16:39 pp., 2019. doi:10.19086/da.11002.

11 Alan Frieze and Ravi Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999. doi:10.1007/s004930050052.

12 Abhibhav Garg, Rafael Oliveira, and Akash Kumar Sengupta. Rank bounds and PIT for depth-4 circuits with top fan-in 3 and constant bottom fan-in via a non-linear Edelstein–Kelly theorem. In *Proceedings of the 66th IEEE Annual Symposium on Foundations of Computer Science*, FOCS 2025, pages 2143–2169. IEEE, 2025. doi:10.1109/FOCS63196.2025.00114.

13 W. T. Gowers and Julia Wolf. Linear forms and higher-degree uniformity for functions on $\mathbb{F}_p^n$. *Geometric and Functional Analysis*, 21(1):36–69, 2011. doi:10.1007/s00039-010-0106-3.

14 Ben Green. Montreal lecture notes on quadratic Fourier analysis. In Andrew Granville, Melvyn B. Nathanson, and József Solymosi, editors, *Additive Combinatorics*, volume 43 of *CRM Proceedings and Lecture Notes*, pages 69–102. American Mathematical Society, 2007. arXiv:math/0604089.

15 Ben Green and Terence Tao. The distribution of polynomials over finite fields, with applications to the Gowers norms. *Contributions to Discrete Mathematics*, 4(2):1–36, 2009. doi:10.55016/ojs/cdm.v4i2.62086.

16 Jaime Gutiérrez, Rosario Rubio, and David Sevilla. On multivariate rational function decomposition. *Journal of Symbolic Computation*, 33(5):545–562, 2002. doi:10.1006/jsco.2000.0529.

17 Hamed Hatami and Shachar Lovett. Higher-order Fourier analysis of $\mathbb{F}_p^n$ and the complexity of systems of linear forms. *Geometric and Functional Analysis*, 21(6):1331–1357, 2011. doi:10.1007/s00039-011-0141-8.

18 Xiang-dong Hou. Permutation polynomials over finite fields—a survey of recent advances. *Finite Fields and Their Applications*, 32:82–119, 2015. doi:10.1016/j.ffa.2014.10.001.

19 Pavel Hrubeš and Amir Yehudayoff. Arithmetic complexity in ring extensions. *Theory of Computing*, 7(8):119–129, 2011. doi:10.4086/toc.2011.v007a008.

20 Oliver Janzer. Polynomial bound for the partition rank vs the analytic rank of tensors. *Discrete Analysis*, 7:18 pp., 2020. doi:10.19086/da.12935.

21 Thomas Karam. Ranges of polynomials control degree ranks of Green and Tao over finite prime fields. *Proceedings of the Edinburgh Mathematical Society*, pages 1–28, 2026. doi:10.1017/S0013091526101382.

22 Tali Kaufman and Shachar Lovett. Worst case to average case reductions for polynomials. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2008, pages 166–175. IEEE, 2008. doi:10.1109/FOCS.2008.17.

23 Tali Kaufman, Shachar Lovett, and Ely Porat. Weight distribution and list-decoding size of Reed–Muller codes. *IEEE Transactions on Information Theory*, 58(5):2689–2696, 2012. doi:10.1109/TIT.2012.2184841.

24 Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC 2014, pages 146–153. ACM, 2014. doi:10.1145/2591796.2591847.

25 David Kazhdan, Amichai Lampert, and Alexander Polishchuk. Schmidt rank and singularities. *Ukrainian Mathematical Journal*, 75(9):1420–1442, 2024. doi:10.1007/s11253-024-02270-6.

26 Mrinal Kumar and Shubhangi Saraf. The limits of depth reduction for arithmetic formulas: it's all about the top fan-in. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC 2014, pages 136–145. ACM, 2014. doi:10.1145/2591796.2591827.

27 Amichai Lampert. Small ideals in polynomial rings and applications, 2023. arXiv:2309.16847.

28 Amichai Lampert and Tamar Ziegler. On rank in algebraic closure. *Selecta Mathematica. New Series*, 30(1), 2024. doi:10.1007/s00029-023-00903-5.

29 Amichai Lampert and Tamar Ziegler. Relative rank and regularization. *Forum of Mathematics, Sigma*, 12, 2024. doi:10.1017/fms.2024.15.

30 Shachar Lovett. Computing polynomials with few multiplications. *Theory of Computing*, 7(13):185–188, 2011. doi:10.4086/toc.2011.v007a013.

31 Luka Milićević. Polynomial bound for partition rank in terms of analytic rank. *Geometric and Functional Analysis*, 29(5):1503–1530, 2019. doi:10.1007/s00039-019-00505-4.

32 Guy Moshkovitz and Daniel G. Zhu. Quasi-linear relation between partition and analytic rank, 2022. arXiv:2211.05780.

33 Eric Naslund. The partition rank of a tensor and k-right corners in $\mathbb{F}_q^n$. *Journal of Combinatorial Theory, Series A*, 174, 2020. doi:10.1016/j.jcta.2019.105190.

34 Rafael Oliveira and Akash Kumar Sengupta. Strong algebras and radical Sylvester–Gallai configurations. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 95–105. ACM, 2024. doi:10.1145/3618260.3649617.

35 Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(7):135–177, 2010. doi:10.4086/toc.2010.v006a007.

36 Wolfgang M. Schmidt. The density of integer points on homogeneous varieties. *Acta Mathematica*, 154:243–296, 1985. doi:10.1007/BF02392473.

37 Hans-Jörg Stoß. Lower bounds for the complexity of polynomials. *Theoretical Computer Science*, 64(1):15–23, 1989. doi:10.1016/0304-3975(89)90094-7.

38 Terence Tao and Tamar Ziegler. The inverse conjecture for the Gowers norm over finite fields in low characteristic. *Annals of Combinatorics*, 16(1):121–188, 2012. doi:10.1007/s00026-011-0124-3.

39 Joachim von zur Gathen and Konstantin Ziegler. Survey on counting special types of polynomials. In Jaime Gutiérrez, Josef Schicho, and Martin Weimann, editors, *Computer Algebra and Polynomials*, volume 8942 of *Lecture Notes in Computer Science*, pages 50–75. Springer Cham, 2015. doi:10.1007/978-3-319-15081-9_3.

Derandomised Tensor Product Gap Amplification for Quantum Hamiltonians

Thiago Bergamaschi
University of California, Berkeley, CA, USA

Tony Metger
ETH Zurich, Zurich, Switzerland
Simons Institute for the Theory of Computing, Berkeley, CA, USA

Thomas Vidick
Weizmann Institute of Science, Rehovot, Israel
EPFL, Lausanne, Switzerland

Tina Zhang
Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

The quantum PCP conjecture asks whether it is QMA-hard to distinguish between high- and low-energy Hamiltonians even when the gap between "high" and "low" energy is large (constant). A natural proof strategy is gap amplification: start from the fact that high- and low-energy Hamiltonians are hard to distinguish if the gap is small (inverse polynomial) [21] and *amplify* the Hamiltonians to increase the energy gap while preserving hardness. Such a gap amplification procedure is at the heart of Dinur's proof of the *classical* PCP theorem [17]. In this work, following Dinur's model, we introduce a new quantum gap amplification procedure for Hamiltonians which uses random walks on expander graphs to derandomise (subsample the terms of) the *tensor product amplification* of a Hamiltonian. Curiously, our analysis relies on a new technique inspired by quantum de Finetti theorems, which have previously been used to rule out certain approaches to the quantum PCP conjecture [11].

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases quantum PCP conjecture, gap amplification, local Hamiltonians, tensor product amplification, expander random walks, quantum de Finetti theorems

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.15

Related Version *Full Version*: `https://arxiv.org/abs/2510.01333` [10]

Funding *Tony Metger*: acknowledges support from SNSF Grant No. 20QU-1_225171 and NCCR SwissMAP.
Thomas Vidick: supported by AFOSR Grant No. FA9550-22-1-0391 and ERC Consolidator Grant VerNisQDevS (101086733).
Tina Zhang: was partially supported by NSF CAREER grant number 2339948.

Acknowledgements We thank Anurag Anshu for useful discussions and suggesting the DL-based construction of a streaming quantum PCP presented in Appendix A. We thank Dorit Aharonov, Niko Breuckmann, Sevag Gharibian, Louis Golowich, Anand Natarajan, Quynh The Nguyen, Nikhil Srivastava, and John Wright for helpful discussions. Part of this work was completed while the authors were visiting the Simons Institute for the Theory of Computing.

1 Introduction

The classical PCP theorem [8] is one of the landmark achievements of classical complexity theory, and the quantum PCP conjecture [2] is one of the major open problems in quantum complexity theory. The classical PCP theorem was proven for the first time using algebraic techniques, and later reproven using elementary combinatorial techniques in the celebrated 2007 paper of Dinur [17]. Both approaches have, so far, resisted quantisation.



© Thiago Bergamaschi, Tony Metger, Thomas Vidick, and Tina Zhang;
licensed under Creative Commons License CC-BY 4.0



The algebraic proof of the PCP theorem seems difficult to quantise because of its reliance on the properties of polynomial codes, which have no clear quantum analogue [2]. The combinatorial proof, which pares the approach down to essentials, has seemed more promising as a guide for quantum PCP. Very broadly speaking, Dinur's proof of the PCP theorem consists of the iterated application of two steps:

▶ **Definition 1.1** (Dinur's template for proving the PCP theorem)**.**
1. *Gap amplification, in which the promise gap of a constraint satisfaction problem family is amplified at the cost of blowing up its alphabet size,*
2. *Alphabet reduction, in which the locality of the constraint satisfaction problem family is reduced back to a constant, and the gap is shown not to shrink too much in the process.*

Applying these two steps iteratively to an NP-complete constraint satisfaction problem (CSP) with inverse polynomial promise gap yields another NP-complete CSP with constant promise gap. Thus, given a purported witness for the amplified CSP, a verifier can check that the witness satisfies the amplified CSP (with constant probability of error) by randomly selecting and checking constantly many clauses, which only requires a logarithmic number of random bits. (The number of bits that the verifier uses in this process is referred to as the *randomness complexity* of the verifier.)

Dinur's classical gap amplification procedure (step (i)) starts from a certain "naïve" gap amplification procedure, which amplifies the promise gap effectively but blows up the randomness complexity of the verifier. To control the randomness complexity of the verifier, Dinur uses expander random walks to derandomise this "naïve" procedure and achieve amplification without paying a steep cost in randomness complexity.

In this work, following Dinur's model, we introduce a new quantum gap amplification procedure for Hamiltonians which uses random walks on expander graphs to derandomise (subsample the terms of) the *tensor product amplification* of a Hamiltonian. The tensor product amplification of a Hamiltonian H (normalized such that $0 \leq H \leq I$) is simply $I - (I - H)^{\otimes t}$. It can be easily shown that tensoring H with itself in this way amplifies the promise gap, but this process blows up the number of terms in the amplified Hamiltonian exponentially in t (and consequently also blows up the amount of randomness required to sample a term at random). By contrast, our derandomised version of tensor product amplification achieves gap amplification without increasing the number of terms in the original Hamiltonian by too much. The following is a formal statement of our gap amplification theorem:

▶ **Theorem 1.2** (Gap amplification)**.** *Let H be a k–local Hamiltonian with m terms acting on n qubits, satisfying the conditions in Definition 1.6. Then, its* derandomised $2t$–fold tensor product amplification $H^{(2t)}$ *(formally defined in Definition 1.9) satisfies:*
1. *$H^{(2t)}$ is $(2t \cdot k)$–local, acts on nt qubits and has $d^{2t} \cdot m$ terms, where d is the degree of some fixed expander graph family.*
2. ***Completeness.*** *The lowest eigenvalue of $H^{(2t)}$ is bounded above by*

$$\lambda_{\min}(H^{(2t)}) \leq 2t \times \lambda_{\min}(H) \tag{1.1}$$

3. ***Soundness.*** *The eigenvalue of $H^{(2t)}$ is bounded below by*

$$\lambda_{\min}(H^{(2t)}) \geq \min\left[\Theta\left(\frac{\log t}{t}\right), \Theta\left(\sqrt{\frac{t}{\log t}} \times \lambda_{\min}(H)\right)\right], \tag{1.2}$$

where the Θ notation obscures constants dependent on the choice of expander graph family and the original Hamiltonian H.

To understand Theorem 1.2, one should consider the case that t is a constant and the completeness parameter of the original Hamiltonian family is exponentially small: in this context, Theorem 1.2 says that we can amplify the additive promise gap of a Hamiltonian family by a constant multiplicative factor, at the cost of increasing both the locality and the number of terms by a constant multiplicative factor. These parameters are comparable, except for one important difference, to those which Dinur achieves with her gap amplification procedure (see [17, Section 6]). We elaborate on this difference more thoroughly in Section 1.1.

In order to prove Theorem 1.2, we devise a new technique inspired by *quantum de Finetti theorems* [27, 28], which we believe could be of independent interest. Curiously, while quantum de Finetti theorems have previously been used to rule out approaches to the quantum PCP conjecture [11], here we use methods inspired by these statements to argue the soundness of our amplification scheme. We give more context and details about our application of de Finetti theorems in the related work Section 1.1 below, and in our technical overview (Section 1.3). We dedicate Section 1.2 to a discussion on potential applications of Theorem 1.2.

1.1 Related work

Quantum gap amplification

In 2008, Aharonov, Arad, Landau and Vazirani proposed a quantum gap amplification procedure that was also based on Dinur's gap amplification [1]. The key ingredient in their work was the *detectability lemma*, which in some sense states that "if the ground state energy of some (non-commuting) Hamiltonian H is at least E, then sequentially measuring all the terms of H on the ground state should detect at least $const \cdot E$ violations (with constant probability)". Although their original motivation lay in the pursuit of the quantum PCP (QPCP) conjecture [2], their work subsequently found many applications to quantum many-body physics and quantum information theory, including area laws [7], k-designs [13], and Gibbs samplers [20].

There are two key differences between the parameters of Dinur's classical amplification procedure and those of the quantum gap amplification procedure proposed by [1]. The first is that (unlike Theorem 1.2) the gap amplification procedure of [1] requires the Hamiltonian family it amplifies to have constant locality to start with.[1] Since [1]'s gap amplification procedure also increases the locality of the Hamiltonian which it amplifies (where the increase in locality is proportional to the increase in promise gap which it achieves, like in Theorem 1.2), this means that [1]'s gap amplification cannot be iterated by itself (without composing it with a locality reduction procedure).

We resolve this issue in this work. Because our gap amplification procedure (Theorem 1.2) can be iterated by itself (albeit only a logarithmic number of times), we can achieve applications that, as far as we know, cannot be achieved with the tools developed by [1]. For example, we are able to get a *locality-gap tradeoff theorem* for QMA-complete Hamiltonians:

▶ **Theorem 1.3** (Locality-gap tradeoff; informal). *Let $\mathcal{H}$ be a k-local Hamiltonian family that is* QMA*-complete with promise gap ϵ.*[2] *Then, given a deterministic construction of $\mathcal{H}$, there is a deterministic construction of a family of Hamiltonians $\mathcal{H}'$ such that $\mathcal{H}'$ has locality $k \cdot 1/\epsilon$ and is* QMA*-complete with some universal constant promise gap c.*

[1] Newer versions of the detectability lemma exist which do not require constant locality of the Hamiltonian to which they are applied (see e.g. [3]); however, we do not know of any detectability lemma that both has this property and is sufficiently strong to complete the proof from [1].

[2] The members of $\mathcal{H}$ also need to satisfy certain technical conditions, such as being a sum of polynomially many projective terms, which we are suppressing in this informal statement. For a formal version, see the full version [10].

In particular, Theorem 1.3 could be applied (for example) to boost the promise gap of a "weak QPCP". Recent progress [4] suggests potential avenues for achieving a so-called *weak quantum PCP*, namely, a QPCP with polylog(n) locality and $1/\text{polylog}(n)$ promise gap instead of constant locality and promise gap. Applying Theorem 1.3 to such an object would immediately yield a PCP with polylog(n) locality and *constant* promise gap.

While our gap amplification procedure can be iterated by itself, like Dinur's, there remains a second important difference between the two. Dinur's gap amplification procedure is *non-locality-increasing*, in the sense that one round of amplification blows up the *alphabet size* of the constraint satisfaction problem but not the number of variables that each constraint involves. All attempted quantisations of Dinur's gap amplification, including ours and [1], are locality-increasing, which is a significant obstacle to applying the kinds of PCP composition theorems that make alphabet reduction possible. [1] gap amplification (and ours) is in some sense a quantum version of classical derandomised *sequential repetition*, and in the classical setting derandomised *parallel repetition* seems necessary to make Dinur's template from Definition 1.1 work: see [19, 18] for a discussion of this issue. As such, producing a quantisation of Dinur's gap amplification procedure which reproduces all of its important characteristics remains an open problem.

Achieving derandomised sequential (locality-increasing) amplification is already non-trivial in the quantum setting, however, and we view one of our primary contributions as that of analysing a natural but so far unstudied locality-increasing derandomised quantum amplification procedure using techniques that have not previously been applied in this context. As far as we know, our techniques are the first to yield iterability. Moreover, the techniques which [1] used to analyse their own amplification procedure have since found widespread application outside of their original context, and we hope the introduction of a new approach for analysing low-randomness locality-increasing quantum gap amplification might be found to have similar utility. We may have particular grounds for hope because the work of [1] can in some ways be viewed as a *black-box reduction* from the quantum problem to the classical one, while our approach more directly interprets certain classical gap amplification strategies in a quantum setting, and as such may more clearly illuminate the differences between the classical and the quantum problems.

Quantum de Finetti theorems

A defining property of quantum entanglement is that it is *monogamous*. That is, a quantum system cannot be very entangled with a large number of other systems. Quantum *de Finetti theorems* attempt to quantify this intuition, and more broadly offer a versatile tool to understand quantum correlations, with applications to quantum information theory [27], cryptography [15, 28], algorithms [11, 9] and complexity theory [22, 12]. Roughly speaking, quantum de Finetti theorems capture the fact that a random k qubit marginal of any $n \gg k$ qubit state should be close to a convex combination of product states, and therefore unentangled.

Most relevant to us are the results of [11], who showed that local Hamiltonians defined on dense (degree d) constraint graphs admit product state approximations, in that there exists a product state whose energy is an additive inverse poly(d) close to the ground state energy.[3] In some sense, their results offer a concrete route to *disproving* the QPCP conjecture: if there

[3] Note that the existence of a product state approximation implies that determining the ground state energy up to said approximation error is in NP, since the description of the product state is a succinct classical witness.

existed a procedure which increased the degree of any local Hamiltonian, without decreasing its ground-state energy (or increasing locality), then deciding the ground state energy of the resulting Hamiltonian up to a constant would be in NP [11, Corollary 11].

Here, we issue two remarks. First, the conclusions of [11] become exponentially weaker if the degree and the locality of the Hamiltonian are allowed to increase simultaneously during amplification [6], which allows us to evade their no-go. Second, despite these "negative" results, de Finetti techniques will nevertheless play a central role in our approach in understanding when quantum correlations can adversarially decrease the ground state energy of our amplified Hamiltonians. A comprehensive discussion is presented in the full version [10].

1.2 Discussion and applications

In this section we discuss a number of potential applications of our gap amplification procedure.

Streaming quantum PCP

Theorem 1.2 can be applied iteratively $1/\log(n)$ times to achieve what we call a "streaming quantum PCP":

▶ **Theorem 1.4** (Streaming quantum PCP)**.** *There exists a deterministic construction of a family $\{H_n\}$ of Hamiltonians on n qubits and an explicit constant c, such that each Hamiltonian H_n is a sum of $\mathrm{poly}(n)$ many terms, each summand is an $O(n)$-fold tensor product of $O(1)$-local projections, and it is QMA-complete to decide whether the ground state energy of $\{H_n\}$ is $\leq \mathsf{negl}(n)$ or $\geq c$.*

Theorem 1.4 can be interpreted as a quantum version of what one would get in the classical setting by iterating Dinur's gap amplification procedure from Definition 1.1 *without* locality or alphabet reduction reduction. Theorem 1.4 can also be viewed as a very preliminary version of quantum PCP. Indeed, the family of Hamiltonians guaranteed by Theorem 1.4 satisfies the conditions of the quantum PCP conjecture – including the condition that there are polynomially many terms in each member Hamiltonian – except for the fact that the member Hamiltonians do not have constant locality, which is of course a substantial omission.

Nonetheless, the terms in each H_n are individually easy to verify: since each term is the $O(n)$-fold tensor product of $O(1)$-local projections, each term can be measured using a constant-depth quantum circuit (and classical post-processing), or by a quantum algorithm that reads a constant number of qubits at a time. There are also only polynomially many such terms in each H_n, which implies that the energy of H_n can be measured on a witness state by a verifier who samples a random term and measures it, and in the process uses only $O(\log n)$ many classical random bits and very limited quantum resources. (We call Theorem 1.4 a "streaming quantum PCP" for this reason.) It also implies that time evolution under H_n can be efficiently simulated (see the full version [10]). Ease of verification is one of the chief motivations for quantum PCP, and ease of simulation is one of the chief motivations for the *sparse* quantum PCP conjecture (see below). As such, Theorem 1.4 can be viewed as progress towards both objectives.

▶ Remark 1.5. There is an alternative way to achieve Theorem 1.4 which does not use our Theorem 1.2 and instead uses improvements to the techniques of [1] due to [3] to prove Theorem 1.4 using "one-shot" (non-iterative) amplification. This result, due to Anurag Anshu, is presented in Appendix A. The main drawback of this alternative approach is that it

does not allow the kind of more "fine-grained" amplification that our Theorem 1.2 provides, where the tunable parameter t simultaneously governs the gap amplification and the locality increase of the input Hamiltonian.

Locality-gap tradeoffs for local Hamiltonians

As mentioned above in Theorem 1.3, Theorem 1.2 can be iteratively applied in order to produce a "locality-gap tradeoff" for local Hamiltonians. More explicitly, Theorem 1.2 implies that, if there is a deterministic construction of a QMA-complete family of Hamiltonians with locality ℓ and promise gap ϵ, then (under mild conditions) there is a deterministic construction of a QMA-complete family of Hamiltonians with locality ℓ/ϵ and gap which is some universal constant c.

In particular, we could use Theorem 1.2 to boost a "weak quantum PCP" statement (cf. [4, Open Problems]) – that is, a family of Hamiltonians with $\operatorname{poly}\log n$ locality and inverse $\operatorname{poly}\log n$ promise gap – into a family of Hamiltonians that has $\operatorname{poly}\log n$ locality and *constant* promise gap.

Quantum games PCP

The quantum games PCP conjecture states that there exists an MIP^* protocol (i.e., a protocol between a classical verifier and multiple efficient, non-communicating but potentially entangled provers) with $\operatorname{poly}\log n$ sized questions and answers, constant completeness-soundness gap, and efficient provers that can decide all of QMA [24]. The motivation for this question lies in the connection between PCPs and succinct MIP (classical multi-prover) protocols with constant gap in the classical world [2]. The connection between quantum PCP and succinct MIP^* protocols is less immediate, although, like in the classical case, quantum games PCP is a necessary consequence of Hamiltonian quantum PCP (but this is considerably harder to show than it is classically). A proof of the quantum games PCP conjecture was claimed by [25], but it has since been established that the proof had errors, and as of now the conjecture remains open.

We believe Theorem 1.4 implies a deterministic construction of an MIP^* protocol with efficient provers, constant completeness-soundness gap and $\log n$ sized *questions* ($\operatorname{poly} n$ sized answers) that decides all of QMA. Such a protocol was not known prior to our work, and it would follow by designing a succinct Pauli braiding test [16, 23] that supports measurements of the terms in our amplified Hamiltonian, which could be made into tensor products of 5-local Clifford projections [14]. Unfortunately, our result does not appear to help with achieving succinct answers, and we leave as an open question whether it is possible to build on these results to prove the quantum games PCP conjecture.

Sparse quantum PCP

The sparse quantum PCP conjecture is similar to the original quantum PCP conjecture except that, instead of requiring that the Hamiltonian is a sum of local terms, we require that the Hamiltonian is expressible as a sparse matrix with efficiently computable entries.

This conjecture is of interest primarily for two reasons:

1. It is, like the NLTS theorem [5], a necessary consequence of quantum PCP, and
2. Sparse Hamiltonians are simulable given only oracle access to their entries, which allows their energy to be measured even in the absence of any term decomposition.

This entails that their ground state energies are easy to verify, and as such a sparse quantum PCP can be viewed as a meaningful stepping stone towards the kind of easy verification that full quantum PCP requires.

One might hope that Theorem 1.4 could be useful for sparse QPCP, because our Hamiltonian is a sum of polynomially many relatively structured terms (see below, Definition 1.9). Unfortunately, existing black-box approaches to matrix sparsification appear to be either too randomness-expensive or too computationally inefficient to yield the sparse QPCP conjecture when applied to the terms of our Hamiltonians. We leave it as an interesting open question whether progress can be made in this direction by exploiting the particular tensor-product structure of our Hamiltonians.

1.3 Technical overview

1.3.1 Construction and notation

In this subsection we more formally present our construction of a quantum gap amplification procedure. Our starting point is a Hamiltonian H on n qubits, which can be partitioned into a convex combination of g commuting layers (or colors).

▶ **Definition 1.6** (Layered Hamiltonians). *H is said to be a* layered Hamiltonian *if it admits a decomposition*

$$H = \sum_{\chi\in[g]} w_\chi \cdot H_\chi \,, \qquad \text{where each layer} \qquad H_\chi = \mathop{\mathbb{E}}_{i\in[m_\chi]} \Pi_i^\chi \tag{1.3}$$

is the expectation over m_χ commuting projections, and the weights $\{w_\chi\}_{\chi\in[g]}$ are positive and $\sum_\chi w_\chi = 1$. To quantify imbalance between the layers, we introduce the parameter

$$\omega_{\min} \equiv (\min_\chi w_\chi)^{-1}. \tag{1.4}$$

As we discuss in the full version [10], there are QMA-complete local Hamiltonians which admit such a decomposition with a constant number g of layers (and constant $\omega_{\min}$), so starting from such a decomposition is essentially without loss of generality. Our construction will amplify the individual commuting layers separately. To do so, for each color we impose an expander graph structure on the clauses indexed by $[m_\chi]$.[4]

▶ **Definition 1.7** (Paths of length t in G_χ). *Let G_χ be a d-regular λ-spectral expander graph on m_χ vertices (Definition 2.8). Define a function family $\mathcal{F}_\chi : \{f : [t] \to [m_\chi]\}$ such that there is a one-to-one*[5] *correspondence between member functions $f \in \mathcal{F}_\chi$ and paths of length t in G_χ.*

The amplified Hamiltonian will act on t (tensor) copies of the original n qubit system. For each color χ and path $f \in \mathcal{F}_\chi$, we associate to f a clause Π_f^χ in the amplified Hamiltonian, in the form of a projection onto the intersection of the t clauses on the path. In other words, the clauses in the amplified Hamiltonian are satisfied if all (the "AND") of the clauses on the path f are satisfied:

$$\Pi_f^\chi \equiv \mathbb{I} - \bigotimes_{j\in[t]} \left(\mathbb{I} - \Pi_{f(j)}^\chi\right) \tag{1.5}$$

[4] So long as m_χ is above some constant m_0, there exists a determinstic construction of such a graph, see Lemma 2.9 [26]. The case $m_\chi \le m_0$ is addressed by trivially repeating the clauses in H_χ.

[5] $\forall f \in \mathcal{F}_\chi$, the tuple $(f(1), \ldots, f(t)) \in [m_\chi]^t$ is a path of length t in G_χ, and every such path is represented by some $f \in \mathcal{F}_\chi$.

▶ Remark 1.8. It is easy to check this operator is a projection: it is positive and squares to itself.

The t-fold amplified version of H, which we denote by $H^{(t)}$, can then be expressed as follows:

▶ **Definition 1.9** (Derandomised Tensor Product Amplification)**.** *Let H be layered as in Definition 1.6. We define the t-fold amplification $H^{(t)}$ of H with respect to $\{\mathcal{F}_\chi\}_{\chi\in[g]}$ as*

$$H^{(t)} = \sum_{\chi\in[g]} w_\chi \cdot H_\chi^{(t)}, \quad \textit{where} \quad H_\chi^{(t)} \equiv \mathop{\mathbb{E}}_{f\in\mathcal{F}_\chi} \left(\mathbb{I} - \bigotimes_{j\in[t]} \left(\mathbb{I} - \Pi_{f(j)}^\chi\right)\right). \tag{1.6}$$

If the original H was a k-local Hamiltonian on n qubits and m clauses, then $H^{(t)}$ is $(k\cdot t)$-local Hamiltonian on $n\cdot t$ qubits and $d^t\cdot m$ clauses.

▶ Remark 1.10. Our construction of the amplified Hamiltonian is similar to that of [1], but we use tensor products instead of matrix products.

1.3.2 Main techniques

Here we highlight the main techniques behind the proof of Theorem 1.2.

Background: tensor product amplification

The starting point for our gap amplification procedure is tensor product amplification. The t-fold tensor product amplification of a Hamiltonian H is simply

$$\mathbb{I} - (\mathbb{I} - H)^{\otimes t}. \tag{1.7}$$

If H has a decomposition as a sum of projections, $H = \mathbb{E}_{i\in[m]}\,\Pi_i$, then the tensor product amplification of H can be expanded as

$$\mathop{\mathbb{E}}_{(i_1,\ldots,i_t)\in[m]^t} \left(\mathbb{I} - \bigotimes_{j\in[t]} (\mathbb{I} - \Pi_{i_j})\right). \tag{1.8}$$

If H had m terms, therefore, the t-fold tensor product amplification of H has m^t terms.

Analysing tensor product amplification is straightforward: one simply has to observe that, since the operator $\mathbb{I} - (\mathbb{I} - H)^{\otimes t}$ is diagonal in the basis constituted by tensor products of eigenvectors of H, the ground energy of $\mathbb{I} - (\mathbb{I} - H)^{\otimes t}$ is attained by a tensor product across t registers of ground states of H. If we start with a Hamiltonian family $\{H_n\}_n$ with promise gap of size γ (and completeness sufficiently close to 1), then the gap increases at least to $t\gamma$ after t-fold tensor product amplification.

We would, however, like to achieve a similar amount of gap amplification without increasing the number of terms so steeply. In particular, instead of m^t terms, we would like the t-fold amplification of H to have $c(t)\cdot m$ terms, where $c(t)$ is a function that depends arbitrarily on t but not on m. Then we can set t to be a constant and iterate gap amplification $O(\log n)$ times to get a new Hamiltonian family $\{H'_n\}_n$ such that H'_n only has a multiplicative factor $\mathrm{poly}(n)$ more terms than H_n, but the promise gap of $\{H'_n\}_n$ is also a $\mathrm{poly}(n)$ factor larger than that of $\{H_n\}_n$.

▶ Remark 1.11. If we are content to show QMA-completeness of streaming quantum PCPs under *randomised reductions*, one way that we could reduce the number of terms in Equation (1.8) is to pick some to keep at random and delete the others. More specifically, we could

prove an analogue of our Theorem 1.4 under randomised reductions by randomly subsampling terms of Equation (1.8) using the matrix Chernoff bound: see e.g. [24]. The gap amplification statement that we prove, Theorem 1.2, can be viewed as the "correct" derandomisation of matrix-Chernoff-based subsampling of tensor product amplification. Theorem 1.2 allows us to show Theorem 1.4 under deterministic reductions.

One advantage of derandomisation in view of applications is that Theorem 1.2 is *iterable*, whereas the matrix-Chernoff-based approach is not. That is, Theorem 1.2 holds even when t is a constant, but matrix-Chernoff can only be used to do "one-shot" amplification (to amplify in one go up to constant gap), since it requires the Hamiltonian family being subsampled to have a constant promise gap.

Background: classical sequential repetition and its derandomisation

The classical analogue of tensor product amplification is *sequential repetition*, in which, given a constraint satsifaction problem (CSP) C with m clauses $c_1, \ldots, c_m$, one constructs a new CSP C' whose clauses are the ANDs of clauses from C: that is, every clause in C' is of the form $c_{i_1} \wedge \cdots \wedge c_{i_t}$ for $(i_1, \ldots, i_t) \in [m]^t$. If C is satisfiable, than C' is also satisfiable; and, if γ fraction of clauses in C are unsatisfied by the most satisfying possible assignment to C (the "unsatisfiability" of C is at least γ), then this fraction will be at least $t\gamma$ for C'.

In the classical case, we can also consider *derandomised* sequential repetition, which achieves a similar amplification guarantee but using far fewer clauses. We can achieve derandomised sequential repetition by first constructing C', the full sequential repetition of C, and then *subsampling* the clauses of C' using random walks on expander graphs. More specifically,

▶ **Definition 1.12** (Derandomised sequential repetition)**.** *Given a (classical) CSP C on n bits and m clauses, its t-fold derandomised sequential repetition is a CSP $C^{(t)}$ on $t \cdot n$ bits where the new clauses are the AND of all the clauses that lie along any length t path in some expander graph defined over $[m]$.*

So that this exposition is self-contained, we will sketch how derandomised sequential repetition can be analysed using a particular property of random walks on expander graphs [30, Chapter 4] [17, Section 6]. The only property of random walks which we need is a "set-avoiding" property: For any fixed sets $A, B \subseteq [m]$ of no more than $\delta \cdot m$ in size, the probability that a random walk that starts in A ends up in B after s steps is bounded above by $\delta + \mu^s$, where μ is the second largest eigenvalue of the expander graph. This property follows in a straightforward way from Lemma 2.10.

We sketch how this property would be useful for a classical analysis of derandomised sequential repetition. Recall that $C^{(t)}$ is the derandomised sequential repetition of C, as defined in Definition 1.12. We define a quantity called the *violation number* which will be useful in the analysis:

▶ **Definition 1.13** (Violation number)**.** *The violation number $X_{x,C^{(t)}}$ is a random variable that can be sampled by picking a uniformly random "path clause" $p = c_1 \wedge \cdots \wedge c_t$ from $C^{(t)}$ and evaluating how many out of the t clauses in the AND is violated by x.*

The violation number is a non-negative integer, and $\Pr_p[X_{x,C^{(t)}} > 0]$ precisely captures the fraction of clauses of $C^{(t)}$ which are violated by x. Recall the Second Moment Method, which for any random variable X relates $\Pr[X > 0]$ to the mean and variance of X:

▶ **Fact 1.14** (Second Moment Method). *For any non-negative random variable $X \geq 0$,*

$$\mathbb{P}[X > 0] \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]} \tag{1.9}$$

As such, if we can get a *lower bound* on the expectation of $X_{x,C^{(t)}}$ and an *upper bound* on the expectation of its square, we will obtain a lower bound on $\Pr[X_{x,C^{(t)}} > 0]$ and by extension a lower bound on the unsatisfiability of $C^{(t)}$.

The expectation of $X_{x,C^{(t)}}$ is easy to control using linearity of expectation. We now sketch how to control the expectation of the square using the "set-avoiding" property of random walks. For any classical assignment $x \in \{0,1\}^{nt}$ to $C^{(t)}$, x can be grouped into blocks $x_1, \cdots, x_t \in \{0,1\}^n$ – which in some sense are assignments to the original C – and one can define a set of clauses $\mathsf{Viol}_{x,i} \subset [m]$ which contain the clauses of C violated by x_i. Then we see that the expectation of the square of the violation number is precisely

$$\sum_{i,j \in [t]} \Pr_{p \leftarrow C^{(t)}}[c_i \in \mathsf{Viol}_{x,i} \text{ and } c_j \in \mathsf{Viol}_{x,j}] \tag{1.10}$$

$$= \sum_{i,j \in [t]} \Pr_{p \leftarrow C^{(t)}}[c_i \in \mathsf{Viol}_{x,i}] \cdot \Pr_{p \leftarrow C^{(t)}}[c_j \in \mathsf{Viol}_{x,j} \mid c_i \in \mathsf{Viol}_{x,i}] \tag{1.11}$$

where the probability is over the sampling of the random "path clause" $p \coloneqq c_1 \wedge \cdots \wedge c_t$. The probability $\Pr_{p \leftarrow C^{(t)}}[c_j \in \mathsf{Viol}_{x,j} \mid c_i \in \mathsf{Viol}_{x,i}]$ is exactly the probability that a random walk which starts in $\mathsf{Viol}_{x,i}$ ends up in $\mathsf{Viol}_{x,j}$ after $|j-i|$ steps, which is the type of quantity controlled by the "set avoiding" property.

▶ Remark 1.15. Since our proof strategy revolves around the second moment of $X_{x,C^{(t)}}$, derandomised sequential repetition would be even easier to analyse if the "path clauses" $p \leftarrow C^{(t)}$ were sampled *pairwise independently* instead of using a random walk. Pairwise independent sampling is too randomness-expensive to yield the parameters we want, so we cannot use it in Definition 1.12; however, this observation is the motivation for the proof strategy that we adopt (see "Technique 1: commuting layers" below).

Difficulties in quantisation

Although this outline is sufficient for classical CSPs, unfortunately, we encounter an obstacle when we try to quantise the argument above. We can, analogously with Definition 1.12, define the *derandomised tensor product amplification* of a Hamiltonian H and call it $H^{(t)}$: this is the motivation for Definition 1.9, in which we define our construction of the amplification of a Hamiltonian. However, $H^{(t)}$ may be non-commuting, so there need not be a valid definition of the violation sets $\mathsf{Viol}_{x,i}$ which we relied on in the classical analysis. Moreover, it may have exclusively entangled ground states, so measuring a clause could collapse the state and affect the probabilities that other clauses are violated.

In this overview we will not attempt to give a complete sketch of our analysis: the reader who wants to understand our entire proof can go directly to the technical sections, which contain further exposition. Here we will focus instead on describing at a high level the two main techniques that we use to overcome the difficulty that the ground states of $H^{(t)}$ may be entangled. Both these techniques rely in a sense on reducing the analysis of an entangled state to that of a convex combination of product states, albeit in different ways.

Technique 1: commuting layers

If all the terms in the original Hamiltonian H *commute*, then we can reduce our analysis to the classical case: we simply imagine measuring the potentially entangled ground state of $H^{(t)}$ in a complete basis that diagonalises all the terms in $H^{\otimes t}$ simultaneously. This collapses the state to a mixture over product states, which we can deal with using the classical argument and convexity. Of course, it is not true that all the terms in H commute, but we can conduct some parts of our analysis by partitioning the terms in H into at most constantly many commuting *layers*, analysing each layer separately, and recombining the layers afterwards (up to some amount of loss).[6]

We use this commuting layers technique only in order to prove that random walk subsampling of terms can be approximately by *pairwise independent* subsampling in our setting. We then develop a new technique – our "miser's de Finetti" technique, which we introduce shortly – in order to prove that pairwise independent sampling is similar enough to fully independent sampling for us even on entangled states. Fully independent sampling is equivalent to (non-derandomised) tensor product amplification, so this statement allows us to get good bounds on the amplification guarantees of our procedure. This last step – i.e. relating the pairwise independent case to the fully independent case – is arguably the hardest part of our analysis.

▶ Remark 1.16. The authors of [1] encounter similar issues, and also resolve them by partitioning the original Hamiltonian into commuting layers. However, they carry through their *entire* analysis using the commuting-layer approach to deal with the entanglement problem, while we use both the commuting-layer approach and also our "miser's de Finetti" technique. Their approach gives rise to constants in the amplification bounds that depend on the locality of terms in H, and in particular become unmanageable when the locality of H is larger than constant; ours, perhaps surprisingly, does not. This is the chief reason we are able to iterate our procedure.

Technique 2: miser's de Finetti

Quantum information theory gives us a class of tools for reducing entangled states to convex combinations of product states, in the form of the *quantum de Finetti theorems* [27, 28]. There are many types of quantum de Finetti theorem, but they all say something like the following: if we take a *symmetric* (permutation-invariant) pure quantum state ρ on t registers and trace out all but $k \ll t$ registers, the mixed state left over on those registers is "close" to a convex combination of product states.

Naïvely, we might try to use a de Finetti theorem to solve our problem as follows. Recall that each term Π_{amp} in the amplified Hamiltonian $H^{(t)}$ (Definition 1.9) is a tensor product $\Pi_{\text{amp}} = (\mathbb{1} - \Pi_1) \otimes \cdots \otimes (\mathbb{1} - \Pi_t)$, where $\Pi_1, \ldots, \Pi_t$ are terms in H, and the constituent terms of Π_{amp} as well as the order in which they appear are determined by a walk. Instead of placing the t terms in Π_{amp} in the order that the walk dictates, we could pick a random permutation π and set $\Pi_{\text{amp}} = (\mathbb{1} - \Pi_{\pi(1)}) \otimes \cdots \otimes (\mathbb{1} - \Pi_{\pi(t)})$ instead – or, if we want our reduction to remain deterministic, we could put all $t!$ possible orderings as terms. This is equivalent to randomly permuting the registers of the state τ to which we apply the terms, so this manoeuvre allows us to assume that τ is permutation symmetric. We trace out all but

[6] Note that, since our amplification procedure (see Definition 1.9) preserves the number of commuting layers, we can iterate our procedure as many times as we like as long as we start with an H that has constantly many layers. We show that such Hamiltonians are QMA-complete in the full version [10].

the last $k \ll t$ registers of τ, leaving us with a state which is (mostly) a convex combination of (approximately) product states. We can then apply the classical argument to these k remaining registers.

Unfortunately, this naïve plan does not work for quantitative reasons. The best possible de Finetti theorem requires $k = \Omega(\log d)$, where d is the dimension of the Hilbert space associated with a single register. This would mean that we get no amplification unless we set $t \geq \text{poly}(n)$, which makes random walk sampling untenably expensive. This would also mean that, if we want a deterministic reduction, we need to have order $n!$ terms in the amplified Hamiltonian. Either of these facts is a sufficient obstacle to achieving an amplified Hamiltonian with a polynomial number of terms.

As such, we cannot hope to rely on a quantum de Finetti theorem as a black box in our setting. However, perhaps surprisingly, we are able to make progress with a technique that is inspired by a particular style of proof for a de Finetti theorem. A strategy for proving a de Finetti theorem goes as follows [31]. The keystone of the strategy is the following fact, which can be made quantitative and then proven using elementary techniques:

▷ Claim 1.17. If ρ is any permutation symmetric state on t registers, and we *project* the last $t - k$ registers onto the state $|\psi\rangle^{\otimes(t-k)}$ for some pure single-register state $|\psi\rangle$, then the first k registers of the residual state must be "close" to the pure state $|\psi\rangle^{\otimes k}$, because the state was permutation symmetric.

It then turns out that, for any permutation symmetric state ρ on t registers, we can write ρ with the last $t - k$ registers traced out as a convex combination of states of the form $(\text{id}_{t-k} \otimes \langle\psi|^{\otimes k})\rho(\text{id}_{t-k} \otimes |\psi\rangle^{\otimes k})$. Combining these two deceptively simple observations yields the de Finetti theorem: if we take a symmetric state ρ on t registers and trace out the final $t - k$ registers, the mixed state left over on the first k registers is "close" to a convex combination of product states $|\psi\rangle^{\otimes k}$.

This argument is at the heart of [31], which gives an astonishingly simple proof of the so-called "exponential de Finetti theorem". However, in order to make Claim 1.17 true quantitatively, t has to be quite large. Intuitively, we can understand this as follows: arguing that the first k registers of a permutation-symmetric state ρ must look similar to $|\psi\rangle^{\otimes k}$ if the last $t - k$ registers are in the state $|\psi\rangle^{\otimes(t-k)}$ is rather like collecting *measurement statistics* about ρ from the last $t - k$ registers and then arguing that, since the state is permutation symmetric, the measurement statistics are representative of the first k registers as well. If t is small, the statistics which were collected from the measured registers are unlikely to be informative about the remaining k registers. In particular, requiring the remaining k registers to be in a state close to $|\psi\rangle^{\otimes k}$ is a stringent condition – we are characterising the leftover state very finely if we find that this condition is true – and one would therefore expect to look at a large number of registers t before one could conclude such a thing with any reasonable probability.

In our situation, we don't actually *need* such a fine characterisation of the state. Indeed, we don't care if the state τ is actually product or not: much weaker guarantees suffice for us. As such, we can potentially get away with collecting far fewer "measurement statistics" about τ than we would have to collect if we wanted to know that τ was product. For example, it is sufficient for us that τ "looks product" with respect to measurements of H, the original Hamiltonian (because the quantity we are trying to bound involves only the trace of H on τ in various registers). As such, we could measure τ in an eigenbasis that diagonalises $H^{\otimes t}$, and then use a *classical* de Finetti theorem on the measurement outcomes. It turns out that this approach also requires an unviably large t (since classical de Finetti theorems also scale unfavourably with the alphabet size of the random variables) – but we can do even better

by realising that we don't need to know the precise eigenvalues of the eigenstates we get in each register after the eigenbasis measurement. Indeed, a very coarse approximation ("is the eigenvalue big or small?") will suffice. This coarse-graining, which allows us to reduce the alphabet size of the measurement outcomes to which we apply de Finetti–inspired techniques, is the chief reason that we can reduce t to something manageable.

This is the idea behind the auxiliary measurement we introduce in Section 5. The binary projective measurements $\{\Pi^{<\alpha}, \mathbb{I} - \Pi^{<\alpha}\}$ (Definition 5.1) act on the same Hilbert space as H, and $\Pi^{<\alpha}$ simply projects onto the energy eigenspaces of H with lower energy than α: therefore, $\Pi^{<\alpha}$ commutes with H. Measuring $\Pi^{<\alpha}$ on a random set of $t/2$ registers allows us to estimate the "energy" E of τ (the estimate we use is the number of $\geq \alpha$ outcomes). We then measure the complementary $t/2$ registers, and use Chernoff-bound-like tools to argue that they are likely to land in eigenspaces with similar energy E. After that, our proof proceeds via a careful case analysis, which hinges on whether E is "high" or "low". In effect, our strategy is to partition the Hilbert space in which τ lives into (constantly many) subspaces, each of which has predictable behaviour with respect to H; and we are able to proceed with the analysis after we project ("pinch") τ into one of these subspaces because the "pinching" measurement commutes with the measurements of H, which are the only measurements that we care about.

1.4 Organization

We begin in section Section 2 by presenting basic definitions of local Hamiltonians and expander graphs, as well as some basic probability facts.

We proceed in Section 3 by formally introducing the violation number measurement, an observable which quantifies the number of violated clauses on a random length t path. In Section 4 we argue the completeness of the amplification scheme (Theorem 1.2, Completeness), by applying the first moment method to the amplified Hamiltonian.

In Section 5 we present the auxiliary energy measurement, a central technical tool which captures the de Finetti ingredient of our proof. In Section 6, we present the proof of soundness of our amplification scheme (Theorem 1.2, Soundness), wherein we apply the second moment method to the violation number measurement.

2 Preliminaries

We dedicate this section to basic facts on the local Hamiltonian problem, probability theory, and expander graphs.

2.1 The Local Hamiltonian problem

The central problem of study in Hamiltonian complexity theory is computing the ground state energy of a local Hamiltonian.

▶ **Definition 2.1** (k-$\mathsf{LH}[\mathsf{a},\mathsf{b}]$)**.** *Let $H = \sum_i^m h_i$ be a k-local Hamiltonian on n qubits and $m = \mathrm{poly}(n)$ terms, where each h_i is a hermitian operator acting on k qubits and is described using $\mathrm{poly}(n)$ bits. Let $0 \leq H \leq \mathbb{I}$.*

Further, let $a < b$ be two real parameters described using $\mathrm{poly}(n)$ bits; $b - a$ is referred to as the "promise gap" of H. The k-local Hamiltonian problem k-$\mathsf{LH}[\mathsf{a},\mathsf{b}]$ then consists of the task of deciding whether the ground state energy $\lambda_{min}(H) \leq a$ or $\lambda_{min}(H) \geq b$, promised that H satisfies one of the two cases.

[21] proved that the local Hamiltonian problem is QMA-Complete.

▶ **Theorem 2.2** ([21]). *The* 5-LH$[2^{-\mathrm{poly(n)}}, 1/\mathrm{poly(n)}]$ *problem is* QMA*-complete.*

The Quantum PCP Conjecture stipulates that the local Hamiltonian problem remains just as hard under a constant promise gap.

▶ **Conjecture 2.3** (The Quantum PCP Conjecture [2]). *There exists a constant locality k as well as $a, b \in [0,1]$ satisfying $b - a = \Omega(1)$ such that the k-*LH$[a, b]$ *problem is* QMA*-complete.*

2.2 Probability

Here we recollect a series of simple inequalities.

▶ **Fact 2.4** (Markov's Inequality). *For any non-negative random variable $X \geq 0$ and real parameter $a > 1$,*

$$\mathbb{P}[X > a \cdot \mathbb{E}[X]] \leq \frac{1}{a} \tag{2.1}$$

▶ **Fact 2.5** (First Moment Method). *For any non-negative random variable $X \geq 0$,*

$$\mathbb{P}[X > 0] \leq \mathbb{E}[X] \tag{2.2}$$

▶ **Fact 2.6** (Second Moment Method). *For any non-negative random variable $X \geq 0$,*

$$\mathbb{P}[X > 0] \geq \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]} \tag{2.3}$$

▶ **Fact 2.7** ([29], Lemma 6). *Let $Z_1, \cdots Z_{n+k}$ be a collection of binary random variables. Let $S \subset [n+k], |S| = k$ be a uniformly random subset. Then,*

$$\Pr\left[\sum_{i\in S} Z_i \leq k \cdot \delta \text{ and } \sum_{i\in \bar{S}} Z_i \geq n(\delta + \nu)\right] \leq \exp\left[-\frac{2\nu^2 nk^2}{(n+k)(k+1)}\right] \tag{2.4}$$

2.3 Expander graphs

We rely on a series of basic facts of expander graphs.

▶ **Definition 2.8** (Spectral Expansion). *Let $G = (V, E)$ be an undirected d-regular graph, and let A be its adjacency matrix. Let $d = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be its eigenvalues. Then G is said to be a λ-spectral-expander if*

$$\max_{i\neq 1} |\lambda_i| \leq \lambda. \tag{2.5}$$

We often-times will refer to the transition matrix $P = A/d$ of the random walk on G, and let $\mu = \lambda/d$. For our reduction we require an explicit deterministic construction.

▶ **Lemma 2.9** ([26]). *There exists explicit constants $m_0, d \in \mathbb{N}$ and a parameter $d > \lambda > 0$, such that there exists a deterministic polynomial-time constructable family $\{G_m\}_{m\geq m_0}$ of d-regular λ-spectral-expander graphs on m vertices for every $m \geq m_0$.*

See e.g. [17, Lemma 2.1] on how to achieve the condition $\forall m \geq m_0$, from the infinite family of [26].

The constraints on λ entail $\mu < 1$. We will require a short lemma on quadratic forms of the transition matrix P of an expander graph.

▶ **Lemma 2.10.** *Let $P = A/d$ be the transition matrix of a d-regular $(d \cdot \mu)$-spectral-expander graph on m vertices. Then, for all vectors $a, b \in [0,1]^{\times m}$ and integer t:*

$$a^T P^t b \leq \frac{1}{m}\|a\|_1 \cdot \|b\|_1 + \mu^t\big(\|a\|_1 + \|b\|_1\big) \tag{2.6}$$

Proof. Let us diagonalize $P = \sum_k \mu_k \gamma_k \gamma_k^T$, where $\mu_1 = 1$, $|\mu_{k>1}| \leq \mu$, and the eigenvectors γ_k form an orthonormal basis.

$$a^T P^t b = \frac{1}{m}\|a\|_1 \cdot \|b\|_1 + \sum_k \mu_k^t (a \cdot \gamma_k)(b \cdot \gamma_k) \tag{2.7}$$

$$\leq \frac{1}{m}\|a\|_1 \cdot \|b\|_1 + \mu^t \sum_k |a \cdot \gamma_k||b \cdot \gamma_k| \tag{2.8}$$

$$\leq \frac{1}{m}\|a\|_1 \cdot \|b\|_1 + \mu^t \left(\sum_k |a \cdot \gamma_k|^2\right)^{1/2} \left(\sum_k |b \cdot \gamma_k|^2\right)^{1/2} \tag{2.9}$$

$$\leq \frac{1}{m}\|a\|_1 \cdot \|b\|_1 + \mu^t \|a\|_2 \cdot \|b\|_2 \tag{2.10}$$

Where, in sequence, we used $|\mu_k| \leq \mu$, the Cauchy-Schwartz inequality, and the orthonormality of the γ_k. Now,

$$\|a\|_2 \cdot \|b\|_2 \leq \|a\|_2^2 + \|b\|_2^2 \leq \|a\|_1 + \|b\|_1 \tag{2.11}$$

by the AM-GM inequality and since each entry of a, b is ≤ 1. ◀

3 The Main Quantity of Interest: The Violation Number Measurement

We recollect that we consider families of layered Hamiltonians as in Definition 1.6. Following Definition 1.9, to define the amplified Hamiltonian, we amplify the layers separately:

$$H = \sum_{\chi \in [g]} w_\chi H_\chi \overset{\text{amplify}}{\longrightarrow} H^{(2t)} := \sum_{\chi \in [g]} w_\chi H_\chi^{(2t)} \tag{3.1}$$

Let the number of projective terms in H_χ be m_χ.

For any given color χ and function $f \in \mathcal{F}_\chi$ (the paths of length $2t$, Definition 1.7), it is useful to associate a measurement N_f^χ. N_f^χ counts the number of projections of color χ that are violated when the projectors specified by $f \in \mathcal{F}_\chi$ are measured, i.e. the projector $\Pi_{f(j)}$ is measured on the jth register for all $j \in [2t]$.

▶ **Definition 3.1** (Violation Number Measurement)**.** *Fix a color $\chi \in [g]$. Given a state ρ on $2t \cdot n$ qubits and a function $f : [2t] \to [m_\chi]$:*

1. *Divide the $2t \cdot n$ qubits into $2t$ blocks of n qubits, and measure each block as follows: On the jth block, perform the measurement $\{\mathbb{I} - \Pi_{f(j)}^\chi, \Pi_{f(j)}^\chi\}$ with outcome $b_j \in \{0,1\}$.*
2. *Output $\sum_j b_j$.*

The observable associated with this measurement is denoted

$$N_f^\chi \equiv \sum_{j \in [2t]} (\Pi_{f(j)}^\chi)_{\text{reg}[j]} \otimes \mathbb{I}_{[2t]\setminus j} \ . \tag{3.2}$$

The notation $(\Pi_{f(j)})_{\text{reg}[j]}$ means $\Pi_{f(j)}$ acting on register j. Intuitively, N_f^χ chooses $2t$ projectors from the Hamiltonian H_χ according to the given function f and then measures these projectors in parallel to count how many of them are violated on the state ρ.

▶ Remark 3.2. N_f^χ is a Hermitian operator with eigenvalues $\{0, 1, \ldots, 2t\}$ whose eigenspaces are the projectors onto the different outcomes of the measurement procedure above. However, as written Equation (3.2) is not the eigendecomposition of N_f^χ.

We also define another collection of measurements, this one indexed by χ (color), f (function) and also $S \subseteq [2t]$ (subset). Measuring $N_{f,S}^\chi$ will correspond to measuring the projectors specified by f only in the registers whose indices are inside S, and counting how many violations result.

▶ **Definition 3.3** (Subset Violation Number). *Fix a color $\chi \in [g]$. Given a state ρ on $2t \cdot n$ qubits, a function $f : [2t] \to [m_\chi]$ and a subset $S \subseteq [2t]$:*

1. *Divide the $2t \cdot n$ qubits into $2t$ blocks of n qubits, and for $j \in S$ measure $\{\mathbb{I} - \Pi_{f(j)}^\chi, \Pi_{f(j)}^\chi\}$ on the jth block with outcome $b_j \in \{0, 1\}$.*
2. *Output $\sum_{j \in S} b_j$.*

We denote the associated observable as $N_{f,S}^\chi$ in the natural way. We will be interested in the probability that measuring N_f^χ (or $N_{f,S}^\chi$) on some state ρ does *not* yield outcome 0. We will write this probability as

$$\Pr_\rho\left[N_f^\chi > 0\right].$$

The projector onto the 0-eigenspace of N_f^χ is $\bigotimes_{j \in [2t]} (\mathbb{1} - \Pi_{f(j)}^\chi)$, so we have

$$\Pr_\rho\left[N_f^\chi > 0\right] = 1 - \mathrm{Tr}\left[\bigotimes_{j \in [2t]} (\mathbb{I} - \Pi_{f(j)}^\chi)\rho\right]. \tag{3.3}$$

The following lemmas give the relationship between the energy of $H_\chi^{(2t)}$ and the observables N_f^χ and $N_{f,S}^\chi$.

▶ **Lemma 3.4.** *Consider $H_\chi^{(2t)}$ and N_f^χ as defined above. Then, for any state ρ:*

$$\mathrm{Tr}\left[H_\chi^{(2t)}\rho\right] = \mathop{\mathbb{E}}_{f \in \mathcal{F}_\chi} \Pr_\rho\left[N_f^\chi > 0\right]. \tag{3.4}$$

Proof. This follows by comparing the definition of $H_\chi^{(2t)}$ and Equation (3.3). ◀

▶ **Lemma 3.5.** *Consider $H_\chi^{(2t)}$, N_f^χ and $N_{f,S}^\chi$ as defined above. Then, for any function f, state ρ, and subset S,*

$$\mathrm{Tr}\left[N_f^\chi \rho\right] \geq \mathrm{Tr}\left[N_{f,S}^\chi \rho\right], \quad \Pr_\rho\left[N_f^\chi > 0\right] \geq \Pr_\rho\left[N_{f,S}^\chi > 0\right] \tag{3.5}$$

Proof. The bits b_j that are summed in the measurement procedure for $N_{f,S}^\chi$ are a subset of the bits summed in the measurement procedure for N_f^χ. ◀

4 On the Completeness of the Amplified Hamiltonian

In this section, we present an upper bound on the ground state energy of the amplified Hamiltonian. The upper bound says that if the ground state energy of the original Hamiltonian was very low, then the ground state energy of the amplified Hamiltonian remains fairly low.

▶ **Proposition 4.1.** *Fix a layered Hamiltonian H as in Definition 1.6, and the collection of function families $\{\mathcal{F}_\chi\}$ from Definition 1.7. For any $t \in \mathbb{N}$, write $H^{(2t)} = \sum_\chi w_\chi H_\chi^{(2t)}$ to denote the derandomized $2t$-fold tensor product amplification of H (Definition 1.9). The lowest eigenvalue of $H^{(2t)}$ is bounded from above by*

$$\lambda_{\min}(H^{(2t)}) \leq 2t \cdot \lambda_{\min}(H)\,.$$

Proof. Let ρ be the ground state of the original Hamiltonian $H = \sum_\chi w_\chi H_\chi$. Let us consider the Hamiltonian terms of each color one at a time. Note that, since we are proving an upper bound on the lowest eigenvalue of the amplified Hamiltonian, it suffices to show that the value is upper bounded on some specific state. Consider the state $\rho^{\otimes 2t}$. From Lemma 3.4,

$$\mathrm{Tr}[H_\chi^{(2t)}\rho^{\otimes 2t}] = \mathbb{E}_{f\in\mathcal{F}_\chi} \Pr_{\rho^{\otimes 2t}}\left[N_f^\chi > 0\right] \tag{4.1}$$

From the first moment method (Fact 2.5), we have

$$\Pr_{\rho^{\otimes 2t}}\left[N_f^\chi > 0\right] \leq \mathrm{Tr}\left[N_f^\chi \rho^{\otimes 2t}\right] \tag{4.2}$$

$$\Rightarrow \quad \mathbb{E}_{f\in\mathcal{F}_\chi} \Pr_{\rho^{\otimes 2t}}\left[N_f^\chi > 0\right] \leq \sum_{i\in[2t]} \mathrm{Tr}\left[(H_\chi)_{\mathrm{reg}[i]}\rho^{\otimes 2t}\right] = 2t \cdot \mathrm{Tr}[H_\chi\rho] \tag{4.3}$$

By linearity, we conclude

$$\lambda_{min}(H^{(2t)}) \leq 2t \cdot \sum_\chi w_\chi \mathrm{Tr}[H_\chi\rho] = 2t \cdot \lambda_{min}(H) \tag{4.4}$$

◀

5 A Technical Tool: The Auxiliary Energy Measurement

To simplify notation, for $j \in [2t]$, let us denote by $H_{\mathrm{reg}[j]}$ the application of H on the j-th register.

$$H_{\mathrm{reg}[j]} = \underbrace{\mathbb{I}\otimes\cdots\otimes\mathbb{I}}_{j-1} \otimes H \otimes \underbrace{\mathbb{I}\otimes\cdots\otimes\mathbb{I}}_{2t-j} \tag{5.1}$$

In this section, we introduce a measurement which we refer to as the "auxiliary energy measurement". This measurement is never performed in a real measurement of the amplified Hamiltonian $H^{(2t)}$; it exists purely in the analysis. Loosely speaking, we think of this measurement as a *diagnostic*: we select some random subset of registers on which to perform it, and its outcomes on those "auxiliary" registers will help us determine how to proceed in the analysis of the quantity we care about, namely $N_{f,S}^\chi$, which is defined on the remaining (non-auxiliary) registers. The fact that we select the subset of registers randomly will mean that the prover cannot adversarially bias the outcome of the auxiliary measurement in order to mislead us about the non-auxiliary registers.

▶ **Definition 5.1** (The Auxiliary Energy Measurement)**.** *Given any state ρ on $n \cdot 2t$ qubits, a subset $S \subseteq [2t]$, and a threshold parameter $\alpha \in \mathbb{R}$:*

1. *For $j \in [2t]$ define $\Pi_{\mathrm{reg}[j]}^{\geq\alpha}$ as the projection onto the direct sum of all eigenspaces of $H_{\mathrm{reg}[j]}$ with associated eigenvalue larger than α.*
2. *For all $j \in \overline{S}$, measure $\{\mathbb{I} - \Pi_{\mathrm{reg}[j]}^{\geq\alpha}, \Pi_{\mathrm{reg}[j]}^{\geq\alpha}\}$ and call the outcome $c_j \in \{0,1\}$.*
3. *Output $c = (c_j)_{j\in\overline{S}}$.*

We will write the probability of receiving an outcome c in this measurement as $\Pr_\rho[c]$, where it will be clear from context that we are referring to the auxiliary energy measurement and what the set S and threshold α are. We will denote by $\rho_{|S,\alpha,c}$ the post-measurement state after receiving outcome c in the auxiliary measurement with set S and threshold α. Note that in the measurement procedure, we measure all registers *not* in S, i.e. $\rho_{|S,\alpha,c}$ is a post-measurement state whose registers in S have been left untouched.

Our analysis will hinge on a careful case division, where we classify the possible outcomes of the auxiliary energy measurement into two categories, "high-energy" and "low-energy", and decide what to do based on whether we got a high– or low-energy outcome. For this purpose, let $r \in [t]$ be an integer parameter (to be picked soon). For any subset $S \subseteq [2t]$, we define

$$U_{\bar{S}}^r \equiv \left\{ c \in \{0,1\}^{\bar{S}} : |c| \geq 4r \right\}^7 \tag{5.2}$$

The following lemma captures the intuition that our auxiliary measurement is a good diagnostic because we pick the subset on which to perform it randomly. Suppose we were to extend the auxiliary energy measurement to **all** the registers, i.e., suppose we were to perform it in S as well as in $\bar{S}$. We expect that, since S is picked randomly, it is likely that, when we condition on receiving a high energy outcome in $\bar{S}$, we will also receive a high energy outcome in S with decent probability.

▶ **Lemma 5.2.** *For each $j \in [2t]$, let the binary random variable C_j denote the outcome of the measurement $\{\mathbb{1} - \Pi_{\mathrm{reg}[j]}^{\geq\alpha}, \Pi_{\mathrm{reg}[j]}^{\geq\alpha}\}$ on any $2t \times n$ qubit state ρ. Then, for every $r \in [t]$,*

$$\mathbb{E}_{|S|=t} \Pr_\rho \Big[\sum_{j \in S} C_j < 2r \;\; \textit{and} \;\; \sum_{i \in \overline{S}} C_i \geq 4r \Big] \leq e^{-\frac{2r^2}{t}} \tag{5.3}$$

Proof. Here we remark that the $\{C_j\}$ correspond to the outcomes of a sequence of commuting measurements, so their joint distribution is well-defined. The proof then follows immediately from Fact 2.7. ◀

As we later discuss, this is the central "de Finetti" statement we require in our approach.

▶ Remark 5.3. So long as one picks the same set S in both, the subset violation number measurement and the auxiliary energy measurement commute. We have therefore the following decomposition for any state ρ, color χ, threshold α, subset S and function f:

$$\Pr_\rho \Big[N_{f,S}^\chi > 0 \Big] = \sum_{c \in \{0,1\}^{\overline{S}}} \Pr_\rho[c] \Pr_{\rho_{|S,\alpha,c}} \Big[N_{f,S}^\chi > 0 \Big] \, . \tag{5.4}$$

Our final lower bound on the ground energy of the amplified Hamiltonian, introduced in Section 6, is based solely on the behavior of $N_{f,S}^\chi$, which we manage to control using the auxiliary measurements. We are "sacrificing" the auxiliary registers, in the sense that we do not count violations on them except as a diagnostic tool. We can do this because the outcome of $N_{f,S}^\chi$ always lower bounds that of N_f^χ (Lemma 3.5).

---

[7] We also denote $\overline{U}_{\bar{S}}^r \subset \{0,1\}^{\bar{S}}$ as the complement set. Whenever implicit, the superscript r is omitted for conciseness.

6 On the Soundness of the Amplified Hamiltonian

In this section, we prove a lower bound on the ground energy of the amplified Hamiltonian: we show that one cycle of amplification boosts the ground energy of the original Hamiltonian by at least a constant multiplicative factor which depends on t, i.e. the number of "copies" of the original Hamiltonian. Together with Proposition 4.1, this shows that our amplification procedure amplifies the promise gap of any Hamiltonian with sufficiently good completeness.

The key challenge is that the ground state of the amplified Hamiltonian may not be a product state, and therefore may be entangled in a way that causes its energy to be lower than what a purely classical analysis of a similar gap amplification procedure, such as that of [17], would lead us to expect. Nevertheless, by combining ingredients from [17]'s proof of classical gap amplification with techniques inspired by the proof of the exponential de Finetti theorem [27, 28, 31], we still are able to show amplification of the ground state energy:

▶ **Theorem 6.1** (Simplified). *Fix a layered Hamiltonian H as in Definition 1.6, and the collection of function families $\{\mathcal{F}_\chi\}$ from Definition 1.7. For any $t \in \mathbb{N}$, write $H^{(2t)} = \sum_\chi w_\chi H_\chi^{(2t)}$ to denote the derandomized $2t$-fold tensor product amplification of H (Definition 1.9). The lowest eigenvalue of $H^{(2t)}$ is bounded from below by*

$$\lambda_{\min}(H^{(2t)}) \geq \min\left\{\Theta\Big(\frac{\log t}{t}\Big)\ ,\ \Theta\Big(\lambda_{\min}(H)\cdot\frac{t^{1/2}}{(\log t)^{1/2}}\Big)\right\},$$

where the big-O hides only constants that do not depend on t.

Due to space constraints, the proof of the theorem is available only in the arXiv version [10].

References

1 Dorit Aharonov, Itai Arad, Zeph Landau, and Umesh Vazirani. The detectability lemma and quantum gap amplification. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 417–426, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1536414.1536472.

2 Dorit Aharonov, Itai Arad, and Thomas Vidick. The quantum PCP conjecture, 2013. arXiv:1309.7495.

3 Anurag Anshu, Itai Arad, and Thomas Vidick. Simple proof of the detectability lemma and spectral gap amplification. *Physical Review B*, 93(20), May 2016. doi:10.1103/physrevb.93.205142.

4 Anurag Anshu, N. P. Breuckmann, and Quynh T. Nguyen. Circuit-to-hamiltonian from tensor networks and fault tolerance. *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, 2023. URL: https://api.semanticscholar.org/CorpusID:262935180.

5 Anurag Anshu, Nikolas P. Breuckmann, and Chinmay Nirkhe. NLTS hamiltonians from good quantum codes. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC '23, pages 1090–1096. ACM, June 2023. doi:10.1145/3564246.3585114.

6 Anurag Anshu and Chinmay Nirkhe. Limitations on brandão-harrow limitation for 4-local hamiltonians, 2022. URL: https://anuraganshu.seas.harvard.edu/sites/g/files/omnuum6156/files/anshu/files/bh_4local.pdf.

7 Itai Arad, Zeph Landau, and Umesh Vazirani. Improved one-dimensional area law for frustration-free systems. *Phys. Rev. B*, 85:195145, May 2012. doi:10.1103/PhysRevB.85.195145.

8 S. Arora, C. Lund, R. Motwani, M. Sudan, and Mario Szegedy. Proof verification and hardness of approximation problems. In *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*, pages 14–23, 1992. doi:10.1109/SFCS.1992.267823.

9 Thiago Bergamaschi. Improved product-state approximation algorithms for quantum local hamiltonians. In *International Colloquium on Automata, Languages and Programming*, 2023. doi:10.4230/LIPIcs.ICALP.2023.20.

10 Thiago Bergamaschi, Tony Metger, Thomas Vidick, and Tina Zhang. Derandomised tensor product gap amplification for quantum hamiltonians. *arXiv preprint arXiv:2510.01333*, 2025. doi:10.48550/arXiv.2510.01333.

11 Fernando G. S. L. Brandão and Aram Wettroth Harrow. Product-state approximations to quantum ground states. In *Symposium on the Theory of Computing*, 2013. URL: https://api.semanticscholar.org/CorpusID:9421795.

12 Fernando GSL Brandao and Aram W Harrow. Quantum de finetti theorems under local measurements with applications. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 861–870, 2013.

13 Fernando G. S. L. Brandão, Aram W. Harrow, and Michał Horodecki. Local random quantum circuits are approximate polynomial-designs. *Communications in Mathematical Physics*, 346(2):397–434, August 2016. doi:10.1007/s00220-016-2706-8.

14 Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for qma. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 31–40. IEEE, 2016. doi:10.1109/FOCS.2016.13.

15 Matthias Christandl, Robert König, and Renato Renner. Postselection technique for quantum channels with applications to quantum cryptography. *Physical review letters*, 102 2:020504, 2008. URL: https://api.semanticscholar.org/CorpusID:10062534.

16 Mikael de la Salle. Spectral gap and stability for groups and non-local games, 2022. arXiv:2204.07084.

17 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12–es, June 2007. doi:10.1145/1236457.1236459.

18 Irit Dinur and Or Meir. Derandomized parallel repetition via structured pcps, 2014. arXiv:1002.1606.

19 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the pcp theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006. doi:10.1137/S0097539705446962.

20 Michael J. Kastoryano and Fernando G. S. L. Brandao. Quantum gibbs samplers: the commuting case, 2016. arXiv:1409.3435.

21 Alexei Y. Kitaev, A. H. Shen, and Mikhail N. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate studies in mathematics*. American Mathematical Society, 2002. URL: https://bookstore.ams.org/gsm-47/.

22 Cécilia Lancien and Andreas Winter. Flexible constrained de finetti reductions and applications. *Journal of Mathematical Physics*, 58(9), September 2017. doi:10.1063/1.5003633.

23 Tony Metger, Anand Natarajan, and Tina Zhang. Succinct arguments for qma from standard assumptions via compiled nonlocal games, 2024. doi:10.48550/arXiv.2404.19754.

24 Anand Natarajan and Chinmay Nirkhe. The status of the quantum PCP conjecture (games version), 2024. doi:10.48550/arXiv.2403.13084.

25 Anand Natarajan and Thomas Vidick. Retracted: Two-player entangled games are np-hard. In *33rd Computational Complexity Conference, CCC 2018, San Diego, CA, USA, June 22-24, 2018*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.CCC.2018.20.

26 Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 3–13, 2000. URL: https://api.semanticscholar.org/CorpusID:420651.

27 Renato Renner. Symmetry of large physical systems implies independence of subsystems. *Nature Physics*, 3(9):645–649, 2007.

28 Renato Renner. Security of quantum key distribution. *International Journal of Quantum Information*, 6(01):1–127, 2008.

29 Marco Tomamichel and Anthony Leverrier. A largely self-contained and complete security proof for quantum key distribution. *Quantum*, 1:14, 2017. doi:10.22331/Q-2017-07-14-14.

30 Salil P. Vadhan. Pseudorandomness. *Foundations and TrendsÂ® in Theoretical Computer Science*, 7:1–336, 2012. doi:10.1561/0400000010.

31 Thomas Vidick and Henry Yuen. A simple proof of Renner's exponential de Finetti theorem. *arXiv preprint*, 2016. arXiv:1608.04814.

A Amplification from the Detectability Lemma

In this section we present an alternative "one-shot" construction of a non-local Hamiltonian with constant gap, based on the detectability lemma and its converse.[8] Roughly speaking, for any integer t, we will define an amplification scheme which maps local Hamiltonians on n qubits and m clauses of (sufficiently small) inverse-polynomial promise gap γ, to a new Hamiltonian on $t \cdot n$ qubits and $O(1)$ clauses, with promise gap $\Theta(\gamma \cdot m \cdot t)$. While these results are conceptually similar to our derandomization scheme in Theorem 6.1, the new Hamiltonian will be $O(n \cdot t)$ local (i.e. fully global, see Theorem A.3 below). We refer the reader back to Section 1.2 for a discussion.

We begin with a brief background on the detectability lemma and its converse.

▶ **Lemma A.1** (Detectability Lemma, e.g. [3, Lemma 2]). *Let $H = \mathbb{E}_{i\in[m]}\Pi_i$, where each Π_i is a projection that commutes with all except at most d of the other projections Π_j. Then, if $\lambda_{\mathsf{min}}(H) \geq \gamma$, we have for every state $|\psi\rangle$:*

$$\Big\| \prod_{i\in[m]} (\mathbb{I} - \Pi_i)|\psi\rangle \Big\|^2 \leq (1 + m\gamma/d^2)^{-1} \,. \tag{A.1}$$

▶ **Lemma A.2** (The Converse to the DL, e.g. [3, Lemma 4]). *Let $A_1, \cdots A_g$ be projections. Then, if $|\psi\rangle$ is a state satisfying for some $\epsilon \in [0,1]$,*

$$\Big\| \prod_{\chi\in[g]} A_i|\psi\rangle \Big\|^2 \leq 1 - \epsilon \,, \tag{A.2}$$

then there exists a $\chi \in [g]$ such that $\|A_\chi|\psi\rangle\|^2 \leq 1 - \frac{\epsilon}{4g}$.

For simplicity of presentation, in this section we fix our attention to local Hamiltonians which are expectations over projections $H = \frac{1}{m}\sum_i \Pi_i$; where further we assume each projection does not commute with at most $(g-1)$ other projections. This implies that the local terms of H can be efficiently (and equitably) partitioned into g commuting layers/colors (Definition 1.6). We express the terms in each layer $\chi \in [g]$ as:

$$H^\chi = \mathbb{E}_{i\in[m_\chi]}\Pi_i^\chi \tag{A.3}$$

and associate a "weight" $w_\chi = m_\chi/m = \Omega(g^{-1})$ to be the number of clauses labeled the color χ. For any integer t, we consider the amplified Hamiltonian defined by tensor products of the DL operators applied to each layer:

$$H^{(t)} = \mathbb{I} - \mathbb{E}_\chi \bigotimes_j^t \Big(\prod_{i\in[m_\chi]} (\mathbb{I} - \Pi_i^\chi) \Big) \,, \tag{A.4}$$

[8] We thank Anurag Anshu for pointing us to this construction.

where as before, $\mathbb{E}_\chi$ indicates a sample χ from the distribution $(w_1, w_2, \cdots w_g)$ over $[g]$. The amplified Hamiltonian is Hermitian because the projections appearing in the product $\prod_{i\in[m_\chi]}(\mathbb{I} - \Pi_i^\chi)$ all commute, since they belong to the same layer.

▶ **Theorem A.3** (Amplification from the DL). *Assume that the minimum eigenvalue of H is $\lambda_{\mathsf{min}}(H) \geq \gamma$. Then, the minimum eigenvalue of the amplified Hamiltonian $H^{(t)}$ from Equation A.4 satisfies*

$$\lambda_{\mathsf{min}}(H^{(t)}) \geq \min\left(\Theta\big(g^{-2}\big), \quad \Omega\left(\frac{t\cdot m\cdot\gamma}{g^4}\right)\right). \tag{A.5}$$

Proof. From Lemma A.1, we have that for any state $|\psi\rangle \in \mathbb{C}^{2^n}$ on a single copy of the system,

$$\left\| \prod_{\chi\in[g]} \prod_{i\in[m_\chi]} \left(\mathbb{I} - \Pi_i^\chi\right)|\psi\rangle \right\|^2 \leq (1 + m\gamma/g^2)^{-1}\,.$$

This can be amplified by taking tensor products since the operator norm is sub-multiplicative. That is, for any $|\psi\rangle \in (\mathbb{C}^{2^n})^{\otimes t}$ on t copies of the original system:

$$\left\| \bigotimes_{j=1}^{t} \prod_{\chi\in[g]} \prod_{i\in[m_\chi]} \left(\mathbb{I} - \Pi_i^\chi\right)_{\mathsf{reg}[j]} |\psi\rangle \right\|^2 \leq (1 + m\gamma/g^2)^{-t}\,.$$

Next, we observe that the order of the tensor product can be exchanged with that of the product over colors. As a consequence, we can introduce the operators $A_\chi = \otimes_j^t \prod_{i\in[m_\chi]} \left(\mathbb{I} - \Pi_i^\chi\right)_{\mathsf{reg}[j]}$, and re-express the operator above as

$$\bigotimes_{j=1}^{t} \prod_{\chi\in[g]} \prod_{i\in[m_\chi]} \left(\mathbb{I} - \Pi_i^\chi\right)_{\mathsf{reg}[j]} = \prod_{\chi\in[g]} A_\chi\,.$$

Each A_χ is a product of *commuting* projections; hence, A_χ itself is also a projection. We can therefore apply Lemma A.2, which implies that there must exist one layer labeled by index χ with

$$\left\| \bigotimes_{j}^{t} \prod_{i\in[m_\chi]} \left(\mathbb{I} - \Pi_i^\chi\right)_{\mathsf{reg}[j]} |\psi\rangle \right\|^2 \leq 1 - \frac{1}{4g}\left(1 - \frac{1}{(1 + m\gamma/g^2)^t}\right).$$

To conclude, we obtain that the energy of $|\psi\rangle$ under the amplified Hamiltonian is at least

$$\begin{aligned}\langle\psi|H^{(t)}|\psi\rangle &= 1 - \mathbb{E}_\chi\langle\psi|\bigotimes_{j}^{t} \prod_{i\in[m_\chi]} (\mathbb{I} - \Pi_i^\chi)|\psi\rangle \\ &\geq \frac{\min_\chi w_\chi}{4g}\cdot\left(1 - \frac{1}{(1 + m\gamma/g^2)^t}\right).\end{aligned}$$

The claimed result then follows by considering the regimes of $m\gamma t/g^2 \leq 0.1$ and ≥ 0.1 separately. ◀

Constant-Depth Circuits for Polynomial GCD over Any Characteristic

Somnath Bhattacharjee
University of Toronto, Canada

Mrinal Kumar
Tata Institute of Fundamental Research, Mumbai, India

Shanthanu S. Rai
Tata Institute of Fundamental Research, Mumbai, India

Varun Ramanathan
University of Copenhagen, Denmark

Ramprasad Saptharishi
Tata Institute of Fundamental Research, Mumbai, India

Shubhangi Saraf
University of Toronto, Canada

Abstract

We show that the GCD of two univariate polynomials can be computed by (piece-wise) algebraic circuits of constant depth and polynomial size over any sufficiently large field, regardless of the characteristic. This extends a recent result of Andrews & Wigderson who showed such an upper bound over fields of zero or large characteristic.

Our proofs are based on a recent work of Bhattacharjee, Kumar, Rai, Ramanathan, Saptharishi & Saraf that shows closure of constant depth algebraic circuits under factorization. On our way to the proof, we show that any n-variate symmetric polynomial P that has a small constant depth algebraic circuit can be written as the composition of a small constant depth algebraic circuit with elementary symmetric polynomials. This statement is a constant depth version of a result of Bläser & Jindal, who showed this for algebraic circuits of unbounded depth. As an application of our techniques, we also strengthen the closure results for factors of constant-depth circuits in the work of Bhattacharjee et al. over fields for small characteristic.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases algebraic circuits, polynomial greatest common divisor, symmetric polynomials, finite fields, constant-depth circuits

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.16

Related Version *Previous Version*: https://arxiv.org/abs/2506.23220v1
Previous Version: https://eccc.weizmann.ac.il/report/2025/085/

Funding *Somnath Bhattacharjee*: Research partially supported by an NSERC Discovery Grant.
Mrinal Kumar: Research Supported by the Department of Atomic Energy, Government of India, under project number RTI4014, and in part by research grants from Google Research, Premji Invest and SERB.
Shanthanu S. Rai: Research Supported by the Department of Atomic Energy, Government of India, under project number RTI4014, TCS Research Fellowship and in part by research grants from Google Research, Premji Invest and SERB.
Varun Ramanathan: This work was conducted primarily while Varun Ramanathan was a doctoral student at TIFR Mumbai, and was supported by the Department of Atomic Energy, Government of India, under project number RTI4014, and in part by research grants from Google Research, Premji Invest and SERB. Final revisions were completed as a postdoctoral researcher at the University of Copenhagen, Denmark, supported by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA).



© Somnath Bhattacharjee, Mrinal Kumar, Shanthanu S. Rai, Varun Ramanathan, Ramprasad Saptharishi, and Shubhangi Saraf; licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 16; pp. 16:1–16:21



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Ramprasad Saptharishi: Research Supported by the Department of Atomic Energy, Government of India, under project number RTI4014, and in part by research grants from Google Research, Premji Invest and SERB.
Shubhangi Saraf: Research partially supported by the McLean Award and an NSERC Discovery Grant.

Acknowledgements The discussions leading to this work started when a subset of the authors were at the workshop on Algebraic and Analytic Methods in Computational Complexity (Dagstuhl Seminar 24381) at Schloss Dagstuhl, and continued when they met again during the HDX & Codes workshop at ICTS-TIFR in Bengaluru. We are thankful to the organizers of these workshops and to the staff at these centers for the wonderful collaborative atmosphere that facilitated these discussions. Varun Ramanathan is grateful to Srikanth Srinivasan and Amik Raj Behera at the University of Copenhagen for helpful discussions on the complexity of symmetric polynomials.

1 Introduction

A popular theme in the research on algebraic computation is that of parallelization. A typical result of this flavor shows that a seemingly sequential computational task can also be performed by algebraic circuits of surprisingly small depth. One of the earliest results of this nature is a result of Csanky [6] which shows that the $n \times n$ symbolic determinant polynomial can be efficiently computed by algebraic circuits of depth $O(\log^2 n)$. Subsequently, a significant generalization of this result by Valiant, Skyum, Berkowitz and Rackoff [11] showed that any algebraic circuit of size and degree $\text{poly}(n)$ can be converted to a circuit of depth $O(\log^2 n)$ with a $\text{poly}(n)$ blow up in size. The decades of 1980s saw many such surprising parallelizability results for algebraic problems, and we refer an interested reader to the detailed introduction in the work of Andrews & Wigderson [1] for further pointers to this literature.

Our focus in this work is to study the parallelizability of GCD computation of two univariate polynomials. More formally, we are interested in the following problem – can the GCD of two polynomials be computed by a polynomial size constant depth algebraic circuit that takes the coefficients of the polynomials as input? The aforementioned work of Andrews & Wigderson [1] showed that over fields of characteristic zero or sufficiently large characteristic, this is indeed the case, i.e. the GCD of two univariate polynomials can be computed (piece-wise) by constant depth algebraic circuits of polynomial size. Here, the "piece-wise" part is to account for the fact that the GCD of two polynomials is not a rational function of their coefficients. We define this model of piece-wise algebraic circuits formally in Definition 19, but for this introduction, we encourage the reader to think of this as a standard algebraic circuit. In addition to the GCD, Andrews & Wigderson showed a similar constant depth upper bound for many closely related algebraic problems like the LCM, the resultant, the discriminant, the inverse of the Sylvester matrix and computing square-free decomposition of a polynomial.

The key technical fact underlying the results in [1] was that many of these algebraic problems have a relatively natural constant depth algebraic circuit if the input polynomials were given via their roots, and not their coefficients. Moreover, these circuits really compute functions that are symmetric in the roots of the inputs polynomials. Thus, if one could somehow compute these symmetric functions of roots via a constant depth algebraic circuit that takes as input the coefficients of the polynomials (and not their roots), we would be done. Over fields of zero or sufficiently large characteristic, Andrews & Wigderson showed that this is indeed the case. Their primary technical ingredient for this is the fact that over

such fields, the classical Newton identities can be implemented efficiently: there are constant-depth algebraic circuits of polynomial size that transform the first n power-sum symmetric polynomials into the n elementary symmetric polynomials, and vice versa. Unfortunately, this last fact is not true when the underlying field has small (non-zero) characteristic. In fact, over such fields, power-sum symmetric polynomials of degree $1, 2, \ldots, n$ are not algebraically independent, while the elementary symmetric polynomials are, and hence we cannot expect to compute the n elementary symmetric polynomials from the first n power-sum symmetric polynomials. While this specific issue is fixable to an extent by considering power-sum symmetric polynomials of carefully chosen degrees (e.g. over fields of characteristic 2, we take the first n power-sums of odd degree), it is unclear if the transformation from this set of power-sums to elementary symmetric polynomials can be done efficiently in constant depth. This technical issue obstructs the natural attempt at extending the results in [1] to finite fields of small characteristic and necessitates the need for some new technical ideas.

In this work, we show that the GCD of two univariates can indeed be computed by polynomial size constant depth algebraic circuits (again, piece-wise) over all fields of polynomially large size, but arbitrary characteristic. We now discuss our results in more detail, before moving on to an overview of the proofs.

1.1 Our results

▶ **Theorem 1.** *There is an absolute constant $c \in \mathbb{N}$ such that the following is true.*

For all $n \in \mathbb{N}$ and any field $\mathbb{F}$ with at least n^c elements, the gcd *of two univariates over $\mathbb{F}$ of degree at most n can be computed piece-wise as a ratio of algebraic circuits of size* $\mathrm{poly}(n)$ *and depth $O(1)$.*

A more detailed version of the above theorem statement is provided in Theorem 20. The above (piece-wise) circuit yields an efficient parallel algorithm for computing the GCD of two polynomials. More precisely, there is an algebraic PRAM algorithm with polynomially many processors and constant number of rounds that takes the coefficient vectors of two univariate polynomials as input and computes their GCD. Each processor is allowed to perform basic field operations and branching instructions (based on if a variable is 0 or not).

Our proof techniques also extend to some of the other results in [1] and show that these continue to be true over all large enough fields. Perhaps the most interesting among these is that there is a constant depth algebraic circuit of $\mathrm{poly}(n)$ size that takes as input the coefficients of two univariates of degree n and computes their resultant. From Theorem 1, we also get a piece-wise small constant-depth circuit for the LCM of two polynomials using the fact that $\mathsf{LCM}(f, g) = \frac{f \cdot g}{\gcd(f,g)}$. However, our techniques do not extend to many of the other results in [1], most notably that of computing square-free decomposition and results on arbitrary manipulation of the multiplicities of roots (Section 9 in the full version of [1]).

On our way to the proof of Theorem 1, we prove the following theorem that might be independently interesting.

▶ **Theorem 2** (Complexity of symmetric polynomials)**.** *Let $n, d, s \in \mathbb{N}$ be parameters, and let $\mathbb{F}$ be a sufficiently large field, i.e. $|\mathbb{F}| \geq (nds)^c$ for an absolute constant $c > 0$. Let $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a symmetric polynomial on n variables of degree d and let $Q(\mathbf{z}) \in \mathbb{F}[\mathbf{z}]$ be the unique*[1] *degree d polynomial such that $P(\mathbf{x}) = Q(\mathrm{Esym}_1(\mathbf{x}), \ldots, \mathrm{Esym}_n(\mathbf{x}))$. Then, the following are true:*

[1] The existence and uniqueness of such a Q follows from the Fundamental Theorem of Symmetric Polynomials (see Theorem 11).

- *If P is computable by a circuit of size s and depth Δ, then Q can be computed by a circuit of size* $\mathrm{poly}(s, d, n)$ *and depth* $\Delta + O(1)$.
- *If P is computable by an algebraic formula of size s, then Q can be computed by a formula of size* $\mathrm{poly}(s, d, n)$.

In [3], Bläser & Jindal showed that if an n-variate symmetric polynomial P can be computed by an algebraic circuit of size s and degree d, then it can be expressed as the composition of a circuit Q of size and degree $\mathrm{poly}(s, d, n)$ with elementary symmetric polynomials. However, if P is assumed to be computable by a constant-depth circuit or a formula, it is unclear from the proof in [3] whether Q also has a small constant-depth circuit or a formula respectively. Thus, Theorem 2 is a version of the result of [3] for constant depth algebraic circuits and formulas. Showing such a statement for constant-depth circuits was mentioned as an open problem in [1] and Theorem 2 answers this question affirmatively. A partial generalization of [3] to formulas and constant-depth formulas was shown in [4, Lemma 1.4] in the context of understanding the complexity of Schur polynomials.

Our proofs are based on techniques that were recently used to show the closure of constant-depth circuits and formulas under factorization in [2] over fields of zero or large characteristic. These closure results, in turn, are a consequence of a classical theorem of Furstenberg [9] that gives a computationally simple and explicit expression for the power series roots of a bivariate polynomial (see Theorem 8 and Theorem 25).

Our proof techniques have another consequence of closure under factorization over fields of small characteristic. In [2], while the complete closure result was over fields of zero or large characteristic, a weaker statement was shown to be over fields of small characteristic. More specifically, it was shown (Theorem 1.3 in [2]) that if an n variate degree d polynomial f has a constant-depth circuit of size s over the field $\mathbb{F}_{p^k}$ (for a prime p) and g is a factor of f, then for some $i \in \mathbb{N}$, g^{p^i} has a constant-depth circuit of size $\mathrm{poly}(s, n, d)$ over the algebraic closure (or a very high degree extension) of $\mathbb{F}_{p^k}$. Using the techniques here, we show that the circuit for g^{p^i} is in fact over the field $\mathbb{F}_{p^k}$ itself, provided that this field is polynomially large. We refer to Theorem 22 for a formal statement of this result.

1.2 Proof overview

Proof of Theorem 2

We start with a discussion of the proof of Theorem 2. To this end, we begin with an outline of the proof in [3], who originally showed the version of Theorem 2 for general algebraic circuits. Here the authors consider the polynomial

$$F(\mathbf{z}, y) = y^n - z_1 y^{n-1} + \cdots + (-1)^n z_n - 1,$$

and think of the polynomial F as a univariate in y with coefficients from the ring $\mathbb{F}[\mathbf{z}]$. Since $F(\mathbf{0}, y) = y^n - 1$ has n distinct roots, namely the n^{th} roots of unity, we get by an application of Newton Iteration that each of these roots of $y^n - 1$ can be *lifted* to a unique power series (in $\mathbf{z}$) root of $F(\mathbf{z}, y)$. Let these power series roots be $A_1(\mathbf{z}), A_2(\mathbf{z}), \ldots, A_n(\mathbf{z})$. If $P(\mathbf{x})$ is a symmetric polynomial, then there is a unique n variate polynomial Q of degree at most $\deg(P)$ such that $P(\mathbf{x}) = Q(\mathrm{Esym}_1(\mathbf{x}), \ldots, \mathrm{Esym}_n(\mathbf{x}))$, where $\mathrm{Esym}_i(\mathbf{x})$ is the elementary symmetric polynomial of degree i in $\mathbf{x}$. Let us consider the substitution where the variable x_i is replaced by the power series $A_i(\mathbf{z})$. Note that $\mathrm{Esym}_j(A_1, \ldots, A_n)$ must equal z_j for every $j \in \{1, 2, \ldots, n-1\}$ and equals $z_n + (-1)^{n-1}$ for $j = n$. So, we get

$$P(A_1(\mathbf{z}), \ldots, A_n(\mathbf{z})) = Q(z_1, \ldots, z_{n-1}, z_n + (-1)^{n-1}).$$

We now note that since Q is a polynomial, it suffices to compute the power series $A_i(\mathbf{z})$ modulo $\langle \mathbf{z} \rangle^{\deg(Q)+1}$. Using the fact that if F has a small algebraic circuit, all its truncated power series roots have small circuits, and with a few small modifications, we get a small unbounded depth circuit for Q.

Given this outline, to prove Theorem 2, it would suffice to show that the truncated power series roots of $F(\mathbf{z}, y)$ have small constant-depth circuits, and we do exactly this. Note that F itself has a small constant-depth circuit. We now invoke the closure results for factors of constant-depth circuits shown in [2] (Theorem 4.1) to show that the truncations of the power series A_i can be computed by small constant-depth circuits. This fact together with the above outline gives us Theorem 2.

We note that even though the main closure results in [2] are for fields of zero or large characteristic, the instance we have at hand here is very special – we are trying to compute the power series roots of a polynomial that has no repeated roots. For such instances, essentially the arguments from fields of characteristic zero carry over as is to fields of small positive characteristic. This completes an outline of the proof of Theorem 2.

Computing the resultant in constant depth

To prove Theorem 1, we first observe that Theorem 2 can be generalized for polynomials that are bi-symmetric. More precisely, we observe that if a $2n$ variate polynomial $P(\mathbf{x}, \mathbf{y})$ is symmetric in $\mathbf{x}$ and in $\mathbf{y}$ (although it need not be symmetric in $\mathbf{x} \cup \mathbf{y}$ together), then P equals $Q(\mathrm{Esym}_1(\mathbf{x}), \ldots, \mathrm{Esym}_n(\mathbf{x}), \mathrm{Esym}_1(\mathbf{y}), \ldots, \mathrm{Esym}_n(\mathbf{y}))$ for a unique polynomial Q. We then show that if P has a small constant-depth circuit, then so does Q.

An almost immediate (and independently) interesting consequence of the above discussion is that the resultant of two monic univariates can be computed by a constant-depth circuit that takes the coefficients of the polynomials as inputs. If $f(x) = f_0 + f_1 x + \cdots + x^n$ and $g(x) = g_0 + g_1 x + \cdots + x^n$ are two univariates with roots denoted by $\{\alpha_1, \ldots, \alpha_n\}$ and $\{\beta_1, \ldots, \beta_n\}$ respectively, then their resultant R equals $\prod_{i,j \in [n]} (\alpha_i - \beta_j)$. We note that R is a polynomial that is symmetric in the roots of f and the roots of g, and has a small circuit of constant depth that takes these roots as inputs. Thus, from the generalization of Theorem 2 to bisymmetric polynomials, we get that there is a polynomial Q that can be computed by a small constant-depth circuit, such that Q composed with the elementary symmetric polynomials of $\{\alpha_1, \ldots, \alpha_n\}$ (i.e. the coefficients of f) and the elementary symmetric polynomials of $\{\beta_1, \ldots, \beta_n\}$ (i.e. the coefficients of g) equals R. Thus, we have a small constant-depth circuit for the resultant of two univariates that takes the coefficients as input.

Computing the GCD

It would be helpful to revisit the proof strategy of Andrews and Wigderson [1]. If the polynomials f and g are square-free, then the $\gcd(f, g)$ can be obtained by "filtering" from f only those roots of f that are also roots of g. One of the key aspects of [1] was to execute this "filtering" operation using the efficient transformation (via constant-depth circuits) between elementary and power-sum symmetric polynomials of the roots of f and g. In the more general setting when f and g are not square-free, using some additional ideas, [1] reduce to the square-free setting by first computing the *square-free decomposition*[2] and working with each part in the decomposition.

[2] i.e., computing $f_1, \ldots, f_d$ that are square-free and co-prime such that $f = \prod f_i^i$

In the setting of characteristic $p > 0$, we have to address two issues – (a) we do not have efficient transformations between power-sum and elementary symmetric polynomials, and (b) computing the square-free part of f is a more general operation than computing p^{th}-roots i.e. computing an efficient circuit for a polynomial f using an efficient circuit for f^p (something that we still do not know how to do, and would be very interesting). We give an alternate implementation of the filtering operation using Theorem 2 and also have a simple observation that allows us to completely bypass the calculation of the square-free part. We briefly elaborate on the key steps below.

- **An auxiliary polynomial containing $\gcd(f, g)$:** Our first observation is that, if we consider a *generic* linear combination of f and g given by $F(y, z) := f(y) + z \cdot g(y)$, then each root α of $\gcd(f, g)$ (from $\overline{\mathbb{F}}$) appears as a linear factor $(y - \alpha)$ of F with the *right* multiplicity. Furthermore, there is no other $\alpha \in \overline{\mathbb{F}}$ satisfying $F(\alpha, z) = 0$. Thus, if we could somehow *filter* the y-roots of $F(y, z)$ with their multiplicity, we will obtain $\gcd(f, g)$. This thereby avoids computing square-free components of f and g entirely.
- **Filtering roots of one polynomial using another, à la [1]:** Suppose $\{\sigma_1, \ldots, \sigma_m\}$ is the multi-set of roots of f over $\overline{\mathbb{F}}$ and $g(y) \in \mathbb{F}[y]$ (where $\mathbb{F}$ could be the field of rational functions over other variables), [1] define a "filter" operation to compute

$$\text{Filter}(f \mid \text{non-roots}(g)) = \prod_{i \in [m] \,:\, g(\sigma_i) \neq 0} (y - \sigma_i),$$

$$\text{Filter}(f \mid \text{roots}(g)) = \prod_{i \in [m] \,:\, g(\sigma_i) = 0} (y - \sigma_i) = \frac{f(y)}{\text{Filter}(f \mid \text{non-roots}(g))}.$$

  [1] show that the above can essentially be computed piece-wise via elementary symmetric polynomials of $\{g(\sigma_1), \ldots, g(\sigma_m)\}$ with the advice parameter being the degree of the filter. Rather than proceeding via power-sum symmetric polynomials (which requires large characteristic fields), we once again use Theorem 2 to express this as a constant-depth circuit over the coefficients of f and g. Theorem 1 now follows by interpreting $F(y, z)$ as a univariate in y over the larger field $\mathbb{F}(z)$, and noticing that

$$\gcd(f(y), g(y)) = \text{Filter}(F(y, z) \mid \text{roots}(g)) = \frac{F(y, z)}{\text{Filter}(F(y, z) \mid \text{non-roots}(g))}.$$

Organization of the paper

The rest of the paper is organized as follows. We start with some general notations and preliminaries in Section 2, followed by the proof of Theorem 2 in Section 3. In Section 4, we show that the resultant and GCD can be computed by constant-depth circuits and in Section 5, we use these techniques to strengthen the closure results for constant-depth circuits under factorization.

2 Preliminaries

Notation

- We use boldface letters such as $\mathbf{x}$ to refer to an ordered tuple of variables
- We use $\mathbb{F}, \mathbb{K}$ etc., to refer to fields, and use $\overline{\mathbb{F}}$ to refer to the algebraic closure of $\mathbb{F}$. Also, $\mathbb{F}[x]$ refers to the polynomial ring, $\mathbb{F}(x)$ to the field of fractions, $\mathbb{F}[\![x]\!]$ to the ring of formal power series, and $\mathbb{F}(\!(x)\!)$ refer to ring of Laurent series with respect to the variable x with coefficients from the field $\mathbb{F}$. We will typically use $\mathbb{K}$ to denote the field of fractions $\mathbb{F}(\mathbf{x})$ for an underlying field $\mathbb{F}$ and some tuple of variables $\mathbf{x}$.

- Given a univariate polynomial $f(x) \in \mathbb{F}[x]$ (or more generally in the ring of formal Laurent series $\mathbb{F}(\!(x)\!)$), we denote by $[x^n]\{f\}$ the coefficient of the monomial x^n in f. In the case of a multivariate polynomial, such as $f(x, y)$, we interpret f as an element of $\mathbb{F}[y][x]$ – that is, as a univariate polynomial in x with coefficients in $\mathbb{F}[y]$ – and $[x^n]\{f\}$ then refers to the x^n coefficient viewed as a polynomial in y.
- The notation $\mathrm{Hom}_d(F)$ refers to the degree d homogeneous part of F, and $\mathrm{Hom}_{\leq d}(F)$ refers to the sum of all homogeneous parts of F up to degree d. For any subset $\mathbf{x}_S \subseteq \mathbf{x}$ and any $i \in [d]$, the degree i homogeneous part of F with respect to $\mathbf{x}_S$, denoted by $\mathrm{Hom}_{\mathbf{x}_S,i}(F)$.
- For polynomials (or more generally, power series) $f(\mathbf{x})$ and $g(\mathbf{x})$, and $t \in \mathbb{N}$, we write $f(\mathbf{x}) \equiv g(\mathbf{x}) \bmod \langle \mathbf{x} \rangle^t$ to denote that $\mathrm{Hom}_{\leq t-1}(f) \equiv \mathrm{Hom}_{d \leq t-1}(g)$.
- The i^{th} elementary symmetric polynomial is sum of all multilinear monomials in $\mathbf{x}$ of degree i. Formally, the i^{th} elementary polynomial is $\mathrm{Esym}_i(\mathbf{x}) \coloneqq \sum_{S \subseteq [n]:|S|=i} \prod_{i \in S} x_i$.

2.1 Interpolation

The following uses of polynomial interpolation in the context of algebraic circuits are due to Michael Ben-Or.

▶ **Lemma 3** (Interpolation). *Let $\mathbb{F}$ be a subfield of a commutative ring R, and suppose that $\mathbb{F}$ contains at least $d+1$ distinct elements. Choose distinct points $\alpha_0, \ldots, \alpha_d \in \mathbb{F}$. Then, for any polynomial $f(t) = f_0 + f_1 t + \cdots + f_d t^d \in R[t]$ of degree at most d, and for each index $i \in \{0, \ldots, d\}$, there exist elements $\beta_{i0}, \ldots, \beta_{id} \in \mathbb{F}$ (depending only on i and the α_j's) such that the coefficient f_i can be recovered from the values of f at the interpolation points:*

$$f_i = \beta_{i0} f(\alpha_0) + \beta_{i1} f(\alpha_1) + \cdots + \beta_{id} f(\alpha_d).$$

▶ **Corollary 4** (Applications of interpolation). *Let $\alpha_0, \ldots, \alpha_d$ be distinct elements of the field $\mathbb{F}$. The following statements are immediate consequences of interpolation:*

1. ***[Extracting homogeneous parts]*** *Let $C(\mathbf{x})$ be a polynomial of total degree d, and let $\mathbf{x}_S \subseteq \mathbf{x}$ be a subset of the variables. For any $i \in \{0, \ldots, d\}$, the degree i homogeneous component of C with respect to the variables in $\mathbf{x}_S$, denoted $\mathrm{Hom}_{\mathbf{x}_S,i}(C)$, can be written as*

$$\mathrm{Hom}_{\mathbf{x}_S,i}(C) = \sum_{j=0}^{d} \beta_{i,j} \cdot C(\alpha_j \cdot \mathbf{x}_S, \mathbf{x}_{\overline{S}})$$

   for some scalars $\beta_{i,j} \in \mathbb{F}$ that are independent of the polynomial C.
2. ***[Derivatives via evaluation]*** *Suppose $C(\mathbf{x}, y)$ is a polynomial with degree d in the variable y. Then the i^{th} partial derivative of C with respect to y can be expressed as an $\mathbb{F}[y]$-linear combination of the evaluations $C(\mathbf{x}, \alpha_j)$ for $j = 0, \ldots, d$. Specifically, there exist polynomials $\mu_0(y), \ldots, \mu_d(y) \in \mathbb{F}[y]$, each of degree at most d and independent of C, such that*

$$\partial_{y^i} C(\mathbf{x}, y) = \sum_{j=0}^{d} \mu_j(y) \cdot C(\mathbf{x}, \alpha_j).$$

Moreover, if C is computed by a circuit of size s and depth Δ, then both operations – extracting homogeneous components and computing partial derivatives – can be implemented by circuits of size $\mathrm{poly}(s, d)$ and depth at most $\Delta + O(1)$.

2.2 Resultants

▶ **Definition 5** (Sylvester Matrix and Resultant (for details, see [12, Chapter 6])). *Let $\mathbb{F}$ be a field. Let $P(z)$ and $Q(z)$ be polynomials of degree $a \geq 1$ and $b \geq 1$ in $\mathbb{F}[z]$. Define a linear map $\Gamma_{P,Q} : \mathbb{F}^a \times \mathbb{F}^b \to \mathbb{F}^{a+b}$ that takes polynomials $A(z)$ and $B(z)$ in $\mathbb{F}[z]$ of degree $a-1$ and $b-1$ respectively, and maps them to $AP + BQ$, a polynomial of degree $a+b-1$.*

The Sylvester matrix *of P and Q, denoted by $\mathrm{Syl}_z(P,Q)$, is defined to be the $(a+b)\times(a+b)$ matrix for the linear map $\Gamma_{P,Q}$.*

The Resultant *of P and Q, denoted by $\mathrm{Res}_z(P,Q)$, is the determinant of $\mathrm{Syl}_z(P,Q)$.*

▶ **Fact 6** (see, e.g., Chapter 3 of [5]). *Suppose $f(z), g(z) \in \mathbb{F}[z]$ are monic polynomials, and the multiset of their roots over $\overline{\mathbb{F}}$ are $\alpha_1, \ldots, \alpha_a$ and $\beta_1, \ldots, \beta_b$. Then,*

$$\mathrm{Res}_z(f,g) = \prod_{i=1}^{a}\prod_{j=1}^{b}(\alpha_i - \beta_j).$$

2.3 Factorization of monic polynomials into power series

▶ **Lemma 7** (Factorization into power series). *Let $f(t,y) \in \mathbb{K}[t,y]$ be a polynomial that is monic in y such that $f(0,y)$ is square-free. For each $\alpha \in \overline{\mathbb{K}}$ (the algebraic closure) such that $f(0,\alpha) = 0$, there is a unique power series $\varphi_\alpha(t) \in \overline{\mathbb{K}}[\![t]\!]$ satisfying $\varphi_\alpha(0) = \alpha$ such that $f(t, \varphi_\alpha(t)) = 0$.*

In fact, the polynomial $f(t,y)$ factorizes in $\overline{\mathbb{K}}[\![t]\!][y]$ as

$$f(t,y) = \prod_{\alpha \in Z}(y - \varphi_\alpha(t))$$

where Z is the set of roots of $f(0,y)$ in $\overline{\mathbb{K}}$.

The above lemma is essentially folklore and [7, Section 3] gives a formal proof of the above. We also note that the lemma can be applied in the multivariate setting by taking a multivariate polynomial in variables $x_1, x_2, \ldots, x_n, y$, replacing each x_i by $x_i \cdot t$ (for a new indeterminate t) and viewing the resulting polynomial as a bivariate in t and y with coefficients in the field $\mathbb{F}(x_1, x_2, \ldots, x_n)$.

2.4 Furstenberg's theorem for roots of multiplicity one

We define the following *diagonal* operator for bivariate power series $F(x,y) = \sum_{i,j} F_{i,j} x^i y^j \in \mathbb{F}[\![x,y]\!]$ as follows:

$$\mathcal{D}(F)(t) = \sum_{i \geq 0} F_{i,i} \cdot t^i.$$

▶ **Theorem 8** ([9, Proposition 2]). *Let $P(t,y) \in \mathbb{F}[\![t,y]\!]$ be a power series and $\varphi(t) \in \mathbb{F}[\![t]\!]$ be a power series satisfying $P(t,y) = (y - \varphi(t)) \cdot Q(t,y)$. If $\varphi(0) = 0$ and $Q(0,0) \neq 0$, then*

$$\varphi = \mathcal{D}\left(\frac{y^2 \cdot \partial_y P(ty,y)}{P(ty,y)}\right) \tag{1}$$

Furstenberg [9] states the theorem for $P(t,y)$ being a polynomial and that will be sufficient for our purposes; the proof works even when $P(t,y)$ is a power series. The diagonal expression above can be simplified to a slightly more convenient expression for implicitly defined power series roots.

▶ **Corollary 9.** *Let $P(t, y) \in \mathbb{F}[\![t, y]\!]$ and $\varphi(t) \in \mathbb{F}[\![t]\!]$ such that $\varphi(0) = 0$, $P(t, \varphi(t)) = 0$ and $\partial_y P(0, 0) = \alpha \neq 0$. Then,*

$$\varphi(t) = \sum_{m \geq 1} \frac{1}{\alpha^{m+1}} \cdot [y^{m-1}] \left\{ \partial_y P(t, y) \cdot (\alpha y - P(t, y))^m \right\}.$$

Suppose $P(\mathbf{x}, y)$ is computable by a small constant-depth circuit. Using Corollary 9, [2] showed that truncations of power series roots (of multiplicity 1) of $P(\mathbf{x}, y)$ also have small constant-depth circuits.

▶ **Theorem 10** (Power series roots without multiplicity [2]). *Let $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be a polynomial computed by a circuit C, and let $\varphi(\mathbf{x}) \in \mathbb{F}[\![\mathbf{x}]\!]$ be a power series satisfying $\varphi(\mathbf{0}) = 0$, $P(\mathbf{x}, \varphi(\mathbf{x})) = 0$ and $\partial_y P(\mathbf{0}, 0) \neq 0$. Then, for any $d \in \mathbb{N}$, there is a circuit C' computing $\mathrm{Hom}_{\leq d}[\varphi]$ such that*

$$\mathrm{size}(C') \leq \mathrm{poly}(d, \mathrm{size}(C))$$

$$\mathrm{depth}(C') \leq \mathrm{depth}(C) + O(1)$$

3 Complexity of Symmetric Polynomials

A polynomial $P(x_1, \ldots, x_n)$ is symmetric if for every permutation σ on n elements, $P(x_1, \ldots, x_n) = P(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$. The elementary symmetric polynomials $\{\mathrm{Esym}_i(\mathbf{x})\}_{i \in [n]}$ are ubiquitous examples of symmetric polynomials. Surprisingly, the *fundamental theorem of symmetric polynomials* states that every symmetric polynomial P can be uniquely expressed as polynomial expressions in $\{\mathrm{Esym}_i(\mathbf{x})\}_{i \in [n]}$.

▶ **Theorem 11** (Fundamental theorem of symmetric polynomials). *Let $P(\mathbf{x})$ be a symmetric polynomial on n variables of degree d. Then there is a unique polynomial $Q(y_1, \ldots, y_n)$ such that $P(\mathbf{x}) = Q(\mathrm{Esym}_1(\mathbf{x}), \mathrm{Esym}_2(\mathbf{x}), \ldots, \mathrm{Esym}_n(\mathbf{x}))$. Moreover, the degree of Q is at most d.*

Given this fact, it is natural to ask if complexity of the polynomial Q is related to the complexity of P. Bläser and Jindal [3] showed that if P is computable by a small algebraic circuit, then Q is also computable by a small algebraic circuit.

▶ **Theorem 12** (Theorem 4 in [3]). *Let $P(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$ be a symmetric polynomial on n variables of degree d computed by a circuit C of size s. Let $Q(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$ be the unique polynomial such that $P(\mathbf{x}) = Q(\mathrm{Esym}_1(\mathbf{x}), \ldots, \mathrm{Esym}_n(\mathbf{x}))$. Then, $Q(\mathbf{x})$ is computable by a circuit of size $\mathrm{poly}(s, d, n)$ over $\mathbb{C}$.*

One of the main steps in their proof is to compute the truncations of power series roots using Newton Iteration. Based on the ideas from [2], we would expect that replacing Newton Iteration with Furstenberg's theorem (Theorem 8) should give us an analogous result over constant-depth circuits. Indeed, this turns out to be the case. We now prove the constant-depth version of their result.

▶ **Theorem 13** (Complexity of symmetric polynomials). *Let $n, d, s \in \mathbb{N}$ be parameters, and let $\mathbb{F}$ be a polynomially large field, i.e. $|\mathbb{F}| \geq (nds)^c$ for an absolute constant $c > 0$. Let $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a symmetric polynomial on n variables of degree d computed by a circuit C of size s and depth Δ. If $Q(\mathbf{z}) \in \mathbb{F}[\mathbf{z}]$ is the unique degree d polynomial such that $P(\mathbf{x}) = Q(\mathrm{Esym}_1(\mathbf{x}), \ldots, \mathrm{Esym}_n(\mathbf{x}))$, then $Q(\mathbf{z})$ is computable by a circuit of size $\mathrm{poly}(s, d, n)$ and depth $\Delta + O(1)$ over $\mathbb{F}$.*

Proof. Let $\alpha_1, \ldots, \alpha_n$ be n distinct elements from $\mathbb{F}$ and β_i denote $\mathrm{Esym}_i(\alpha_1, \ldots, \alpha_n)$ for each $i \in [n]$. Define $F(\mathbf{z}, y)$ as:

$$F(\mathbf{z}, y) := y^n - (z_1 + \beta_1)y^{n-1} + \cdots + (-1)^{n-1}(z_{n-1} + \beta_{n-1})y + (-1)^n(z_n + \beta_n).$$

$F[\mathbf{z}, y]$ has a circuit of size $O(n^2)$ and depth 2. Note that $F(\mathbf{0}, y) = y^n + \sum_{i=1}^{n} (-1)^i y^{n-i}\beta_i = \prod_{i\in[n]}(y - \alpha_i)$. Since $\alpha_1, \ldots, \alpha_n$ are distinct, we have that $F(\mathbf{0}, y)$ is square-free and hence (by Lemma 7) $F(\mathbf{z}, y)$ factorizes as

$$F(\mathbf{z}, y) = \prod_{i=1}^{n}(y - A_i(\mathbf{z}))$$

where $A_i(\mathbf{z}) \in \mathbb{F}[\![\mathbf{z}]\!]$ with $A_i(\mathbf{0}) = \alpha_i$ for each $i \in [n]$. Furthermore, $\mathrm{Esym}_k(A_1(\mathbf{z}), \ldots, A_n(\mathbf{z})) = z_k + \beta_k$.

We now apply Theorem 10 on $F(\mathbf{z}, y)$ to get size $\mathrm{poly}(n)$ and depth $O(1)$ circuits $\tilde{A}_1, \ldots, \tilde{A}_n$ such that $\tilde{A}_i(\mathbf{z}) = A_i(\mathbf{z}) \bmod \langle \mathbf{z} \rangle^{d+1}$ for each $i \in [n]$. This implies that for each $i \in [n]$,

$$\mathrm{Esym}_i(\tilde{A}_1, \ldots, \tilde{A}_n) = z_i + \beta_i \bmod \langle \mathbf{z} \rangle^{d+1}.$$
$$\therefore \quad P(\tilde{A}_1, \ldots, \tilde{A}_n) = P(A_1, \ldots, A_n) \bmod \langle \mathbf{z} \rangle^{d+1}$$

Since $P(\mathbf{x}) = Q(\mathrm{Esym}_1(\mathbf{x}), \ldots, \mathrm{Esym}_n(\mathbf{x}))$,

$$\begin{aligned} P(A_1, \ldots, A_n) &= Q(\mathrm{Esym}_1(A_1, \ldots, A_n), \ldots, \mathrm{Esym}_n(A_1, \ldots, A_n)) \\ &= Q(z_1 + \beta_1, \ldots, z_n + \beta_n) \end{aligned}$$

where degree of Q is at most d from Theorem 11. Therefore,

$$Q(z_1 + \beta_1, \ldots, z_n + \beta_n) = P(A_1, \ldots, A_n) = \mathrm{Hom}_{\le d}\, P(\tilde{A}_1, \ldots, \tilde{A}_n)$$

Recall that each $\tilde{A}_i$ is computable by a size $\mathrm{poly}(n)$ and depth $O(1)$ circuit, and P is computable by a size $\mathrm{poly}(s, d, n)$ and depth Δ circuit. Thus, we obtain a $\mathrm{poly}(s, d, n)$ and depth $\Delta + O(1)$ circuit for $Q(\mathbf{z} + \boldsymbol{\beta})$, and thus for $Q(\mathbf{z})$ as well (by shifting $\mathbf{z} \leftarrow \mathbf{z} - \boldsymbol{\beta}$). ◀

3.1 Multi-symmetric polynomials

▶ **Definition 14** (Multi-symmetric polynomials). *Let $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_k$ be a partition of the set of variables $\mathbf{x}$. A polynomial $f(\mathbf{x}) = f(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ is said to be* multi-symmetric *with respect to the above partition if for all $i \in [k]$ we have that $f(\mathbf{x})$ is symmetric with respect to the part $\mathbf{x}_i$. In other words, for every choice of permutations $\pi_1, \ldots, \pi_k$ on $\mathbf{x}_1, \ldots, \mathbf{x}_k$, we have $f(\pi_1(\mathbf{x}_1), \ldots, \pi_k(\mathbf{x}_k)) = f(\mathbf{x}_1, \ldots, \mathbf{x}_k)$.*

The following is a natural extension of Theorem 11. For purely the ease of exposition, we assume the size of each part in the partition is the same; the proof would readily extend to the general case with mere notational changes.

▶ **Theorem 15** (Fundamental theorem for multi-symmetric polynomials). *Let $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_k$ be a partition of the variables with each $|\mathbf{x}_i| = n$, and let $P(\mathbf{x})$ be a multi-symmetric polynomial with respect to the above partition of degree d. Then there is a unique polynomial $Q(\mathbf{y}_1, \ldots, \mathbf{y}_k)$ (where $\mathbf{y}_i = (y_{i1}, \ldots, y_{in})$) such that $P(\mathbf{x}) = Q(\overline{\mathrm{Esym}}(\mathbf{x}_1), \ldots, \overline{\mathrm{Esym}}(\mathbf{x}_k))$ where*

$$\overline{\mathrm{Esym}}(\mathbf{x}_i) \coloneqq (\mathrm{Esym}_1(\mathbf{x}_i), \ldots, \mathrm{Esym}_n(\mathbf{x}_i)).$$

Moreover, the degree of Q is at most d.

Proof. The proof is just a simple induction on k. For $k = 1$, the claim follows from Theorem 11.

Assuming that the theorem is true for $k-1$, and write $P(\mathbf{x})$ as

$$P(\mathbf{x}) = \sum_{\mathbf{e}\geq 0} [\mathbf{x}_k^{\mathbf{e}}]\{P\} \cdot \mathbf{x}_k^{\mathbf{e}}.$$

We will say $\mathbf{e}_i \sim \mathbf{e}_j$ if there is some permutation π on $|\mathbf{x}_k|$ elements such that $\pi(\mathbf{e}_i) = \mathbf{e}_j$; let $\mathcal{E}$ be a distinct set of representatives of the equivalence classes defined by $\sim$. Note that if $\mathbf{e}_i \sim \mathbf{e}_j$, then $[\mathbf{x}_k^{\mathbf{e}_i}]\{P\} = [\mathbf{x}_k^{\mathbf{e}_j}]\{P\}$. Therefore, if we define $H_{\mathbf{e}^*}(\mathbf{x}_k)$ for an $\mathbf{e}^* \in \mathcal{E}$ as

$$H_{\mathbf{e}^*}(\mathbf{x}_k) \coloneqq \sum_{\mathbf{e}' \,:\, \mathbf{e}' \sim \mathbf{e}^*} \mathbf{x}_k^{\mathbf{e}'}$$

then,

$$\begin{aligned} P(\mathbf{x}) &= \sum_{\mathbf{e}^* \in \mathcal{E}} [\mathbf{x}_k^{\mathbf{e}^*}]\{P\} \cdot H_{\mathbf{e}^*}(\mathbf{x}_k) \\ &= \sum_{\mathbf{e}^* \in \mathcal{E}} [\mathbf{x}_k^{\mathbf{e}^*}]\{P\} \cdot \tilde{Q}_{\mathbf{e}^*}(\overline{\mathrm{Esym}}(\mathbf{x}_k)) \qquad \text{(by Theorem 11)} \end{aligned}$$

for some $\tilde{Q}_{\mathbf{e}^*}$ of degree at most $|\mathbf{e}^*|$. Since P is multi-symmetric, we have that $[\mathbf{x}_k^{\mathbf{e}}]\{P\}$ is multi-symmetric with respect to $\mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_{k-1}$. By induction, there exists polynomials $Q_{\mathbf{e}}$ of degree at most $d - |\mathbf{e}|$ such that

$$\begin{aligned} &[\mathbf{x}_k^{\mathbf{e}}]\{P\} = Q_{\mathbf{e}}(\overline{\mathrm{Esym}}(\mathbf{x}_1), \ldots, \overline{\mathrm{Esym}}(\mathbf{x}_{k-1})). \\ \implies &P(\mathbf{x}) = \sum_{\mathbf{e}^* \in \mathcal{E}} Q_{\mathbf{e}^*}(\overline{\mathrm{Esym}}(\mathbf{x}_1), \ldots, \overline{\mathrm{Esym}}(\mathbf{x}_{k-1})) \cdot \tilde{Q}_{\mathbf{e}^*}(\overline{\mathrm{Esym}}(\mathbf{x}_k)) \\ &\qquad =: Q(\overline{\mathrm{Esym}}(\mathbf{x}_1), \ldots, \overline{\mathrm{Esym}}(\mathbf{x}_k)) \end{aligned}$$

for some $Q(y_{11}, \ldots, y_{kn})$ of degree at most d. ◀

▶ **Theorem 16** (Complexity of multi-symmetric polynomials). *Let $\mathbb{F}$ be a polynomially large field with respect to parameters $n, d, s \in \mathbb{N}$, i.e. $|\mathbb{F}| \geq (nds)^c$ for an absolute constant $c > 0$. Let $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_k$ be a partition of the variables, with each $|\mathbf{x}_i| = n$, and let $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a multi-symmetric polynomial with respect to the partition of degree d computed by a circuit C of size s and depth Δ. If $Q(z_{11}, \ldots, z_{kn}) \in \mathbb{F}[\mathbf{z}]$ is the unique degree d polynomial such that*

$$P(\mathbf{x}) = Q(\overline{\mathrm{Esym}}(\mathbf{x}_1), \ldots, \overline{\mathrm{Esym}}(\mathbf{x}_k))$$

where $\overline{\mathrm{Esym}}(\mathbf{x}_i) \coloneqq (\mathrm{Esym}_1(\mathbf{x}_i), \ldots, \mathrm{Esym}_n(\mathbf{x}_i))$, then $Q(\mathbf{z})$ is computable by a circuit of size $\mathrm{poly}(s, d, n)$ and depth $\Delta + O(1)$ over $\mathbb{F}$.

Proof. The proof is almost a direct extension of Theorem 13. As in the proof of Theorem 13, for each $i \in [k]$ define the polynomials for $\mathbf{z}_i = (z_{i,1}, \ldots, z_{i,n})$:

$$F_i(\mathbf{z}_i, y) \coloneqq y^n - (z_{i,1} + \beta_1)y^{n-1} + \cdots + (-1)^{n-1}(z_{i,n-1} + \beta_{n-1})y + (-1)^n(z_{i,n} + \beta_n)$$

where $\beta_i = \mathrm{Esym}_i(\alpha_1, \ldots, \alpha_n)$ for some distinct choice of $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$. By Lemma 7, since $F_i(\mathbf{0}, y) = \prod_{i=1}^n (y - \alpha_i)$ is square-free, we have that $F_i(\mathbf{z}_i, y)$ factors

$$F_i(\mathbf{z}_i, y) = \prod_{j=1}^{n} (y - A_{ij}(\mathbf{z}_i))$$

where $A_{ij}(\mathbf{z}_i) \in \mathbb{F}[\![\mathbf{z}_i]\!]$ and $A_{ij}(\mathbf{0}) = \alpha_j$. Note that $\mathrm{Esym}_r(A_{i1}, \ldots, A_{in}) = z_{i,r} + \beta_r$. By Theorem 10, we have circuits $\tilde{A}_{ij}$ of size $\mathrm{poly}(n)$ and depth $O(1)$ such that

$$\tilde{A}_{ij}(\mathbf{z}_i) = A_{ij}(\mathbf{z}_i) \bmod \langle \mathbf{z}_i \rangle^{d+1}.$$

From Theorem 16, degree of Q is at most d. Thus,

$$\begin{aligned} P(\tilde{A}_{11}(\mathbf{z}_1), \ldots, \tilde{A}_{kn}(\mathbf{z}_k)) &= P(A_{11}(\mathbf{z}_1), \ldots, A_{kn}(\mathbf{z}_k)) \bmod \langle \mathbf{z} \rangle^{d+1} \\ &= Q(\overline{\mathrm{Esym}}(A_{11}, \ldots, A_{1n}), \ldots, \overline{\mathrm{Esym}}(A_{k1}, \ldots, A_{kn})) \\ &= Q(\mathbf{z}_1 + \boldsymbol{\beta}, \ldots, \mathbf{z}_k + \boldsymbol{\beta}) \\ \implies Q(\mathbf{z}_1 + \boldsymbol{\beta}, \ldots, \mathbf{z}_k + \boldsymbol{\beta}) &= \mathrm{Hom}_{\leq d}\, P(\tilde{A}_{11}(\mathbf{z}_1), \ldots, \tilde{A}_{kn}(\mathbf{z}_k)) \end{aligned}$$

By Corollary 4, the RHS is computable by a $\mathrm{poly}(n, d, s)$-size depth $\Delta + O(1)$ circuit, and by un-translating by $\boldsymbol{\beta}$ we also obtain a such a circuit for $Q(\mathbf{z}_1, \ldots, \mathbf{z}_k)$. ◀

Note that the additive $O(1)$ increase in depth does not depend on k (the number of sets in the partition of variables) since we are simultaneously inverting the elementary symmetric polynomials for each set of variables by plugging in A_{ij} for each x_{ij}.

4 Computing resultants and GCD

4.1 A constant-depth circuit for the resultant over all fields

An almost immediate corollary of Theorem 16 is a polynomial size constant-depth circuit for the resultant of two polynomials over any large enough field.

▶ **Corollary 17** (Computing resultants in constant depth). *Let $\mathbb{F}$ be any large enough field (i.e., $|\mathbb{F}| \geq d^c$ for some absolute constant $c > 0$). Given two degree d univariate polynomials $f(y)$ and $g(y)$ that are monic in y and provided as a list of coefficients, we can compute the resultant of the two polynomials, $\mathrm{Res}_y(f, g)$, via a $\mathrm{poly}(d)$-size depth $O(1)$ circuit over the coefficients of f and g over the field $\mathbb{F}$.*

Proof. Let $f(y) = y^{d_1} - f_1 y^{d_1 - 1} + \cdots + (-1)^{d_1} f_0$ and $g(y) = y^{d_2} - g_1 y^{d_2 - 1} + \cdots + (-1)^{d_2} g_0$ and let $d = \max(d_1, d_2)$. If the multiset of roots (in $\overline{\mathbb{F}}$) of f and g are given by $\{\sigma_1, \ldots, \sigma_{d_1}\}$ and $\{\tau_1, \ldots, \tau_{d_2}\}$, then by Fact 6 we have

$$\mathrm{Res}_y(f, g) = \prod_{i=1}^{d_1} \prod_{j=1}^{d_2} (\sigma_i - \tau_j).$$

The RHS is clearly a bi-symmetric polynomial with respect to $\{\sigma_1, \ldots, \sigma_{d_1}\} \sqcup \{\tau_1, \ldots, \tau_{d_2}\}$ and is a $\mathrm{poly}(d)$ sized depth 2 circuit over the σ's and τ's. By Theorem 15, there exists a unique Q of degree at most $d_1 d_2$ such that

$$\mathrm{Res}_y(f, g) = Q(\overline{\mathrm{Esym}}(\sigma_1, \ldots, \sigma_{d_1}), \overline{\mathrm{Esym}}(\tau_1, \ldots, \tau_{d_2})).$$

Note that $\mathrm{Esym}_r(\sigma_1, \ldots, \sigma_{d_1})$ is just f_r and $\mathrm{Esym}_r(\tau_1, \ldots, \tau_{d_2})$ is just g_r. Thus, by Theorem 16, Q is also computable by a circuit of size $\mathrm{poly}(d)$ and depth $O(1)$ over the coefficients $f_1, \ldots, f_{d_1}, g_1, \ldots, g_{d_2}$ over the field $\mathbb{F}$. ◀

If $f(\mathbf{x}, y), g(\mathbf{x}, y)$ are multivariate polynomials that are monic in y and computable by size s depth Δ circuits, then the coefficients of monomials in y is also computable by size $\mathrm{poly}(s, n, d)$, depth $\Delta + O(1)$ circuits and thus $\mathrm{Res}_y(f, g)$ is also computable by a size $\mathrm{poly}(s, n, d)$ depth $\Delta + O(1)$ circuit.

The resultant is the case of computing the product of g's evaluations on the roots of f. A more general statement is the following lemma.

▶ **Lemma 18.** *Let $\mathbb{F}$ be any large enough field (i.e., $|\mathbb{F}| \geq (sd_1d_2n)^c$ for some absolute constant $c > 0$). Suppose $f(\mathbf{x}, y), g(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ are two n-variate polynomial that are monic in y, of degree d_1 and d_2 respectively (in y), that are given by size s, depth Δ circuits. Let $\{\sigma_1, \ldots, \sigma_{d_1}\}$ be the multi-set of y-roots of f over $\overline{\mathbb{F}(\mathbf{x})}$. Then, for any $1 \leq r \leq d_1$, we can obtain a circuit of size $\mathrm{poly}(s, n, d_1, d_2)$ and depth $\Delta + O(1)$ for*

$$\mathrm{Esym}_r(\{g(\mathbf{x}, \sigma_1), \ldots, g(\mathbf{x}, \sigma_{d_1})\}) \in \mathbb{F}[\mathbf{x}].$$

Proof. Let $d = \max(d_1, d_2)$. Note that $\mathrm{Esym}_r(y_1, \ldots, y_m) = [t^r]\{(1 + ty_1) \cdots (1 + ty_m)\}$. By Corollary 4, it is computable by a size $\mathrm{poly}(r, m)$ depth 3 circuit over $\mathbb{F}$.

If $\{\tau_1, \ldots, \tau_{d_2}\}$ is the multi-set of y-roots of g, then

$$\mathrm{Esym}_r(\{g(\mathbf{x}, \sigma_1), \ldots, g(\mathbf{x}, \sigma_{d_1})\}) = \mathrm{Esym}_r\left(\left\{\prod_{j=1}^{d_2}(\sigma_i - \tau_j) \;:\; i \in [d_1]\right\}\right)$$

is clearly a bi-symmetric polynomial with respect to $\{\sigma_1, \ldots, \sigma_{d_1}\} \sqcup \{\tau_1, \ldots, \tau_{d_2}\}$. Thus, if $f(\mathbf{x}, y) = y^{d_1} + f_1 y^{d_1 - 1} + \cdots + f_{d_1}$ and $g(\mathbf{x}, y) = y^{d_2} + g_1 y^{d_2 - 1} + \cdots + g_0$, then each f_i, g_i are the elementary symmetric polynomials of $\{\sigma_1, \ldots, \sigma_{d_1}\}$ and $\{\tau_1, \ldots, \tau_{d_2}\}$, and Corollary 4 shows that we can obtain circuits of size $\mathrm{poly}(s, d)$ and depth $\Delta + O(1)$ for them. By Theorem 16, we can express the above as $Q(f_1, \ldots, f_{d_1}, g_1, \ldots, g_{d_2})$ where $f(\mathbf{x}, y) = y^{d_1} + f_1 y^{d_1 - 1} + \cdots + f_{d_1}$ and $g(\mathbf{x}, y) = y^{d_2} + g_1 y^{d_2 - 1} + \cdots + g_0$ with Q computable by a size $\mathrm{poly}(s, n, d)$, depth $\Delta + O(1)$ circuit. ◀

The above lemma can also be easily generalized to compute elementary symmetric polynomials of a *rational function* $g(x)/h(x)$ evaluated at the roots of $f(x)$, as shown in Lemma 29.

4.2 Computing the GCD of two polynomials

Unlike operations such as computing the resultant of two polynomials, the GCD of two polynomials $f(y), g(y)$ is not a continuous function of the coefficients and therefore cannot be expressed as a polynomial (or even a rational) function of the coefficients of f and g. Andrews and Wigderson [1] show that it can be expressed as a *piece-wise rational function* of the coefficients of f, g where we have a sequence of rational functions computed by small algebraic circuits over the coefficients of f and g such that one of them computes $\gcd(f, g)$, and we know which of them via appropriate *zero-tests*.

▶ **Definition 19** (Piece-wise rational circuit families)**.** *A* piece-wise rational circuit *for a function $P : \mathbb{F}^n \to \mathbb{F}^m$ is a family of "computation circuits" $\{A_1, \ldots, A_s : \mathbb{F}^n \to \mathbb{F}^m\}$, $\{B_1, \ldots, B_s : \mathbb{F}^n \to \mathbb{F}^m\}$ and "test circuits" $\{T_1, \ldots, T_s : \mathbb{F}^n \to \mathbb{F}\}$ such that for every $\mathbf{a} \in \mathbb{F}^n$ we have*

$$P(\mathbf{a}) = \frac{A_r(\mathbf{a})}{B_r(\mathbf{a})}$$

where the advice parameter $r = \max\{k \in \mathbb{N} \;:\; T_k(\mathbf{a}) \neq 0\}$. *The size of the family is said to be the sum of the sizes of A_i's, B_i's and T_i's, and the depth of the family is maximum depth of any A_i, B_i, T_i that constitute the family.*

Situations where $P(\mathbf{a}) \in \mathbb{F}[y]$ (such as computing the GCD of two polynomials given by a list of coefficients) can be equivalently viewed as a multi-output function that outputs the list of coefficients of the polynomial $P(\mathbf{a})$. For a more detailed discussion on this perspective and the definition of *piece-wise rational* AC^0 *circuits*, we refer the reader to [1, Section 2 in the full version].

The following is the analogue of the main result of [1] for computing the GCD of two polynomials over all sufficiently large fields expressed in terms of piece-wise rational circuit families.

▶ **Theorem 20** (Computing GCD of polynomials in terms of coefficients). *Let $\mathbb{K}$ be any polynomially large field (i.e., $|\mathbb{K}| \geq (d_1 d_2)^c$ for some absolute constant $c > 0$). There are explicit families of algebraic circuits $\{P_{d_1,d_2,k}\}$, $\{Q_{d_1,d_2,k}\}$, $\{T_{d_1,d_2,k}\}$ of size $\mathrm{poly}(d_1, d_2)$ and depth $O(1)$ such that, for every $f(y), g(y) \in \mathbb{K}[y]$ that are monic in y with $\deg(f) = d_1, \deg(g) = d_2$, we have*

$$\gcd(f(y), g(y)) = \frac{P_{d_1,d_2,r}(\overline{\mathrm{coeff}}(f), \overline{\mathrm{coeff}}(g))}{Q_{d_1,d_2,r}(\overline{\mathrm{coeff}}(f), \overline{\mathrm{coeff}}(g))}$$

where $r = \max\left\{k \leq \mathrm{poly}(d_1, d_2) \ : \ T_{d_1,d_2,k}(\overline{\mathrm{coeff}}(f), \overline{\mathrm{coeff}}(g)) \neq 0\right\}$. (Here will use $\overline{\mathrm{coeff}}(f)$ to refer to the vector of coefficients of f.)

In fact, the "advice" r above is essentially just $\deg(\gcd(f, g))$. The key idea is to consider the polynomial $F(y, z) = f(y) + z \cdot g(y) = \gcd(f, g) \cdot F'(y, z)$ and somehow "filter out" the $F'(y, z)$. We now describe how to accomplish this.

Let $f(y) \in \mathbb{K}[y]$ be a monic polynomial that factorizes over $\overline{\mathbb{K}}$ as $(y - \sigma_1)^{e_1} \cdots (y - \sigma_m)^{e_m}$, with $e_i \geq 1$. For any polynomial $g(y) \in \mathbb{K}[y]$, we define $\mathrm{Filter}(f \mid \mathrm{roots}(g))$ and $\mathrm{Filter}(f \mid \text{non-roots}(g))$ as

$$\mathrm{Filter}(f \mid \text{non-roots}(g)) = \prod_{\substack{i \in [m] \\ g(\sigma_i) \neq 0}} (y - \sigma_i)^{e_i},$$

$$\mathrm{Filter}(f \mid \mathrm{roots}(g)) = \prod_{\substack{i \in [m] \\ g(\sigma_i) = 0}} (y - \sigma_i)^{e_i} = \frac{f}{\mathrm{Filter}(f \mid \text{non-roots}(g))}.$$

▶ **Lemma 21** (Computing Filter from coefficients). *Let $\mathbb{K}$ be any large enough field (i.e., $|\mathbb{K}| \geq (d_1 d_2)^c$ for some absolute constant $c > 0$). There are explicit families of circuits $\{A_{d_1,d_2,k}\}$, $\{B_{d_1,d_2,k}\}$, $\{T_{d_1,d_2,k}\}$ of size $\mathrm{poly}(d_1, d_2)$ and depth $O(1)$ such that, for every pair of monic polynomial $f, g \in \mathbb{K}[y]$ with $\deg f = d_1$ and $\deg g = d_2$, we have*

$$\mathrm{Filter}(f \mid \text{non-roots}(g)) = \frac{A_{d_1,d_2,r}(\overline{\mathrm{coeff}}(f), \overline{\mathrm{coeff}}(g))}{B_{d_1,d_2,r}(\overline{\mathrm{coeff}}(f), \overline{\mathrm{coeff}}(g))}$$

where the advice parameter $r = \max\left\{k \leq \mathrm{poly}(d_1, d_2) \ : \ T_{d_1,d_2,k}(\overline{\mathrm{coeff}}(f), \overline{\mathrm{coeff}}(g)) \neq 0\right\}$.

Proof. Let $f(y) = \prod_{i=1}^{m} (y - \sigma_i)^{e_i}$ with $\sigma_i \in \overline{\mathbb{K}}$ and $e_i \geq 1$, and let Σ be the multiset of roots of $f(y)$ in $\overline{\mathbb{K}}$. If $r = \deg(\mathrm{Filter}(f \mid \text{non-roots}(g)))$, then

$$\mathrm{Esym}_r\left(\{g(\sigma) \ : \ \sigma \in \Sigma\}\right) = \prod_{i \in [m] \,:\, g(\sigma_i) \neq 0} g(\sigma_i)^{e_i} \neq 0,$$

$$\mathrm{Esym}_k\left(\{g(\sigma) \ : \ \sigma \in \Sigma\}\right) = 0 \quad \text{for all } k > r$$

as any subset of size greater than r among Σ must necessarily contain a σ such that $g(\sigma) = 0$. By Lemma 18, the above is expressible as circuit of size $\mathrm{poly}(d_1, d_2)$ and depth $O(1)$ over the coefficients of f and g and that gives us the claimed family $\{T_{d_1,d_2,k}\}$.

To obtain the filter as a ratio of algebraic circuits, define the polynomial $h(x,y) = (y-x)\cdot g(x)$ for a fresh variable x. Note that, for any $\sigma \in \overline{\mathbb{K}}$, we have $h(\sigma, y) = 0$ if and only if $g(\sigma) = 0$. Therefore,

$$\begin{aligned}\mathrm{Esym}_r\left(\{h(\sigma,y) \;:\; \sigma\in\Sigma\}\right) &= \prod_{i\in[m]\,:\,g(\sigma_i)\neq 0} \left((y-\sigma_i)\cdot g(\sigma_i)\right)^{e_i} \\ &= \left(\prod_{i\in[m]\,:\,g(\sigma_i)\neq 0} (y-\sigma_i)^{e_i}\right)\cdot \mathrm{Esym}_r\left(\{g(\sigma) \;:\; \sigma\in\Sigma\}\right) \\ \therefore \mathrm{Filter}(f \mid \text{non-roots}(g)) &= \frac{\mathrm{Esym}_r\left(\{h(\sigma,y) \;:\; \sigma\in\Sigma\}\right)}{\mathrm{Esym}_r\left(\{g(\sigma) \;:\; \sigma\in\Sigma\}\right)}\end{aligned}$$

Once again by Lemma 18, both the numerator and denominator on the RHS can be expressed as circuits of size $\mathrm{poly}(d_1,d_2)$ and depth $O(1)$ over the coefficients of f and g and that gives us the circuit families $\{A_{d_1,d_2,k}\}$ and $\{B_{d_1,d_2,k}\}$. ◀

We are now ready to prove Theorem 20.

Proof of Theorem 20. Without loss of generality, let us assume that $d_1 := \deg f > \deg g =: d_2$ (by working with $f, g-f$ if their degrees were the same). Let $\{\sigma_1,\ldots,\sigma_m\} \in \overline{\mathbb{K}}$ be the union of roots of f and g, and let

$$\begin{aligned} f(y) &= (y-\sigma_1)^{e_1}\cdots(y-\sigma_m)^{e_m}, \\ g(y) &= (y-\sigma_1)^{e_1'}\cdots(y-\sigma_m)^{e_m'} \\ \implies \gcd(f,g) &= (y-\sigma_1)^{e_1^*}\cdots(y-\sigma_m)^{e_m^*} \quad \text{where } e_i^* = \min(e_i,e_i').\end{aligned}$$

Consider the polynomial $F(y,z) = f(y) + z\cdot g(y)$ for a fresh variable z, which is monic in y since $\deg f > \deg g$. The key observation is that $\gcd(f,g) = \mathrm{Filter}(F(y,z) \mid \mathrm{roots}(g))$ and the theorem essentially follows from Lemma 21 and this observation. We formalize this below.

Since $F(y,z)$ is clearly divisible by $\gcd(f,g)$, we can write $F(y,z)$ as

$$F(y,z) = \gcd(f,g)\cdot F'(y,z)$$

for some $F'(y,z) \in \mathbb{K}[y,z]$. Note that, for every σ_i, we have $F'(\sigma_i,z) \neq 0$ (for otherwise $(y-\sigma_i)^{e_i^*+1}$ divides both f and g). Therefore,

$$\begin{aligned} &\mathrm{Filter}(F(y,z) \mid \text{non-roots}(g)) = F'(y,z) \\ \implies &\mathrm{Filter}(F(y,z) \mid \mathrm{roots}(g)) = \frac{F(y,z)}{\mathrm{Filter}(F(y,z) \mid \text{non-roots}(g))} = \gcd(f,g).\end{aligned}$$

By Lemma 21, we have circuit families $\{A_{d_1,d_2,k}\}$, $\{B_{d_1,d_2,k}\}$, $\{T_{d_1,d_2,k}\}$ such that

$$\begin{aligned} \mathrm{Filter}(F(y,z) \mid \text{non-roots}(g)) &= \frac{A_{d_1,d_2,r}(\overline{\mathrm{coeff}_y}(F), \overline{\mathrm{coeff}}(g))}{B_{d_1,d_2,r}(\overline{\mathrm{coeff}_y}(F), \overline{\mathrm{coeff}}(g))} \\ \text{where } r &= \max\left\{k \le \mathrm{poly}(d_1,d_2) \;:\; T_{d_1,d_2,k}(\overline{\mathrm{coeff}_y}(F), \overline{\mathrm{coeff}}(g)) \neq 0\right\} \\ \implies \gcd(f,g) &= \mathrm{Filter}(F(y,z) \mid \mathrm{roots}(g)) \\ &= \frac{F(y,z)\cdot B_{d_1,d_2,r}(\overline{\mathrm{coeff}_y}(F), \overline{\mathrm{coeff}}(g))}{A_{d_1,d_2,r}(\overline{\mathrm{coeff}_y}(F), \overline{\mathrm{coeff}}(g))}.\end{aligned}$$

Since each coefficient of F can be expressed as a polynomial function of the coefficients of f and g, the above circuit families $\{A_{d_1,d_2,k}\}$, $\{B_{d_1,d_2,k}\}$, $\{T_{d_1,d_2,k}\}$ can also be expressed as $\mathrm{poly}(d_1, d_2)$ size depth $O(1)$ circuit families over the coefficients of f and g. It can also be checked that the advice parameter $r = d_1 - \deg(\gcd(f, g))$.

As stated, although $\gcd(f(y), g(y)) \in \mathbb{K}[y]$ and is thus independent of the variable z, the RHS of the above equations consist of numerators and denominators that depend on z. To eliminate z from the RHS, we can remove any powers of z dividing the numerator and denominator and then set z to zero. This can be achieved by expanding the test circuits to also find the smallest i such that $[z^i]\left\{A_{d_1,d_2,r}(\overline{\mathrm{coeff}_y}(F), \overline{\mathrm{coeff}}(g))\right\} \neq 0$ and set the numerator and denominator circuits as

$$P_{d_1,d_2,(r,i)}(\overline{\mathrm{coeff}}(f), \overline{\mathrm{coeff}}(g)) := [z^i]\left\{F(y,z) \cdot B_{d_1,d_2,r}(\overline{\mathrm{coeff}_y}(F), \overline{\mathrm{coeff}}(g))\right\},$$
$$Q_{d_1,d_2,(r,i)}(\overline{\mathrm{coeff}}(f), \overline{\mathrm{coeff}}(g)) := [z^i]\left\{A_{d_1,d_2,r}(\overline{\mathrm{coeff}_y}(F), \overline{\mathrm{coeff}}(g))\right\}$$

respectively. Thus, overall, we obtain size $\mathrm{poly}(d_1, d_2)$ and depth $O(1)$ circuits over the coefficients of f and g as claimed by the theorem. ◀

5 Improved closure results for fields of small characteristic

In this section, we use the Theorem 2 to prove the following closure result for constant-depth circuits over fields of characteristic p. This statement is similar to a result in [2] with one key difference – the circuit for $g(\mathbf{x})^{p^\ell}$ in Theorem 22 is over the underlying field itself and not over the algebraic closure (or a high degree extension) of the base field $\mathbb{F}_q$, provided q is polynomially large. This extends the results of Kaltofen [10] for fields of characteristic p for natural subclasses of algebraic circuits such as constant-depth circuits, formulas, branching programs etc. The key technical ingredient in the proof again is the use of Theorem 2.

▶ **Theorem 22** (Complexity of factors). *Let $\mathbb{F}_q$ be a field of positive characteristic p that is polynomially large (i.e., there is an absolute constant c such that $|\mathbb{F}| \geq (snd)^c$). Let $P(\mathbf{x}) \in \mathbb{F}_q[\mathbf{x}]$ be a polynomial on n variables of degree d computed by a circuit C of size s and depth Δ. Further, let $g(\mathbf{x})$ be a factor of $P(\mathbf{x})$ with multiplicity $p^\ell \cdot e$ where $\gcd(p, e) = 1$. Then, $g(\mathbf{x})^{p^\ell}$ is computable by a circuit of size $\mathrm{poly}(s, d, n)$ and depth $\Delta + O(1)$ over $\mathbb{F}_q$.*

We start with a few preliminaries necessary for the proof of the theorem. We shall interpret $P(\mathbf{x})$ as an element of $\mathbb{F}[\mathbf{x}, y]$, and apply a map $\Psi : x_i \mapsto tx_i + a_i y + b_i$ on $P(\mathbf{x})$ for randomly chosen a_is and b_is, to ensure certain properties[3]. Thus, without loss of generality, we can work with a bivariate polynomial $P(t, y)$ over a field that contains the other variables. Some preliminaries will also be stated for bivariate polynomials.

5.1 Hasse derivatives

We shall work with the notion of Hasse derivative, which is the standard alternative to partial derivatives in the small characteristic setting. We state the definition and the product rule for Hasse derivatives. For more details, we recommend the reader to refer to [8, Appendix C].

▶ **Definition 23** (Hasse derivatives). *The* Hasse Derivative of order i *of $F(t, y) \in \mathbb{F}[t, y]$ with respect to y, denoted as $\mathrm{D}_y^{(i)}(F)$, is defined as the coefficient of z^i in the polynomial $F(t, y + z)$.*

[3] We expand on this in the beginning of Subsection 5.4. For more details, please refer to the preliminaries of [2].

▶ **Lemma 24** (Product rule for Hasse derivatives). *Let $G(t,y), H(t,y) \in \mathbb{F}[t,y]$ be bivariate polynomials and let $k \geq 0$. Then,*

$$\mathrm{D}_y^{(k)}(GH) = \sum_{i+j=k} \mathrm{D}_y^{(i)}(G) \cdot \mathrm{D}_y^{(j)}(H)$$

5.2 Furstenberg's theorem over small characteristic fields

While Theorem 8 states Furstenberg's theorem for power series roots that appear as a linear factor with multiplicity one, there is a version of the theorem that works even when the roots appear with arbitrary multiplicity e (see [2, Theorem 3.1]) if e is invertible in the field $\mathbb{F}$. Over a field of characteristic p, p might divide e and thus e might not be invertible. The following version of Furstenberg's theorem over small characteristic is very similar to Theorem 8, with some key differences. The theorem expresses an appropriate power of a power series root of a polynomial as a diagonal of a rational expression involving the polynomial and its derivatives.

▶ **Theorem 25** (Furstenberg's theorem over small characteristic fields (Theorem A.3 in [2])). *Let $\mathbb{F}$ be a field of characteristic p. Let $P(t,y) \in \mathbb{F}[\![t,y]\!]$ be a power series and $\varphi(t) \in \mathbb{F}[\![t]\!]$ be a power series satisfying*

$$P(t,y) = (y - \varphi(t))^{p^\ell e} \cdot Q(t,y)$$

for some $\ell \geq 0, e \geq 1$ such that $\gcd(p,e) = 1$. If $\varphi(0) = 0$ and $Q(0,0) \neq 0$, then

$$\varphi^{p^\ell} = \mathcal{D}\left(\frac{y^{2p^\ell} \cdot \mathrm{D}_y^{(p^\ell)}(P)(ty,y)}{e \cdot P(ty,y)}\right) \tag{2}$$

We can further simplify the expression in Theorem 25 to get a version of Corollary 9 over small characteristic fields.

▶ **Corollary 26** (Corollary A.5 in [2]). *Let $P(t,y), Q(t,y) \in \mathbb{F}[\![t,y]\!]$ and $\varphi(t) \in \mathbb{F}[\![t]\!]$ satisfy*

$$P(t,y) = (y - \varphi(t))^{p^\ell e} \cdot Q(t,y)$$

with $\gcd(p,e) = 1$, $\varphi(0) = 0$ and $Q(0,0) = \alpha \neq 0$. Then,

$$\varphi(t)^{p^\ell} = \sum_{m \geq 0} [y^{p^\ell(e(m+1)-2)}] \left\{ \frac{\mathrm{D}_y^{(p^\ell)}(P)(t,y)}{e \cdot \alpha^{m+1}} \Big(\alpha y^{p^\ell \cdot e} - P(t,y)\Big)^m \right\}.$$

Moreover,

$$\mathrm{Hom}_{\leq d}[\varphi(t)^{p^\ell}] = \mathrm{Hom}_{\leq d}\left[\sum_{m \geq 0}^{2e(d+p^\ell)} [y^{p^\ell(e(m+1)-2)}] \left\{ \frac{\mathrm{D}_y^{(p^\ell)}(P)(t,y)}{e \cdot \alpha^{m+1}} \Big(\alpha y^{p^\ell \cdot e} - P(t,y)\Big)^m \right\} \right].$$

The following claim would be helpful to express $\alpha = Q(0,0)$ as an appropriate Hasse derivative of P.

▷ Claim 27. Suppose $P(t,y) = (y - \varphi(t))^{p^\ell e} \cdot Q(t,y) \in \mathbb{F}[\![t,y]\!]$ with $\gcd(p,e) = 1$, $\varphi(0) = 0$ and $Q(0,0) = \alpha \neq 0$. Then,

$$\alpha = \mathrm{D}_y^{(p^\ell \cdot e)} P(0,0)$$

Proof. Let $F(t,y) = (y-\varphi(t))^{p^\ell \cdot e}$. Then, by Lemma 24,

$$\mathrm{D}_y^{(p^\ell \cdot e)} P(t,y) = \sum_{i+j=p^\ell \cdot e} \mathrm{D}_y^{(i)} F(t,y) \cdot \mathrm{D}_y^{(j)} Q(t,y)$$

$$\implies \mathrm{D}_y^{(p^\ell \cdot e)} P(0,0) = \sum_{i+j=p^\ell \cdot e} \mathrm{D}_y^{(i)} F(0,0) \cdot \mathrm{D}_y^{(j)} Q(0,0)$$

Note that $\mathrm{D}_y^{(i)} F = [z^i]\left\{(y^{p^\ell} + z^{p^\ell} - \varphi^{p^\ell})^e\right\}$ and is a nonzero polynomial only for i of the form $p^\ell \cdot e'$ for some $0 \le e' \le e$, and $\mathrm{D}_y^{(p^\ell e)} F = 1$. Furthermore, for every $e' < e$, we have $\mathrm{D}_y^{(p^\ell e')} F$ is divisible by $(y-\varphi)^{p^\ell}$ and hence $\mathrm{D}_y^{(p^\ell e')} F(0,0) = 0$ for all $e' < e$. Therefore,

$$\begin{aligned}\mathrm{D}_y^{(p^\ell \cdot e)} P(0,0) &= \sum_{i+j=p^\ell \cdot e} \left(\mathrm{D}_y^{(i)} F(0,0)\right) \cdot \left(\mathrm{D}_y^{(j)} Q(0,0)\right) \\ &= \left(\mathrm{D}_y^{(p^\ell e)} F(0,0)\right) \cdot \left(\mathrm{D}_y^{(0)} Q(0,0)\right) = Q(0,0) = \alpha. \end{aligned}$$ ◁

5.3 Complexity of power series roots over small characteristic fields

Using Theorem 25 and Corollary 26, we get the following analogue of Theorem 10 over arbitrary fields of small characteristic.

▶ **Theorem 28** (Power series roots with multiplicity over small characteristic)**.** *Let $\mathbb{F}$ be a field of positive characteristic p. Suppose $P(\mathbf{x},y) \in \mathbb{F}[\mathbf{x},y]$ is a polynomial computed by a circuit C, and $\varphi(\mathbf{x}) \in \mathbb{F}[\![\mathbf{x}]\!]$ is a power series satisfying $P(\mathbf{x},y) = (y-\varphi(\mathbf{x}))^{p^\ell e} \cdot Q(\mathbf{x},y)$ where $\varphi(\mathbf{0}) = 0$, $\gcd(p,e) = 1$ and $Q(\mathbf{0},0) \neq 0$. Then, for any $d \in \mathbb{N}$, there is a circuit C' over $\mathbb{F}$ computing $\mathrm{Hom}_{\le d}\left[\varphi^{p^\ell}\right]$ such that*

$\mathrm{size}(C') \le \mathrm{poly}(d, \mathrm{size}(C))$

$\mathrm{depth}(C') \le \mathrm{depth}(C) + O(1)$

Proof. The theorem follows from Corollary 26 that asserts that

$$\mathrm{Hom}_{\le d}\left[\varphi^{p^\ell}\right] = \mathrm{Hom}_{\le d}\left[\sum_{m \ge 0}^{2e(d+p^\ell)} [y^{p^\ell(e(m+1)-2)}]\left\{\frac{\mathrm{D}_y^{(p^\ell)}(P)(t,y)}{e \cdot \alpha^{m+1}}\left(\alpha y^{p^\ell \cdot e} - P(t,y)\right)^m\right\}\right],$$

and Corollary 4 shows that the RHS is computable by a size $\mathrm{poly}(s,n,d)$ and depth $\Delta + O(1)$ circuit. ◀

5.4 Proof of Theorem 22

We will need the following generalization of Lemma 18, whose proof we will defer to the end of this section.

▶ **Lemma 29.** *Let $\mathbb{F}$ be any large enough field (i.e., $|\mathbb{K}| \ge (sdn)^c$ for some absolute constant $c > 0$). Let $f(\mathbf{x},y), g(\mathbf{x},y), h(\mathbf{x},y) \in \mathbb{F}[\mathbf{x},y]$ be polynomials that are monic in y of degrees d_f, d_g and d_h respectively, with $\gcd(f,h) = 1$, that are given by size s, depth Δ circuits; let $d = \max(d_f, d_g, d_h)$. If $\{\sigma_1, \ldots, \sigma_{d_f}\}$ be the multi-set of y-roots of f over $\overline{\mathbb{F}(\mathbf{x})}$, then for any $1 \le r \le d_f$, we can get circuits $A(\mathbf{x}), B(\mathbf{x})$ of size $\mathrm{poly}(s,n,d)$ and depth $\Delta + O(1)$ such that*

$$\mathrm{Esym}_r\left(\left\{\frac{g(\mathbf{x},\sigma_i)}{h(\mathbf{x},\sigma_i)} \;:\; i \in [d_f]\right\}\right) = \frac{A(\mathbf{x})}{B(\mathbf{x})}.$$

We are now ready to complete the proof of Theorem 22.

Proof of Theorem 22. By interpreting $P(\mathbf{x})$ as an element of $\mathbb{F}[\mathbf{x}, y]$, let $\tilde{P}(t, y) = \Psi(P(\mathbf{x}, y)) \in \mathbb{F}[\mathbf{x}][t, y]$ for a map $\Psi : x_i \mapsto tx_i + a_i y + b_i$ where the a_is and b_is are chosen at random. This will ensure that $\tilde{P}(t, y)$ is monic in y, the square-free part of $\tilde{P}(t, y)$ is also square-free at $t = 0$, and that $\tilde{P}(t, y) \in \mathbb{F}[\mathbf{x}][t, y]$ is also computable by a size $\text{poly}(s, d, n)$, depth $\Delta + O(1)$ circuit over $\mathbb{F}$. From Gauss' Lemma, we know that all the factors of $\tilde{P}$ can now be assumed to be monic in y without loss of generality, and viewing $\tilde{P}$ as a bivariate in t and y with coefficients in $\mathbb{F}(\mathbf{x})$ maintains the factorization pattern of the polynomial. Furthermore, the map can be inverted using the transformation $x_i \mapsto x_i - a_i y - b_i$ and $t \mapsto 1$ and this gives us a way to argue about factors of P using factors of $\tilde{P}$. Thus[4], it suffices to show that for an arbitrary factor $g(t, y) \in \mathbb{F}[\mathbf{x}][t, y]$ of $\tilde{P}(t, y)$, if the multiplicity of $g(t, y)$ is $p^\ell \cdot e$ with $\gcd(e, p) = 1$, then $g(t, y)^{p^\ell}$ is computable by a $\text{poly}(s, d, n)$ size, depth $\Delta + O(1)$ circuit over the field $\mathbb{F}$.

First we note that the existence of a power series factorization of $\tilde{P}(t, y)$ follows *almost* immediately from Lemma 7. We cannot apply Lemma 7 directly because the preprocessing map Ψ only ensures that the square-free part of $\tilde{P}(t, y)$ remains square-free when t is set to zero. Applying Lemma 7 on the square-free part of $\tilde{P}(t, y)$, denoted by $\hat{P}(t, y)$, gives us the factorization $\hat{P}(t, y) = \prod_{i \in [m]}(y - \varphi_i(t))$, where each $\varphi_i(t)$ has a distinct constant term $\varphi_i(0) \in \overline{\mathbb{F}}$.

Suppose $g(t, y)$ is a factor with multiplicity $p^\ell \cdot e$. For each power series y-root $\varphi(t)$ of $g(t, y)$, we can write $\tilde{P}(t, y) = (y - \varphi(t))^{p^\ell \cdot e} Q(t, y)$ for some $Q(t, y)$ that satisfies $Q(0, \varphi(0)) \neq 0$. Translating y to $y + \varphi(0)$, we can use Corollary 26 on $\tilde{P}(t, y + \varphi(0))$ to show that $\text{Hom}_{\leq d}[\varphi(t)^{p^\ell}]$ is equal to

$$\text{Hom}_{\leq d}\left[\varphi(0)^{p^\ell} + \sum_{m \geq 0}^{2e(d+p^\ell)} [y^{p^\ell(e(m+1)-2)}] \left\{ \frac{\mathrm{D}_y^{(p^\ell)}(\tilde{P})(t, y + \varphi(0))}{e \cdot \alpha^{m+1}} \Big(\alpha (y + \varphi(0))^{p^\ell \cdot e} - \tilde{P}(t, y + \varphi(0)) \Big)^m \right\} \right]$$

where $\alpha = Q(0, \varphi(0)) \neq 0$, which is also equal to $\mathrm{D}_y^{(p^\ell \cdot e)} \tilde{P}(0, \varphi(0))$ by Claim 27. Thus, we define the rational function $R(z)$ as

$$z^{p^\ell} + \sum_{m \geq 0}^{2e(d+p^\ell)} [y^{p^\ell(e(m+1)-2)}] \left\{ \frac{\mathrm{D}_y^{(p^\ell)}(\tilde{P})(t, y + z)}{e \cdot \mathrm{D}_y^{(p^\ell \cdot e)}(\tilde{P})(0, z)^{m+1}} \Big(\mathrm{D}_y^{(p^\ell \cdot e)}(\tilde{P})(0, z) \cdot (y + z)^{p^\ell \cdot e} - \tilde{P}(t, y + z) \Big)^m \right\}.$$

If $\hat{P}(0, y) = \prod_{i=1}^m (y - \varphi_i(0))$, and $g(0, y)^{p^\ell} = \prod_{i \in S}(y - \varphi_i(0))^{p^\ell}$ for some subset $S \subset [m]$, then the power series roots of $g(t, y)$ are precisely $\varphi_j(t)$ for $j \in S$ and hence

$$\begin{aligned}
g(t, y)^{p^\ell} &= \prod_{j \in S} (y^{p^\ell} - \varphi_j(t)^{p^\ell}) = \prod_{j \in S} (y^{p^\ell} - R(\varphi_j(0))) \bmod t^{d+1} \\
\implies g(t, y)^{p^\ell} &= \text{Hom}_{\leq d}\left[g'(t, y)\right] \\
\text{where} \quad g'(t, y) &= \prod_{j \in S} (y^{p^\ell} - R(\varphi_j(0))) \\
&= \sum_{k=0}^{|S|} y^{p^\ell(|S|-k)} \cdot (-1)^{|S|-k} \cdot \text{Esym}_k(\{R(\varphi_j(0)) \ : \ j \in S\})
\end{aligned}$$

[4] For more details, we encourage the reader to refer to the preliminaries in [2].

Thus, it suffices to show that $\mathrm{Esym}_k(\{R(\varphi_j(0)) \ : \ j \in S\})$ for all $1 \le k \le d$ can be computed by a $\mathrm{poly}(s,n,d)$ size depth $\Delta + O(1)$ circuit as that would immediately yield similar size and depth bounds for $g(t,y)^{p^\ell}$ by Corollary 4.

Note that the rational function $R(z)$ can be easily expressed as $\frac{R_{\mathrm{num}}(z)}{R_{\mathrm{denom}}(z)}$ where $R_{\mathrm{num}}(z)$ and $R_{\mathrm{denom}}(z)$ are both computable by a $\mathrm{poly}(s,d,n)$ sized depth $\Delta + O(1)$ circuits since $\tilde{P}(t,y)$ is given by a $\mathrm{poly}(s,d,n)$ sized depth $\Delta + O(1)$ circuit (using Lemma 3 and Corollary 4). Furthermore, $R_{\mathrm{denom}}(z) = (\mathrm{D}_y^{(p^\ell \cdot e)}(\tilde{P})(0,z))^{2e(d+p^\ell)+1}$, so it remains nonzero element of $\overline{\mathbb{F}_q}$ when z is set to any root of $\tilde{P}(0,y)$ (Claim 27).

Note that $\mathrm{Esym}_k(\{\varphi_j(0) \ : \ j \in S\})$ is in $\mathbb{F}$ since they are the (signed) coefficients of $g(0,y)$. Thus, by Lemma 29, we can express $\mathrm{Esym}_k(\{R(\varphi_j(0)) \ : \ j \in S\})$ as a ratio of two $\mathrm{poly}(s,d,n)$ size depth $\Delta + O(1)$ circuits over $\mathbb{F}$. By eliminating the division gate using the standard technique of Strassen (which works in constant depth), we obtain a $\mathrm{poly}(s,d,n)$ size depth $\Delta + O(1)$ circuits over $\mathbb{F}$ for $\mathrm{Esym}_k(\{R(\varphi_j(0)) \ : \ j \in S\})$ for each $1 \le k \le d$, and thus a similar bound for $g(t,y)^{p^\ell}$. ◀

Proof of Lemma 29. Note that

$$\mathrm{Esym}_r\left(\frac{u_1}{v_1}, \ldots, \frac{u_{d_f}}{v_{d_f}}\right) = [t^r]\left\{(v_1 + tu_1)\cdots(v_{d_f} + tu_{d_f})\right\} \cdot \frac{1}{v_1 \ldots v_{d_f}}. \tag{3}$$

Let $\left\{\tau_1, \ldots, \tau_{d_g}\right\}, \{\rho_1, \ldots, \rho_{d_h}\} \subset \overline{\mathbb{F}(\mathbf{x})}$ be the multi-set of y-roots of g and h respectively. Thus, setting $u_i = g(\mathbf{x}, \sigma_i)$ and $v_i = h(\mathbf{x}, \sigma_i)$ in (3), we see that

$$\begin{aligned}\Gamma(\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\rho}) &= [t^r]\left\{(h(\mathbf{x},\sigma_1) + tg(\mathbf{x},\sigma_1))\cdots(h(\mathbf{x},\sigma_{d_f}) + tg(\mathbf{x},\sigma_{d_f}))\right\} \\ &= [t^r]\left\{\left(\prod_{j=1}^{d_h}(\sigma_1 - \rho_j) + t\prod_{k=1}^{d_g}(\sigma_1 - \tau_k)\right)\cdots\left(\prod_{j=1}^{d_h}(\sigma_{d_f} - \rho_j) + t\prod_{k=1}^{d_g}(\sigma_{d_f} - \tau_k)\right)\right\}\end{aligned}$$

is multi-symmetric with respect to $\left\{\sigma_1, \ldots, \sigma_{d_f}\right\} \sqcup \left\{\tau_1, \ldots, \tau_{d_g}\right\} \sqcup \left\{\rho_1, \ldots, \rho_{d_h}\right\}$ that is computable by a $\mathrm{poly}(s,d)$ size depth $\Delta + O(1)$ circuit (over $\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\rho}$). Thus, by Theorem 16, we have a circuit of size $\mathrm{poly}(s,d)$ and depth $\Delta + O(1)$ over the elementary symmetric polynomials of $\boldsymbol{\sigma}, \boldsymbol{\tau}, \boldsymbol{\rho}$, which are the coefficients of f, g and h, and they are also computable by size $\mathrm{poly}(s,d)$ depth $\Delta + O(1)$ circuits (by Corollary 4). That gives us the circuit $A(\mathbf{x})$.

The circuit $B(\mathbf{x})$ is to compute $\prod_{i=1}^{d_f} h(\mathbf{x}, \sigma_i) = \mathrm{Esym}_{d_f}(h(\mathbf{x},\sigma_1), \ldots, h(\mathbf{x},\sigma_{d_f}))$ and Lemma 18 shows that $B(\mathbf{x})$ is also computable by a size $\mathrm{poly}(s,d)$ depth $\Delta + O(1)$ circuit. Since $\gcd(f,h) = 1$, they do not share any y-roots and hence the division $A(\mathbf{x})/B(\mathbf{x})$ is well-defined. ◀

References

1 Robert Andrews and Avi Wigderson. Constant-depth arithmetic circuits for linear algebra problems. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2367–2386, 2024. Full version at `arXiv:2404.10839`. `doi:10.1109/FOCS61266.2024.00138`.

2 Somnath Bhattacharjee, Mrinal Kumar, Shanthanu Rai, Varun Ramanathan, Ramprasad Saptharishi, and Shubhangi Saraf. Closure under factorization from a result of furstenberg. *Electronic Colloquium on Computational Complexity (ECCC)*, TR25-084, 2025. URL: `https://eccc.weizmann.ac.il/report/2025/084`.

3 Markus Bläser and Gorav Jindal. On the Complexity of Symmetric Polynomials. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 47:1–47:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2019.47`.

4 Prasad Chaugule, Mrinal Kumar, Nutan Limaye, Chandra Kanta Mohapatra, Adrian She, and Srikanth Srinivasan. Schur polynomials do not have small formulas if the determinant does not. *computational complexity*, 32(1):3, 2023. doi:10.1007/s00037-023-00236-x.

5 David A Cox, John Little, and Donal O'shea. *Using Algebraic Geometry*. Springer New York, NY, 2005. doi:10.1007/b138611.

6 Laszlo Csanky. Fast parallel matrix inversion algorithms. *SIAM J. Comput.*, 5(4):618–623, 1976. doi:10.1137/0205040.

7 Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. *J. ACM*, 69(3), June 2022. doi:10.1145/3510359.

8 Michael A. Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2014. URL: https://hdl.handle.net/1721.1/89843.

9 Harry Furstenberg. Algebraic functions over finite fields. *Journal of Algebra*, 7(2):271–277, 1967. doi:10.1016/0021-8693(67)90061-0.

10 Erich Kaltofen. Factorization of Polynomials Given by Straight-Line Programs. In *Randomness and Computation*, pages 375–412. JAI Press, 1989. URL: https://users.cs.duke.edu/~elk27/bibliography/89/Ka89_slpfac.pdf.

11 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM Journal of Computing*, 12(4):641–644, 1983. Preliminary version in the *6th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1981)*. doi:10.1137/0212043.

12 Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3 edition, 2013. doi:10.1017/CBO9781139856065.

Rank Bounds and Polynomial-Time PIT for $\Sigma^k\Pi\Sigma\Pi^2$ Circuits

Abhibhav Garg
University of Waterloo, Canada

Rafael Oliveira
University of Waterloo, Canada

Akash Kumar Sengupta
Rutgers University, Piscataway, NJ, USA

Nir Shalmon
Tel Aviv University, Israel

Amir Shpilka
Tel Aviv University, Israel

Abstract

A depth-4 algebraic circuit with top fan-in k and bottom fan-in 2 is a circuit Φ of the form $\Phi = \sum_{i=1}^{k} \prod_{j=1}^{m_i} Q_{ij}$, where the polynomials $Q_{ij} \in \mathbb{K}[x_1, \ldots, x_n]$ have degree at most 2. The class of all such circuits is denoted by $\Sigma^k\Pi\Sigma\Pi^2$. We say that the circuit Φ is an identity if it formally computes the zero polynomial. An important parameter of $\Sigma^k\Pi\Sigma\Pi^2$ circuits Φ is their (linear) rank, which is defined as the vector space dimension of the polynomials $\{Q_{ij}\}_{i\in[k],j\in[m_i]}$.

We prove that, when the base field $\mathbb{K}$ is of characteristic zero, the rank of any (simple and minimal) $\Sigma^k\Pi\Sigma\Pi^2$ identity is upper bounded by a function which depends only on the top fan-in k. This result makes progress on [11, Conjecture 28], being the first work to establish a bound on the rank of such identities that depends only on the top fan-in. Moreover, when combined with [11, Theorem 2], our main result yields the first deterministic, *polynomial time* PIT algorithm for $\Sigma^k\Pi\Sigma\Pi^2$ circuits.

One of the key components of our proof of the rank bounds is the derivation of an approximate Hansen-type result, which is interesting in its own right. This result can be seen as an algebraic and higher-dimensional analogue of the approximate Sylvester-Gallai result of [5], and a distinct approximate fractional Sylvester-Gallai result than the one from [27]. Additionally, we prove a robust version of it, in the spirit of the generalization of Hansen's theorem by [9].

This paper is an extended abstract of the full version of the paper, which can be found at [28].

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Sylvester-Gallai Theorems, Polynomial Identity Testing, Strong Algebras

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.17

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2026/084` [28]

Funding NS and AS research was funded by the European Union (ERC, EACTP, 101142020). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgements Part of this work took place while RO was supported by the Simons Institute for the Theory of Computing, and conducted when the author was visiting the Institute. The author would like to thank the generosity, hospitality, and excellent research environment provided by the Simons Institute. Part of this work was done while AKS was postdoctoral fellow at the University of Waterloo.

© Abhibhav Garg, Rafael Oliveira, Akash Kumar Sengupta, Nir Shalmon, and Amir Shpilka;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 17; pp. 17:1–17:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Polynomial Identity Testing (PIT) is the problem of checking whether a given algebraic circuit over a field $\mathbb{K}$ computes the zero polynomial. When the base field $\mathbb{K}$ is sufficiently large, the PIT problem is equivalent to asking whether the given algebraic circuit evaluates to zero on all inputs. This equivalent formulation suggests a very simple randomized algorithm: simply evaluate the circuit at a randomly chosen point. The correctness of this algorithm is given by the polynomial identity lemma.

Derandomizing PIT is a foundational problem in theoretical computer science: it is intrinsically related to proving lower bounds for algebraic circuits [39, 41, 2, 49, 50] and the derandomization of PIT even for restricted classes of circuits has resulted in the derandomization of important problems in mathematics and computer science [3, 24, 53, 22, 37, 68]. For an overview on PIT and its connections, we refer the reader to [61, 67, 62, 16].

Given the difficulty of derandomizing the (full) PIT problem, much of the focus has shifted to natural subclasses of circuits, such as sparse polynomials (equivalently, depth-2 circuits) [12, 32, 47] and depth-3 circuits with bounded top fan-in [19, 45, 44, 42, 63, 64]. Recent advances in depth reduction [1, 48, 70, 36] have shown that a deterministic PIT algorithm for unrestricted depth-3 circuits or homogeneous depth-4 circuits would yield a quasipolynomial-time deterministic PIT algorithm for general circuits. These insights have renewed interest in these classes and spurred substantial research activity on PIT within these settings, as seen in [8, 43, 60, 11, 56, 23, 4, 33, 15, 58, 31, 34] and references therein.

The work [19], which first studied the PIT problem for the class of depth-3 circuits of bounded top fan-in (denoted $\Sigma^k\Pi\Sigma$), defined the *rank* of a depth-3 circuit as the dimension of the vector space of the linear forms appearing in the bottom layer of the circuit. Their high-level strategy for obtaining a deterministic, polynomial-time PIT algorithm for this class was as follows. Suppose there exists a function $R : \mathbb{N} \to \mathbb{N}$ such that any "non-trivial" (simple and minimal) $\Sigma^k\Pi\Sigma$ identity has rank less than $R(k)$. If one can devise a variable reduction procedure that maps the variables into an $R(k)$-dimensional space while preserving the rank of any $R(k)$ linear forms, then composing this map with the circuit yields a nonzero circuit in only $R(k)$ variables. A simple interpolation on this reduced circuit suffices to verify nonzeroness.

While [19] was not able to prove the existence of such a function R, they managed to prove a rank bound which depended both on the top fan-in k and on the logarithm of the degree of the circuit. This bound, combined with the variable reduction procedure was enough to yield a quasi-polynomial time PIT algorithm for $\Sigma^k\Pi\Sigma$ circuits. In [19] the above variable reduction strategy was achieved in the *white-box* setting, where the circuit is given explicitly as input and one can inspect the linear functions appearing in it. Later, building on [25], Karnin and Shpilka obtained a corresponding variable reduction procedure in the *black-box* setting, where the algorithm only has query access to the circuit [42].

The work [19] also highlighted a connection between rank bounds for depth-3 identities and discrete geometry, specifically to colored versions of the classical Sylvester-Gallai theorem. In particular, [19] suggested that insights from discrete geometry could lead to improved rank bounds. This direction was effectively pursued by Kayal and Saraf [44], who observed that one of the conjectures posed in [19] corresponds to the Edelstein-Kelly theorem [20]. By applying higher dimensional incidence theorems, [44] obtained the first rank bounds which depended only on the top fan-in k. Subsequent research further improved the upper bound on $R(k)$ culminating in $R(k) = 3k^2$ over fields of characteristic zero [64].

Motivated by the success of rank-bounds for depth-3 circuits, Beecken, Mittmann and Saxena [11] conjectured that if one defines an analogous notion of rank for depth-4 circuits, replacing linear dimension with algebraic rank, then "nontrivial" depth-4 identities should have small rank. Motivated by the connection between depth-3 identities and questions in incidence geometry, Gupta suggested far-reaching generalizations of the Sylvester-Gallai theorem and its colored and high-dimensional variants [35]. These conjectures suggest a program of generalizing incidence geometry from studying intersections of lines, hyperplanes or curves, to studying intersections between zero sets of low-degree polynomials (or more generally sparse polynomials).

The first progress on these conjectures was made by Shpilka [66], who established an analogue of the Sylvester-Gallai theorem for quadratic polynomials. Subsequent work gradually confirmed several other conjectures of Gupta [57, 58, 59, 54, 29, 27, 55, 31, 30]. The results of [58, 31], combined with [11, Theorem 2], immediately yield black-box PIT algorithms for $\Sigma^3\Pi\Sigma\Pi^d$ circuits, for $d = 2$ and more generally for any $d = O(1)$, respectively.

In this work we obtain the first polynomial time black-box PIT algorithm for $\Sigma^k\Pi\Sigma\Pi^2$ circuits, for any constant top fan-in k. We achieve this by combining the conceptual approach of [44] with the techniques developed in [55]. This allows us to prove an upper bound on the rank of $\Sigma^k\Pi\Sigma\Pi^2$ identities. Before stating our main results we discuss the relation between PIT and (non-linear) incidence geometry.

1.1 Edelstein-Kelly Theorem and Small Depth Identities

The Sylvester-Gallai theorem, posed independently by Sylvester [69] and Erdös [21], and solved independently by Melchior [52] and Gallai [26], asserts that if a finite set of points in $\mathbb{R}^n$ has the property that every line passing through any two points in the set also contains a third point from the set, then all the points in the set are collinear. In [46], Kelly observed that a result of Hirzebruch [40] proves that over $\mathbb{C}^n$ the points must lie on a 2-dimensional affine space, thereby resolving a question of Serre [65]. Many variants of this theorem were studied: extensions to higher dimensions, colored versions, robust versions and many more. For a survey on the Sylvester-Gallai theorem and its variants see [13]. Of particular relevance to PIT is the Edelstein-Kelly theorem [20], a colored variant of the Sylvester-Gallai theorem. We begin by defining the notion of an Edgelstein-Kelly configuration.

▶ **Definition 1** (Edelstein-Kelly configurations). *Let $\mathbb{K}$ be a field, $\mathcal{A} \coloneqq \{u_1, \ldots, u_a\}$, $\mathcal{B} \coloneqq \{v_1, \ldots, v_b\}$ and $\mathcal{C} \coloneqq \{w_1, \ldots, w_c\}$ be pairwise disjoint subsets of $\mathbb{P}(\mathbb{K}^N)$. We say that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ forms a* linear Edelstein-Kelly configuration *if any pair of points taken from distinct sets is collinear with a point from the third set. The* rank *of an Edelstein-Kelly configuration $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is given by* $\dim \operatorname{span}_{\mathbb{K}}\{\mathcal{A} \cup \mathcal{B} \cup \mathcal{C}\}$.

The Edelstein-Kelly theorem asserts that, for $\mathbb{K} = \mathbb{R}$, the rank of any linear Edelstein-Kelly configuration is at most 3. Over $\mathbb{C}$, the fractional Sylvester-Gallai theorems of [10, 18, 17] imply that the dimension is bounded by some absolute constant.

Consider now the simplest nontrivial $\Sigma^3\Pi\Sigma$ identity over $S \coloneqq \mathbb{K}[\mathbf{x}]$:

$$\prod_i \ell_{1,i}(\mathbf{x}) + \prod_j \ell_{2,j}(\mathbf{x}) + \prod_k \ell_{3,k}(\mathbf{x}) \equiv 0\,, \tag{1}$$

where the $\ell_{a,b}$ are linear forms. By factoring out the common gcd we may assume that the products in (1) are pairwise prime. Indeed, if a form divides two terms then it must divide the third term as well. The *rank* of the identity is defined as $\dim \operatorname{span}_{\mathbb{K}}\{\ell_{a,b}\}_{a,b}$. Since

the three terms sum to zero, if we pick any two linear forms from distinct terms, say $\ell_{1,i}$ and $\ell_{2,j}$, and restrict to the subspace $U = \{\mathbf{a} : \ell_{1,i}(\mathbf{a}) = \ell_{2,j}(\mathbf{a}) = 0\}$, we obtain that also $(\Pi_k \ell_{3,k}(\mathbf{x}))\,|_U \equiv 0$. In other words, if we consider the corresponding zero sets,[1]

$$V(\ell_{1,i}, \ell_{2,j}) \subseteq V\left(\prod_k \ell_{3,k}(\mathbf{x})\right) = \bigcup_k V(\ell_{3,k}(\mathbf{x})). \tag{2}$$

Since $V(\ell_{1,i}, \ell_{2,j})$ is irreducible[2] there exists some $k = k(i,j)$ such that $\ell_{3,k} \in \mathrm{span}_{\mathbb{K}}\{\ell_{1,i}, \ell_{2,j}\}$. Thus, any two linear forms from distinct terms span a linear form from the third term.

To fit this into the setting of Definition 1, identify each linear form $\ell_{a,b}$ with the point $p_{a,b} \in \mathbb{P}(\mathbb{K}^N)$ spanned by its coefficient vector $\mathbf{a}_{a,b}$. Then, $\ell_{3,k} \in \mathrm{span}\{\ell_{1,i}, \ell_{2,j}\}$ if and only if $p_{3,k}$ is collinear with $p_{1,i}$ and $p_{2,j}$. Hence, the sets $\mathcal{A} = \{p_{1,i}\}$, $\mathcal{B} = \{p_{2,j}\}$ and $\mathcal{C} = \{p_{3,k}\}$ satisfy the conditions of Definition 1. The Edelstein-Kelly theorem then implies that, for $\mathbb{K}$ of characteristic zero, all the points $p_{a,b}$ lie on an $O(1)$-dimensional space, and consequently, $\dim_{\mathbb{K}} \mathrm{span}\{\ell_{a,b}\} = O(1)$. In conclusion, this argument proves that any nontrivial (i.e., gcd-free) depth-3 identity depends essentially on only $O(1)$ variables, thereby enabling an efficient PIT algorithm.

Consider now a nontrivial $\Sigma^4\Pi\Sigma$ identity. Henceforth, by nontrivial we mean gcd-free and that any 3 terms are linearly independent (i.e., there is no shorter nontrivial identity):

$$\prod_{i=1}^{d} \ell_{1,i}(\mathbf{x}) + \prod_{i=1}^{d} \ell_{2,i}(\mathbf{x}) + \prod_{i=1}^{d} \ell_{3,i}(\mathbf{x}) + \prod_{i=1}^{d} \ell_{4,i}(\mathbf{x}) \equiv 0\,. \tag{3}$$

The natural attempt to bound the rank is to reduce to the case of three terms as in (1). For this we can consider the restriction to $U = \{\mathbf{a} : \ell_{1,1}(\mathbf{a}) = 0\}$. This restriction cancels out the first term. However, it may now be the case that after restricting to U, the resulting identity

$$\prod_{i=1}^{d} \ell_{2,i}(\mathbf{x})|_U + \prod_{i=1}^{d} \ell_{3,i}(\mathbf{x})|_U + \prod_{i=1}^{d} \ell_{4,i}(\mathbf{x})|_U \equiv 0$$

is no longer nontrivial. This complication becomes more evident as we increase the number of terms, and we will discuss it more in Section 1.3. Thus, moving from 3 terms to 4 already poses nontrivial challenges.

Let us now consider a $\Sigma^3\Pi\Sigma\Pi^d$ identity:

$$\prod_i A_i(\mathbf{x}) + \prod_j B_j(\mathbf{x}) + \prod_k C_k(\mathbf{x}) \equiv 0\,, \tag{4}$$

where each A_i, B_j, C_k is a homogeneous polynomial of degree $\leq d$, and, as before, we assume that (4) is gcd-free. In analogy to (2) we obtain[3]

$$V(A_i, B_j) \subseteq V\left(\prod_k C_k(\mathbf{x})\right). \tag{5}$$

Trying to repeat the previous argument, we immediately encounter a new difficulty. Since A_i and B_j are forms of degree higher than 1, their variety may not be irreducible. Equivalently,

[1] By symmetry, a similar containment holds for any two forms from distinct terms and the third term.
[2] This can be shown directly by simple linear algebra, but we use the more general terminology for later purposes.
[3] By symmetry, a similar containment holds for any two forms from distinct terms and the third term.

the ideal $I = (A_i, B_j) \subset \mathbb{K}[\mathbf{x}]$ is not necessarily prime. As a result, it may no longer be true that there exists $C_k \in \text{span}\{A_i, B_j\}$. Even worse, we are not guaranteed that there is a C_k such that $V(A_i, B_j) \subseteq V(C_k)$.

When $d = 2$, i.e., the forms are quadratic or linear polynomials, Peleg and Shpilka [57, 58] proved that if the ideal generated by two quadratic polynomials is not prime, then some of its associated primes have very simple form, and with that information [58] managed to extend the Edelstein-Kelly theorem to quadratic polynomials. Oliveira and Sengupta [54] generalized this approach to the case of cubic forms by characterizing non-radical ideals generated by two cubic forms. However, when $d > 3$ the situation becomes significantly more complex and there is no such "structure theorem" for ideals generated by two higher degree forms. On the other hand, structure theorems such as the primality conditions in [30], allowed [30] to control the number and structure of "bad events". To use this result inductively, Garg, Oliveira and Sengupta [55, 31] asked more difficult questions, but which are more amenable to induction. Instead of considering dependencies over $S = \mathbb{K}[\mathbf{x}]$, they considered dependencies over quotient rings $R = S/(U)$ where U is a vector space of forms up to degree d, such that R is a unique factorization domain (UFD). In addition, and to capture (and generalize) the technique of projection introduced in [44, 66] and generalized in [55], [31] gave the following definition that greatly generalizes Definition 1

▶ **Definition 2** (High-degree EK configurations). *Let $U \subseteq S$ be a graded, finitely generated vector space such that $R := S/(U)$ is a UFD, and let $z \in R_1$. Let $\mathcal{A}, \mathcal{B}, \mathcal{C} \subset R$ be pairwise disjoint finite sets of irreducible forms of degree at most d. We say that $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ is a* (d, z, R)–EK configuration *if the following hold:*

1. *$z \notin \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$, and the union $\{z\} \cup \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}$ consists of pairwise non-associate forms.*
2. *For every $A \in \mathcal{A}$ and $B \in \mathcal{B}$, we have*

$$z \cdot \prod_{C \in \mathcal{C}} C \in \text{rad}(A, B),$$

and similarly for any two forms from any two distinct sets, where $\text{rad}(A, B)$ is the radical of the ideal (A, B) in the quotient ring R.

While this definition is more intricate than Definition 1, it has the important advantage of greater flexibility. In particular, it allows one to add forms to U while remaining within the same framework. Their high level approach was to prove an upper bound on the rank using induction. The case $d = 1$ can be handled as in previous work, even though the setting is more general. Now, if all the forms in the identity Equation 4 are of degree at most $d - 1$, then we are done by induction. Otherwise, they consider two cases. The first is when there is a form $A \in \mathcal{A}$ such that for any other form $B \in \mathcal{B}$ (similarly, for $C \in \mathcal{C}$), $\text{span}\{A, B\}$ does not contain any form of small "strength". That is, no nonzero linear combination of A, B can be written as a sum of a few terms of the form $P_i P_i'$, where $\deg(P_i), \deg(P_i') < d$. [31] prove that in such a case, we can add A to U and by going to $R' = S/(U, A)$, obtain a simpler identity. In the other extreme case, assume that every form in the identity has small strength. Moreover, let us assume that it depends on a few variables. Consider now a forms $A \in \mathcal{A}$. If for many forms $B_j \in \mathcal{B}$, the ideal (A, B_j) is prime, then A spans a form C_k with each of these "good" B_j. If this holds for most $A \in \mathcal{A}$ then we can conclude by a *robust* version of the linear Edelstein-Kelly theorem. If this is not the case, then the following important structural result of [30, 31] is needed: there is a universal upper bound, depending only on the degree d, on the number of irreducible forms P such that (A, P) is not a prime ideal. Thus, modulo A, many forms in $\mathcal{B}$ become reducible, and we've made progress. Furthermore, since A depends on a few variables, we can pass to a smaller polynomial ring in which these forms factor.

While the description above greatly simplifies the approach of [31], it highlights some challenges that arise when trying to go from three terms to more terms and from linear forms to higher degree forms.

1.2 Our results

In this work, we overcome the challenges posed by depth-4 identities of constant top fan-in, at the expense of restricting the bottom fan-in to be bounded by 2.

Before we state our main result, we need to establish standard definitions of "nontrivial identities" for $\Sigma^k\Pi\Sigma\Pi^2$ circuits. As we mentioned in the previous subsection, one way in which an identity $\Phi = T_1 + \cdots + T_k$ can be trivial is if each gate is a multiple of a quadratic (or linear) polynomial. To disallow such triviality, we say that Φ is a simple circuit if $\gcd(T_1, \ldots, T_k) = 1$. Another way to obtain a trivial identity is by adding two smaller identities together. To disallow such triviality, we say that Φ is a minimal circuit if no subset of its gates sum to zero.

We also need to define the rank of a circuit. In this work, the rank of a $\Sigma^k\Pi\Sigma\Pi^2$ circuit $\Phi = T_1 + \cdots + T_k$ is the dimension of the $\mathbb{K}$-span of all the forms appearing in any of the gates T_i. While in [11] the authors use an algebraic notion of rank of a depth-4 circuit, we use a stricter (and simpler to state) notion of rank of a circuit, which in particular upper bounds the notion of rank in [11]. With these definitions in mind, we can state our main theorem.

▶ **Theorem 3.** *Let $\mathbb{K}$ be an algebraically closed field of characteristic zero and $S := \mathbb{K}[x_1, \ldots, x_n]$. There is a function $R : \mathbb{N} \to \mathbb{N}$, independent of $\mathbb{K}, n$, such that for any simple and minimal $\Sigma^k\Pi\Sigma\Pi^2$ identity Φ over S, we have* $\operatorname{rank}(\Phi) \leq R(k)$.

The above theorem makes progress on [11, Conjecture 28], being the first work to obtain bounds for circuits with top fan-in larger than 3. Combining the above theorem with [11, Theorem 2], we obtain the following corollary.

▶ **Corollary 4.** *There is a deterministic, polynomial-time black-box PIT algorithm for $\Sigma^k\Pi\Sigma\Pi^2$ circuits.*

Theorem 3 is a corollary of our main technical result [28, Theorem 6.1]. In [28, Section 6] we prove our main technical result and prove Theorem 3 as a corollary. An important ingredient in the proof of Theorem 3 (more accurately, of [28, Theorem 6.1]) is an extension of Hansen's theorem to quadratic polynomials [38], which we prove in [28, Section 4].

Our result both extends Hansen's theorem to the quadratic setting and relaxes its exact linear dependence condition to an approximate one. This can also be viewed as a high-dimensional quadratic analogue of the approximate collinear Sylvester–Gallai theorem of Ai, Dvir, Saraf, and Wigderson [5].

To state the theorem, we first introduce the notion of an (M, b)-strong configuration. The following simplified version captures the main idea, while the precise definition and theorem appear in [28, Section 4].

▶ **Definition 5.** *We say that a finite set of quadratic polynomials $\mathcal{F}$ is an (M, b)-strong span configuration if, for any $b-1$ polynomials $Q_1, \ldots, Q_{b-1} \in \mathcal{F}$, every other $Q \in \mathcal{F}$ satisfies one of the following:*

1. $|\operatorname{span}_{\mathbb{K}}\{Q_1, \ldots, Q_{b-1}, Q\} \cap \mathcal{F}| > b$, *or*
2. *Q is M-close to* $\operatorname{span}_{\mathbb{K}}\{Q_1, \ldots, Q_{b-1}\}$.

Here, M-closeness means that there exists $F \in \operatorname{span}_{\mathbb{K}}\{Q_1, \ldots, Q_{b-1}\}$ such that the quadratic form $Q - F$ has rank less than M.

We now present a simplified version of [28, Theorem 4.6].

▶ **Theorem 6.** *For any (M, b)-strong span configuration $\mathcal{F}$, there exists a subspace V of dimension $O_b(1)$ such that every quadratic form in $\mathcal{F}$ is $O_b(M)$-close to V. In particular, the dimension of V is independent of the number of variables, of $|\mathcal{F}|$, and of M.*

Interestingly, for the proof of Theorem 6 we actually first need to prove a *robust* version of it, in the spirit of the robust version of Hansen's theorem proved in [9]. We think that this result is also interesting on it own. We now state a simplified version of [28, Definition 4.7].

▶ **Definition 7.** *Let $0 < \nu \leq 1$ We say that a finite set of quadratic polynomials $\mathcal{F}$ is a (ν, M, b)-strong span configuration if, for any $b-1$ polynomials $Q_1, \ldots, Q_{b-1} \in \mathcal{F}$, for ν fraction of the remaining $Q \in \mathcal{F}$ the following holds:*

1. $|\mathrm{span}_{\mathbb{K}} \{Q_1, \ldots, Q_{b-1}, Q\} \cap \mathcal{F}| > b$, *or*
2. Q *is M-close to* $\mathrm{span}_{\mathbb{K}} \{Q_1, \ldots, Q_{b-1}\}$.

We now give a simplified version of [28, Theorem 4.10].

▶ **Theorem 8.** *For any (ν, M, b)-strong span configuration $\mathcal{F}$, there exists a subspace V of dimension $O_b(1/\nu)$, such that $O_b(\nu)$ fraction of the forms in $\mathcal{F}$ are M-close to V.*

We now proceed to describe an outline of our proof of Theorem 3.

1.3 Proof outline

As we previously mentioned, our approach generalizes the framework of [44] via the techniques developed in [55, 30]. In this subsection, we present our proof strategy and its implementation, emphasizing the conceptual contributions and the technical challenges that we overcome in developing the inductive framework that underlies our rank bound proof for $\Sigma^k\Pi\Sigma\Pi^2$ identities. To better understand our proof, we briefly outline the inductive approach of [44] to construct their rank upper bound function $R : \mathbb{N} \to \mathbb{N}$ for $\Sigma^k\Pi\Sigma$ circuits.

1.3.1 Overview of the framework in [44]

When $k \leq 2$, only the trivial circuit is a simple and minimal identity, by the unique factorization property in polynomial rings. This gives us a trivial starting upper bound on the rank of such identities. In particular, we can define $R(0) = R(1) = R(2) = 1$.

Now, suppose that we have constructed our function R up to $k-1$, where $k \geq 3$, and let $\Phi = T_1 + \cdots + T_k$ be our simple and minimal $\Sigma^k\Pi\Sigma$ identity, where each $T_i = \prod_{j=1}^{m} \ell_{ij}$ (note that ℓ_{ij}'s may or may not be scalar multiples of one another). We can still use unique factorization of the polynomial ring to quantify "how different two distinct gates are." This motivated the authors to define the "symmetric difference" of two gates as follows: $T_i \Delta T_j$ equals the set of linear forms (up to scalar multiples) which appear in T_i and T_j with different multiplicities (if $\ell \nmid T_i$ then we say that the multiplicity of ℓ in T_i is zero). Note that the size of the symmetric difference is not too important as a parameter, but a better choice of parameter is the dimension of the span of this set, i.e. $\dim \mathrm{span}_{\mathbb{K}} \{T_i \Delta T_j\}$, as this quantifies the "distinct number of variables" that distinguish the two gates. This leads to the definition of *pairwise rank* of Φ: it is defined as $\min_{1 \leq i < j \leq k} \dim \mathrm{span}_{\mathbb{K}} \{T_i \Delta T_j\}$.

If two gates T_i and T_j "are almost the same," that is, $\dim \mathrm{span}_{\mathbb{K}} \{T_i \Delta T_j\}$ is "small enough," then it is possible to "merge" these two gates via a *general quotient* by the linear forms in $T_i \Delta T_j$. This essentially preserves simplicity and minimality of the circuit, and since we are quotienting by linear forms the new circuit is still over a polynomial ring, in which

case we would get an $\Sigma^{k-1}\Pi\Sigma$ identity, for which we know rank bounds by induction. This would allow us to lift the rank bounds. If one believes that such rank bounds ought to exist, then in particular it must be the case that any two gates should "depend on almost the same variables", in which case the above scenario must always happen. The main strategy of [44] is to show that this case always happens, and it takes up most of the work in [44].

Assume we are not in the above case, that is, we have a simple and minimal circuit $\Phi = T_1 + \cdots + T_k$ in which *every* pair of gates T_i, T_j are "far enough apart." We would like to show that Φ is not an identity. Since we have an inductive proof, we assume that we know rank bounds for $\Sigma^{k-1}\Pi\Sigma$ identities. One way to show this is via a *fan-in reduction lemma*: to show the existence of a linear form ℓ which appears in one of the gates of the circuit, such that by "zeroing out" ℓ – in other words, work over the quotient ring $\mathbb{K}[\mathbf{x}]/(\ell)$, which is still a polynomial ring – the gates of our new circuit $\overline{\Phi}$ (which will now have at most $k-1$ gates) are still "far enough apart from one another", which implies that $\operatorname{rank}(\overline{\Phi})$ is too large for it to be an identity.

This linear form ℓ, if exists, is rather special, as it could happen that by taking a quotient (which is not "general"), the new circuit is no longer simple and minimal. One terrible thing that could happen because of failure of simplicity is that the simple part of the new circuit "is missing" many linear forms from the original circuit – this could happen if many forms "go into the GCD" after the quotient. For instance, if all gates but two in our circuit were divisible by ℓ and the remaining two gates are $\prod_{i=1}^m x_i - \prod_{i=1}^m (\ell + x_i)$, when we quotient by ℓ these two gates become an identity with trivial simple part, but our bounds say nothing about the rank of the GCD (which in this case is everything!).

One ingredient in the fan-in reduction lemma of [44] is Hansen's incidence theorem [38], which in essence states that if a set $\mathcal{F}$ of linear forms is such that $\dim \operatorname{span}_{\mathbb{K}} \{\mathcal{F}\}$ is large, then it must contain a $(k+1)$-subset of forms $\ell_1, \ldots, \ell_{k+1} \in \mathcal{F}$ such that $|\operatorname{span}_{\mathbb{K}} \{\ell_1, \ldots, \ell_{k+1}\} \cap \mathcal{F}| = k+1$.[4] In other words, these forms do not span any other form in $\mathcal{F}$. With this theorem in mind, [44] proceed as follows: since the pairwise rank of Φ is large enough (as a function of k), one can construct a large enough "core" $\mathcal{C} := \bigcup_{i,j} \mathcal{C}_{ij}$ of linear forms, where each $\mathcal{C}_{ij} \subset T_i \Delta T_j$ has "enough distinct variables" appearing in the symmetric difference. The objective of having this core is that we want to prove that there is a linear form ℓ in the circuit that "does not affect any subcore $\mathcal{C}_{ij}$" if $\ell \nmid T_i \cdot T_j$. If one achieves this, we will be able to quotient by (ℓ), thereby reducing fan-in of Φ, and preserving "too many variables" in the symmetric differences, which implies that the new circuit $\overline{\Phi}$ will have too large of a rank, in which case it cannot be an identity.

The above summarizes the core ideas behind (a simplification of) the approach of [44], but several challenges present themselves in implementing such approach. In general they are not always able to preserve the core itself, but a "proxy core," which emerges from the interaction of the core with the remainder of the circuit. This will be the same for us in our generalization of their approach.

1.3.2 Generalizing the above approach

Several new challenges arise in the above approach when trying to generalize it to the setting of $\Sigma^k\Pi\Sigma\Pi^2$ circuits, as we have to deal with quadratic forms, in addition to linear forms. These challenges are both algebro-geometric and combinatorial. We now outline the main conceptual and technical challenges we face, and our approaches to overcome them.

[4] Note that Hansen's theorem is over $\mathbb{K} = \mathbb{R}$ only, but a generalization of it is proved over any characteristic zero field in [10, 18].

1. In order to reduce the top fan-in, we may have to "zero out" a set of quadratic forms, that is, work over a polynomial ring modulo an ideal with quadratic generators. This forces us to work with rings which are not polynomial rings. If one is not careful, such rings will not even be unique factorization domains (UFDs), which makes the entire approach collapse.

   We overcome this issue by building on the concept of *general quotients* developed in previous works [44, 66, 55, 30, 31]. While the general quotients developed in these previous works enable us to reduce the top fan-in in certain situations, they are not able to handle the case where we want to set a *particular* quadratic form to zero. To achieve this, we develop the notion of *targeted quotients* and establish some of its algebro-geometric properties, which will allow us to handle all the necessary quotients that "zero out" a particular quadratic form. We establish the properties that we need from our quotients in [28, Section 3.5].

   Moreover, to be able to always work within "nice enough rings", we invoke the strong algebras and vector spaces developed in [6, 55]. Analogously to previous works, we generalize the notion of $\Sigma^k\Pi\Sigma\Pi^2$ identities to rings of the form $S/(U)$ where S is a polynomial ring and U is a strong enough vector space in S. This generalized notion of identities allows our inductive approach more flexibility to handle the general and targeted quotients that we need to apply in order to reduce the top fan-in. However, unlike in previous works, it is important for us to be able to define the concept of "essential variables" inside of quotient rings, as these variables allow us to exploit the geometric techniques coming from absolute irreducibility. We define strong vector spaces and algebras in [28, Section 3], and define and establish properties of essential variables in quotient rings in [28, Section 3.4].

2. In [44], the drop in pairwise rank of the circuit can only happen due to linear relations between linear forms. This allows the authors to both argue about large pairwise rank with Hansen's theorem and to control the drop in pairwise rank by preserving non-colinearity relations. However, the presence of quadratic forms yields more intricate algebro-geometric phenomena, which makes the understanding of the behaviour of pairwise rank more complex.

   We overcome this issue by proving structural results on primality of certain special ideals generated by quadratic forms using absolute irreducibility over algebras, in a similar way as was done in [55, 30, 31]. Since we are dealing with quadratic forms only, the primality results we need are simpler and stronger, and thus we do not need to use the primality results in these works. Moreover, we use resultant-based methods to control the drop in pairwise rank under targeted quotients. Once we find a desired set of forms that we want to quotient out, both general quotients and targeted quotients will take care of pairwise rank preservation under quotients. It is important to note that, unlike for general quotients, we do not need to prove any lifting results for the targeted quotients. This phenomenon happens because of the same reasons as its linear analogue in [44]: if the pairwise rank is large enough (by assumption), then our circuit must not be an identity. Hence, to obtain the rank bounds one only need to lift the general quotients that one applies when dealing with the cases of small pairwise rank.

3. Unlike the linear case, we do not have at hand an algebro-geometric analog of Hansen's theorem. Note that the main result of [27] does not seem to be the right analog to what is needed in our case, since they only handle radical dependencies (whereas we would need product dependencies), and they do not handle the same combinatorial conditions as Hansen's theorem.

We overcome this issue in a 2-step approach: in the first step, we "control" the high strength quadratics (the ones that "depend on too many variables") in any $\Sigma^k\Pi\Sigma\Pi^2$ identity by reducing our identity to a non-trivial $\Sigma^k\Pi\Sigma\Pi^2$ identity where all the quadratics have bounded strength (i.e., "depend on few variables"). Interestingly enough, to control the high strength quadratics, we need to combine the approach of [44] to reduce the top fan-in – which is done in [28, Section 4.1] – together with a Hansen-like result, which we establish in [28, Theorem 4.6]. Note that [28, Theorem 4.6] is different from the statements of [27, Section 6.2] (where they consider configurations of strong forms), since in our case we have to handle a Hansen-like condition, whereas [27] only requires to handle a stronger condition, similar to the SG_k^* conditions from [10]. We establish our Hansen-like result by reducing our strong span configurations to fractional configurations, which then we can further reduce to several fractional Sylvester-Gallai configurations, with an approach inspired by [10]. For details, see [28, Section 4] for the entire approach and [28, Section 4.2] for our Hansen-like result.

Once we complete the strength reduction above, the second step is to prove rank bounds for $\Sigma^k\Pi\Sigma\Pi^2$ identities where all quadratics "depend on few variables." In this case we are able to combine our primality structure theorems similar to the ones from [30, 31] and the integral sequences developed in [27] with the approach of [44] described above and construct a core, along with a "proxy core," which allows us to upper bound the rank of such $\Sigma^k\Pi\Sigma\Pi^2$ identities.

While the above outlines how we overcome the challenges posed by working with quadratic and linear forms, we now elaborate on our technical contributions and how they compare with previous works.

1.3.3 Elaboration of our techniques

The two main conceptual and technical contributions of this work are the concept of *targeted quotients* and *essential variables in (strong) quotient rings*. Once we have these two concepts and their properties established, we can execute our strategy above with a substantial amount of intricate combinatorial work.

- **Targeted quotients:** Previous works, such as [44, 66, 58, 55, 31], only required working with general quotients – where forms are mapped to general multiples of (appropriate powers of) a special linear form z. These quotients have very useful properties, and allowed these works to reduce the complexity of their main objects of study (polynomial identities or Sylvester-Gallai type configurations). However, in [44] the authors need to also quotient by especially chosen linear forms. This type of quotient was not captured (and not needed) in subsequent works [66, 58, 55, 31], but it is crucial here. While the definition is clear on what quotient we will need for linear forms and for high strength quadratics, as these essentially behave "as variables in a polynomial ring," it is much less clear what to do in the case where we need to make a quadratic of low strength vanish "in a generic way" that does not substantially affect the other forms in the circuit. The latter case of a low strength quadratic F is addressed by our notion of targeted quotients, whose idea is as follows: if F only depends on the variables $x_1, \ldots, x_r$, to "make F vanish", we need to set the variables $x_1, \ldots, x_r$ to a zero of F (i.e. a point in the variety defined by F) in "a generic way." The way to obtain such a generic zero of F is to simply pick a general point $(\alpha_1, \ldots, \alpha_r)$ in the zero set of F and set $x_i \mapsto \alpha_i z$, where z is a new variable. It turns out that this choice of quotients map F to zero, and at the same time "preserves" the algebraic and geometric properties of other forms that we need. We also show that

the only instance where another form G is not preserved under such quotients is when G also depends only on the variables $x_1, \ldots, x_r$, and in this case, we also show that G will be sent to zero iff it is a multiple of F. [28, Proposition 3.42] establishes the properties we need from targeted quotients and in [28, Section 5] we show how such quotients interact with the other forms in our circuit.

- **Essential variables in quotient rings:** previous works on Sylvester-Gallai type theorems involving quadratics [66, 57, 58, 59, 29] heavily used the fact that they were working over polynomial rings, which in particular was very useful when defining the space of linear forms that defined a quadratic. In this setting, the uniqueness of representation of any polynomial in the polynomial ring was key in their definitions of the relative linear space of forms of a quadratic of low strength over an algebra. However, in our more general setting, where we must work with quotients of a polynomial ring by an ideal generated by a strong vector space, it is much less clear how to define the relative space of a quadratic element. The main difficulty stems from the fact that an element of the quotient ring has many distinct representatives in the corresponding polynomial ring, and in our setting we must make sure that any such representative must be compatible with a unique space of linear forms. To address this issue, in [28, Section 3.4], we establish some necessary results needed in order to define two key concepts: the *pseudo-distance* of a quadratic to an algebra in a quotient ring ([28, Definition 3.28]) and then after establishing some results about the pseudo-distance, we can finally define the *relative space of essential variables* ([28, Definition 3.32]), which generalize the related notions in previous works. With these notions and their properties at hand, we can analyze the case where all of our quadratics have bounded strength, which we do in [28, Section 5]. In this section, we crucially use the relative space of essential variables, combined with the geometric properties established in previous sections, to obtain our rank bounds.
- **Controlling Hansen-like configurations:** in our reduction from general $\Sigma^k\Pi\Sigma\Pi^2$ identities to the case of $\Sigma^k\Pi\Sigma\Pi^2$ identities where all quadratics are close to a small strong algebra, one important step to control the strong quadratics (i.e. the ones depending on many variables) is our Hansen-like result [28, Theorem 4.6]. This result bounds the size of a subset needed to bring a set of forms satisfying certain Hansen-like conditions (see [28, Definition 4.2]) "close to a given algebra." Although the result of [28, Theorem 4.6] is stated in a manner which seems specific to depth-4 identities, note that [28, Definition 4.2] can be stated independently of the setting of depth-4 identities. We believe that this result is also an interesting geometric-combinatorial result on its own right, as it allows one to control such sets of forms. A weaker statement was proved in [27, Section 6], where the authors handle a geometric-combinatorial condition similar to the SG_k^* configurations defined in [10, Definition 5.1]. In our case, our configurations are more related to the fractional configurations generalizing Hansen, given in [10, Definition 5.2].

1.3.4 Choice of approach, challenges of higher degree and open questions

Given the above overview and the history about rank bounds in the depth-3 case, one may wonder what makes the approach of [44] the preferred one in this paper. One of the main reasons that motivated this choice is the fact that the approach of [44] proves rank bounds by controlling pairs of gates, whereas the other approaches make simultaneous use of more gates in their analysis. The latter approaches require geometric results involving ideals generated by several forms (of potentially unbounded degree), such as the ideal chinese remaindering results of [45, 64], whereas the approach of [44] only requires the use of geometric statements

about ideals generated by forms of constant degree. Extending these other approaches is an interesting open problem, and the geometric statements needed to do so may be important in their own right.

Another question that may arise, in light of the recent rank bounds for $\Sigma^3\Pi\Sigma\Pi^d$ circuits and structural results for ideals generated by pairs of degree d forms from [30, 31], is what makes the approach above not work for bottom fan-in larger than 2. The main issue in extending the above result to bottom fan-in larger than 2 is the fact that we currently lack the correct analogue of "essential variables" when decomposing a low strength form of degree larger than 2, Note that, even for the case of degree 3, a minimum decomposition of a low strength form may have quadratic factors, which may interact non-trivially with the remainder of the forms in the circuit. On the other hand, in our case, once we establish the uniqueness of the space of linear forms that we will quotient by, their infinite strength (as they are truly variables in the polynomial ring) allows us to argue that they do not interact non-trivially with any other linear form lying outside of the given space. Generalizing our notion of essential variables to the higher degree setting is an interesting open question, as it would yield a valuable tool which can have uses not only for PIT but also for other problems in the computational complexity of other computer algebra problems.

Lastly, we conclude by emphasizing that a full resolution of [11, Conjecture 28] is the biggest problem left open by this work: proving rank bounds for $\Sigma^k\Pi\Sigma\Pi^d$ circuits which depend only on k, d.

1.3.5 Flow of argument for rank bound

Starting assumption: rank bounds for simple and minimal $\Sigma^{k-1}\Pi\Sigma\Pi^2$ identities over rings $S/(U)$, where U is a strong enough vector space (the required lower bound on the strength of U is a function of $k-1$).

Suppose we are given a simple minimal $\Sigma^k\Pi\Sigma\Pi^2$ identity $\Phi = T_1 + \cdots + T_k \equiv 0$ in the ring $R = S/(U)$, where U is a strong enough vector space.

- Step 1: Are two gates close to each other? If so then apply a general quotient (see [28, Section 3.5]) to merge two gates, and invoke inductive rank bounds for circuits with top fan-in $k-1$. Then lift the rank bound to deduce a rank bound for Φ using [28, Proposition 3.41].
  Here the measure of closeness of two gates is "pairwise rank" or the size of their symmetric difference (see [28, Lemma 4.1], item 1 for quantitative details).
- Step 2: Suppose that no two gates are close to each other, i.e. all pairwise ranks are high.
  Case 2.1. (Fanin-reduction) If there exists a strong form H in some gate T_i such that after applying a quotient by H, the resulting circuit has high pairwise rank, then we obtain a contradiction to the starting assumption.
  Case 2.2. If there is no strong form H with the property above, then we have a (M, b)-strong configuration (see [28, Section 4.1] where we prove the contrapositive: a fanin-reduction lemma). This argument follows the strategy of Kayal and Saraf [44].
- Step 3: If a set of forms is an (M, b)-configuration, then there is a vector space V such that all forms in the set are close to V (see [28, Section 4.2]). This result is itself proved by induction. Refer to [28, Lemma 4.1], item 3, [28, Theorem 4.6], [28, Theorem 4.10] for qualitative versions that prove approximate and robust Hansen-type results.
  Therefore, all the forms in the circuit Φ are close to some vector space V.
- Step 4: Now, all the forms in the circuit Φ are close to some vector space V, i.e. Φ is of bounded pseudo-distance with respect to V (see [28, Section 3.4] for the notions of essential variables and pseudo-distance in quotient rings).

In [28, Section 5] we handle such circuits and [28, Lemma 5.3] shows that we will land in one of the following cases, each handled separately. These statements are proved using a potential function (see [28, Section 5.1]) and an iterative process that decreases this potential (see [28, Section 5.3]).

Case 4.1. There exists a vector space Y such that some symmetric difference lies in the algebra generated by Y. Apply a general quotient with respect to Y and lift the inductive bounds.

Case 4.2. There exists a vector space Y such that either every quadratic form in Φ is absolutely reducible with respect to Y. Apply a general quotient with respect to Y to reduce the degree of forms in the configuration and use inductive rank bounds on $\Sigma^k\Pi\Sigma$ circuits similar to [44].

Case 4.3. (Targeted fanin-reduction). There exists a form H such that the targeted quotient with respect to H (see [28, Proposition 3.42]) results in a circuit with high pairwise rank (see [28, Section 5.4]). Then we have a contradiction to the starting assumption.

At the end of these steps, we have obtained a rank bound for $\Sigma^k\Pi\Sigma\Pi^2$ circuits.

1.4 Previous and related work

1.4.1 Polynomial identity testing

Depth-4 circuits with bounded top and bottom fan-in are the simplest known circuit class for which no deterministic polynomial-time algorithm for PIT is currently known. Consequently, this model has been studied using techniques that extend beyond the incidence geometry–based methods discussed earlier.

In [15], the authors devise and prove correctness of a quasipolynomial-time deterministic PIT algorithm for $\Sigma^k\Pi\Sigma\Pi^d$ circuits, leveraging the Jacobian method introduced in [4]. Their key insight involves applying the logarithmic derivative and its associated power series expansion to transform the top summation gate of the circuit into a powering gate. While this transformation technically violates the bounded top fan-in constraint, circuits with powering gates are well studied, and efficient PIT algorithms are known for such models. This reduction enables the use of existing tools for PIT in powering-gate circuits, ultimately leading to their quasipolynomial-time result.

The breakthrough work of [51] on proving lower bounds for bounded-depth arithmetic circuits offers another route toward derandomizing PIT for this model. The *hardness-versus-randomness paradigm* has been a powerful tool for establishing connections between circuit lower bounds and derandomization [2, 41]. Notably, the results of [14] extended these tradeoffs to the bounded-depth regime, showing that analogous principles continue to hold even in this restricted setting. By leveraging these developments, one obtains a *subexponential-time* PIT algorithm for depth-4 circuits. Building further on the lower bound techniques of [51], [7] constructed another *hitting set generator* tailored to bounded-depth circuits, yielding an alternative subexponential-time PIT algorithm with improved parameters. This construction exploits the fact that such circuits are inherently unable to detect low-rank matrices, due to the algebraic hardness of computing the determinant in low-depth settings. It is worth emphasizing, however, that neither of these approaches appears capable of yielding a *fully polynomial-time* PIT algorithm for $\Sigma^k\Pi\Sigma\Pi^d$ circuits.

Karnin et al. [43] and Saraf and Volkovich [60] studied *multilinear* $\Sigma^k\Pi\Sigma\Pi$ circuits and obtained quasi-polynomial- and subsequently polynomial-sized hitting sets, respectively. However, as their techniques crucially rely on multilinearity, none of these works can handle even $\Sigma^3\Pi\Sigma$ circuits.

Peleg and Shpilka [58] gave the first polynomial sized hitting sets for $\Sigma^3\Pi\Sigma\Pi^2$ circuits. As their technique relied on their structural theorem, which argues about the structure of ideals generated by two quadratic polynomials, they were not able to extend their proof beyond three terms, as more terms would involve ideals generated by three or more quadratics.

Garg et al. [27] studied high-dimensional Sylvester-Gallai configurations and obtained a generalization of both [38, 66]. This can be viewed as a problem analogous to ours, but in the *single color class* case, where each and every k-tuple of forms satisfies some nontrivial relation. In contrast, to obtain a PIT algorithm, we need to consider the case where forms belong to different sets and we can only argue about k-tuples coming from k disjoint sets.

Recently, two independent works [31, 34] have made progress in the PIT problem for $\Sigma^3\Pi\Sigma\Pi^d$ circuits. Note that progress in these works is made on the *bottom fan-in* of the circuit, while the current work focuses on the *top fanin* of depth-4 circuits. These two settings present different challenges, which must be overcome in order to fully resolve the PIT problem for $\Sigma^k\Pi\Sigma\Pi^d$ circuits.

In [34], the authors use normalization of algebraic varieties to give a deterministic, polynomial time PIT algorithm for $\Sigma^3\Pi\Sigma\Pi^d$ circuits over *any base field* (of any characteristic), but with the *additional assumption* that one summand of the top gate is *square-free* (that is, a multiplication of coprime polynomials). Their approach circumvents the need of incidence theorems and their generalizations, which provides an advantage over finite fields, as it is known that even in the case of depth-3 circuits over fields of positive characteristic, incidence-based approaches can only yield quasi-polynomial time algorithms. On the other hand, the requirement on one summand to be square-free seems to be very restrictive.

In [31, Theorem 1.4], as discussed above, the authors prove that there is a function $R_{EK} : \mathbb{N} \to \mathbb{N}$ such that any degree d EK configurations over the polynomial ring $\mathbb{K}[x_1, \ldots, x_n]$, where $\operatorname{char} \mathbb{K} = 0$, has rank upper bounded by $R_{EK}(d)$. As any simple and minimal $\Sigma^3\Pi\Sigma\Pi^d$ identity gives rise to a degree d EK configuration over $\mathbb{K}[x_1, \ldots, x_n]$, the EK rank bound implies a rank bound for these identities, and when one combines this result with [11, Theorem 2], one gets a deterministic, polynomial time algorithm for the PIT problem for this class.

1.4.2 High-dimensional and approximate variants of Sylvester-Gallai theorem

Let us now elaborate on the relation between Theorem 6, Hansen's theorem [38], the strong sequences of [27] and the result of Ai, Dvir, Saraf, and Wigderson [5].

Both [38, 5] consider configurations of points in Euclidean space, so in that context $\mathcal{F}$ should be viewed as a set of points rather than quadratic polynomials. Hansen requires that for every $b-1$ points $p_1, \ldots, p_{b-1}$ and any other point $p \in \mathcal{F}$, either the flat they define contains another point of $\mathcal{F}$, or that p lies exactly in the flat defined by $p_1, \ldots, p_{b-1}$. Thus, Hansen's condition enforces exact containment in the span rather than proximity to it.

Ai et al. consider an approximate version of this condition. Roughly speaking, an ϵ-approximate collinear Sylvester–Gallai configuration is a finite set of points $\mathcal{F} \subset \mathbb{R}^n$ that are neither "too close" nor "too far" from one another, such that for any two points in $\mathcal{F}$, there exists a third point in $\mathcal{F}$ that is ϵ-close to the line spanned by them. In other words, the standard Sylvester–Gallai requirement that the third point lie exactly on the line is replaced by an approximate one. The conclusion in this setting is that there exists a subspace V of dimension $O(1)$ such that all the points in $\mathcal{F}$ are $O(\epsilon)$-close to V. This is similar in spirit to our result, except that it concerns points rather than polynomials and corresponds to the case $b = 3$.

Hence, our Theorem 6 (and the formal [28, Theorem 4.6]) generalizes both Hansen's theorem and the result of Ai et al., yielding a quadratic, high-dimensional, approximate Sylvester–Gallai theorem.

Barak, Dvir, Wigderson, and Yehudayoff [9] proved a robust version of Hansen's theorem. They call a $(b-1)$-dimensional flat spanned by b points of $\mathcal{F}$ *elementary* if its intersection with $\mathcal{F}$ does not contain any other point. They showed that if, for every choice of $b-1$ points in $\mathcal{F}$, at least a ν-fraction of the remaining points are such that each of them either lies in the flat spanned by these $b-1$ points or spans with them a flat that is not elementary, then $\mathcal{F}$ is contained in an $O_b(1/\nu^2)$-dimensional flat. This was improved in [18, 17] to $O_b(1/\nu)$.

If we now replace points by quadratic polynomials, and the condition of belonging to a flat by being close to it, we obtain a setting similar to that of Theorem 8. The conclusion we derive is also analogous to that of [9]: to approximate all the quadratics, we must repeat the argument about $1/\nu$ times, resulting in an $O_b(1/\nu^2)$-dimensional subspace. The work of [27] generalizes the version of *ordinary* spaces defined by [10, Definition 5.1].

References

1 M. Agrawal and Manindra. Vinay. Arithmetic circuits: A chasm at depth four. *In 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28*, 2008.

2 Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science: 25th International Conference, Hyderabad, India, December 15-18, 2005. Proceedings 25*, pages 92–105. Springer, 2005. doi:10.1007/11590156_6.

3 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Annals of mathematics*, pages 781–793, 2004.

4 Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: Hitting sets, lower bounds for depth-d occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM Journal on Computing*, 45(4):1533–1562, 2016. doi:10.1137/130910725.

5 Albert Ai, Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Sylvester–gallai type theorems for approximate collinearity. In *Forum of Mathematics, Sigma*, volume 2, page e3. Cambridge University Press, 2014.

6 Tigran Ananyan and Melvin Hochster. Strength conditions, small subalgebras, and stillman bounds in degree ≤ 4. *Transactions of the American Mathematical Society*, 373(7):4757–4806, 2020.

7 Robert Andrews and Michael A. Forbes. Ideals, determinants, and straightening: proving and using lower bounds for polynomial ideals. In *54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 389–402, 2022. doi:10.1145/3519935.3520025.

8 Vikraman Arvind and Partha Mukhopadhyay. The ideal membership problem and polynomial identity testing. *Inf. Comput.*, 208(4):351–363, 2010. doi:10.1016/J.IC.2009.06.003.

9 Boaz Barak, Zeev Dvir, Avi Wigderson, and Amir Yehudayoff. Fractional sylvester-gallai theorems. *Proceedings of the National Academy of Sciences*, 110, 2013.

10 Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 519–528, 2011. doi:10.1145/1993636.1993705.

11 Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Information and Computation*, 222:2–19, 2013. doi:10.1016/J.IC.2012.10.004.

12 Michael Ben-Or and Prasoon Tiwari. A Deterministic Algorithm for Sparse Multivariate Polynominal Interpolation (Extended Abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 301–309. ACM, 1988. doi:10.1145/62212.62241.

13 Peter Borwein and William OJ Moser. A survey of sylvester's problem and its generalizations. *Aequationes Mathematicae*, 40(1):111–135, 1990.

14 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure results for polynomial factorization. *Theory of Computing*, 15(1):1–34, 2019. doi:10.4086/TOC.2019.V015A013.

15 Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. Deterministic Identity Testing Paradigms for Bounded Top-Fanin Depth-4 Circuits. In *Proceedings of the 36th Computational Complexity Conference*, CCC '21, 2021. doi:10.4230/LIPIcs.CCC.2021.11.

16 Pranjal Dutta and Sumanta Ghosh. SIGACT News Complexity Theory Column 121. *SIGACT News*, 55(2):53–88, June 2024. doi:10.1145/3674159.3674165.

17 Zeev Dvir, Ankit Garg, Rafael Oliveira, and József Solymosi. Rank bounds for design matrices with block entries and geometric applications. *Discrete Analysis*, 5(2018):1–24, 2018.

18 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Improved rank bounds for design matrices and a new proof of kelly's theorem. In *Forum of Mathematics, Sigma*, volume 2. Cambridge University Press, 2014.

19 Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007. doi:10.1137/05063605X.

20 Michael Edelstein and Leroy M Kelly. Bisecants of finite collections of sets in linear spaces. *Canadian Journal of Mathematics*, 18:375–380, 1966.

21 Paul Erdos, Richard Bellman, Hubert S Wall, James Singer, and Victor Thébault. Problems for solution: 4065-4069. *The American Mathematical Monthly*, 50(1):65–66, 1943.

22 Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Communications of the ACM*, 62(3):109–115, 2019. doi:10.1145/3306208.

23 Michael A. Forbes. Deterministic Divisibility Testing via Shifted Partial Derivatives. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 451–465. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.35.

24 Michael A Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 527–542. Springer, 2013. doi:10.1007/978-3-642-40328-6_37.

25 Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Comb.*, 28(4):415–440, 2008. doi:10.1007/S00493-008-2259-3.

26 Tibor Gallai. Solution of problem 4065. *American Mathematical Monthly*, 51:169–171, 1944.

27 Abhibhav Garg, Rafael Oliveira, Shir Peleg, and Akash Kumar Sengupta. Radical Sylvester-Gallai Theorem for Tuples of Quadratics. In *38th Computational Complexity Conference (CCC 2023)*, pages 20:1–20:30. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.CCC.2023.20.

28 Abhibhav Garg, Rafael Oliveira, Akash Sengupta, Nir Shalmon, and Amir Shpilka. Rank bounds and polynomial-time PIT for $\Sigma^k\Pi\Sigma\Pi^2$ circuits. *ECCC*, 2026. URL: https://eccc.weizmann.ac.il/report/2026/084.

29 Abhibhav Garg, Rafael Oliveira, and Akash Kumar Sengupta. Robust Radical Sylvester-Gallai Theorem for Quadratics. In *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:13, 2022. doi:10.4230/LIPIcs.SoCG.2022.42.

30 Abhibhav Garg, Rafael Oliveira, and Akash Kumar Sengupta. Uniform bounds on product sylvester-gallai configurations. In *41st International Symposium on Computational Geometry (SoCG 2025)*, pages 52:1–52:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.SoCG.2025.52.

31 Abhibhav Garg, Rafael Oliveira, and Akash Kumar Sengupta. Rank bounds and pit for $\Sigma^3\Pi\Sigma\Pi^d$ circuits via a non-linear edelstein-kelly theorem. *arXiv preprint*, 2025. doi:10.48550/arXiv.2504.14729.

32 Dima Grigoriev, Marek Karpinski, and Michael F. Singer. Fast Parallel Algorithms for Sparse Multivariate Polynomial Interpolation over Finite Fields. *SIAM J. Comput.*, 19(6):1059–1063, 1990. doi:10.1137/0219073.

33 Zeyu Guo. Variety Evasive Subspace Families. In Valentine Kabanets, editor, *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:33, 2021. doi:10.4230/LIPIcs.CCC.2021.20.

34 Zeyu Guo and Siki Wang. Deterministic Depth-4 PIT and Normalization. *arXiv preprint arXiv:2504.15143*, 2025. doi:10.48550/arXiv.2504.15143.

35 Ankit Gupta. Algebraic Geometric Techniques for Depth-4 PIT & Sylvester-Gallai Conjectures for Varieties. In *Electron. Colloquium Comput. Complex.*, volume 21, page 130, 2014.

36 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Unexpected power of low-depth arithmetic circuits. *Commun. ACM*, 60(6):93–100, 2017. doi:10.1145/3065470.

37 Rohit Gurjar and Thomas Thierauf. Linear Matroid Intersection is in Quasi-NC. *Comput. Complex.*, 29(2):9, 2020. doi:10.1007/S00037-020-00200-Z.

38 Sten Hansen. A generalization of a theorem of sylvester on the lines determined by a finite point set. *Mathematica Scandinavica*, 16(2):175–180, 1965.

39 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute. In *Proceedings of the twelfth annual ACM Symposium on Theory of Computing*, pages 262–272, 1980.

40 Friedrich Hirzebruch. Arrangements of lines and algebraic surfaces. In *Arithmetic and geometry*, pages 113–140. Springer, 1983.

41 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *computational complexity*, 13:1–46, 2004. doi:10.1007/S00037-004-0182-6.

42 Zohar S. Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333–364, 2011. doi:10.1007/s00493-011-2537-3.

43 Zohar Shay Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. Deterministic Identity Testing of Depth-4 Multilinear Circuits with Bounded Top Fan-in. *SIAM J. Comput.*, 42(6):2114–2131, 2013. doi:10.1137/110824516.

44 Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 198–207. IEEE, 2009. doi:10.1109/FOCS.2009.67.

45 Neeraj Kayal and Nitin Saxena. Polynomial Identity Testing for Depth 3 Circuits. *Computational Complexity*, 16(2):115–138, 2007. doi:10.1007/s00037-007-0226-9.

46 Leroy Milton Kelly. A resolution of the sylvester-gallai problem of j.-p. serre. *Discrete & Computational Geometry*, 1(2):101–104, 1986. doi:10.1007/BF02187687.

47 Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 216–223, 2001. doi:10.1145/380752.380801.

48 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. doi:10.1016/J.TCS.2012.03.041.

49 Mrinal Kumar and Ramprasad Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bulletin of EATCS*, 3(129), 2019.

50 Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-Optimal Bootstrapping of Hitting Sets for Algebraic Models. *Theory Comput.*, 19:1–30, 2023. doi:10.4086/TOC.2023.V019A012.

51 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 804–814. IEEE, 2022.

52 Eberhard Melchior. Uber vielseite der projektiven ebene. *Deutsche Math*, 5:461–475, 1940.

53 Ketan Mulmuley. Geometric complexity theory v: Efficient algorithms for noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.

54 Rafael Oliveira and Akash Kumar Sengupta. Radical sylvester-gallai theorem for cubics. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 212–220, 2022. doi:10.1109/FOCS54457.2022.00027.

55 Rafael Oliveira and Akash Kumar Sengupta. Strong algebras and radical sylvester-gallai configurations. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 95–105, 2024. doi:10.1145/3618260.3649617.

56 Rafael Oliveira, Amir Shpilka, and Ben Lee Volk. Subexponential Size Hitting Sets for Bounded Depth Multilinear Formulas. *Comput. Complex.*, 25(2):455–505, 2016. doi:10.1007/S00037-016-0131-1.

57 Shir Peleg and Amir Shpilka. A Generalized Sylvester-Gallai Type Theorem for Quadratic Polynomials. In *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:33, 2020. doi:10.4230/LIPIcs.CCC.2020.8.

58 Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via edelstein-kelly type theorem for quadratic polynomials. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 259–271. ACM, 2021. doi:10.1145/3406325.3451013.

59 Shir Peleg and Amir Shpilka. Robust sylvester-gallai type theorem for quadratic polynomials. In *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPIcs*, pages 43:1–43:15, 2022. doi:10.4230/LIPIcs.SoCG.2022.43.

60 Shubhangi Saraf and Ilya Volkovich. Black-Box Identity Testing of Depth-4 Multilinear Circuits. *Comb.*, 38(5):1205–1238, 2018. doi:10.1007/S00493-016-3460-4.

61 Nitin Saxena. Progress on Polynomial Identity Testing. *Bull. EATCS*, 99:49–79, 2009.

62 Nitin Saxena. Progress on polynomial identity testing-II. *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, pages 131–146, 2014.

63 Nitin Saxena and Comandur Seshadhri. Blackbox Identity Testing for Bounded Top-Fanin Depth-3 Circuits: The Field Doesn't Matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012. doi:10.1137/10848232.

64 Nitin Saxena and Comandur Seshadhri. From Sylvester-Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *Journal of the ACM (JACM)*, 60(5):1–33, 2013. doi:10.1145/2528403.

65 Jean-Pierre Serre. Advanced problem 5359. *Amer. Math. Monthly*, 73(1):89, 1966.

66 Amir Shpilka. Sylvester-Gallai type theorems for quadratic polynomials. *Discrete Anal.*, pages 34, Paper No. 13, 2020. doi:10.19086/da.

67 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010. doi:10.1561/0400000039.

68 Ola Svensson and Jakub Tarnawski. The Matching Problem in General Graphs Is in Quasi-NC. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.70.

69 James Joseph Sylvester. Mathematical question 11851. *Educational Times*, 59(98):256, 1893.

70 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Information and Computation*, 240:2–11, 2015. doi:10.1016/J.IC.2014.09.004.

Multiplicative Pseudorandom Generators for Nondeterministic Circuits

Alon Dermer
Department of Computer Science, University of Haifa, Israel

Ronen Shaltiel
Department of Computer Science, University of Haifa, Israel

Abstract

The hardness vs. randomness paradigm aims to construct pseudorandom generators (PRGs) based on complexity theoretic hardness assumptions. A seminal result in this area is a PRG construction by [27, 24]. A sequence of works [25, 30, 36, 31] generalized the result of [27, 24] to nondeterministic circuits, and showed that if $\mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ requires nondeterministic circuits of size $2^{\Omega(n)}$, then for every sufficiently large s, and every $\epsilon \geq \frac{1}{s}$, there is an ϵ-PRG $G : \{0,1\}^{r=O(\log s + \log \frac{1}{\epsilon})} \to \{0,1\}^s$ that runs in time $\mathsf{poly}(s)$, and fools size s nondeterministic circuits. In particular, for every size s nondeterministic circuit C,

$$\Pr[C(G(U_r)) = 1] \leq \Pr[C(U_s) = 1] + \epsilon.$$

Applebaum et al. [1] showed that "black-box techniques" cannot achieve such results for $\epsilon = s^{-\omega(1)}$. In order to circumvent this problem, Artemenko et al. [2] suggested a "multiplicative" version of PRGs, which requires that:

$$\Pr[C(G(U_r)) = 1] \leq 2 \cdot \Pr[C(U_s) = 1] + \epsilon.$$

This still gives that $\Pr[C(G(U_r)) = 1]$ is very small, if $\Pr[C(U_s) = 1]$ is very small, and is therefore suitable for applications that only require this consequence. [2] constructed such multiplicative PRGs for $\epsilon = s^{-\omega(1)}$ (based on very strong hardness assumptions).

In this paper, we give an optimal construction of multiplicative PRGs for nondeterministic circuits. More specifically, under the same hardness assumption used for (standard) PRGs for nondeterministic circuits, we show that for every $\epsilon \geq \frac{1}{2^s}$, there is a multiplicative PRG $G : \{0,1\}^{r=O(\log s + \log \frac{1}{\epsilon})} \to \{0,1\}^s$ that runs in time $\mathsf{poly}(s)$ and fools size s nondeterministic circuits.

This gives the optimal seed length under a hardness assumption that is necessary, and provides improvements in several applications of multiplicative PRGs. Our result improves upon the previous multiplicative PRG construction of [2], which uses a stronger hardness assumption against Σ_3-circuits, and where the seed length is the suboptimal $r = O(\log s) + O(\log \frac{1}{\epsilon})^2$. Our result also improves upon the recent multiplicative PRG of Shaltiel [28] that only achieves very small stretch (the output length in [28] is less than twice the seed length).

Our PRG construction borrows ideas from the recent "low stretch" PRG of Shaltiel [28], and the (standard) PRG construction of Shaltiel and Umans [30]. Loosely speaking, we aim to get the "multiplicativity" of the former, and the "large stretch" of the latter. While both approaches generalize the list-decoding results of Sudan, Trevisan and Vadhan [34], the two results are tailored to two very different parameter regimes, and we introduce several new ideas to make the two approaches co-exist.

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Hardness vsRandomness, Pseudorandomness, Nondeterministic Circuits

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.18

Related Version *Extended Version*: https://eccc.weizmann.ac.il/report/2026/017/



© Alon Dermer and Ronen Shaltiel;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 18; pp. 18:1–18:20



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Funding *Alon Dermer*: The research was supported by ISF grant 1006/23.
Ronen Shaltiel: The research was supported by ISF grant 1006/23 and also co-funded by the European Union (ERC, NFITSC, 101097959). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgements We want to thank anonymous referees for helpful comments.

1 Introduction

1.1 Pseudorandom Generators With Respect To a Relation

Pseudorandomness is a viewpoint that says that a distribution Z over $\{0,1\}^m$ is "similar" to the uniform distribution U_m, from the point of view of some function $C : \{0,1\}^m \to \{0,1\}$, if the quantities $p_1 = \Pr[C(U_m) = 1]$ and $p_2 = \Pr[C(Z) = 1]$ are "similar". Typically, this similarity is measured by choosing a parameter $0 < \epsilon \leq 1$ and using the standard (additive, double-sided) relation $\overset{ad}{\sim}_\epsilon$ on $[0,1]$ defined as follows:

$$p_1 \overset{ad}{\sim}_\epsilon p_2 \iff |p_2 - p_1| \leq \epsilon,$$

or in other words, Z is pseudorandom for C if:

$$|\Pr[C(Z) = 1] - \Pr[C(U_m) = 1]| \leq \epsilon.$$

In this paper, following [2, 28] we will also consider other relations (as we explain below in Section 1.3). In the definition below, we define pseudorandomness, and pseudorandom generators (PRGs) with respect to an arbitrary relation, and obtain the standard notion as a special case.

▶ **Definition 1** (Pseudorandomness with respect to a relation). *Let $\sim$ be a relation on $[0,1]$. Given a function $C : \{0,1\}^m \to \{0,1\}$, a distribution Z over $\{0,1\}^m$ is* pseudorandom for C with respect to $\sim$, *if*

$$\Pr[C(U_m) = 1] \sim \Pr[C(Z) = 1].$$

We will abbreviate "with respect to" as "w.r.t." for brevity. Given a class $\mathcal{C}$ of functions $C : \{0,1\}^m \to \{0,1\}$, we say that Z is pseudorandom for $\mathcal{C}$ w.r.t. $\sim$, *if it is pseudorandom for every C in the class $\mathcal{C}$ w.r.t. $\sim$.*

We say that Z is ϵ-pseudorandom for $\mathcal{C}$, *if it is pseudorandom for $\mathcal{C}$ w.r.t. $\overset{ad}{\sim}_\epsilon$.*[1]

▶ **Definition 2** (Pseudorandom generators w.r.t. a relation). *Let $\sim$ be a relation on $[0,1]$. A function $G : \{0,1\}^r \to \{0,1\}^m$ is a* PRG *for a class $\mathcal{C}$ w.r.t. $\sim$, if $G(U_r)$ is pseudorandom for $\mathcal{C}$ w.r.t. $\sim$. G is an* ϵ-PRG *for $\mathcal{C}$ if it is a PRG for $\mathcal{C}$ w.r.t. $\overset{ad}{\sim}_\epsilon$.*

[1] We remark that if the class $\mathcal{C}$ is closed under complement then the standard notion is also obtained when using the (one sided) relation

$$p_1 \overset{a}{\sim}_\epsilon p_2 \iff p_2 \leq p_1 + \epsilon,$$

in which the absolute value is removed.

1.2 PRGs for Nondeterministic Circuits from Hardness Assumptions

In this paper we will consider PRGs for the class of functions computable by either deterministic or nondeterministic circuits of a specified size s. We will consider PRGs in the "Nisan-Wigderson setting" in which the PRG will be allowed to run in time $p(s) > s$ for some polynomial $p(s)$. It is well known that such PRGs imply circuit lower bounds, and therefore, explicit constructions of such PRGs (e.g. [24]) require complexity theoretic hardness assumptions. We now discuss a well known (family of) hardness assumptions.

We say that *"E is hard for exponential size circuits of some type"*, if there exists a problem $L \in \mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ and a constant $\beta > 0$, such that for every sufficiently large n, circuits of size $2^{\beta \cdot n}$ (of the specified type) fail to compute the characteristic function of L on inputs of length n. (See full version for a more formal definition).

The assumption that E is hard for exponential size (deterministic) circuits was used by the celebrated paper of Impagliazzo and Wigderson [24] to imply PRGs for (deterministic) circuits. Subsequent work [25, 30, 36, 31] shows how to obtain PRGs for nondeterministic circuits, assuming that E is hard for exponential size nondeterministic circuits.[2] These two results are stated below.

▶ **Theorem 3** ([24]). *If E is hard for exponential size (deterministic) circuits, then for every sufficiently large s, there is an ϵ-PRG, $G : \{0,1\}^{r=O(\log s)} \to \{0,1\}^s$ for size s (deterministic) circuits, with $\epsilon = \frac{1}{s}$. Furthermore, G is computable in time $\mathsf{poly}(s)$.*[3]

▶ **Theorem 4** ([30]). *If E is hard for exponential size nondeterministic circuits, then for every sufficiently large s, there is an ϵ-PRG, $G : \{0,1\}^{r=O(\log s)} \to \{0,1\}^s$ for size s nondeterministic circuits, with $\epsilon = \frac{1}{s}$. Furthermore, G is computable in time $\mathsf{poly}(s)$.*

We remark that in both theorems, it is known that the assumption is necessary for the conclusion. Note that both theorems obtain error $\epsilon = \frac{1}{s}$. It is natural to ask whether it is possible to obtain smaller error of $\epsilon = s^{-\omega(1)}$ (at the cost of increasing the seed length). Unfortunately, we have evidence that "black-box techniques" cannot produce such results, even when starting from significantly stronger assumptions.

▶ **Theorem 5** ((Informal) [1, 20]). *"Black box techniques" cannot give a version of Theorem 3 or Theorem 4 in which $\epsilon = s^{-\omega(1)}$ even if:*

- *The seed length r is increased from $r = O(\log s)$ to $r = s - 1$.*
- *The hardness assumption is strengthened to "E is hard for exponential size Σ_i-circuits", for an arbitrary constant i. Here a Σ_i-circuit is a circuit with special gates that compute a complete language in the complexity class Σ_i^{P}. See precise definition in full version.*
- *The PRG is only w.r.t. $\overset{a}{\sim}_\epsilon$ rather than w.r.t. $\overset{ad}{\sim}_\epsilon$.*

The reader is referred to [1, 20] for a precise formulation of these results. We stress that all proofs in the PRG literature use "black-box techniques".

We remark that "cryptographic PRGs" can do better, and achieve error that is "negligible" in s. However, cryptographic PRGs (even w.r.t. $\overset{a}{\sim}_\epsilon$) do not exist against nondeterministic circuits, because in the cryptographic setting, the adversary has the computational resources to run the PRG, and can easily use nondeterminism to break the PRG.

---

[2] The assumption that E is hard for exponential size nondeterministic circuits is standard, and was used in many results [4, 25, 26, 30, 9, 17, 21, 31, 32, 15, 1, 10, 2, 22, 14, 5, 12, 6, 7, 29, 28]. It can be viewed as a scaled, nonuniform version of the widely believed assumption that $\mathsf{EXP} \neq \mathsf{NP}$.

[3] Here, and in the remainder of the paper, we often set $m = s$, as a size s circuit can receive at most s input bits, and can choose to ignore some of them.

1.3 Multiplicative PRGs

A useful property of a δ-PRG, $G : \{0,1\}^r \to \{0,1\}^m$ with very low error $\delta = s^{-\Omega(1)}$ against size s nondeterministic circuits, is that for every nondeterministic circuit C of size s,

$$p_1 = \Pr[C(U_m) = 1] \leq \delta \;\Rightarrow\; p_2 = \Pr[C(G(U_r)) = 1] \leq 2\delta = O(\delta).$$

That is, if p_1 is guaranteed to be very small, then p_2 is also guaranteed to be very small. We will loosely refer to this property as "preserving small probabilities". Some applications of PRGs for nondeterministic circuits (that we will survey later on) require $\delta = s^{-\omega(1)}$, but only rely on "preserving small probabilities".

Motivated by such applications (and the negative results of Theorem 5) Artemenko et al. [2] and Shaltiel [28] defined a notion of "multiplicative PRGs" that implies "preserving small probabilities" (and can be constructed under hardness assumptions). More specifically, inspired by differential privacy, they consider PRGs w.r.t. the following "one-sided" relation (which combines multiplicative and additive terms).

$$p_1 \overset{m}{\sim}_{(\epsilon,\delta)} p_2 \iff p_2 \leq e^{\epsilon} \cdot p_1 + \delta.$$

▶ **Definition 6** (Multiplicative PRGs [28]). [4] *A function $G : \{0,1\}^r \to \{0,1\}^m$ is an (ϵ,δ)-multiplicative PRG for a class $\mathcal{C}$ if it is a PRG for $\mathcal{C}$ w.r.t. $\overset{m}{\sim}_{(\epsilon,\delta)}$.*

Note that even if we take $\epsilon = 1$, then for every $\delta > 0$, we have that a $(1,\delta)$-multiplicative PRG $G : \{0,1\}^r \to \{0,1\}^m$ "preserves small probabilities", because

$$\Pr[C(U_m) = 1] \leq \delta \;\Rightarrow\; \Pr[C(G(U_r)) = 1] \leq e^1 \cdot \Pr[C(U_m) = 1] + \delta = O(\delta).$$

Recent work [28, 8] shows that multiplicative PRGs (with this definition) are useful when constructing extractors for samplable distributions. For the applications considered in this paper, it will always suffice to take $\epsilon = 1$, and very small δ, and so, we advise the reader to focus on the parameter δ.

1.4 An Optimal Multiplicative PRG for Nondeterministic Circuits

The main result of this paper is a new construction of multiplicative PRGs for nondeterministic circuits.

▶ **Theorem 7** (Optimal multiplicative PRGs for nondeterministic circuits). *If* E *is hard for exponential size nondeterministic circuits, then for every sufficiently large s, and every $\frac{1}{2^s} \leq \delta \leq \frac{1}{s}$, there is an $(\frac{1}{s},\delta)$-multiplicative PRG,*

$$G : \{0,1\}^{r=O(\log \frac{1}{\delta})} \to \{0,1\}^s$$

for size s nondeterministic circuits. Furthermore, G is computable in time poly(s).

---

[4] We remark that [2] use a different relation, more specifically the "double sided" relation:

$$p_1 \overset{md}{\sim}_{(\epsilon,\delta)} p_2 \iff p_1 \overset{m}{\sim}_{(\epsilon,\delta)} p_2 \text{ and } p_2 \overset{m}{\sim}_{(\epsilon,\delta)} p_1.$$

However, as we will see, in applications we typically only need the one sided version of $\overset{m}{\sim}_{(\epsilon,\delta)}$.

1.4.1 Optimality of the multiplicative PRG

Our multiplicative PRG is optimal in the following features:

- *The hardness assumption is necessary*: The assumption that E is hard for exponential size nondeterministic circuits, easily *follows* from the conclusion of Theorem 7. In fact, it follows for much weaker parameters, see full version for details. Another way to view this is that we are able to get a multiplicative PRG using the same hardness assumption that is necessary and sufficient for standard PRGs in Theorem 4.[5]
- *The seed-length is optimal up to a constant*: For every $\frac{1}{2^s} \leq \delta \leq \frac{1}{s}$, we obtain seed length $r = O(\log \frac{1}{\delta})$. This is obviously best possible (except for the constant hidden in the $O(\cdot)$ notation) as a probability space defined using r random bits cannot have nonzero probabilities that are smaller than 2^{-r}.
- We achieve $\epsilon = \frac{1}{s}$ which by Theorem 5 is best possible for black-box techniques. While we do achieve this, we remark again that for many applications $\epsilon = 1$ suffices.

1.4.2 Comparison to previous multiplicative PRG constructions

Artemenko et al. [2] constructed a multiplicative PRG which is suboptimal precisely in the two features mentioned above. They rely on the stronger assumption that E is hard for exponential size Σ_3-circuits (rather than the necessary assumption that we rely on). Furthermore, the seed length that they use is quadratically larger. More specifically, they obtain $r = O(\log \frac{1}{\delta})^2$ rather than the optimal $r = O(\log \frac{1}{\delta})$.[6]

Shaltiel [28] constructed a multiplicative PRG that has similar parameters to our Theorem 7, with the difference that the output length m is much smaller. More specifically, there exists a constant $\alpha > 0$, such that the output length is $m = (1 + \alpha) \cdot r$. Note that this means that this PRG has small stretch (it cannot even double its seed length) whereas the PRG of Theorem 7 achieves the optimal output length $m = s$ (and a circuit of size s cannot receive more than s bits as input). In particular, the PRG of Theorem 7 achieves superlinear stretch for $\delta = 2^{-o(s)}$, polynomial stretch if $\delta \leq 2^{-s^{0.9}}$, and almost exponential stretch if $\delta = 2^{-\log^{O(1)} s}$.

The proof of Theorem 7 builds on ideas from the two previous constructions [2, 28] (as well as other ideas) as we explain in detail in Section 2.

1.4.3 Black-box impossibility for a version of Theorem 7 for deterministic circuits

Given the similarity between Theorem 3 and Theorem 4 (which are identical except for replacing "deterministic circuits" with "nondeterministic circuits") it is natural to ask whether we can prove a version of Theorem 7 in which nondeterministic circuits are replaced by deterministic circuits. The next theorem shows that (in a similar sense to Theorem 5) "black-box techniques" cannot prove such a result.

---

[5] It is easy to see that an (ϵ, δ)-multiplicative PRG for a class $\mathcal{C}$ that is closed under complement, is also an $O(\epsilon + \delta)$-PRG for $\mathcal{C}$. The class of nondeterministic circuits of size s is not known (and not expected) to be closed under complement. Therefore, the aforementioned implication does not apply. Nevertheless our PRG has the property that for every relevant δ in Theorem 7, the multiplicative PRG that we construct is a $\frac{1}{s}$-PRG for nondeterministic circuits. Note that by Theorem 5 this is the best we can hope for.

[6] The PRG of [2] is only stated for $\delta = 2^{-s^{\Omega(1)}}$, however, it seems to us that it can be extended to any $\frac{1}{2^s} \leq \delta \leq \frac{1}{s}$.

▶ **Theorem 8** ((Informal) Black-box impossibility for a deterministic version of Theorem 7)**.** *"Black box techniques" cannot give a version of Theorem 7 in which* $\epsilon = 1$*,* $\delta = s^{-\omega(1)}$*, and all occurrences of "nondeterministic circuits" are replaced by "deterministic circuits". Furthermore, this holds even if the seed length* r *is increased to* $r = \Omega(s)$*.*

This shows that if one wants a multiplicative PRG for deterministic circuits of size s, with $\delta = s^{-\omega(1)}$, at the moment, we only know how to give such a result as a consequence of Theorem 7, and require the assumption that E is hard for exponential size nondeterministic circuits.

The proof of Theorem 8 follows by carefully adapting the black-box impossibility result of Grinberg, Shaltiel and Viola [20], and is deferred to the full version.

1.5 Randomness Reduction in Explicit Constructions for coNP/poly Properties

Explicit construction problems have the following form: We are given a property (namely a language $\mathcal{P} \in \{0,1\}^*$) and a number m, and want to efficiently produce a string $z \in \mathcal{P}$ of length m. For many famous properties $\mathcal{P}$ it is known that $\Pr[U_m \in \mathcal{P}] \geq 1 - \mu(m)$, for an exponentially small $\mu(m)$. In many cases (like producing a rigid matrix, or a generator matrix for a linear code that meets the Gilbert-Varshamov bound) it is also known that $\mathcal{P} \in \mathsf{coNP}$.

This holds whenever the probabilistic argument showing that $\Pr[U_m \in \mathcal{P}] \geq 1 - \mu(m)$ works as follows: It uses a union bound over exponentially many "bad events" B_i, where for each individual "bad event" B_i, there is a deterministic (or even nondeterministic) circuit $C_i(z)$ of size $s = \mathsf{poly}(m)$, that checks whether z is in the "bad event" B_i. Note that in this case, the union of bad events is in NP, as checking whether a given z is in the union of "bad events", amounts to checking whether there *exists* an i such that $C_i(z)$ accepts. Consequently, $\mathcal{P}$ (which is the complement of the union of bad events) is in coNP.[7]

We are interested in designing a randomized polynomial time algorithm $\mathsf{Construct}$ that on input 1^m, produces a string of length m that has the required property w.h.p. Our focus is the tradeoff between randomness complexity (the number of random bits used) and the error (the probability that the obtained string is not in $\mathcal{P}$). Multiplicative PRGs $G : \{0,1\}^r \to \{0,1\}^m$ for nondeterministic circuits of size $\mathsf{poly}(m)$ immediately apply for this problem. More specifically, by defining $\mathsf{Construct}(1^m) = G(U_r)$ and using Theorem 7 we obtain the following result.

▶ **Theorem 9** (Randomized explicit construction with error δ, using $r = O(\log \frac{1}{\delta})$ random bits)**.** *Let* $\mathcal{P}$ *be a language in* $\mathsf{coNP/poly}$ *such that for every sufficiently large* m*,*

$$\Pr[U_m \in \mathcal{P}] \geq 1 - \delta(m),$$

for $0 < \delta(m) \leq \frac{1}{m}$*. If* E *is hard for exponential size nondeterministic circuits, then there exists a randomized polynomial time algorithm* $\mathsf{Construct}$ *such that for every sufficiently large* m*,* $\mathsf{Construct}(1^m)$ *produces a string of length* m*, using* $r = O(\log \frac{1}{\delta(m)})$ *random bits, and we have that:*

$$\Pr[\mathsf{Construct}(1^m) \in \mathcal{P}] \geq 1 - 3 \cdot \delta(m).$$

[7] To demonstrate this point, consider the problem of explicitly producing an m bit string z that encodes a generator matrix A_z for a binary code that meets the Gilbert-Varshamov bound. Here, for every "message" i, the "bad event" B_i checks whether the string $A_z \cdot i$ has low Hamming weight.

Theorem 9 gives the "correct" tradeoff between the amount of random bits r, and the error δ. This improves upon the previous work of Artemenko et al. [2] (that relied on the multiplicative PRG of that paper) and used a stronger hardness assumption (against Σ_4-circuits) and larger randomness of $r = O(\log \frac{1}{\delta(m)})^2$ rather than $r = O(\log \frac{1}{\delta(m)})$.

1.5.1 Reducing communication when agreeing on a string with a certain coNP/poly property

Let $\mathcal{P}$ be a language as in Theorem 9. Imagine a "cryptographic scenario" where two parties Alice and Bob want to agree on some string $z \in \mathcal{P}$ of a specified length m (for some later use). The obvious approach is for Alice to choose a random $z \leftarrow U_m$ and send it to Bob. In this approach, the communication used is m.

Assuming the hardness assumption, and using Theorem 9, Alice can choose a random $x \in U_r$ for $r = O(\log \frac{1}{\delta(m)})$, send it to Bob, and both players can apply Construct on their own. This reduces the communication from m to r. Typically in cryptographic scenarios one expects error that is negligible in m, and we can take any function $\delta = m^{-\omega(1)}$ and reduce the communication complexity from m to say $r = \log(\frac{1}{\delta(m)})$ which is only slightly larger than $\log m$. Furthermore, note that $\delta = m^{-\omega(1)}$ is precisely where standard PRGs are inapplicable, whereas multiplicative PRGs obtain the "correct" tradeoff.[8]

1.6 Multiplicative PRGs For Nonboolean Circuits

Dubrov and Ishai [16] suggested a natural generalization of PRGs, which extends the standard definition (in which the circuit C outputs one bit) to a setting where C outputs ℓ bits. Thinking ahead, in the definition below, we consider PRGs w.r.t. a relation, and the notion of [16] is obtained for the relation $\overset{ad}{\sim}_\epsilon$.

▶ **Definition 10** (nb-PRGs w.r.t. a relation). *A function $G : \{0,1\}^r \to \{0,1\}^m$ is a ℓ-nb-PRG for a class $\mathcal{C}$ w.r.t. a relation $\sim$ on $[0,1]$, if for every function $C : \{0,1\}^m \to \{0,1\}^\ell$ in the class $\mathcal{C}$, and for every (arbitrary) function $D : \{0,1\}^\ell \to \{0,1\}$,*

$$\Pr[D(C(U_m)) = 1] \sim \Pr[D(C(G(U_r))) = 1].$$

1.6.1 Motivation for nb-PRGs

It is easy to see that for the relation $\overset{ad}{\sim}_\epsilon$ and $\ell = 1$, this definition coincides with the standard definition. For sufficiently large ℓ, and $m > \ell$ (a good choice to keep in mind is $m = \ell^{O(1)}$) we may view a function $C : \{0,1\}^m \to \{0,1\}^\ell$ as a "sampling procedure" which uses m random bits to produce a sample from the distribution $P = C(U_m)$. It is natural to ask whether it is possible to produce a sample from a distribution P' that is ϵ-close to P in statistical distance, using fewer random bits.[9]

nb-PRGs (w.r.t. $\overset{ad}{\sim}_\epsilon$) were introduced by Dubrov and Ishai [16] specifically for this application. More specifically, the distribution $P' = G(U_r)$ is generated using only r random bits, and Definition 10 immediately implies that P and P' are ϵ-close in statistical distance.

[8] We remark that recent work of [11] gives a pseudo-deterministic randomized algorithm for constructing strings with properties $\mathcal{P} \in \mathsf{P}$, and this allows two parties to agree on such a string (w.h.p) with no communication.

[9] Note that here, we want P and P' to be statistically indistinguishable and not just computationally indistinguishable. The latter goal is easier to obtain, but in some settings it is crucial for the two distributions to be statistically indistinguishable, so that even computationally unbounded adversaries D cannot distinguish between them.

Consequently, an ℓ-nb-PRG $G : \{0,1\}^r \to \{0,1\}^s$ for (deterministic circuits) of size s (w.r.t $\overset{ad}{\sim}_\epsilon$) can be used to reduce the randomness complexity of every sampling procedure C that runs in time s, and produces a distribution $P = C(U_s)$ on ℓ-bit strings. More specifically, the randomness complexity is reduced from s to r, while producing a distribution P' that is ϵ-close to P in statistical distance.

Note that we cannot hope for $r < \ell$, because a possible procedure $C(z)$ is one that outputs the first ℓ bits of z, and samples the distribution $P = U_\ell$. It is impossible to sample a distribution P' that is close in statistical distance to U_ℓ using fewer than ℓ random bits.

1.6.2 Constructions of nb-PRGs for polynomial size circuits

Applebaum et al. [1] (improving upon [16, 3]) showed that for every sufficiently large s, and every $\ell \geq \log s$ there is an ℓ-nb-PRG $G : \{0,1\}^{r=O(\ell)} \to \{0,1\}^s$ for circuits of size s (w.r.t. $\overset{ad}{\sim}_{\frac{1}{s}}$). This was obtained assuming E is hard for exponential size nondeterministic circuits. It is not known whether this hardness assumption is necessary, and as far as we know it may be possible to replace "nondeterministic circuits" with "deterministic circuits" in the hardness assumption. This result does achieve optimal seed length of $r = O(\ell)$, and error $\epsilon = \frac{1}{s}$, which is best possible in light of Theorem 5, using the observation that every ℓ-nb-PRG is also a 1-nb-PRG which is a PRG.

1.6.3 Multiplicative nb-PRGs for polynomial size circuits

In some cryptographic applications of nb-PRGs presented in [16, 2] (that we survey in the full version) the desired statistical distance ϵ should be "negligible". Motivated by these applications Artemenko et al. [2] observed that these applications only require "preserving small probabilities", and it is sufficient to use "multiplicative nb-PRGs" (that is nb-PRGs w.r.t $\overset{m}{\sim}_{(1,\delta)}$ for $\delta = s^{-\omega(1)}$) for these applications. We will refer to such ℓ-nb-PRGs as "(ℓ, ϵ, δ)-multiplicative nb-PRGs", and define these explicitly below.

▶ **Definition 11** (Multiplicative nb-PRGs). *A function $G : \{0,1\}^r \to \{0,1\}^m$ is an (ℓ, ϵ, δ)-multiplicative nb-PRG for a class $\mathcal{C}$, if for every function $C : \{0,1\}^m \to \{0,1\}^\ell$ in the class $\mathcal{C}$, and for every (arbitrary) function $D : \{0,1\}^\ell \to \{0,1\}$,*

$$\Pr[D(C(G(U_r))) = 1] \leq e^\epsilon \cdot \Pr[D(C(U_m)) = 1] + \delta.$$

In this paper we give an improved construction of multiplicative nb-PRGs.

▶ **Theorem 12** (Improved multiplicative nb-PRGs). *If E is hard for exponential size nondeterministic circuits, then for every sufficiently large s, every $\ell \geq \log s$, and every $\frac{1}{2^s} \leq \delta \leq \frac{1}{s}$ there is an $(\ell, \frac{1}{s}, \delta)$-multiplicative nb-PRG,*

$$G : \{0,1\}^{r=O(\ell+\log\frac{1}{\delta})} \to \{0,1\}^s$$

for size s deterministic circuits. Furthermore, G is computable in time $\mathsf{poly}(s)$.

We remark that Theorem 12 also holds if one replaces "deterministic circuits" by "single valued nondeterministic circuits".

Theorem 12 achieves the optimal seed length of $r = O(\ell + \log\frac{1}{\delta})$, and in particular, the right dependence on both ℓ and δ (up to constants). The previous construction of Artemenko et al. [2] achieved seed length $r = \ell + O(\log(1/\delta))^2$ which has incorrect dependence on δ.

While we do not know whether the hardness assumption in Theorem 12 is necessary, the hardness assumption used is the same hardness assumption that is used for (standard) nb-PRGs. In contrast, the previous construction of Artemenko et al. [2] assumes the much stronger assumption that E is hard for exponential size Σ_6-circuits.

The proof of Theorem 12 appears in the full version and shows that a multiplicative PRG with correct dependence on δ (as we obtain in Theorem 7) immediately implies a multiplicative nb-PRG.

2 Technique

In this section, we give a detailed informal overview of the main ideas that we use. This informal overview is intended to help the reader to understand the later technical sections (which contain full definitions, statements and proofs and do not build on the informal explanation of this section).

The readers can skip to the technical section if they wish.

2.1 An Overview of the Structure of (Standard) PRG Constructions

Let us start by revisiting the high level structure of (standard) PRGs for deterministic circuits, and specifically, the one by [24, 34] that give the PRG that is stated in Theorem 3. In this setting, we are given a parameter s, and aim to construct a δ-PRG $G : \{0,1\}^{O(\log s)} \to \{0,1\}^s$ for (deterministic) circuits of size s, w.r.t. $\overset{ad}{\sim}_\delta$, for $\delta = \frac{1}{s}$, under a hardness assumption. Thinking ahead, we think of δ as a parameter, and will later consider the case where $\delta = s^{-\omega(1)}$.

2.1.1 Low degree extension

When we aim to construct a δ-PRG for size s circuits and $\delta = \frac{1}{s}$ we first choose a large constant c_0 and set $d = \frac{c_0}{\beta}$ where β is the constant from the harnmess assumption. We choose the input length ℓ for the hardness assumption to be $\ell = d \log s = O(\log s)$. Invoking the hardness assumption, we obtain a function $f : \{0,1\}^\ell \to \{0,1\}$ that cannot be computed by circuits of size $2^{\beta \cdot \ell} = s^{c_0}$, but can be computed in time $2^{O(\ell)} = \mathsf{poly}(s^{c_0})$. We then "extend" the function f into its "low degree extension" $\hat{f}$. This is a degree $\mathsf{poly}(s)$ multivariate polynomial $\hat{f} : \mathbb{F}_q^d \to \mathbb{F}_q$, where $q = \mathsf{poly}(\frac{s}{\delta})$, and is an extension of f, in the following sense: For some fixed (and pre-determined) subset $H \subseteq \mathbb{F}_q$ of size s, there is an efficiently computable bijective map $\phi : \{0,1\}^\ell \to H^d$ such that for every $x \in \{0,1\}^\ell$, we have that $\hat{f}(\phi(x)) = f(x)$. In the terminology of error-correcting codes, this amounts to encoding (the truth table of) f using the Reed-Muller code to obtain the codeword that is (the truth table of) $\hat{f}$.

This step can be viewed as (nonboolean) *hardness amplification*: More specifically, Sudan, Trevisan and Vadhan [34] gave a local list-decoding algorithm showing that a circuit $\hat{C}$ of size s such that $\Pr_{\hat{x} \leftarrow \mathbb{F}_q^d}[\hat{C}(\hat{x}) = \hat{f}(\hat{x})] \geq \delta$, can be used to obtain a circuit B of size $\mathsf{poly}(s, q)$ that computes f correctly. For $\delta = \frac{1}{s}$ (which is the setting in Theorem 3) we have that $q = \mathsf{poly}(s)$ and can choose c_0 to be sufficiently large so that $\mathsf{poly}(s, q) \leq s^{c_0}$, contradicting the hardness

of f. Consequently, we can conclude that for every circuit $\hat{C}$ of size s, $\Pr[\hat{C}(X) = \hat{f}(X)] \leq \delta$. This argument fails if $\delta = s^{-\omega(1)}$, as then $q = s^{\omega(1)}$ and the size of the obtained circuit B is $\mathsf{poly}(s, q) = s^{\omega(1)}$ and does not contradict the hardness assumption.[10]

2.1.2 Concatenating with a boolean code (or Goldreich-Levin)

The next step is to transform the nonboolean function $\hat{f} : \mathbb{F}_q^d \to \mathbb{F}_q$ to a boolean function $g : \mathbb{F}_q^d \times \mathbb{F}_q \to \{0, 1\}$. For this purpose it is helpful to assume w.l.o.g. that q is a power of 2 (so that $v \in \mathbb{F}_q$ can be thought of as a $\log q$ bits string). One then defines:

$$g(w, v) = \langle \hat{f}(w), v \rangle,$$

where the inner product is taken over $\mathbb{F}_2$. In coding theoretic terms, this can be viewed as code-concatenation, namely, encoding each symbol of the "codeword" $\hat{f}$ by the Hadamard code, so that (the truth table of) g is the encoding of f by the code which is the concatenation of Reed-Muller and Hadamard.

The overall process of obtaining g from f is often referred to as "boolean hardness amplification" as one can show that the function g has the property that for $\delta = \frac{1}{s}$, it follows that every circuit P of size s,

$$\Pr_{w \leftarrow \mathbb{F}_q^d, v \leftarrow \mathbb{F}_q}[P(w, v) = g(w, v)] \leq \delta,$$

by showing that otherwise, there exists a small circuit $\hat{C}$ which breaks the (nonboolean hardness) of $\hat{f}$.

We stress that unlike the case of nonboolean hardness amplification, there are black-box limitations by Applebaum et al. [1] showing that assuming hardness against nondeterministic circuit (or even Σ_i-circuits for large i) does not help to obtain *boolean* hardness amplification for $\delta = s^{-\omega(1)}$. This is closely related to the limitations mentioned in Theorem 5, and is the essence of why we cannot expect (standard) PRGs with error $\delta = s^{-\omega(1)}$.

2.1.3 PRGs with one bit-stretch

At this point (at least for $\delta = \frac{1}{s}$) one can construct a (standard) PRG with one bit stretch by:

$$G(x) = x, g(x)$$

Note that the seed length of G is the input length of g which is $n = (d+1) \cdot \log q = O(\log q) = O(\log s + \log \frac{1}{\delta})$, which is indeed $O(\log s)$ for $\delta = \frac{1}{s}$. The correctness of this PRG G follows by first showing that for a uniform $x \leftarrow U_n$, if a circuit D of size s distinguishes $(x, g(x))$ from (x, U_1) (w.r.t. $\overset{ad}{\sim}_\delta$) then by a "distinguisher to predictor" argument, there exists a circuit P of roughly the same size of D, such that $\Pr[P(x) = g(x)] \geq \frac{1}{2} + \delta$, which is a contradiction to the boolean hardness amplification.

[10] We remark that Trevisan and Vadhan [35] showed how to "speed up the local decoding using nondeterminism" and obtain a *nondeterministic* C of size $\mathsf{poly}(s, \log q)$. This allows taking $\delta = s^{-\omega(1)}$ (in fact even exponentially small in s) and still obtain nonboolean hardness amplification, under a hardness assumption against nondeterministic circuits. While this result will not be useful for us directly as stated, we will later use a similar approach to "speed up" a size $\mathsf{poly}(s, q)$ deterministic circuit, and replace it by a size $\mathsf{poly}(s, \log q)$ nondeterministic circuit.

2.1.4 PRGs with large stretch

At this point, the original proof of Theorem 3 uses the Nisan-Wigderson generator [27] and "combinatorial designs" to take a seed x of length $r = O(\frac{n^2}{\log s})$ and stretch it to s strings $x_1, \ldots, x_s \in \{0,1\}^n$ such that the construction $G(x) = g(x_1), \ldots, g(x_s)$ is a PRG. Note that for $\delta = \frac{1}{s}$ this PRG indeed has seed length $r = O(\log s)$

2.2 Multiplicative PRG: The Construction

We now aim to prove Theorem 7 and construct a $(\frac{1}{s}, \delta)$-multiplicative PRG $G : \{0,1\}^{O(\log \frac{1}{\delta})} \to \{0,1\}^s$ for size s nondeterministic circuits, under the assumption that E is hard for exponential size nondeterministic circuits. The main thing to remember is that now δ can be much smaller than $\frac{1}{s}$, and a good choice to keep in mind is $\delta = 2^{-\sqrt{s}}$. Our plan is to follow the steps in Section 2.1. More specifically, given s and $\frac{1}{2^s} \leq \delta \leq \frac{1}{s}$, we set $q = \mathsf{poly}(s/\delta) = \mathsf{poly}(1/\delta)$, exactly as done earlier. However, in contrast to the standard setting in which $\delta = \frac{1}{s}$ and $q = \mathsf{poly}(s)$, we will be dealing with $q = s^{\omega(1)}$ which may be *exponential* in s.

We proceed with the first two steps described in the previous section (low-degree extension and Goldreich-Levin) for this modified choice of δ and q and obtain the function $g : \mathbb{F}_q^d \times \mathbb{F}_q \to \{0,1\}$, defined by $g(w,v) = \langle \hat{f}(w), v \rangle$ over $\mathbb{F}_2$. Recall that the input length of g (in bits) is $n = O(\log s + \log \frac{1}{\delta}) = O(\log \frac{1}{\delta})$.

2.2.1 "Standard approach" fails even for one bit stretch

Note that unlike the previous section, we do not expect to show that $G(x) = (x, g(x))$ is a PRG w.r.t. $\overset{a}{\sim}_\delta$, as by Theorem 5, black-box techniques (like we use here) cannot give such a result. Nevertheless, it was recently shown by Shaltiel [28] that G is a PRG w.r.t. $\overset{m}{\sim}_{(\frac{1}{s},\delta)}$. Our goal is to construct a multiplicative PRG with much larger stretch, that outputs s bits.

2.2.2 Using the Shaltiel-Umans PRG construction

Extending the number of random bits using the Nisan-Wigderson generator [27] (as done in the previous section) gives a seed length that does not have the correct dependence on δ. This is because, as explained in the previous section, using the Nisan-Wigderson generator (and combinatorial designs) one can at best get a seed length of $r = O(\frac{n^2}{\log s}) = \frac{O(\log \frac{1}{\delta})^2}{\log s} \approx (\log \frac{1}{\delta})^2$, for δ that is exponentially small in s. This limitation in the seed length of the Nisan-Wigderson generator is a focus of some earlier work (see [23, 30] for a discussion) and Shaltiel and Umans [30] gave a different PRG construction (in the standard setting) that avoids this type of loss.

Our plan is to use the approach of [30] and adjust it to the multiplicative setting. The Shaltiel-Umans PRG [30] relies on a $d \times d$ matrix A over $\mathbb{F}_q$ with some specific properties that we will soon explain. For every $0 \leq j \leq d-1$, they define $G_j : \mathbb{F}_q^d \times \mathbb{F}_q \to \{0,1\}^s$ as follows:

$$G_j(w,v) = g\left(\left(A^{s^j}\right)^0 \cdot w, v\right), \ldots, g\left(\left(A^{s^j}\right)^{s-1} \cdot w, v\right).$$

Note that the seed length of each G_j is $(d+1)\log q = O(\log \frac{1}{\delta})$. The final PRG $G : \left(\{0,1\}^{(d+1)\log q}\right)^d \to \{0,1\}^s$ is given by:

$$G(x_0, \ldots, x_{d-1}) = \bigoplus_{j=0}^{d-1} G_j(x_j)$$

and note that its seed length is $O(d \cdot \log \frac{1}{\delta}) = O(\log \frac{1}{\delta})$ as required.

2.2.3 Adjusting for an exponentially large field size

We will use this construction. A difficulty is that the construction and analysis of [30] are tailored to a parameter regime in which $q = \mathsf{poly}(s)$. This is because several components in the construction and analysis run in time $\mathsf{poly}(q)$. While, this is acceptable in the original construction where $q = \mathsf{poly}(s)$, it is a deal breaker in our case in which q may be exponential in s.[11]

Jumping ahead, we mention that a lot of the modifications that we make in the construction and analysis of [30] are required to reduce the running time of certain procedures from $\mathsf{poly}(q)$ to $\mathsf{poly}(s)$. The first such example is in the choice of the matrix A.

2.2.4 A modified property for the matrix A

As the matrix A is used by the PRG construction, that is required to run in time $\mathsf{poly}(s)$, we need to use a matrix A that can be found in time $\mathsf{poly}(s)$. The PRG construction of [30] relies on a matrix that we only know to find in time $\mathsf{poly}(q)$. As we cannot afford this, we will use a matrix A with a modified property, that can be found in time $\mathsf{poly}(s)$. We now elaborate on this issue.

The property of the $d \times d$ matrix A over $\mathbb{F}_q$ that is used in the analysis of [30] is that it is regular, and for every $w \in \mathbb{F}_q^d$ such that $w \neq 0$,

$$\left\{A^i \cdot w : 1 \leq i \leq q^d - 1\right\} = \mathbb{F}_q^d \setminus \{0\}.$$

In addition, the PRG construction of [30] also relies on the fact that (by appropriately choosing the relationship between s and q) the matrix $B = A^{\frac{q^d-1}{s^d-1}}$ satisfies that for every $w \in H^d$ such that $w \neq 0$, $\left\{B^i \cdot w : 1 \leq i \leq s^d - 1\right\} = H^d \setminus \{0\}$. We will refer to this property as the "additional property".

In [30] it was shown that a matrix A with the original property can be found in time $\mathsf{poly}(q^d) = \mathsf{poly}(q)$, but we do not know how to find it in time $\mathsf{poly}(s)$. Because of this reason (as well as additional reasons that we will explain later on) we will use a matrix A that only has the "additional property". Namely, in our construction we will use a matrix A that satisfies that A is regular, and for every $w \in H^d$ such that $w \neq 0$,

$$\left\{A^i \cdot w : 1 \leq i \leq s^d - 1\right\} = H^d \setminus \{0\}.$$

Recall that H is of size s, and using similar arguments to the ones used in [30], it follows that a matrix A that satisfies only the modified property can be produced in time $\mathsf{poly}(s^d, \log q) = \mathsf{poly}(s)$. The precise statement appears in the full version. With this choice, computing G indeed takes time $\mathsf{poly}(s)$ as required. Jumping ahead, we will need to modify the proof of [30] to work with this modified property of A.

[11] We remark that Umans [36] gave a modified PRG construction which uses a more complicated low degree extension, but avoids the need to xor the outputs of d "candidate PRGs". This yields reduced seed length, as one avoids the need to choose d independent seeds. Note however that in our setting, $d = c_0/\beta$ is a constant, and so this saving is insignificant. In any case, it seems to us that extending the approach of [36] to the multiplicative setting, is more difficult than extending the approach of [30]. Loosely speaking, this is because there are more technical issues to handle when adjusting the modified low degree extension of Umans [36] to exponentially large field size. It seems to us that some of the ideas used in this paper to adjust for exponentially large can be extended to the framework of Umans [36], but we did not pursue this direction, as at the moment, there is no clear motivation to do so.

2.2.5 How to find a matrix A with the modified property in time $\mathsf{poly}(s)$

For completeness we briefly survey the algorithm of [30] for producing a matrix A. In [30], the matrix A with the "original property" is obtained by identifying the vector space $\mathbb{F}_q^d$ with the extension field $\mathbb{F}_{q^d}$ (in the natural way). The multiplicative group of this field is cyclic, and a generator α for this group can be found by exhaustive search in time $\mathsf{poly}(q^d)$. The function $x \to \alpha \cdot x$ over $\mathbb{F}_q^d$ is an invertible $\mathbb{F}_q$-linear map, and therefore, can be represented by an invertible $d \times d$ matrix A over $\mathbb{F}_q$. This indeed gives that A is regular, and $\left\{A^i \cdot v : 1 \leq i \leq q^d - 1\right\} = \mathbb{F}_q^d \setminus \{0\}$.

Shaltiel and Umans [30] were able to argue that $B = A^{\frac{q^d-1}{s^d-1}}$ satisfies the "additional property" by carefully selecting the parameters s and q, to guarantee that there is a subfield of size s^d in $\mathbb{F}_{q^d}$ that coincides with (the set) H^d. By the same argument, one can use brute force search to find a generator of the multiplicative group of the *subfield*, rather than the *big field*. As this subfield is only of size s^d (rather than q^d), the time of the brute force search for the generator can be reduced from $\mathsf{poly}(q^d)$ to $\mathsf{poly}(s^d, \log q) = \mathsf{poly}(s)$.

2.3 Multiplicative PRG: The Analysis

We will now outline the proof of Theorem 7. We assume for the purpose of contradiction that G is not a multiplicative PRG with the required parameters, and therefore, there exists a size s nondeterministic circuit $D : \{0,1\}^s \to \{0,1\}$, that "distinguishes" a pseudorandom output from a uniform output, meaning that $\Pr[D(U_s) = 1] \overset{m}{\not\sim}_{(\frac{1}{s},\delta)} \Pr[D(G(U_r)) = 1]$.

Our goal is to contradict the hardness assumption by constructing a nondeterministic circuit B of size s^{c_0} for f. On a high level, we will try to imitate the analysis of Shaltiel and Umans [30] (that applies in the standard case) by also using ideas from the approach of Shaltiel [28] (that shows that $G(x) = x, g(x)$ is a multiplicative PRG). We will have to deal with the fact that q is no longer polynomial in s, and that we use a matrix A with the modified property.

The first step in the [30] analysis is to argue that as G is the xor of d "candidates" $G_0, \ldots, G_{d-1}$, a nondeterministic "distinguisher" D for G, yields d nondeterministic distinguishers $D_0, \ldots, D_{d-1}$ for $G_0, \ldots, G_{d-1}$. This step can easily be adjusted to our multiplicative setting (see full version for details).

The second step in [30] (as in many PRG constructions) is to use a hybrid argument [18] to argue that for every $0 \leq j \leq d-1$, the circuit D_j distinguishes between two distributions which only differ in the last bit. This hybrid argument can be easily adapted to the multiplicative setting (see full version for details). To make the notations simpler, we concentrate only on $j = 0$, and then the first $s-1$ bits of $G_0(w, v)$ are given by:

$$\mathsf{Prv}(w, v) = \langle \hat{f}(A^0 \cdot w), v\rangle, \ldots, \langle \hat{f}(A^{s-2} \cdot w), v\rangle,$$

whereas the last output bit of $G_0(w, v)$ is $\mathsf{Nxt}(w, v) = \langle \hat{f}(A^{s-1} \cdot w), v\rangle$.

For a uniformly chosen seed (w, v) for G_0, this "multiplicative hybrid argument" gives that D_0 distinguishes $(\mathsf{Prv}(w, v), U_1)$ from $(\mathsf{Prv}(w, v), \mathsf{Nxt}(w, v))$ w.r.t. $\overset{m}{\sim}_{(\frac{1}{s},\delta)}$. More formally, we have that:

$$\Pr[D_0\left(\mathsf{Prv}(w, v), U_1\right) = 1] \overset{m}{\not\sim}_{(\frac{1}{s},\delta)} \Pr[D_0\left(\mathsf{Prv}(w, v), \mathsf{Nxt}(w, v)\right) = 1].^{12}$$

[12] Here, we are slightly cheating as (just like in the case of the standard hybrid argument) there are small quantitative losses the multiplicative hybrid argument incurs, and both $\frac{1}{s}$ and δ should be divided by s (See full version for details). However, this quantitative loss is insignificant for our purposes, and we ignore it in this high level overview.

2.3.1 The multiplicative setting does not have a "distinguisher to predictor" argument

So far, the analysis we sketched closely follows the proof of [30] by finding "multiplicative analogs" for the steps in the original proof. Now, however, we arrive at a step for which there does not seem to be a multiplicative analog. In the standard setting, one can use a "distinguisher to predictor" argument to transform the distinguisher D_0 into a predictor $P_0(\mathsf{Prv}(w,v))$ which predicts $\mathsf{Nxt}(w,v)$ correctly with probability roughly $\frac{1}{2}+\delta$.

In our setting, $\delta = s^{-\omega(1)}$ and such a predictor is not useful. Loosely speaking, had we been able to use such a predictor to contradict the hardness of f, we would have broken the limitations of Theorem 5. Instead, we will need to proceed without using a "distinguisher to predictor" step, and instead we will try to imitate the next steps of the analysis of [30] using a distinguisher D_0 rather than a predictor. Loosely speaking, following [28], we will try to circumvent this difficulty using *nondeterminism*. Before we explain how to do this, let us briefly survey how the original proof of [30] proceeds after obtaining a predictor P_0.

2.3.2 How [30] use the predictor

Note that for every w, $(\langle \hat{f}(w), v\rangle)_{v\in\mathbb{F}_q}$ is the Hadamard encoding of $\hat{f}(w)$. Loosely speaking, by using list decoding techniques, [30] obtain a (nonboolean) predictor P_0' such that:

$$P_0'\left(\hat{f}(A^0\cdot w),\ldots,\hat{f}(A^{s-2}\cdot w)\right) = \hat{f}(A^{s-1}\cdot w)$$

with probability roughly δ. Loosely speaking, Shaltiel and Umans [30] show how to "error-correct" P_0' to an errorless predictor P_0^* that succeeds with probability one. We will explain some of this argument later.

Once the nonboolean predictor is errorless, one can obtain a circuit B that computes f as follows: The circuit B will be hardwired with some $w_0 \in \mathbb{F}_q^d$, and the evaluations $\hat{f}(A^0\cdot w_0),\ldots,\hat{f}(A^{s-2}\cdot w_0)$. When given $x\in\{0,1\}^\ell$, the circuit B will compute a number i^* such that $A^{i^*}\cdot w_0 = \phi(x)$ (where ϕ is the map $\phi:\{0,1\}^\ell \to H^d$ of the low degree extension) and then we know that $f(x) = \hat{f}(\phi(x)) = \hat{f}(A^{i^*}\cdot w_0)$.

Note that B can use P_0^* to predict the "next element" $\hat{f}(A^{s-1}\cdot w_0)$ from "previous elements" $\hat{f}(A^0\cdot w_0),\ldots,\hat{f}(A^{s-2}\cdot w_0)$, and then continue to predict $\hat{f}(A^s\cdot w_0)$ from $\hat{f}(A^1\cdot w_0),\ldots,\hat{f}(A^{s-1}\cdot w_0)$ that are now available to it. By using this process iteratively, one can reach i^* and compute $\hat{f}(A^{i^*}\cdot w_0) = f(x)$.

It should be noted that as described above, this iterative process takes too much time. This is because i^* could be very large. Using a matrix A with the original property, it could be that $i^* = q^d - 1$ and this could take q^d steps. With the modified property the number of steps can be reduced to s^d (assuming we can arrange that $w_0 \in H^d$ so that the starting point is in H^d). We will later explain how to modify the approach of [30] so that we can arrange that $w_0 \in H^d$. The reduced number of steps follows because by the modified property of A we have that $i^* \leq s^d$.

However, even with this reduction in the number of steps, this takes too much time, as $s^d = 2^\ell$, and it is trivial to obtain a size 2^ℓ circuit B for a function f on ℓ bits. This issue is the reason that [30] use many "generators" $G_0,\ldots,G_{d-1}$ in their construction. Loosely speaking, when using t generators, [30] shows (by a complicated argument that we will not explain in this high level overview) how to reduce the number of steps from i^* to $(i^*)^{1/t}$. This gives that in our setup, using a matrix with modified property, so that $i^* \leq s^d$, by using $t = d$ candidates, we can reduce the number of steps from s^d to s, and the obtained circuit B will end up being of size s^{c_0} that indeed contradicts the hardness of f.

2.3.3 Learning the next curve

As we explained above, the proof of [30] uses error-correcting techniques to error-correct a "noisy predictor" into an "errorless predictor". We now give an overview of this argument (and later explain how to imitate it using a distinguisher instead of a predictor). The main technical step in this process considers the following probabilistic setting: Let r be a large constant, and consider a uniformly chosen degree r polynomial $C : \mathbb{F}_q \to \mathbb{F}_q^d$ (which we will refer to as a curve). Loosely speaking, the approach of [30] is to design an errorless predictor P_0^* which predicts all the evaluations of the "next curve"

$$\left(\hat{f}(A^{s-1} \cdot C(t))\right)_{t \in \mathbb{F}_q}$$

from evaluations on previous curves, namely from:

$$\left(\hat{f}(A^0 \cdot C(t)), \ldots, \hat{f}(A^{s-2} \cdot C(t))\right)_{t \in \mathbb{F}_q}.$$

Note that the evaluations on the next curve, are evaluations of the univariate polynomial

$$\hat{p}(t) = \hat{f}(A^{s-1} \cdot C(t)).$$

This polynomial is a nonzero polynomial with low degree (as $\hat{f}, C$ are low degree, and A is a regular matrix). For every $t \in \mathbb{F}_q$, one can use the "noisy predictor" P_0' for $w = C(t)$, using the previous evaluations, and obtain a prediction for $\hat{p}(t)$. For every $t \in \mathbb{F}_q$, $C(t)$ is uniform over $\mathbb{F}_q^d$, and this means that we expect a δ-fraction of the q predictions to be correct. At this point, [30] use Sudan's list-decoding algorithm for the Reed-Solomon code [33] to obtain a list of $L = \mathsf{poly}(1/\delta)$ polynomials such that one of them is the correct polynomial $\hat{p}$. For the sake of explaining the argument, let us cheat and assume that $L = 1$, and one can "uniquely decode" to the correct polynomial.[13]

2.3.4 Arranging that a random curve intersects H^d

Recall that earlier, we promised to show that "errorless prediction" can start from a point w_0 that is in H^d rather than $\mathbb{F}_q^d$. This does not follow from the argument of [30]. Loosely speaking, this is because the random curve C is unlikely to intersect the set H^d, and so it may be the case that no "starting point" is in H^d.

One of the modifications that we introduce in this paper (which already applies in the original setting of [30]) is a modified probability space for the analysis of [30]. In this modified probability space, we insist that the random curve C intersects the set H^d. More specifically, rather than choosing the curve C uniformly at random, we choose it *conditioned* on the event that $\{C(0) = w_0\}$ for some $w_0 \in H^d$. Fortunately, it is possible to carry out the analysis of [30] with this modification. Loosely speaking, this is because even in the modified probability space, the random variables $(C(t))_{t \leftarrow \mathbb{F}_q \setminus \{0\}}$ are distributed like $q-1$ elements that are $(r-1)$-wise independent.

[13] Shaltiel and Umans are able to achieve $L = 1$, by considering a significantly more complicated random experiment in which two curves C^1 and C^2 are chosen at random, so that each one is uniform, but the two curves are "interleaved" (meaning that they are correlated in a specially designed way that allows list-decoding to imply unique decoding). We will not explain this argument in this high level overview. Our full proof repeats this argument (see full version for details) with some changes that we explain below.

2.3.5 Using a distinguisher instead of a predictor

We now return to the multiplicative case. Recall that we stopped after obtaining a distinguisher D_0 such that

$$\Pr[D_0\left(\mathsf{Prv}(w,v),U_1\right)=1] \overset{m}{\not\sim}_{(\frac{1}{s},\delta)} \Pr[D_0\left(\mathsf{Prv}(w,v),\mathsf{Nxt}(w,v)\right)=1].$$

As we explained earlier, in our setting when δ is very small, and we cannot use a "distinguisher to predictor" argument. Instead, we will have to imitate the proof of [30] using the distinguisher D_0.

A key idea is that while we don't have a predictor, we are in a setting where we are allowed to use nondeterminism. Inspired by [6, 28], our high level idea is to "learn the next curve" in two steps: We first use nondeterminism to guess the coefficients of the "correct polynomial" $\hat{p}(t)=\hat{f}(A^{s-1}\cdot C(t))$. We then design an efficient test such that the correct polynomial $\hat{p}$ passes the test, but no other low-degree polynomial does. This means that using nondeterminism, we can reduce the task of "learning the next curve" to the task of "testing the next curve".

Note that in the framework, the verification step of "testing the next curve" can *itself* be *nondeterministic*, as even then, the entire "guess and test" computation can be implemented by a small nondeterministic circuit. This additional freedom of using nondeterminism twice will be crucial in the argument.

More specifically, we will be able to show that there exists a number $\gamma>0$ (which can be exponentially small in s) such that for a random $t,v\leftarrow\mathbb{F}_q$, the correct polynomial $\hat{p}$ satisfies:

$$\Pr[D_0(\mathsf{Prv}(C(t),v),\langle\hat{p}(t),v\rangle)=1]\geq\gamma.$$

On the other hand, for every low degree polynomial $p\neq\hat{p}$,

$$\Pr[D_0(\mathsf{Prv}(C(t),v),\langle p(t),v\rangle)=1]\leq e^{-\frac{1}{s}}\cdot\gamma.$$

Loosely speaking, this "multiplicative distance" between γ and $e^{-\frac{1}{s}}\cdot\gamma$ is inherited from the fact that the distinguisher D_0 distinguishes w.r.t. a multiplicative relation. We will not go into precise details of this proof that builds on ideas from [6, 28] that are adapted to the setting considered in [30]. In fact, in the actual proof, we can only show that the second inequality holds for p which agrees with $\hat{p}$ on r choices of $t\in\mathbb{F}_q$. In this high level overview, we will cheat and ignore this technicality (which introduces significant complications).

In order to test whether a given low degree polynomial p is the correct polynomial $\hat{p}$, we will prepare the size $\mathsf{poly}(s)$ nondeterministic circuit:

$$B(t,v)=D_0(\mathsf{Prv}(C(t),v),\langle p(t),v\rangle).$$

All that is left is to design a test that given a size $\mathsf{poly}(s)$ circuit B, distinguishes the case that B accepts at least a γ-fraction of its inputs, from the case that B accepts at most $e^{-\frac{1}{s}}\cdot\gamma$-fraction of its inputs.

2.3.6 Using the Goldwasser-Sipser "set lower bound" protocol

We now explain how to distinguish the two cases. To gain some intuition, note that we do not have time to go over all inputs to B as their number is larger than q, and is not polynomial in s. It is also not feasible to distinguish the two cases by taking a sample of $\mathsf{poly}(s)$ random inputs to B, because the "additive distance" $\gamma-e^{-\frac{1}{s}}\cdot\gamma$ may be exponentially small in s, and a $\mathsf{poly}(s)$ size sample cannot be used to distinguish.

Fortunately, distinguishing between these two cases is precisely what nondeterministic circuits are good at. Following [28], we use the "set lower bound" of Goldwasser and Sipser [19] for this purpose. The Goldwasser-Sipser "set lower bound" is typically stated as an AM protocol, where the input is a size s deterministic circuit B, and Merlin claims that B accepts a γ-fraction of the inputs (for some given $\gamma > 0$). We can adapt this protocol to our setting by observing that:

- The Goldwasser-Sipser AM protocol works not just for deterministic circuits, but also for nondeterministic circuits. (This essentially follows because AM with a constant number of messages can be collapsed to two messages).
- $\mathsf{AM} \subseteq \mathsf{NP/poly}$, and so, the Goldwasser-Sipser AM protocol can be implemented by a nondeterministic circuit of size $\mathsf{poly}(s)$.
- The error of the Goldwasser-Sipser protocol is *multiplicative*. More specifically, even when γ is exponentially small, when given a size $\mathsf{poly}(s)$ circuit B, the Goldwasser-Sipser protocol can distinguish the case that B accepts at least a γ-fraction of the inputs, from the case that B accepts at most a $e^{-\frac{1}{s}} \cdot \gamma$-fraction of its inputs.[14]

Note that in the argument above we use nondeterminism three times: We use nondeterminism as D_0 is a nondeterministic circuit. Nevertheless, even if we were constructing a multiplicative PRG for deterministic circuits and D_0 is deterministic, we need nondeterminism to guess the correct polynomial $\hat{p}$, and then we need additional nondeterminism to run the Goldwasser-Sipser protocol.

Organization of this paper

Due to space constraints, this version is an extended abstract and does not contain full statements and proof. The reader is referred to [13] for a full version of this paper.

3 Conclusion and Open Problems

In this paper we give an optimal construction of multiplicative PRGs for nondeterministic circuits, as well as multiplicative nb-PRGs. Both are constructed under the hardness assumption that E is hard for exponential size nondeterministic circuits.

A natural open problem is to find more applications of multiplicative PRGs. In addition to the applications mentioned in this paper, recent work [28, 8] utilizes multiplicative PRGs as building blocks in explicit constructions of extractors for samplable distributions, as well as other types of "hard to sample functions". It is natural to ask whether the multiplicative PRGs of this paper can provide improvement in these applications.

Theorem 5 shows that we cannot expect to obtain (standard) ϵ-PRGs with error $\epsilon = s^{-\omega(1)}$ under assumptions of the form: there exists an i, such that E is hard for exponential size Σ_i-circuits. Is there a way to bypass these limitation by introducing another type of plausible assumption? We remark that it is possible to bypass this limitations if one is willing to

[14] Loosely speaking, it is this property of nondeterministic circuits that avoids black-box limitations and enables us to obtain multiplicative PRGs with $\delta = s^{-\omega(1)}$ under a hardness assumption against nondeterministic circuits. Loosely speaking, the black-box impossibility results of [1] follow by showing that size $\mathsf{poly}(s)$ nondeterministic circuits (or even Σ_i-circuits for any large i) cannot distinguish the case where a circuit B of size s accepts $\frac{1}{2} + s^{-\omega(1)}$ of its inputs from the case that it accepts $\frac{1}{2}$ of its inputs. This is essentially why we are not able to use predictors in our proof. However, we can use "multiplicative" distinguishers, as this leads to the need to distinguish between a circuit that accepts at least γ-fraction of inputs, from a circuit that accepts at most $e^{-\frac{1}{s}} \cdot \gamma$ fraction of inputs, and we've just seen that nondeterministic circuits *can* perform this task.

assume the very strong assumption that E is hard for exponential size PSPACE-circuits (as also mentioned in [2]). However, maybe there's a completely different line of plausible assumptions that can bypass such limitations.

A related open problem is that we don't know whether (standard) nb-PRGs for deterministic circuits require a hardness assumption against nondeterministic circuits. This assumption is used by current constructions [1], but unlike other cases mentioned in this paper, there are no black-box impossibility results that prevent such an nb-PRG from being based on the assumption that E is hard for exponential size deterministic circuits.

References

1 B. Applebaum, S. Artemenko, R. Shaltiel, and G. Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. In *30th Conference on Computational Complexity*, pages 582–600, 2015. doi:10.4230/LIPIcs.CCC.2015.582.

2 S. Artemenko, R. Impagliazzo, V. Kabanets, and R. Shaltiel. Pseudorandomness when the odds are against you. In *31st Conference on Computational Complexity, CCC*, volume 50, pages 9:1–9:35, 2016. doi:10.4230/LIPIcs.CCC.2016.9.

3 S. Artemenko and R. Shaltiel. Pseudorandom generators with optimal seed length for non-boolean poly-size circuits. In *Symposium on Theory of Computing, STOC*, pages 99–108, 2014. doi:10.1145/2591796.2591846.

4 V. Arvind and J. Köbler. New lowness results for ZPP^{NP} and other complexity classes. *J. Comput. Syst. Sci.*, 65(2):257–277, 2002. doi:10.1006/jcss.2002.1835.

5 M. Ball, D. Dachman-Soled, and J. Loss. (nondeterministic) hardness vs. non-malleability. In *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference*, volume 13507, pages 148–177, 2022. doi:10.1007/978-3-031-15802-5_6.

6 M. Ball, E. Goldin, D. Dachman-Soled, and S. Mutreja. Extracting randomness from samplable distributions, revisited. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1505–1514, 2023. doi:10.1109/FOCS57990.2023.00092.

7 M. Ball, R. Shaltiel, and J. Silbak. Non-malleable codes with optimal rate for poly-size circuits. In *Advances in Cryptology - EUROCRYPT*, volume 14654 of *Lecture Notes in Computer Science*, pages 33–54, 2024. doi:10.1007/978-3-031-58737-5_2.

8 M. Ball, R. Shaltiel, and J. Silbak. Extractor for samplable distributions with low min-entropy. *To appear in STOC*, 2025.

9 B. Barak, S. J. Ong, and S. P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007. doi:10.1137/050641958.

10 N. Bitansky and V. Vaikuntanathan. A note on perfect correctness by derandomization. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 10211, pages 592–606, 2017. doi:10.1007/978-3-319-56614-6_20.

11 L. Chen, Z. Lu, I. C. Oliveira, H. Ren, and R. Santhanam. Polynomial-time pseudodeterministic construction of primes. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1261–1270. IEEE, 2023. doi:10.1109/FOCS57990.2023.00074.

12 L. Chen and R. Tell. When arthur has neither random coins nor time to spare: Superfast derandomization of proof systems. *Electron. Colloquium Comput. Complex.*, TR22-057, 2022. URL: https://eccc.weizmann.ac.il/report/2022/057.

13 A. Dermer and R. Shaltiel. Multiplicative pseudorandom generators for nondeterministic circuits. *Electron. Colloquium Comput. Complex.*, TR26, 2026. URL: https://eccc.weizmann.ac.il/report/2026/017.

14 D. Doron, D. Moshkovitz, J. Oh, and D. Zuckerman. Nearly optimal pseudorandomness from hardness. *J. ACM*, 69(6):43:1–43:55, 2022. doi:10.1145/3555307.

15 Andrew Drucker. Nondeterministic direct product reductions and the success probability of SAT solvers. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 736–745, 2013. doi:10.1109/FOCS.2013.84.

16 B. Dubrov and Y. Ishai. On the randomness complexity of efficient sampling. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 711–720, 2006. doi:10.1145/1132516.1132615.

17 O. Goldreich and A. Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *APPROX-RANDOM*, pages 209–223, 2002. URL: http://link.springer.de/link/service/series/0558/bibs/2483/24830209.htm.

18 S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984. Preliminary version appeared in STOC' 82. doi:10.1016/0022-0000(84)90070-9.

19 S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 59–68, 1986. doi:10.1145/12130.12137.

20 A. Grinberg, R. Shaltiel, and E. Viola. Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 956–966. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00094.

21 Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. Uniform hardness versus randomness tradeoffs for arthur-merlin games. *Computational Complexity*, 12(3-4):85–130, 2003. doi:10.1007/s00037-003-0178-7.

22 P. Hubácek, M. Naor, and E. Yogev. The journey from NP to TFNP hardness. In *8th Innovations in Theoretical Computer Science Conference, ITCS*, volume 67, pages 60:1–60:21, 2017. doi:10.4230/LIPIcs.ITCS.2017.60.

23 R. Impagliazzo, R. Shaltiel, and A. Wigderson. Reducing the seed length in the nisan-wigderson generator. *Combinatorica*, 26(6):647–681, 2006. doi:10.1007/s00493-006-0036-8.

24 R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *STOC*, pages 220–229, 1997.

25 A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002. doi:10.1137/S0097539700389652.

26 P. Bro Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005. doi:10.1007/s00037-005-0197-7.

27 N. Nisan and A. Wigderson. Hardness vs. randomness. *JCSS: Journal of Computer and System Sciences*, 49, 1994.

28 R. Shaltiel. Multiplicative extractors for samplable distributions. In *40th Computational Complexity Conference, CCC*, volume 339 of *LIPIcs*, pages 22:1–22:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CCC.2025.22.

29 R. Shaltiel and J. Silbak. Explicit codes for poly-size circuits and functions that are hard to sample on low entropy distributions. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC*, pages 2028–2038, 2024. doi:10.1145/3618260.3649735.

30 R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005. doi:10.1145/1059513.1059516.

31 R. Shaltiel and C. Umans. Pseudorandomness for approximate counting and sampling. *Computational Complexity*, 15(4):298–341, 2006. doi:10.1007/s00037-007-0218-9.

32 R. Shaltiel and C. Umans. Low-end uniform hardness versus randomness tradeoffs for am. *SIAM J. Comput.*, 39(3):1006–1037, 2009. doi:10.1137/070698348.

33 M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13, 1997.

34 M. Sudan, L. Trevisan, and S. P. Vadhan. Pseudorandom generators without the xor lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001. doi:10.1006/JCSS.2000.1730.

35 L. Trevisan and S. P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science*, pages 32–42, 2000. doi:10.1109/SFCS.2000.892063.

36 C. Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences*, 67:419–440, 2003. doi:10.1016/S0022-0000(03)00046-1.

When Hilbert Approximates: A Strong Nullstellensatz for Approximate Polynomial Satisfiability

Sanyam Agarwal
Universität des Saarlandes, Saarbrücken, Germany

Sagnik Dutta
Max-Planck-Institut für Informatik, Saarbrücken, Germany

Anurag Pandey
Universität des Saarlandes, Saarbrücken, Germany

Himanshu Shukla
Universität Regensburg, Germany

Abstract

Guo, Saxena, and Sinhababu (TOC'18, CCC'18) defined a natural, approximative analog of the polynomial system satisfiability problem, which they called approximate polynomial satisfiability (APS). They proved algebraic and geometric properties of it and showed an NP-hardness lower bound and a PSPACE upper bound for it. They further established how the problem naturally occurs in border complexity and Geometric complexity theory (GCT) and used the problem to construct hitting sets for $\overline{\mathsf{VP}}$ in PSPACE, hence greatly mitigating the GCT chasm.

The starting point of this work is the observation that Guo, Saxena, and Sinhababu's criterion for non-existence of approximative solution can be interpreted as an analog of Weak Hilbert's Nullstellensatz in the approximative setting. We extend their work by proving an analog of Strong Hilbert's Nullstellensatz in the approximative setting. Concretely, we give an algebraic criterion for containment between approximative solution sets defined by systems of polynomials. In fact, this characterization turns out to be equivalent to membership in the integral closure over a maximal ideal of a local subring of $\mathbb{C}(x_1, \ldots, x_n)$ determined by the given polynomials. In addition, we use our proof to provide a PSPACE algorithm for testing this containment, exponentially better than the EXPSPACE bounds for polynomial subalgebra membership testing and the polynomial integral closure membership testing, hence matching the complexity bound of Guo, Saxena, and Sinhababu's Weak Approximative Nullstellensatz.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Approximate Polynomial Satisfiability, Algebraic Geometry, PSPACE

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.19

Funding *Himanshu Shukla*: HS was supported by the European Union (ERC, CountHom, 101077083). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgements We would like to thank Bernd Sturmfels for pointing us towards Ganzstellensatz. AP would like to thank Nitin Saxena for initial discussions, and Karl Bringmann and Raimund Seidel for hosting him during this work. We thank the anonymous reviewers for their helpful comments. Finally, we would like to thank Markus Bläser for his unconditional support throughout.



© Sanyam Agarwal, Sagnik Dutta, Anurag Pandey, and Himanshu Shukla;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 19; pp. 19:1–19:18



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

COMPUTATIONAL COMPLEXITY CONFERENCE

1 Introduction

Deciding whether a multivariate polynomial system has a common zero is a central problem in mathematics, engineering, and computer science. Over an algebraically closed field, the classical *Weak Hilbert Nullstellensatz* (WHN) characterizes *non-emptiness* of an affine algebraic set via the existence of an explicit polynomial certificate, while the *Strong Hilbert Nullstellensatz* (SHN) characterizes *containment* of varieties by relating vanishing on a solution set to membership in a radical ideal (equivalently, by allowing powers). Formally, WHN says that polynomials $f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n]$ don't have a common root if and only if there exists $g_1, \ldots, g_n \in \mathbb{C}[x_1, \ldots, x_n]$ such that $\sum_i f_i g_i = 1$. Similarly, SHN says that $h \in \mathbb{C}[x_1, \ldots, x_n]$ vanishes on the common root set of $f_1, \ldots, f_m$ if and only if there exists $r \in \mathbb{Z}_{\geq 1}$ such that $h^r = \sum_i f_i g_i$ for some $g_1, \ldots, g_n \in \mathbb{C}[x_1, \ldots, x_n]$ [10, 31]. From the algorithmic viewpoint, the complexity of such feasibility and containment questions has been studied for decades, with PSPACE upper bounds appearing already in the context of algebraic decision problems and quantifier elimination [8, 26, 25]. In fact, over complex numbers, the feasibility/containment question can be decided in the low levels of the polynomial hierarchy [25], assuming the Generalized Riemann Hypothesis (GRH). Recently, Andrews, Garg, and Schost showed that existence of a solution over the algebraic closure of certain fields like the rational numbers, and number fields can be decided in the Counting Hierarchy [1].

In many settings, exact satisfiability is often too strong an ask. Hence, one cares about *infinitesimal approximation* rather than equality. An example that clearly demonstrates this is the following.

▶ **Example 1.1.** Let $f_1, f_2 \in \mathbb{C}[x, y]$ such that $f_1 = xy - 1$ and $f_2 = x$. Clearly, these polynomials have no common root over $\mathbb{C}^2$. However, geometrically, the curves seem to converge on the y-axis. To capture such convergence we can allow the variables to take values in $\mathbb{C}(\varepsilon)$ where ε is an *infinitesimally* small parameter. In this case, the point $p = (\varepsilon, \frac{1}{\varepsilon}) \in \mathbb{C}(\varepsilon)^2$ is an approximative common root the system, as $\lim_{\varepsilon \to 0} f_1(p) = \lim_{\varepsilon \to 0} f_2(p) = 0$.

Guo, Saxena, and Sinhababu (GSS) formalized this perspective by introducing *approximate polynomial satisfiability* (APS) [18].

▶ **Problem 1** (Approximate Polynomial Satisfiability (APS)). *Given polynomials* $\mathbf{f} = f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n]$*, does there exist* $\beta \in \mathbb{C}(\varepsilon)^n$ *such that for all* i*,* $f_i(\beta)$ *is in the ideal* $\varepsilon\mathbb{C}[\varepsilon]$ *of* $\mathbb{C}[\varepsilon]$*? If yes, then we say that* $\mathbf{f}$ *is in* APS*.*

▶ **Example 1.2.** While this notion of approximation allows us to capture more solutions, it doesn't mean that every system of polynomials now has a solution. For example, take $f_1, f_2 \in \mathbb{C}[x, y]$ such that $f_1 = xy$ and $f_2 = xy + 1$. No assignment to x, y from $\mathbb{C}(\varepsilon)^2$ can make this satisfiable in the above approximative sense.

▶ Remark 1.3. As shown in Example 1.1, approximative solutions differ from solutions in the affine space. Not only that, GSS showed that these approximative solutions *also* differed from solutions in the projective space. They showed that while every approximative solution is a projective solution, the converse is not true.

Informally, APS asks whether a polynomial system admits solutions that satisfy the constraints *to all orders* in an infinitesimal parameter (equivalently, whether a target point lies in the Zariski closure of a polynomial map's image, as made precise in [18]). This captures the "border" viewpoint pervasive in algebraic complexity, where one studies limits of efficiently computable objects. Further, they showed that APS captures canonical border phenomena such as tensor *border rank* (see [7, 27]).

Along with this, GSS also showed that computationally this problem is NP-Hard and, in fact, lies in PSPACE when the input polynomials are given as algebraic circuits. They do this by showing that existence of an approximative solution to a set of polynomials is equivalent to testing whether all the *annihilators* of the input polynomials have constant term 0 (Note that given a field k and polynomials $\mathbf{f} \coloneqq \{f_1, \ldots, f_m\} \in k[x_1, \ldots, x_n]$, an annihilator of $\mathbf{f}$ is a polynomial $A(y_1, \ldots, y_m) \in k[y_1, \ldots, y_m]$ such that $A(f_1, \ldots, f_m) = 0$).

We observe that this can be reformulated as a weak "approximative" Nullstellensatz. Intuitively, the non-existence of an approximate solution is equivalent to the existence of an annihilator having a non-zero constant (equivalently, constant term being 1) which is analogous to a variety defined by polynomials being empty if 1 lies in their ideal. Hence, we call their characterization the *Weak Approximative Nullstellensatz* (WAN), and formally define it below. Before that, we need to define the approximative vanishing set of a given system of polynomials.

▶ **Definition 1.4** (Approximative Vanishing Set (AV)). *Given polynomials* $\mathbf{f} \coloneqq f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n]$ *the approximative vanishing set* $\mathsf{AV}(\mathbf{f})$ *is defined as:*

$$\mathsf{AV}(\mathbf{f}) \coloneqq \{\beta \in \mathbb{C}(\varepsilon)^n \mid \forall i,\ f_i(\beta) = \varepsilon \frac{p_{i,\beta}(\varepsilon)}{q_{i,\beta}(\varepsilon)} \text{ with } p_{i,\beta}, q_{i,\beta} \in \mathbb{C}[z], \text{and } q_{i,\beta}(0) \neq 0\}.$$

▶ **Example 1.5.** Using Example 1.1, $\mathsf{AV}(f_1, f_2)$ contains the point $(\varepsilon, \frac{1}{\varepsilon})$. But it also contains many other points like $(\varepsilon^2, \frac{1}{\varepsilon^2}), (\frac{\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{\varepsilon})$, and so on.

▶ Remark 1.6. In [18], APS is defined as containing those $\beta \in \mathbb{C}(\varepsilon)^n$ such that $f_i(\beta)$ is in the ideal $\varepsilon\mathbb{C}[\varepsilon]$ of $\mathbb{C}[\varepsilon]$, while in our case $f_i(\beta)$ has numerators in the same ideal, but denominators are polynomials in ε with non-zero constant terms. These two conditions are equivalent when talking about non-emptiness of approximate vanishing sets. To see this assume for some $\beta \in \mathbb{C}(\varepsilon)^n$, for all i, $f_i(\beta) = \varepsilon \frac{p_{i,\beta}(\varepsilon)}{q_{i,\beta}(\varepsilon)}$ with $q_{i,\beta}(0) \neq 0$. Then, we can scale β to get $\beta' = \beta \prod_{i=1}^{m} q_{i,\beta}(\varepsilon)$ such that $f_i(\beta') = f_i(\beta) + \varepsilon h_i(\varepsilon)$ for some polynomials $h_i(z) \in \mathbb{C}[z]$. Hence, $f_i(\beta')$ lies in the ideal $\varepsilon\mathbb{C}[\varepsilon]$ of $\mathbb{C}[\varepsilon]$.

▶ Remark 1.7. We choose all such $\beta \in \mathbb{C}(\varepsilon)^n$ as geometrically all these points also ensure that $f_i(\beta)$ converges to 0 in the limit. Further, this is more aligned with the literature on valued fields (see [14] for more details on valued fields and Ganzstellensatz in § 1.2).

▶ **Theorem 1.8** (Weak Approximative Nullstellensatz (WAN), [18]). *Given polynomials* $\mathbf{f} = f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n]$, $\mathsf{AV}(\mathbf{f})$ *is empty* if and only if *there exists an annihilator of* $\mathbf{f}$ *with constant-term* 1.

Thus, GSS showed that computationally testing WAN is in PSPACE. Not only that, they showed that this enables the construction of hitting sets for $\overline{\mathsf{VP}}$ in PSPACE (over any field), thereby substantially narrowing the "GCT chasm" studied before in [33, 15]. For complex numbers, PSPACE-constructible hitting sets for the closure of small algebraic circuits already appear in the work of Forbes and Shpilka [16].

1.1 Our Contributions

The starting point of this work is the observation that the GSS algebraic criterion for *non-existence* of approximative solutions can be viewed as an analog of the *Weak Hilbert's Nullstellensatz* (WHN) in the approximative setting. In classical algebraic geometry, the next structural step is the *Strong Hilbert's Nullstellensatz* (SHN), which answers questions about containment of solutions. Formally, it asks whether polynomial constraints defining one variety *force* another polynomial to vanish? Motivated by this, we study the following natural problem about containment of approximative vanishing sets.

▶ **Problem 2.** *Given two systems of polynomials* $\mathbf{f} = (f_1, \ldots, f_m)$ *and* $\mathbf{g} = (g_1, \ldots, g_t)$ *over* $\mathbb{C}$*, decide whether the* approximative solution set *of* $\mathbf{f}$ *is contained in that of* $\mathbf{g}$*, i.e., whether every infinitesimal solution to* $\mathbf{f}$ *necessarily satisfies* $\mathbf{g}$ *(in the same approximative sense).*

The containment condition mentioned above is not always obvious to check, as demonstrated in the following example.

▶ **Example 1.9.** Given $f_1, f_2, g, h \in \mathbb{C}[x, y]$ where $f_1 = x, f_2 = xy - 1, g = x^2y, h = xy^2 - 1$.

- *Containment of* AV*:* It can be verified that $\mathsf{AV}(f_1, f_2) \supseteq \mathsf{AV}(g, f_2)$.
- *Non-containment of* AV*:* It is easy to see that $\mathsf{AV}(f_1, h) \supsetneq \mathsf{AV}(f_1, f_2)$ as $(\varepsilon, \frac{1}{\varepsilon})$ lies in $\mathsf{AV}(f_1, f_2)$ but not $\mathsf{AV}(f_1, h)$, and similarly $\mathsf{AV}(f_1, h) \subsetneq \mathsf{AV}(f_1, f_2)$ as $(\varepsilon^2, \frac{1}{\varepsilon})$ lies in $\mathsf{AV}(f_1, h)$ but not $\mathsf{AV}(f_1, f_2)$.

▶ Remark 1.10. We do not need to always check the containment for the whole system of polynomials together. Instead, we can check containment for each individual polynomial. This is because $\mathsf{AV}(g_1, \ldots, g_t) = \cap_{i=1}^{t} \mathsf{AV}(g_i)$. Hence, checking $\mathsf{AV}(g_1, \ldots, g_t) \supseteq \mathsf{AV}(f_1, \ldots, f_m)$ is equivalent to checking $\mathsf{AV}(g_i) \supseteq \mathsf{AV}(f_1, \ldots, f_m)$ for each $i \in [t]$.

In the approximative setting, this is the natural strengthening of WAN. To resolve this question, we prove our main result.

▶ **Theorem 1.11** (Strong Approximative Nullstellensatz (SAN))**.** *Given a set of polynomials* $f_1, \ldots, f_m, g \in \mathbb{C}[x_1, \ldots, x_n]$*,*

$$\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, \ldots, f_m) \quad \text{if and only if} \quad g^r = \sum_{i=0}^{r-1} \frac{F_i(f_1, \ldots, f_m)}{1 + G_i(f_1, \ldots, f_m)} g^i,$$

for some $r \in \mathbb{N}$*, and constant-free polynomials* $F_i, G_i \in \mathbb{C}[y_1, \ldots, y_m]$*, for* $0 \leq i \leq r - 1$*.*

where a polynomial $h(x_1, \ldots, x_n) \in \mathbb{C}[x_1, \ldots, x_n]$ is constant-free if $h(0, \ldots, 0) = 0$. Note that, we can always take the common denominator in the above formulation to ensure that the above formulation is equivalent to $g^r = \sum_{i=0}^{r-1} \frac{F'_i(f_1, \ldots, f_m)}{1 + G(f_1, \ldots, f_m)} g^i$ where F'_i, G are constant-free.

▶ **Example 1.12.** Consider polynomials $f_1, f_2 \in \mathbb{C}[x, y]$ where $f_1 = xy^2 - y + x^2y$ and $f_2 = xy - 1$. Then, it can be verified that $g = x$ such that $\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, f_2)$. On first look, its hard to see why this is true. But if we use our algebraic criterion, then $g^2 = \frac{f_1}{1 + f_2} g - f_2$ certifies this containment.

Clearly, SAN implies WAN: if $\mathsf{AV}(f_1, \ldots, f_m)$ was empty then 1 would lie in their integral closure. Using the characetrization above, we can rearrange to get $1 = A(f_1, \ldots, f_m)$ where $A(y_1, \ldots, y_m)$ was constant-free in $\mathbb{C}[y_1, \ldots, y_m]$. But this simply means that there exists an annihilator with constant-term 1. Further, our containment criterion can be reformulated as a membership condition in the *integral closure* over a maximal ideal of a local subring of $\mathbb{C}(x_1, \ldots, x_n)$ determined by $\mathbf{f}$, see Section 2 for more details. This connects approximative containment to classical commutative-algebraic notions surrounding integral dependence and closure* [20, 38]. We informally state our correspondence theorem here. For a formal statement, see Theorem 2.5.

▶ **Theorem 1.13** (Approximate Vanishing to Integral Closure correspondence (Informal))**.** *For* $f_1, \ldots, f_m, g \in \mathbb{C}[x_1, \ldots, x_n]$*,* $\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, \ldots, f_m)$ if and only if g *lies in the integral closure of a local subring of* $\mathbb{C}(x_1, \ldots, x_n)$ *determined by* $f_1, \ldots, f_m$*.*

* Please refer to Atiyah and Macdonald [3, Chapter 5] for an overview of these concepts.

As discussed before, GSS also gave a PSPACE algorithm for testing WAN. In the same spirit, we also look at the computational problem defined by SAN. We assume the polynomials are given as algebraic circuits. Formally, we define the computational problem of testing SAN as the following:

▶ **Problem 3** (Computational Problem of testing SAN). *Given polynomials* $f_1, \ldots, f_m, g \in \mathbb{C}[x_1, \ldots, x_n]$, SAN *is the decision problem asking whether* $\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, \ldots, f_m)$.

Depending on the context, we will use SAN interchangeably for both the decision problem and the containment criteria. We show that testing containment of approximate vanishing sets can also be done in PSPACE, thus matching the upper bound of WAN.

▶ **Theorem 1.14.** SAN *is in* PSPACE.

This beautifully coincides with the known matching upper bounds of the WHN and SHN in conventional algebraic geometry, where it is typically achieved through the use of "Rabinowitsch's trick" [37].

1.2 Related works

In this section, we survey the known related works and their implications in our setting.

Hilbert's Nullstellensatz. It is well-known that the Weak Hilbert's Nullstellensatz (WHN) and the Strong Hilbert's Nullstellensatz (SHN) are equivalent [10]. Typically, this equivalence is achieved using Rabinowitsch's trick [37]. The trick works like this: given polynomials $f_1, .., f_m, g \in \mathbb{C}[x_1, .., x_n]$, checking whether the zeroset of g contains the zeroset of $f_1, \ldots, f_m$ is equivalent to checking if the zeroset of the polynomials $f_1, \ldots, f_m, 1 - yg \in \mathbb{C}[x_1, .., x_n, y]$ is empty or not. Further, this also ensures that WHN and SHN are computationally equivalent. Once this equivalence is established in the classical setting, we can use *effective* Nullstellensatz bounds [26] to obtain efficient algorithms. Hence, a natural starting point is to see if this correspondence also holds in the approximative setting. However, this is not the case and Rabinowitsch's trick can't be directly used to get equivalence between WAN and SAN. This can be intuitively seen as follows: Suppose at a point $p \in \mathbb{C}^n$, $f_i(p) = g(p) = 0$ for all i. Then, clearly the polynomial $1 - yg$ is non-zero. However, in the approximative setting this does not hold. Since we are now looking at approximative vanishing instead of exact vanishing, it might be the case that at $p \in \mathbb{C}(\varepsilon)^n$, $g(p) = \varepsilon \frac{p(\varepsilon)}{q(\varepsilon)}$ where $q(0) \neq 0$ and $p(\varepsilon) \neq 0$. Then setting $y = \frac{1}{\varepsilon} \frac{q(\varepsilon)}{p(\varepsilon)} \in \mathbb{C}(\varepsilon)$ ensures that the polynomial $1 - yg$ approximatively vanishes. Thus, this extra polynomial doesn't fully capture the notion of approximative vanishing. hence, the conventional Rabinowitsch trick can't be used to show equivalence of WAN and SAN. Thus, this also makes the path to obtain efficient runtime for the decision problem SAN unclear.

ε-approximate Nullstellensatz. In an interesting parallel direction, Göös et al. [17] defined the notion of *ε-approximate Nullstellensatz*. In particular, they used a robust version of the WHN to show separations in proof complexity where they show that *Resolution proofs* cannot be efficiently simulated by *Sherali-Adams proofs* when the coefficients are written in unary.

Subalgebra Membership. Looking at the geometric containment $\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, \ldots, f_m)$, it is natural to ask whether the notion of subalgebra membership can be used to capture this algebraically. Given polynomials $g, f_1, \ldots, f_m \in \mathbb{C}[x_1, .., x_n]$, the *subalgebra membership* problem asks if g can be written as $g = \sum a_k f_1^{k_1} \ldots f_m^{k_m}$, where the sum is finite. Recently,

it was shown that, in the general case, this problem is EXPSPACE-complete, see [23] for definitions and more details. However, while subalgebra membership implies containment of approximative vanishing sets, the converse is not true. This can be seen by the following example.

▶ **Example 1.15.** Given $f_1, f_2 \in \mathbb{C}[x, y]$, such that $f_1 = x^2$ and $f_2 = x^2y - 1$. Let $g = x$. It is easy to verify that $\mathsf{AV}(g) = \mathsf{AV}(f_1)$, and hence $\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, f_2)$. However, g doesn't lie in the subalgebra formed by $\mathbb{C}[f_1, f_2]$.

Ganzstellensatz. The closest related notions to SAN in literature comes from the work on *Ganzstellensatz*, see [24, 36, 19, 28] for more details. These works approach this question through the point of view of valuations. Let $K \coloneqq \mathbb{C}\{\{\varepsilon\}\}$, i.e., field of *Puiseux series* over $\mathbb{C}$ with the ε-adic valuation val_ε of ε normalized to 1, and let $g, f_1, \ldots, f_m \in K[x_1, \ldots, x_n]$. Consider the set $S \subset K^n$ such that for all $s \in S$ and for all $i \in [m]$, $\mathrm{val}_\varepsilon(f_i(s)) \geq 0$. Then they give an algebraic criterion on g such that for all $s \in S$, $\mathrm{val}_\varepsilon(g(s)) \geq 0$. Restricting to our setting, i.e., when $g, f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n]$, the criterion of Haskell and Yaffe [19] gives that g is integral over $\mathbb{C}[f_1, \ldots, f_m]$, i.e., $g^r = \sum_{i=0}^{r-1} F_i g^i$ for some $r \in \mathbb{N}$ and $F_i \in \mathbb{C}[f_1, \ldots, f_m]$. Clearly, if g is integral over $\mathbb{C}[f_1, \ldots, f_m]$, then $\mathrm{val}_\varepsilon(g(s)) \geq 0$, for all $s \in K^n$ such that for all $i \in [m]$, $\mathrm{val}_\varepsilon(f_i(s)) \geq 0$. .

Another way to interpret this is the following: they look at all points such that when $f_1, \ldots, f_m$ is "well-behaved", then so is g. That is, for all points s in $S = \{\beta \in K^n \mid \lim_{\varepsilon \to 0} f_i(\beta) = a,\ a \in \mathbb{C}\}$, they want $\lim_{\varepsilon \to 0} g(s) = b$ for some $b \in \mathbb{C}$. In particular, they don't enforce the condition of "vanishing". While close to our setting, it is a different characterization. The following example demonstrates this.

▶ **Example 1.16.** Using Example 1.15, we know that for $f_1 = x^2y$ and $f_2 = xy - 1$, $g = x$ contains their approximative vanishing set. However, for $s = (\frac{1}{\epsilon}, \epsilon^3)$, we have $\lim_{\varepsilon \to 0} f_1(s) = 0, \lim_{\varepsilon \to 0} f_2(s) = -1$, but $\lim_{\varepsilon \to 0} g(s)$ does not exist. Hence, g does not lie in the *Ganzstellensatz* criterion for f_1, f_2 defined above.

1.3 Our methods

In this section, our aim is to convey the fundamental ideas behind the proofs of our main results. Hence, we occasionally gloss over the finer technical arguments needed to formally prove the claims.

Proof ideas for Theorem 1.11 (SAN). One direction is easy to show, that is if g lies in the defined integral closure, then it will contain the approximative vanishing set of $f_1, \ldots, f_m$. Intuitively, when all the f_i vanish approximatively, so will F_i, G_i in Theorem 1.11, hence so will g, see Proof 2 for details. For the converse direction, as discussed above, the conventional Rabinowitsch's trick doesn't work in the approximative setting. However, when geometrically looking at the containment of approximative vanishing sets, simultaneous approximative vanishing of $f_1, \ldots, f_m$ forces g to also approximatively vanish, which is equivalent to saying that g neither diverges nor converges to a non-zero constant $c \in \mathbb{C}$ (see Lemma 2.1 for more details). We call these two conditions the "*approximative* Rabinowitsch tricks". Using these two conditions along with GSS criterion (Theorem 1.8) we can obtain the desired formulation as follows. Using the first trick (the non-divergence condition), we first obtain a univariate polynomial P of degree r with coefficients from $\mathbb{C}[f_1, ..., f_m]$, such that g satisfies P (see Lemma 2.2). Using the obtained polynomial recursively along with the second approximative Rabinowitsch trick (non-convergence to any non-zero constant), we can get the claimed characterization (refer to Lemma 2.3 and subsequent discussion).

Proof ideas for Theorem 1.14 (SAN in PSPACE). Apriori, it is not clear how to put the decision problem of SAN in PSPACE using the geometric conditions obtained in Lemma 2.1. This is because the second approximative Rabinowitsch's condition (non-convergence to any non-zero constant) requires the elimination of a *universal* quantifier. Quantifier elimination for algebraically closed fields and its complexity has a long history, with foundational algorithms due to Chistov and Grigor'ev [9]. Ierardi developed an exponential-space decision procedure for the first-order theory of algebraically closed fields and showed improved space bounds under bounded alternations [21]. Hence, a naive elimination-based approach to SAN would suggest an EXPSPACE upper bound. Even if we use the algebraic criterion devised in Theroem 1.11, it is not clear how to put the decision problem in PSPACE. de Jong [12] gave an algorithm for computing the integral closure which requires computation of ideals in the intermediate steps. Typically this involves the use of Gröbner Basis methods, for which doubly-exponential degree bounds are known in the worst case [30, 13, 2]. Consequently, one should expect an EXPSPACE algorithm.

However, we are looking for polynomials in a specific integral closure defined by a local subring determined by the input polynomials. This additional structure, along with known annihilator bounds (see Theorem 3.1) help us in attaining a PSPACE algorithm. We can interpret our algebraic characterization in Theorem 1.11 as a univariate polynomial P of degree r with coefficients from $\mathbb{C}[f_1, ..., f_m]$, such that g satisfies P. Hence, this means that the polynomials $g, f_1, \ldots, f_m$ have a non-zero annihilator. However, currently, we cannot say anything about the degree bounds of this annihilator, as Theorem 3.1 does not guarantee an exponential degree bound for *all* the annihilators, but the minimal one, and the annihilator obtained through P might not be the minimal annihilator of $g, f_1, \ldots, f_m$. To overcome this, we now use the random reduction trick introduced by GSS (see Theorem 3.2), where they show that given a system of polynomials $f_1, \ldots, f_m$ of transcendence degree ρ, we can reduce to $\rho+1$ random linear combinations of these f_i such that the non-emptiness of AV is preserved. The advantage of doing this is the following: the annihilator ideal of ℓ polynomials is principal if the transcendence degree is $\ell - 1$ (see [18, 22]). Once we have this, we can prove the *effective* version of Lemma 2.2 and Lemma 2.3, both of which only deal with *non-emptiness* of approximative vanishing set, thanks to our *approximative* Rabinowitsch tricks, and hence, GSS' random reduction lemma can indeed be invoked. Repeated application of this helps us to finally upper bound, in Theorem 1.11, the exponent r and the degrees of F_i's and G_i's to at most singly exponential in the input parameters using the degree bound on the minimal annihilating polynomial [34, 22]. Finally, we only need to solve a system of exponentially many linear equations in exponentially many variables, which is known to be in PSPACE [5, 11, 32], see Section 3 for details.

2 Proof of Strong Approximative Nullstellensatz

In this section we prove our main characterization theorem, the Strong Approximative Nullstellensatz, where analogous to the ideal-variety correspondence in algebraic geometry, we have the correspondence between the integral closure (over a maximal ideal of a local/fractional/natural subring of $\mathbb{C}(x_1, \ldots, x_n)$ determined by $f_1, \ldots, f_m$) and the approximative vanishing set (recall Definition 1.4).

It is easy to see the correspondence with WHN in the classical setting: the vanishing set $V(f_1, \ldots, f_m)$ of $f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n]$ is empty $\iff 1 \in \langle f_1, \ldots, f_m \rangle$. Similarly, in the classical setting SHN says that the ideal generated by all functions vanishing on $V(f_1, \ldots, f_m)$ is exactly the radicle ideal $\sqrt{\langle f_1, \ldots, f_m \rangle}$, or in other words, if $f \in \mathbb{C}[x_1, \ldots, x_n]$, then $f^r \in \langle f_1, \ldots, f_m \rangle \iff V(f_1, \ldots, f_m) \subseteq V(f)$. We now present the natural analogue of SHN in the approximative setting.

▶ **Theorem** (Strong Approximative Nullstellensatz (SAN)). *Given* $f_1, \ldots, f_m, g \in \mathbb{C}[x_1, \ldots, x_n]$,

$$\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, \ldots, f_m) \quad \text{if and only if} \quad g^r = \sum_{i=0}^{r-1} \frac{F_i}{1+G_i} g^i,$$

for some $r \in \mathbb{N}$*, where* F_i, G_i *are constant-free polynomials in* $\mathbb{C}[f_1, \ldots, f_m]$.

We first prove the easier direction of the theorem, that the algebraic condition implies the geometric condition: g being in the integral closure implies that the approximative vanishing set of g contains the common approximative vanishing set of $f_1, \ldots, f_m$. Before moving forward, we give some definitions. For a nonzero $h(\varepsilon) \in \mathbb{C}(\varepsilon)$, define $\mathrm{ord}_\varepsilon(h)$ to be the *highest power of* ε *dividing* h, i.e., the unique integer $t \in \mathbb{Z}$ such that

$$h(\varepsilon) = \varepsilon^t \cdot \frac{p(\varepsilon)}{q(\varepsilon)} \quad \text{with } p, q \in \mathbb{C}[\varepsilon],\ p(0) \neq 0,\ q(0) \neq 0.$$

Set $\mathrm{ord}_\varepsilon(0) = +\infty$. Then $\mathrm{ord}_\varepsilon(h_1 + h_2) \geq \min\{\mathrm{ord}_\varepsilon(h_1), \mathrm{ord}_\varepsilon(h_2)\}$, where equality holds if $\mathrm{ord}_\varepsilon(h_1) \neq \mathrm{ord}_\varepsilon(h_2)$, and $\mathrm{ord}_\varepsilon(h_1 h_2) = \mathrm{ord}_\varepsilon(h_1) + \mathrm{ord}_\varepsilon(h_2)$. We say $h \equiv 0 \mod \varepsilon \iff \mathrm{ord}_\varepsilon(h) \geq 1$, and $h \equiv c \mod \varepsilon$, for some $c \in \mathbb{C} \setminus \{0\} \iff \mathrm{ord}_\varepsilon(h) = 0$ and $c = h(0)$.

Proof. ($\Leftarrow$) Assume that for some $r \in \mathbb{N}$ and $F_i, G_i \in \mathbb{C}[f_1, \ldots, f_m]$ constant-free polynomials, for $0 \leq i \leq r-1$, the identity

$$g^r = \sum_{i=0}^{r-1} \frac{F_i}{1+G_i}\, g^i$$

holds. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{C}(\varepsilon)^n$ be such that for all $1 \leq j \leq m$, $f_j(\mathbf{x}) \equiv 0 \mod \varepsilon$. We show that $g(\mathbf{x}) \equiv 0 \mod \varepsilon$. Since each F_i and G_i is constant-free in $\mathbb{C}[f_1, \ldots, f_m]$, every monomial in F_i (resp. G_i) contains at least one f_j. Therefore, for $0 \leq i \leq r-1$, substituting $\mathbf{x}$ gives $\mathrm{ord}_\varepsilon(F_i) \geq 1$ and $\mathrm{ord}_\varepsilon(G_i) \geq 1$. In particular, $\mathrm{ord}_\varepsilon(1 + G_i(\mathbf{x})) = \min\{\mathrm{ord}_\varepsilon(1), \mathrm{ord}_\varepsilon(G_i(\mathbf{x}))\}$. Let $u := g(\mathbf{x}) \in \mathbb{C}(\varepsilon)$ and define

$$c_i := \frac{F_i(\mathbf{x})}{1+G_i(\mathbf{x})} \in \mathbb{C}(\varepsilon).$$

Therefore, $\mathrm{ord}_\varepsilon(c_i) = \mathrm{ord}_\varepsilon(F_i(\mathbf{x}) - \mathrm{ord}_\varepsilon(1 + G_i(\mathbf{x})) \geq 1$, and $u^r = \sum_{i=0}^{r-1} c_i\, u^i$. Comparing the orders one obtains

$$r\,\mathrm{ord}_\varepsilon(u) \geq \min_{0 \leq i \leq r-1} \big(\mathrm{ord}_\varepsilon(c_i u^i)\big) = \min_{0 \leq i \leq r-1} \big(\mathrm{ord}_\varepsilon(c_i) + i\,\mathrm{ord}_\varepsilon(u)\big) \geq \min_{0 \leq i \leq r-1} (1 + i\,\mathrm{ord}_\varepsilon(u)) \geq 1.$$

Since $\mathrm{ord}_\varepsilon(u) \in \mathbb{Z}$, so $\mathrm{ord}_\varepsilon(u) \geq 1$, or equivalently $g(\mathbf{x}) \equiv 0 \mod \varepsilon$. ◀

The converse direction – showing that the geometric condition implies the algebraic condition – is significantly more involved. The first step towards this is to reduce the containment problem for approximative vanishing sets to the problem of emptiness of approximative vanishing set, taking inspiration from the classical *Rabinowitsch trick*. In contrast to the classical Rabinowitsch trick, we do not obtain a single augmented system whose WAN-infeasibility is equivalent to containment. Instead, we obtain one fixed APS-infeasibility instance (see Condition 2) together with a *family* of APS-infeasibility instances indexed by all nonzero scalars $c \in \mathbb{C}$ (see Condition 1). A key additional step is a finiteness reduction: although, Condition 1 is stated universally over all $c \neq 0$, we show that it suffices to check it for only *finitely many* values of c, given by bounded-degree algebraic dependencies arising in our proof. This finiteness reduction is essential for turning the apparent "for-all" obstacle into an efficient (ultimately PSPACE) decision procedure.

The following lemma shows that the geometric condition $\mathsf{AV}(f_1, \ldots, f_m) \subseteq \mathsf{AV}(g)$ can be intuitively rephrased as follows: *Whenever $f_1, \ldots, f_m$ simultaneously approximatively vanish, the value of g can neither diverge nor converge to a non-zero constant $c \in \mathbb{C}$, consequently, it must converge to 0.*

▶ **Lemma 2.1.** $\mathsf{AV}(f_1, \ldots, f_m) \subseteq \mathsf{AV}(g)$ if and only if *both the following conditions hold.*

1. *For all $c \in \mathbb{C} \setminus \{0\}$, $f_1, \ldots, f_m, g - c$ have no common* APS*, and*
2. *$f_1, \ldots, f_m, y, 1 - yg$ have no common* APS*.*

Proof. We assume $\mathsf{AV}(f_1, \ldots, f_m) \neq \emptyset$ as otherwise $\mathsf{AV}(g)$ always contains $\mathsf{AV}(f_1, ..., f_m)$. Recall that for $g \in \mathbb{C}[x_1, \ldots, x_n]$, $\mathbf{x} \in \mathsf{AV}(g) \iff g(\mathbf{x}) \equiv 0 \mod \varepsilon \iff \mathrm{ord}_\varepsilon(g(\mathbf{x})) \geq 1$. ($\Rightarrow$) Assume, first, that $\mathsf{AV}(f_1, \ldots, f_m) \subseteq \mathsf{AV}(g)$.

- *Proof of condition 1*: Let $c \in \mathbb{C}$. If $f_1, \ldots, f_m, g - c$ had a common APS at some $\mathbf{x} \in \mathbb{C}(\varepsilon)^n$, then for all $1 \leq i \leq m$, $f_i(\mathbf{x}) \equiv 0 \mod \varepsilon$, and $(g - c)(\mathbf{x}) \equiv 0 \mod \varepsilon$. Therefore, $g(\mathbf{x}) \equiv c \mod \varepsilon$. Therefore, $c = 0$, and $f_1, \ldots, f_m$ and $g - c$ have no common APS, for $c \neq 0$.
- *Proof of condition 2.* If $f_1, \ldots, f_m, y, 1 - yg$ had a common APS $(\mathbf{x}, h) \in \mathbb{C}(\varepsilon)^{n+1}$. Then $h \equiv 0 \mod \varepsilon$, and for all $1 \leq i \leq m$, $f_i(\mathbf{x}) \equiv 0 \mod \varepsilon$, and $hg(\mathbf{x}) \equiv 1 \mod \varepsilon$. Since $\mathsf{AV}(f_1, \ldots, f_m) \subseteq \mathsf{AV}(g)$, $\mathrm{ord}_\varepsilon(g(\mathbf{x})) \geq 1$, which contradicts $hg(\mathbf{x}) \equiv 1 \mod \varepsilon$.

($\Leftarrow$) Assume both conditions 1 and 2 hold, and let $\mathbf{x} \in \mathsf{AV}(f_1, \ldots, f_m) \setminus \mathsf{AV}(g)$. Then $\mathrm{ord}_\varepsilon(g(\mathbf{x})) \leq 0$. If $\mathrm{ord}_\varepsilon(g(\mathbf{x})) = 0$, then there is a $c \in \mathbb{C}^\times$ such that $g(\mathbf{x}) \equiv c \mod \varepsilon$. Therefore, $(g - c)(\mathbf{x}) \equiv 0 \mod \varepsilon$, which contradicts the condition 1. On the other hand, if $\mathrm{ord}_\varepsilon(g(\mathbf{x})) = t \leq -1$. Then $(\mathbf{x}, h)$, where $h = \frac{\varepsilon^{-t}}{(\varepsilon^{-t} g(\mathbf{x}))(0)} \equiv 0 \mod \varepsilon$, is in $\mathsf{AV}(f_1, \ldots, f_m, y, 1 - yg)$, thus contradicting the condition 2. ◀

While Lemma 2.1 gives us geometric connection in relation to common solutions, it still doesn't give any insight algebraically into the problem. We now show how we can use the two conditions defined in the lemma to get an algebraic relation between the polynomials $f_1, \ldots, f_m, f$.

▶ **Lemma 2.2.** *Suppose $f_1, \ldots, f_m, y, 1 - yg$ have no common* APS*. Then for some $r \geq 1$,*

$$g^r = \sum_{i=0}^{r} g^i Q_i(f_1, \ldots, f_m),$$

where $Q_r \in \mathbb{C}[y_1, \ldots, y_m]$ is a constant-free polynomial.

Proof. Using Theorem 1.8, if $f_1, \ldots, f_m, y, 1 - yg$ have no common APS, then there exists a constant-free polynomial A in $\mathbb{C}[y_1, \ldots, y_{m+2}]$ such that $A(f_1, \ldots, f_m, y, 1 - yg) = 1$, (the polynomial $A(y_1, \ldots, y_{m+2}) - 1$ is an annihilator of $f_1, \ldots, f_m, y, 1 - yg$). Now the polynomial $A(f_1, \ldots, f_m, y, 1 - yg) - 1 \in \mathbb{C}[x_1, \ldots, x_n, y]$ is the 0 polynomial and remains so in $\mathbb{C}(x_1, \ldots, x_n)[y]$. We can set $y = \frac{1}{g}$, to get

$$A(f_1, \ldots, f_m, \frac{1}{g}, 0) = 1 \tag{1}$$

Therefore, $A(y_1, \ldots, y_{m+2}) \neq 0 \mod y_{m+2}$, otherwise Equation 1 gives us $0 = 1$. Consequently, there is an $r \geq 1$ and $G_i \in C[f_1, \ldots, f_m]$ with G_0 constant-free (since A was constant-free) such that

$$\sum_{i=0}^{r} \frac{1}{g^i} G_i(f_1, \ldots, f_m) = 1$$

Rearranging, we get $g^r = \sum_{i=0}^{r} g^i Q_i(f_1, \ldots, f_m)$, where Q_r is constant-free. ◀

Thus, we have shown how we can use condition (2) to get a polynomial relation. Using this result along with ondition (1), we can show the following.

▶ **Lemma 2.3.** *Assume that for all $c \in \mathbb{C}^\times$, $f_1, \ldots, f_m, g - c$ have no common* APS*. Let $r \geq 1$, and for $1 \leq i \leq r$ let $Q_i \in \mathbb{C}[y_1, \ldots, y_m]$ be polynomials with Q_r constant-free such that*

$$g^r = \sum_{i=0}^{r} g^i Q_i(f_1, \ldots, f_m).$$

Then there are $1 \leq k \leq r$, $l \geq 1$, and constant-free polynomials $F_i \in \mathbb{C}[y_1, \ldots, y_m]$, for $1 \leq i \leq l$, such that

$$g^k = \sum_{i=0}^{l} g^i F_i(f_1, \ldots, f_m).$$

Proof. If Q_i is constant-free for all i, then we set $k = l = r$ and $F_i = Q_i$ and we are done. Otherwise, there is atleast some Q_i with non-zero constant term. Consider the polynomial $A \in \mathbb{C}[y_1, \ldots, y_m, z]$ given by $z^{r+1}(1 - Q_r) - \left(\sum_{i=1}^{r} Q_{i-1} z^i\right)$. Clearly A is constant-free and $A(f_1, \ldots, f_m, g) = 0$. We rewrite each $Q_i = Q_i' + c_i$ such that Q_i' is constant-free for each i. We know that $c_r = 0$ and for some $i < r$, $c_i \neq 0$. Then $A = h(z) - \left(\sum_{i=1}^{r+1} Q'_{r-1} z^i\right)$, where $h(z) = z^{r+1} - c_{r-1} z^r - \ldots - c_0 z$ is a polynomial in $\mathbb{C}[z]$.

Clearly $h(z) \neq z^{r+1}$, so there is a finite non-empty subset $S \subset \mathbb{C}^\times$, positive integers e_s, for each $s \in S$, and $1 \leq k \leq r$ such that $h(z) = z^k \prod_{s \in S} (z - s)^{e_s}$. Therefore, $A(f_1, \ldots, f_m, g) = 0$ gives $h(f) = A'(f_1, \ldots, f_m, g)$, where $A' \in \mathbb{C}[y_1, \ldots, y_m, z]$ has no term of the form z^i, as Q_i' are constant-free.

Now we use condition (1) from Lemma 2.1. For all $s \in \mathbb{C}^\times$, $f_1, \ldots, f_m, g - s$ have no APS. In particular, this holds for $s \in S$. Then, by Theorem 1.8, for each $s \in S$ there exists a constant-free polynomial $F_s \in \mathbb{C}[y_1, \ldots, y_m, z]$ such that $F_s(f_1, \ldots, f_m, g - s) = 1$. Write $F_s = G_s(y_1, \ldots, y_m) + z H_s(y_1, \ldots, y_m, z)$ with G_s constant-free. Then

$$\left(\prod_{s \in S} (z - s)^{e_s} H_s(y_1, \ldots, y_m, z - s)^{e_s} - \prod_{s \in S} (1 - G_s(y_1, \ldots, y_m))^{e_s}\right)(f_1, \ldots, f_m, g) = 0.$$

Hence, there are $H \in \mathbb{C}[y_1, \ldots, y_m, z]$ and constant-free $G \in \mathbb{C}[y_1, \ldots, y_m]$ such that

$$H(f_1, \ldots, f_m, g)\, h(g) = g^k + g^k \cdot G(f_1, \ldots, f_m).$$

Therefore, $g^k = h(g) H(f_1, \ldots, f_m, g) - g^k \cdot G(f_1, \ldots, f_m)$, and hence

$$g^k = A'(f_1, \ldots, f_m, g) H(f_1, \ldots, f_m, g) - g^k \cdot G(f_1, \ldots, f_m),$$

and the polynomial $A'H - z^k G \in \mathbb{C}[y_1, \ldots, y_m, z]$ has no term of the form z^i. To see this note that the least degree of coefficient of z^i in $A'H$ and G is greater than 1. Therefore, there is an $l \geq 1$, and for $1 \leq i \leq l$, there are constant-free polynomials $F_i \in \mathbb{C}[y_1, \ldots, y_m]$ such that $g^k = \sum_{i=0}^{l} F_i(f_1, \ldots, f_m)$. ◀

To summarize, we have obtained the following equivalence $\mathsf{AV}(f_1, \ldots, f_m) \subseteq \mathsf{AV}(g) \iff$ there are $r, k, l \geq 1$ with $k \leq r$, and polynomials Q_i and F_j, for $0 \leq i \leq r$ and $0 \leq j \leq l$ in

$\mathbb{C}[y_1, \ldots, y_m]$ with Q_r and all F_j constant-free such that

$$g^r = \sum_{i=0}^{r} g^i Q_i(f_1, \ldots, f_m), \quad \text{and} \tag{2}$$

$$g^k = \sum_{i=0}^{l} g^i F_i(f_1, \ldots, f_m). \tag{3}$$

Using this, we now show how we can prove the other direction of Theorem 1.11.

Proof of Theorem 1.11 continued. Suppose $l \leq k$, then we can simply rearrange Equation 3 to get $g^k = \sum_{i=0}^{k-1} \frac{F_i}{1+F_k} g^i$ and we have the desired characterization. Now assume that $l > k$.

By Lemma 2.3, we can see that $1 \leq k \leq r$. Then we multiply Equation 3 by g^{r-k} on both sides so that the left hand side has exponent r and right hand side has maximum exponent $l' = l + (r - k)$. Since $l > k$, we have $l' = l + r - k > r$. We will now rename $l' = l$ and thus, we always have $l > r$. We can rewrite Equation 3 as

$$g^r = \sum_{i=0}^{l} g^i F_i(f_1, \ldots, f_m) \tag{4}$$

where $F_0 = \cdots = F_{r-k-1} = 0$. Now, both equations have the same exponent r on the left hand side. Rewrite Equation 2 to get $g^r(1 - Q_r) = \sum_{i=0}^{r-1} g^i Q_i$. Multiplying Equation (4) by $1 - Q_r$, we get

$$\begin{aligned}(1 - Q_r)g^r &= \sum_{i=0}^{l-1} g^i F_i(1 - Q_r) + g^{l-r} F_l g^r (1 - Q_r) \\ &= \sum_{i=0}^{l-1} g^i F_i(1 - Q_r) + g^{l-r} F_l (\sum_{j=0}^{r-1} g^j Q_j) = \sum_{i=0}^{l-1} g^i (F_i(1 - Q_r) + B_i) = \sum_{i=0}^{l-1} g^i H_i\end{aligned}$$

where for $0 \leq i \leq l - r - 1$, $B_i = 0$, and for all $l - r \leq i \leq l - 1$, $B_i = F_l Q_{i-(l-r)}$. Since F_i are constant-free, it is easy to see that H_i will be constant-free. Hence, starting from Equation 4, where g^r was a linear combination of powers of g up to l we have reduced it to linear combination of powers of g up to $l - 1$ over $\mathbb{C}[f_1, \ldots, f_m]$ such that $g^r = Q_r g^r + \sum_{i=0}^{l-1} H_i(f_1, \ldots, f_m) g^i$ with each H_i, and $Q_r \in \mathbb{C}[y_1, \ldots, y_m]$ being constant-free. We can do this inductively as long as $l \geq r$ to eventually get

$$g^r = \sum_{i=0}^{r} g^i F_i(f_1, \ldots, f_m), \tag{5}$$

where $F_i \in \mathbb{C}[y_1, \ldots, y_m]$ is constant-free for all i. This can be restated as

$$g^r = \sum_{i=0}^{r-1} g^i \frac{F_i(f_1, \ldots, f_m)}{1 - F_r(f_1, \ldots, f_m)},$$

which is exactly the formulation we desired in Theorem 1.11. ◀

In order to restate our theorem in the language of integral closures, consider the subring $R \coloneqq \mathbb{C}[f_1, \ldots, f_m]$. Note that $1 \in R$. Now, $\mathfrak{m} \coloneqq \langle f_1, \ldots, f_m \rangle$ is the maximal ideal of R.

▶ Remark 2.4. Notice that the above ring R differs from the standard polynomial ring $\mathbb{C}[x_1, \ldots, x_n]$. In fact, $R \subseteq \mathbb{C}[x_1, \ldots, x_n]$. Further while $\mathfrak{m}$ is a maximal ideal of R, it might not even be an ideal in $\mathbb{C}[x_1, \ldots, x_n]$.

Let $S \coloneqq R \setminus \mathfrak{m}$. Intuitively, S will contain evaluations at $f_1, \ldots, f_m$ of all polynomials $h \in \mathbb{C}[y_1, \ldots, y_m]$ with constant term not 0. Consider the ring $R_{\mathfrak{m}} \coloneqq S^{-1}R$. It is well known (see [3, Chapter 3]) that in this case $R_{\mathfrak{m}}$ is a local ring, i.e., it will have a unique maximal ideal $\mathfrak{m}R_{\mathfrak{m}}$ which is exactly the set of non units in $R_{\mathfrak{m}}$. It is given by

$$\mathfrak{m}R_{\mathfrak{m}} = \left\{ \frac{h(f_1, \ldots, f_m)}{1 + g(f_1, \ldots, f_m)} \mid h, g \in \mathbb{C}[y_1, \ldots, y_m] \text{ are both constant-free} \right\}.$$

We say that an element $g \in \mathbb{C}(x_1, \ldots, x_n)$ is integral over $\mathfrak{m}R_{\mathfrak{m}} \iff$ there is a monic polynomial $h \in R_{\mathfrak{m}}[T]$ with coefficients in the ideal $\mathfrak{m}R_{\mathfrak{m}}$ such that $h(g) = 0$. We are now ready to state our algebro-geometric correspondence in the approximative setting. Thus, Theorem 1.11 can be restated as follows.

▶ **Theorem 2.5.** *Let $f_1, \ldots, f_m, g \in \mathbb{C}[x_1, \ldots, x_n]$. Then $\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, \ldots, f_n)$ if and only if g is integral over $\mathfrak{m}R_{\mathfrak{m}}$.*

▶ Remark 2.6. A natural question, is, can we get even simpler characterizations that the one in Theorem 2.5? However, we show two natural candidates do not suffice to fully characterize AV containment.

- *g can't just be characterized by an integral closure without localization:* We show that this is not possible. Consider, $f_1 = xy, f_2 = wz, f_3 = yz - 1, g = xz$. Then $\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1, f_2, f_3)$ but $g(1 + f_3) = f_1 f_2$.
- *g can't just be characterized by a radical ideal in the local ring:* A "radical-type" characterization would be of the form $g^r = \frac{F(f_1, \ldots, f_m)}{1 + G(f_1, \ldots, f_m)}$. However, we show that this is also not always possible. Consider $f_1 = x^2 - xy, f_2 = y, g = x$, then $g^2 = f_1 + g \cdot f_2$. In fact, $\mathfrak{m}$ is maximal so its radical is itself. Therefore, we really need to use integral closure, just using the radical will not be enough.

We can now see that Theorem 2.5 gives us a very nice characterization analogous to the strong Hilbert's Nullstellensatz (SHN). Recall that SHN says that given polynomials $f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n]$, a polynomial g vanishes on vanishing set of $f_1, \ldots, f_m$ is equivalent to saying g lies in the radical ideal of $f_1, \ldots, f_m$. Similarly, Theorem 2.5 says that g approximatively vanishes on the common approximative vanishing set of $f_1, \ldots, f_m$ if and only if g is integral over a maximal ideal of a natural fractional subring of $\mathbb{C}(x_1, \ldots, x_n)$ determined by $f_1, \ldots, f_m$.

▶ Remark 2.7. Conventionally, an element in the integral closure of an ideal implies the element is in the radical of the ideal. Suppose we have some ring R and its ideal I. Let

$$g^r = \sum_{i=0}^{r-1} a_i g^i$$

where $a_i \in I$. Then, it simply implies $g^r \bmod I = 0$ and g lies in the radical of ideal I. This is because g, a_i are coming from the same underlying structure R. However, in our case while the maximal ideal $\mathfrak{m}R_{\mathfrak{m}}$ originates from the ring $R = \mathbb{C}[f_1, \ldots, f_m]$, the polynomial g is instead coming from the larger polynomial ring $\mathbb{C}[x_1, \ldots, x_n]$. Hence, $a_i g^i$ is not necessarily a part of ideal I and hence it might be the case that $a_i g^i \bmod I \neq 0$. Thus, g might not lie in the radical.

Integral closure over $\mathfrak{m}R_{\mathfrak{m}}$ forms an $R_{\mathfrak{m}}$-module. Define $\overline{\mathfrak{m}R_{\mathfrak{m}}}$ to be the integral closure of $\mathfrak{m}R_{\mathfrak{m}}$ in $\mathbb{C}(x_1,\ldots,x_n)$. Let $f,g \in \overline{\mathfrak{m}R_{\mathfrak{m}}}$, then $\mathsf{AV}(f+g) \supseteq \mathsf{AV}(f)\cap\mathsf{AV}(g) \supseteq \mathsf{AV}(f_1,\ldots,f_m)$. Hence $f+g \in \overline{\mathfrak{m}R_{\mathfrak{m}}}$. Now for $r \in R_{\mathfrak{m}}$, $\mathsf{AV}(rf) \supseteq \mathsf{AV}(f) \cup \mathsf{AV}(r) \supseteq \mathsf{AV}(f_1,\ldots,f_m)$. Hence $rf \in \overline{\mathfrak{m}R_{\mathfrak{m}}}$ and $\overline{\mathfrak{m}R_{\mathfrak{m}}}$ is an $R_{\mathfrak{m}}$-module. Clearly, the natural inclusion $R \hookrightarrow R_{\mathfrak{m}}$ makes $\overline{\mathfrak{m}R_{\mathfrak{m}}} \cap \mathbb{C}[x_1,\ldots,x_n]$ an R-module.

Integral closure over $\mathfrak{m}R_{\mathfrak{m}}$ is finitely generated. The next proposition proves that $\overline{\mathfrak{m}R_{\mathfrak{m}}}$ is a finitely generated as an $R_{\mathfrak{m}}$-module. In order to do so, we show that the set of elements of $\mathbb{C}(x_1,\ldots,x_n)$ algebraic over $C(f_1,\ldots,f_m)$ is a finite extension of $C(f_1,\ldots,f_m)$. Then we use that $R_{\mathfrak{m}}$ is an integral Noetherian Nagata domain, i.e., the integral closure of $R_{\mathfrak{m}}$ in a finite extension of its field of fractions is finitely-generated $R_{\mathfrak{m}}$-module (for details see [29, Definition 31(A)]).

▶ **Proposition 2.8.** *Let $R := \mathbb{C}[f_1,\ldots,f_m] \subset \mathbb{C}[x_1,\ldots,x_n]$ and let $\mathfrak{m} = (f_1,\ldots,f_m) \subset R$ be a maximal ideal. Let $\overline{R_{\mathfrak{m}}}$ be the integral closure of $R_{\mathfrak{m}}$ inside $\mathbb{C}(x_1,\ldots,x_n)$. Then $\overline{R_{\mathfrak{m}}}$ is a finitely generated $R_{\mathfrak{m}}$-module.*

Furthermore, the $R_{\mathfrak{m}}$-module

$$\overline{\mathfrak{m}R_{\mathfrak{m}}} \coloneqq \{f \in \overline{R_{\mathfrak{m}}} \mid \exists\ r \geq 1 \text{ and } a_0,\ldots,a_{r-1} \in \mathfrak{m}R_{\mathfrak{m}} \text{ s.t. } f^r = \sum_{i=0}^{r-1} a_i f^i\}$$

is a finitely generated $R_{\mathfrak{m}}$-module.

Proof. Let $K := \mathrm{Frac}(R) = \mathrm{Frac}(R_{\mathfrak{m}}) = \mathbb{C}(f_1,\ldots,f_m) \subset L := \mathbb{C}(x_1,\ldots,x_n)$. Let

$$E := \{\ell \in L :\ \ell \text{ is algebraic over } K\}$$

be the algebraic closure of K in L. Let $r := \mathrm{trdeg}_{\mathbb{C}} K$ (equivalently, $r = \mathrm{trdeg}_{\mathbb{C}}\mathbb{C}(f_1,\ldots,f_m)$). Since $\mathrm{trdeg}_{\mathbb{C}} L = n$, we have $\mathrm{trdeg}_K L = n-r$. Choose a transcendence basis $g_1,\ldots,g_{n-r} \in L$ of L over K. Then L is algebraic over $K(g_1,\ldots,g_{n-r})$. Since the field $L = \mathbb{C}(x_1,\ldots,x_n)$ is a finitely generated algebra over $K(g_1,\ldots,g_{n-r})$, by [3, Proposition 7.9] it is a finite algebraic extension. However, $K(g_1,\ldots,g_{n-r}) \cap E = K$ and therefore, $[E:K] = [EK(g_1,\ldots,g_{n-r}) : K(g_1,\ldots,g_{n-r})] \leq [L : K(g_1,\ldots,g_{n-r})] < \infty$.

If $t \in L$ is integral over $R_{\mathfrak{m}}$, then t is algebraic over K, i.e., $t \in E$. Therefore, the elements of L integral over R are exactly the elements of E integral over R.

Since R is a finitely generated $\mathbb{C}$-algebra, it is a Nagata ring, and localizing preserves the Nagata property [29, Definition 31(A)]; hence $R_{\mathfrak{m}}$ is Noetherian integral domain that is also Nagata. By the Nagata property, $\overline{R_{\mathfrak{m}}}$ is a finite (hence finitely generated) $R_{\mathfrak{m}}$-module.

Now for the final part, we note that $\overline{\mathfrak{m}R_{\mathfrak{m}}}$ is a submodule of a finitely generated module $\overline{R_{\mathfrak{m}}}$ over the Noetherian ring $R_{\mathfrak{m}}$, and we are done. ◀

3 Proof of SAN ∈ PSPACE

In this section, we prove our other main result where we show that the computational problem SAN can be decided in PSPACE. We do this by essentially proving *effective* versions of Lemma 2.2 and Lemma 2.3 by providing a singly exponential bound on the exponent and the degree of polynomials that appear in these characterizations. Degree bounds for annihilating polynomials go back to Perron [34] and have been extended further in [4, 22]. We use these bounds together with the "random reductions trick" of GSS to obtain our algorithm. We now present these two key results for completeness.

▶ **Theorem 3.1** (Annihilator bound, Theorem 11, [22]). *Let $h_1, \ldots, h_\ell \in \mathbb{C}[x_1, \ldots, x_n]$ be of degree at most d, and suppose the algebraic rank (transcendence degree) of $\{h_1, \ldots, h_\ell\}$ is $r < \ell$. Then there exists a nonzero annihilating polynomial for $(h_1, \ldots, h_\ell)$ of degree at most $(r+1)d^r$, and there are examples where the minimal annihilator has degree at least d^r.*

▶ **Theorem 3.2** (Random reductions, Theorem 4.6, [18]). *Let $f_1, \ldots, f_m \in \mathbb{C}[x_1, \ldots, x_n]$ with transcendence degree k. Let S be a finite subset of $\mathbb{C}$. Let $c_{i,j} \in S$ be randomly chosen for $i \in [k+1], j \in [m]$ such that $g_i = \sum_j c_{i,j} f_j$. Then, with high probability, we have*

1. *Transcendence degree of $g_1, \ldots, g_{k+1}$ over $\mathbb{C}$ is k.*
2. *The constant term of every annihilator for $g_1, \ldots, g_{k+1}$ is zero* if and only if *the constant term of every annihilator of $f_1, \ldots, f_m$ is zero*

▶ **Theorem.** SAN *is in* PSPACE.

As discussed earlier in Section 1.3, apriori it is not clear how to use the geometric criterion obtained in Lemma 2.1, or the integral closure formulation obtained in Theorem 1.11 to obtain a PSPACE upper bound. However, leveraging the additional structure of our formulation along with Theorem 3.1, 3.2 we can match the PSPACE upper bound of WAN. We cannot directly use the annihilator obtained from the algebraic characterization in Theorem 1.11 (shown in Equation 5), as we have no guarantees about the degree bound there, since Theorem 3.1 only establishes the singly exponential degree bound for the minimal annihilator of the system of polynomials, not *all* annihilators. However, we can now use Theorem 3.2 to ensure that it is sufficient to look at the minimal annihilator. Given any system of polynomials, we first calculate its transcendence degree ρ in PSPACE using linear algebra [34, 35, 5]. Then, we reduce to $\rho + 1$ polynomials which are random linear combinations which preserve the non-emptiness of AV. Since the annihilator ideal of ℓ polynomials $h_1, \ldots, h_\ell$ is principal if the transcendence degree of the system is $\ell - 1$ [18, 22], we can now bound the degree of this generator by Theorem 3.1.

Proof. We achieve the proof by giving a singly-exponential degree upper bound in the equivalent criterion of SAN restated below (see Equation 5).

$$g^r = \sum_{i=0}^{r} g^i F_i(f_1, \ldots, f_m),$$

We got this by combining the criteria obtained in Lemma 2.2 and Lemma 2.3, and we show that both of these can be reformulated to have singly exponential degree bounds.

First, we re-inspect the proof of Lemma 2.2. We started with the condition: $f_1, \ldots, f_m, y, 1 - yg$ have no common APS, and using the above discussion we can reduce to a system of polynomials $g_1, \ldots, g_{\rho+1}$ where $g_j = c_{j,1} f_1 + \cdots + c_{j,m} f_m + c_{j,m+1} y + c_{j,m+2}(1 - yg)$, and $A(g_1, \ldots, g_{\rho+1}) = 1$ such that $A(z_1, \ldots, z_{\rho+1})$ is constant-free. Let d' be the maximum degree of the polynomials $f_1, \ldots, f_m, g$, then clearly degree of each g_j is bounded above by $d = d' + 1$. Hence, A has degree bounded by $D = (\rho + 1)d^\rho$ by Theorem 3.1, where ρ is the transcendence degree of $f_1, \ldots, y, 1 - yg$. Similar to proof of Lemma 2.2, since $A(g_1, \ldots, g_{\rho+1}) = 1$ holds in $\mathbb{C}[x_1, .., x_n, y]$, it holds in $\mathbb{C}(x_1, \ldots, x_n)[y]$. Hence, we can set $y = \frac{1}{g}$. After this substitution, each

$$g_j = \frac{1}{g}\Big(g(\sum_{i=1}^{m} c_{j,i} f_i) + b_j\Big) = \frac{g q_j(f_1, \ldots, f_m) + b_j}{g}$$

where $b_j \in \mathbb{C}, q_j \in \mathbb{C}[y_1, \ldots, y_m]$ such that q_j is constant-free. Abusing notation, we will now call such polynomials q_j constant-free in $\mathbb{C}[f_1, \ldots, f_m]$. Any monomial of $A(z_1, \ldots, z_{\rho+1})$ is of the form $\alpha \prod_{i=1}^{\rho+1} z_i^{a_i}$ where $t = \sum_i a_i \leq D$ and $\alpha \in \mathbb{C}$. On substitution, this monomial becomes $\alpha \frac{1}{g^t}(gq_j + b_j)^{a_j} = \alpha \frac{1}{g^t}(g^t v_t + g^{t-1} v_{t-1} + \ldots + v_0)$ where, v_j are constant-free elements in $\mathbb{C}[f_1, \ldots, f_m]$ for all $1 \leq j \leq t$, and $v_0 = b_1^{a_1} \cdots b_{\rho+1}^{a_{\rho+1}} \in \mathbb{C}$. Further, degree of each v_j is bounded by D in terms of f_i. In particular, $A(g_1, \ldots, g_{k+1}) = 1$ under this substitution gives $\sum_{t=1}^{D} \frac{1}{g^t}(g^t p_{t,t} + \cdots + p_{t,0}) = 1$ where $p_{t,1}, ..., p_{t,t}$ are constant-free polynomials in $\mathbb{C}[f_1, ..., f_m]$ and $p_{t,0} \in \mathbb{C}$ for all $1 \leq t \leq D$. Further, degree of each p_i is bounded by D. Thus, this becomes

$$g^D = (g^D p_{D,D} + \cdots + p_{D,0}) + g(g^{D-1} p_{D-1,D-1} + \cdots + p_{D-1,0}) + \cdots + g^{D-1}(g p_{1,1} + p_{1,0})$$

In particular, the coefficient of g^D is $p_{D,D} + p_{D-1,D-1} + \ldots + p_{1,1}$. Since each of these $p_{a,a}$ is constant-free, there will be no term of cg^D for some $c \in \mathbb{C}$ in the right hand side of the equation. In particular, we get an annihilator $g^D = g^D Q_D + g^{D-1} Q_{D-1} \cdots + Q_0$ where Q_D is constant-free and degree of each Q_i is bounded by D. This is exactly like the formulation we obtained in Lemma 2.2, with the *additional constraint* that D and degree of each Q_i is singly exponential in the input parameters.

Now, we re-inspect the proof of Lemma 2.3 using this annihilator as a starting point. The corresponding $h(z)$ we get will be a degree $D+1$ polynomial in z and hence has at most D non-zero roots. For each $F_s(f_1, \ldots, f_m, g - s) = 1$ in the proof, we now instead use $F_s(g'_1, \ldots, g'_{\rho+1}) = 1$ where the g'_j are now random linear combinations of $f_1, \ldots, f_m, g - s$ and ρ is their transcendence degree. The degree of F_s will be bounded by D. In particular, each g'_j can be written as $g'_j = p_j + b'_j(g - s)$ where each $p_j \in \mathbb{C}[f_1, \ldots, f_m]$ is linear in f_i and is constant-free, and $b'_j \in \mathbb{C}$. Any monomial of F_s would be of the form $\gamma \prod_{j=1}^{\rho+1}(p_j + b'_j(g - s))^{a'_j}$ where $\sum a'_j = t \leq D$ and $\gamma \in \mathbb{C}$. Thus, any monomial is of the type $(g-s)^t a'_t + (g-s)^{t-1} q'_{t-1} + \ldots + q'_0$ where each q'_i is constant-free in $\mathbb{C}[f_1, \ldots, f_m]$ with degree bounded by $D - i$, and $a'_t \in \mathbb{C}$. Using this we can rewrite $F_s(g'_1, \ldots, g'_{\rho+1})$ as $Q(f_1, \ldots, f_m) + (g - s)P(f_1, \ldots, f_m, g)$ where degree of Q in f_i is bounded by D and degree of P in f_i, g is bounded by D. Also, Q is constant-free in $\mathbb{C}[f_1, \ldots, f_m]$. Using this annihilator in the proof of Lemma 2.3, the next step is to multiply these for all the roots of $h(z)$, which are at most D. Further, we then multiply with g^k where k is also bounded by D (since k was the multiplicity of 0 as a root of $h(z)$). Thus, finally we get $H(f_1, \ldots, f_m, g)h(g) = g^k + g^k G(f_1, \ldots, f_m)$ where degree of H, G is bounded above by D^2 and G is constant-free in $\mathbb{C}[f_1, \ldots, f_m]$. We can again replace $h(g)$ like in the original proof and get

$$g^k = \sum_{i=0}^{l} g^i F_i(f_1, .., f_m)$$

where l and degree of F_i is bounded by $D' = D^2 + D$, and each F_i is constant-free.

We now show that the obtained bounds are sufficient to put SAN in PSPACE. First thing to note is that $D = (\rho+1)d^\rho$. Since ρ was the transcendence degree of the a system of at most $m+2$ polynomials in $n+1$ variables, $\rho \leq n+1$. Hence, k, D, D' is atmost singly exponential in the input parameters. We can thus set up an exponential-sized system of linear equations in the same way as for computing annihilator using Perron's bound [18, 35, 34], which can be solved in PSPACE [5, 32, 11, 6] ◀

4 Conclusion

In this work, we provide an analog of the SHN for approximative roots of a system of polynomials. Further, we provide a PSPACE upper bound for testing the same, matching the best known upper bound for the Weak Approximative Nullstellensatz. We believe that this work opens up several interesting areas of study in the geometry of approximative roots. Most prominently, defining the notion of dimension to estimate the size of a set of approximative roots is particularly interesting. Similarly, examining how intersection theory of curves translates to this setting is also very intriguing. From a computational perspective, improving the complexity bounds for WAN and SAN analgous to recent improvements in the classical setting [1], and algorithmically obtaining the generators of the integral closure obtained in Theorem 1.11 is also interesting.

References

1 Robert Andrews, Abhibhav Garg, and Éric Schost. Hilbert's nullstellensatz is in the counting hierarchy, 2026. doi:10.48550/arXiv.2602.17904.

2 Matthias Aschenbrenner and Anton Leykin. Degree bounds for gröbner bases in algebras of solvable type. *Journal of Pure and Applied Algebra*, 213(8):1572–1588, 2009. doi:10.1016/j.jpaa.2008.11.022.

3 Michael Francis Atiyah and I. G. MacDonald. *Introduction to commutative algebra*. Addison-Wesley-Longman, Reading, MA, 1969.

4 Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Inf. Comput.*, 222:2–19, 2013. doi:10.1016/j.ic.2012.10.004.

5 Stuart J Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information processing letters*, 18(3):147–150, 1984. doi:10.1016/0020-0190(84)90018-8.

6 Allan Borodin, Joachim von zur Gathem, and John Hopcroft. Fast parallel matrix and gcd computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 65–71. IEEE, 1982. doi:10.1109/SFCS.1982.17.

7 Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1997.

8 John F. Canny. Some algebraic and geometric computations in PSPACE. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 460–467. ACM, 1988. doi:10.1145/62212.62257.

9 A. L. Chistov and D. Yu. Grigor'ev. Complexity of quantifier elimination in the theory of algebraically closed fields. In *Mathematical Foundations of Computer Science 1984 (MFCS'84)*, volume 176 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 1984. doi:10.1007/BFb0030287.

10 David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.)*. Undergraduate texts in mathematics. Springer, 1997.

11 L. Csanky. Fast parallel matrix inversion algorithms. In *16th Annual Symposium on Foundations of Computer Science, Berkeley, California, USA, October 13-15, 1975*, pages 11–12. IEEE, IEEE Computer Society, 1975. doi:10.1109/SFCS.1975.14.

12 Theo de Jong. An algorithm for computing the integral closure. *Journal of Symbolic Computation*, 26(3):273–277, 1998. doi:10.1006/jsco.1998.0211.

13 Thomas W. Dubé. The structure of polynomial ideals and gröbner bases. *SIAM Journal on Computing*, 19(4):750–773, 1990. doi:10.1137/0219053.

14 Antonio J Engler and Alexander Prestel. *Valued fields*. Springer, 2005.

15 Michael A Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 527–542. Springer, 2013. doi:10.1007/978-3-642-40328-6_37.

16 Michael A. Forbes and Amir Shpilka. A PSPACE construction of a hitting set for the closure of small algebraic circuits. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018)*, pages 1180–1192, 2018. doi:10.1145/3188745.3188792.

17 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in proof complexity and tfnp. *Journal of the ACM*, 71(4):1–45, 2024. doi:10.1145/3663758.

18 Zeyu Guo, Nitin Saxena, and Amit Sinhababu. Algebraic dependencies and pspace algorithms in approximative complexity over any field. *Theory of Computing*, 15(1):1–30, 2019. doi:10.4086/toc.2019.v015a016.

19 Deirdre Haskell and Yoav Yaffe. Ganzstellensätze in theories of valued fields. *Journal of Mathematical Logic*, 8(01):1–22, 2008. doi:10.1142/S0219061308000695.

20 Craig Huneke and Irena Swanson. *Integral Closure of Ideals, Rings, and Modules*, volume 336 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 2006.

21 Doug Ierardi. Quantifier elimination in the theory of an algebraically-closed field. Technical Report TR89-1030, Cornell University, 1989. doi:10.1145/73007.73020.

22 Neeraj Kayal. The complexity of the annihilating polynomial. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 184–193. IEEE, 2009. doi:10.1109/CCC.2009.37.

23 Elisabeth Leonie Kayser. Computational complexity of polynomial subalgebras. In Carlos D'Andrea, Sonia Pérez-Díaz, and Santiago Laplagne, editors, *Proceedings of the 2025 International Symposium on Symbolic and Algebraic Computation, ISSAC 2025, Guanajuato, Mexico, 28 July 2025 - 1 August 2025*, ISSAC '25, pages 337–344, New York, NY, USA, 2025. ACM. doi:10.1145/3747199.3747578.

24 Simon Kochen. Integer valued rational functions over the p-adic numbers: A p-adic analogue of the theory of real fields. In *Proc. Symp. Pure Math*, volume 12, pages 57–73, 1969.

25 Pascal Koiran. Hilbert's nullstellensatz is in the polynomial hierarchy. *Journal of complexity*, 12(4):273–286, 1996. doi:10.1006/jcom.1996.0019.

26 János Kollár. Sharp effective nullstellensatz. *Journal of the American Mathematical Society*, pages 963–975, 1988.

27 J. M. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

28 Noa Lavi. A ganzstellensatz for semialgebraic sets and a boundedness criterion for rational functions. *Communications in Algebra*, 44(1):26–39, 2016. doi:10.1080/00927872.2014.930473.

29 Hideyuki Matsumura. *Commutative Algebra*. 2022. Revised and modernized edition; typeset by the TEXromancers. Original edition: 1980. URL: https://aareyanmanzoor.github.io/assets/matsumura-CA.pdf.

30 Ernst W. Mayr and Albert R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. In *Advances in Mathematics*, 1982.

31 James S Milne. *Algebraic geometry*. Allied Publishers, 2012.

32 Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 338–339, 1986. doi:10.1145/12130.12164.

33 Ketan Mulmuley. Geometric complexity theory V: equivalence between blackbox derandomization of polynomial identity testing and derandomization of noether's normalization lemma. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 629–638. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.15.

34 Oskar Perron. Algebraische abhängigkeit und ihre anwendung in der algebra. *Mathematische Annalen*, 1927.

35 Arkadiusz Płoski. Algebraic dependence of polynomials after o. perron and some applications. *Computational Commutative and Non-Commutative Algebraic Geometry*, pages 167–173, 2005.

36 Alexander Prestel and Cydara C Ripoll. Integral-valued rational functions on valued fields. *manuscripta mathematica*, 73(1):437–452, 1991.

37 JL Rabinowitsch. Zum hilbertschen nullstellensatz. *Mathematische Annalen*, 102(1):520–520, 1930.

38 Wolmer V. Vasconcelos. *Integral Closure: Rees Algebras, Multiplicities, Algorithms*. Springer Monographs in Mathematics. Springer, 2005.

On Factorization of Sparse Polynomials of Bounded Individual Degree

Aminadav Chuyoon
School of Mathematics, Tel Aviv University, Israel

Amir Shpilka
Blavatnik School of Computer Science and AI, Tel Aviv University, Israel

Abstract

We study sparse polynomials with bounded individual degree and the class of their factors. In particular we obtain the following algorithmic and structural results:

1. A deterministic polynomial-time algorithm for finding *all the sparse divisors* of a sparse polynomial with bounded individual degree. As part of this, we establish the first upper bound on the number of non-monomial irreducible factors of such polynomials.
2. A $\mathrm{poly}(n, s^{d \log \ell})$-time algorithm for recovering ℓ irreducible s-sparse polynomials of bounded individual degree d from blackbox access to their product (which is not necessarily sparse). This partially resolves a question posed in [10]. In particular, when $\ell = O(1)$, the algorithm runs in polynomial time.
3. Deterministic algorithms for factoring a *product* of s-sparse polynomials of bounded individual degree d from blackbox access. Over fields of characteristic zero or sufficiently large, the algorithm runs in $\mathrm{poly}(n, s^{d^3 \log n})$-time; over arbitrary fields it runs in $\mathrm{poly}(n, (d^2)!, s^{d^5 \log n})$-time. This improves upon the algorithm of [2], which runs in $\mathrm{poly}(n, s^{d^7 \log n})$-time and applies only to a single sparse polynomial of bounded individual degree. In the case where the input is a single sparse polynomial, we give an algorithm that runs in $\mathrm{poly}(n, s^{d^2 \log n})$-time.
4. Given blackbox access to a *product* of (not necessarily sparse or irreducible) *factors of sparse polynomials of bounded individual degree*, we give a deterministic polynomial-time algorithm for finding all irreducible sparse multiquadratic factors of it (along with their multiplicities). This generalizes the algorithms of [26] and [27]. We also show how to decide whether such a product is a complete power (in case it is defined over a field of zero or large enough characteristic), extending the algorithm of [5].

Our algorithms most naturally apply over fields of zero or sufficiently large characteristic. To handle arbitrary fields, we introduce the notion of *primitive divisors* for a class of polynomials, which may be of independent interest. This notion enables us to adapt ideas of [5] and remove characteristic assumptions from most of our algorithms.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases algebraic complexity theory, sparse polynomials, factorization, reconstruction

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.20

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2026/036/`

Funding This research was funded by the European Union (ERC, EACTP, 101142020). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

© Aminadav Chuyoon and Amir Shpilka;
licensed under Creative Commons License CC-BY 4.0

1 Introduction

1.1 Background and Motivation

This paper is motivated by numerous open problems concerning the factorization and complexity of factors of multivariate sparse polynomials. These are n-variate polynomials of degree $\text{poly}(n)$ whose number of monomials is also $\text{poly}(n)$. In particular, this is the class of polynomials that have polynomial size $\Sigma\Pi$ circuits, which is one of the simplest classes from the algebraic complexity point of view.

Despite being structurally simple, sparse polynomials form a rich and non-trivial class of polynomials that has been extensively studied. Efficient deterministic algorithms for polynomial identity testing and reconstruction were developed for this class [1, 13, 7, 18]. In particular, the work of Klivans and Spielman [18] introduced a special reduction from multivariate sparse polynomials to univariate polynomials of polynomial degree, over any field, that has since found many applications, mainly for questions concerning polynomial identity testing and reconstruction. See the surveys [21, 24, 22, 9].

In this work we are mostly interested in understanding factors of sparse polynomials. The seminal work of Kaltofen [16] shows that factors of sparse polynomials have polynomial-size algebraic circuits. Over fields of characteristic zero, or sufficiently large characteristic, the recent breakthrough of [3] implies that factors of sparse polynomials have polynomial-size bounded-depth circuits.

However, an intriguing question that remains unsolved for over forty years concerns the complexity of factors of sparse polynomials in terms of their number of monomials. In other words, how many monomials can a factor of a sparse polynomial have. The following example of von zur Gathen and Kaltofen demonstrates that the sparsity of irreducible factors of polynomials may be super-polynomial.

▶ **Example 1** (Example of [28]). Consider

$$g(\mathbf{x}) = \prod_{i=1}^{n}(x_i - 1) \quad \text{and} \quad h(\mathbf{x}) = \prod_{i=1}^{n}(1 + x_i + \cdots + x_i^{n-1}) + n.$$

It is not hard to prove that h is irreducible over $\mathbb{Q}$. Let

$$f(\mathbf{x}) = g \cdot h = \prod_{i=1}^{n}(x_i^n - 1) + n\prod_{i=1}^{n}(x_i - 1).$$

Then, f has $(2^{n+1} - 1)$ monomials, and an irreducible factor h of sparsity n^n, which is super-polynomial in the sparsity of f.

This led them to pose the following question.

▶ **Question 2** ([28]). *Can the output size for the factoring problem actually be more than quasi-polynomial in the sparsity of the input?*

This question is still wide open with only several special cases solved. In [26] Volkovich proved that multilinear factors of sparse polynomials are sparse. In a follow-up work [27] he proved that all factors of multiquadratic sparse polynomials are themselves sparse. This result was extended to higher individual degrees by Bhargava, Saraf, and Volkovich [2], who considered sparse polynomials with bounded individual degree d, and proved a quasi-polynomial upper bound on the sparsity of factors of such polynomials. In particular, if f is an s-sparse n-variate polynomial with individual degrees bounded by d, then every

factor of f has at most $s^{O(d^2 \log n)}$ monomials. This is the first nontrivial sparsity bound for factors in the regime $d > 2$. While the result of [2] gives a quasi-polynomial upper bound on sparsity of factors of polynomials with bounded individual degree, it is believed that the "real" bound should be polynomial. In [5], the authors provide some evidence for that by giving polynomial-time algorithms for various problems for which the existence of such algorithms is implied by the conjectured polynomial upper bound on sparsity of factors.

In their work von zur Gathen and Kaltofen gave a randomized algorithm for factorization of sparse polynomials and noted another obstacle towards obtaining deterministic factorization algorithms:

> "Also note that there does not seem to be a feasible way of checking deterministically whether the output is correct, i.e., whether $f = \prod_i f_i^{e_i}$" [28].

In particular, Dutta, Sinhababu and Thierauf asked the following question.

▶ **Question 3** ([10, Question 2]). *Can we find the factorization of polynomials whose irreducible factors are sparse?*

We note that [3] gave a subexponential time algorithm for Question 3, but obtaining a polynomial-time or even quasi-polynomial-time algorithm is still an open question.

In fact, Dutta et al. also asked the following special case of Question 3.

▶ **Question 4** ([10, Section 6, item 4]). *Given a blackbox computing the product of sparse irreducible polynomials f_i with bounded individual degree, find the f_i's in deterministic polynomial-time.*

The aforementioned results of [26, 27, 2] also have consequences for Question 3 and Question 4. In [26] Volkovich gave a deterministic polynomial-time algorithm for factorization of sparse polynomials whose irreducible factors are all multilinear (hence sparse) polynomials. In [27] he gave an efficient deterministic algorithm for factoring multiquadratic sparse polynomials. In [2] the authors combined their sparsity bound with techniques similar to [26, 27] to obtain a deterministic quasi-polynomial-time algorithm for factoring sparse polynomials of bounded individual degree. Dutta, Sinhababu, and Thierauf [10] gave efficient deterministic reconstruction of constant degree factors of sparse polynomials.

One notable limitation of the algorithm of [2] is that it does not give an efficient way to find the *sparse factors* of the input polynomial, even if such factors exist. This problem is closely related to the following question asked by Dutta, Sinhababu and Thierauf.

▶ **Question 5** ([10, Question 1]). *Design a better-than-exponential time deterministic algorithm that outputs all the sparse irreducible factors of a sparse polynomial.*

The result of Bhattacharjee et al. [3] gives a subexponential time algorithm for the problem, but obtaining a polynomial-time (or quasi-polynomial) time algorithm is still an open problem.

Another outstanding open problem concerns the reconstruction of rational functions whose numerator and denominator are sparse polynomials. Surprisingly, the best known deterministic algorithm for this question runs in exponential time [14]. While seemingly unrelated, special cases of this question lie at the heart of our algorithms, and we solve it in two special cases where more information is provided.

Many more open problems have been asked for sparse polynomials and the reader is referred to [10] and [12].

1.2 Our Results

Before presenting our results, let us introduce the following notation.

▶ **Definition 6.** *We denote the class of n-variate, s-sparse polynomials of bounded individual degree d, by $\mathcal{P}(n,s,d)$. We refer to polynomials in this class as (n,s,d)-sparse polynomials. It turns out that in many cases it makes sense to talk about the class of* divisors *of (n,s,d)-sparse polynomials; we denote this class of polynomials by $\mathcal{P}_{Factors}(n,s,d)$.*

We first summarize our main contributions informally as follows:

1. We prove a sharp upper bound on the number of irreducible factors of (n,s,d)-sparse polynomials (Theorem 8 and Remark 18).
2. Using this bound, we give the first deterministic polynomial-time algorithm that constructs *all* sparse divisors of (n,s,d)-sparse polynomials (Theorem 9).
3. We give a deterministic blackbox factorization algorithm for products of (n,s,d)-sparse polynomials in quasi-polynomial-time in the number of terms. This solves Question 4 when the number of terms is constant (Theorem 10). We note that the running time of this algorithm is polynomial in the number of variables n.

▶ Remark 7. In stating our results we suppress the precise dependence on the field (beyond its characteristic) and on bit complexity. As usual in factorization algorithms, an additional factor reflecting the complexity of univariate factorization over the field should be included.

We now give the formal statement of our results. Our first main result gives an upper bound for the number of sparse divisors of (n,s,d)-sparse polynomials. While its proof is not difficult, it is instrumental in our algorithms and demonstrates the strong structural limitations that sparsity imposes.

▶ **Theorem 8** (Divisor bound for (n,s,d)-sparse polynomials)**.** *Let $f \in \mathcal{P}(n,s,d)$. The number of non-monomial irreducible factors of f, counted with multiplicities, is at most $d \log s$. In particular, the number of divisors of an s-sparse polynomial f with bounded individual degree d, that are not divisible by any monomial, is at most $2^{d \log s} = s^d$.*

Our first main algorithmic result shows how to deterministically output all sparse divisors of such polynomials in polynomial-time. Earlier, only a quasi-polynomial-time algorithm was known [2].

▶ **Theorem 9** (Finding all sparse divisors of (n,s,d)-sparse polynomials)**.** *There is a deterministic, $\mathrm{poly}(n, d!, s^d)$-time algorithm which, given $f \in \mathcal{P}(n,s,d)$, outputs all the s-sparse divisors of f that are not divisible by any monomial, as well as the monomial of maximal degree dividing f.*

Our second main algorithmic result answers Question 4 when there are constantly many polynomials in the product, and in the general case it provides a quasi-polynomial-time algorithm.

▶ **Theorem 10** (Factoring products of (n,s,d)-sparse polynomials)**.** *Let $\mathbb{F}$ be a field of characteristic zero or larger than $2d$. There is a deterministic $\mathrm{poly}(n, d^d, s^{d \log \ell}, \ell^d)$-time algorithm which, given blackbox access to a product $f = \prod_{i=1}^{\ell} \phi_i$, where $\phi_1, \ldots, \phi_\ell \in \mathcal{P}(n,s,d)$ are all (not necessarily different) irreducible, returns the ϕ_i-s. Moreover, when the assumption on the field is removed, there is an algorithm that solves this problem in $\mathrm{poly}(n, (d^2)!, s^{d \log \ell + d^3}, \ell^d)$-time.*

This theorem will follow as a consequence of the next, more general, result.

▶ **Theorem 11** (Factoring products of factors of sparse polynomials)**.** *Let $\mathbb{F}$ be a field with* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\mathrm{char}(\mathbb{F}) > 2d$*. There is a deterministic* $\mathrm{poly}(n, d^d, s^{d\log \ell}, \ell^d)$*-time algorithm which, given blackbox access to a product f of ℓ polynomials in $\mathcal{P}_{Factors}(n, s, d)$ over the field $\mathbb{F}$, returns all the s-sparse divisors of bounded individual degree d of f that are not divisible by any monomial, as well as the multiplicity k_i of each x_i as a factor of f. Moreover, if f is in fact a product of polynomials in $\mathcal{P}(n, s, d)$, there is such an algorithm that works over arbitrary fields, whose running time is* $\mathrm{poly}(n, (d^2)!, s^{d\log \ell + d^3}, \ell^d)$*.*

Observe that this result is slightly weaker for fields of small characteristic. Indeed, as will be discussed in Section 1.4.1.2 we will see that some of our methods only work for zero characteristic fields, or for fields with large enough characteristic.

In addition to our main results we provide several improvements and generalizations of earlier work. We start by stating results concerning factorization of sparse polynomials of bounded individual degree into irreducible factors. The first improves the best known factorization algorithm of such polynomials [2].

▶ **Theorem 12.** *There is a deterministic* $\mathrm{poly}(n, s^{d^2 \log n})$*-time algorithm which, given an s-sparse polynomial of bounded individual degree d in n-variables, returns its factorization into irreducible factors.*

The second theorem generalizes the result of [2] to a significantly larger class of polynomials, while achieving improved running time compared to their algorithm.

▶ **Theorem 13.** *Let $\mathbb{F}$ be a field with* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\mathrm{char}(\mathbb{F}) > 2d$*. There is a deterministic* $\mathrm{poly}(n, s^{d^3 \log n}, (d\ell)^d)$*-time algorithm which, given blackbox access to a product f of ℓ polynomials in $\mathcal{P}_{Factors}(n, s, d)$ over $\mathbb{F}$, returns its factorization into irreducible factors. Moreover, there is an algorithm that works over arbitrary fields, whose running time is* $\mathrm{poly}(n, (d^2)!, s^{d^5 \log n}, \ell^d)$*.*

Note that this last result considers the class of *factors of sparse polynomials of bounded individual degree*. Indeed, we will see that it is more natural to consider this class in many cases, and several results of ours actually hold for this class rather than just for the class of sparse polynomials.

Finally, we generalize results of [26, 27, 5] in a similar manner to our generalization of the results of [2].

Our next algorithm finds all the multiquadratic irreducible factors of a product of polynomials from $\mathcal{P}_{\mathrm{Factors}}(n, s, d)$. Earlier, only factorization of polynomials whose irreducible factors are all multiquadratic and s-sparse was known [27, Theorem 47].

▶ **Theorem 14.** *Let $\mathbb{F}$ be a field of characteristic zero or larger than $2d$. There is a deterministic* $\mathrm{poly}(n, d!, s^d, \ell)$*-time algorithm that, given blackbox access to a product f of ℓ elements of $\mathcal{P}_{Factors}(n, s, d)$ defined over $\mathbb{F}$, returns all the s-sparse multiquadratic irreducible factors of f. In particular, the algorithm returns all the multilinear factors of f (as these are s-sparse, see for example [15, Lemma 10]). Moreover, there exists such an algorithm that works over arbitrary fields, whose running time is* $\mathrm{poly}(n, (d^2)!, s^{d^2}, \ell)$*.*

We give a similar improvement to the perfect power testing of [5].

▶ **Theorem 15.** *Let $\mathbb{F}$ be a field of characteristic zero or larger than $2d$. There is a deterministic* $\mathrm{poly}(n, d!, s^d, \ell)$*-time algorithm that, given blackbox access to a polynomial f that is a product of ℓ polynomials in $\mathcal{P}_{Factors}(n, s, d)$ (defined over $\mathbb{F}$) and a positive integer e, decides whether f is a complete e-th power.*

Finally, we extend the polynomial identity testing result of [5].

▶ **Theorem 16.** *Let $\mathbb{F}$ be a field of characteristic zero or larger than $2d$. There exists a deterministic* $\mathrm{poly}(n, d!, s^d, \ell)$*-time algorithm that, given blackbox access to two polynomials (over $\mathbb{F}$) f and g that are products of ℓ polynomials in $\mathcal{P}_{Factors}(n, s, d)$, decides if $f|g$. Moreover, if f and g are actually products of ℓ polynomials in $\mathcal{P}(n, s, d)$, there is an algorithm that works over arbitrary fields, whose running time is* $\mathrm{poly}(n, (d^2)!, s^{d^3}, \ell)$

As mentioned in Section 1.1, we will have to solve some rational interpolation problems. Somewhat surprisingly, it is not known how to deterministically reconstruct two coprime s-sparse polynomials f and g from blackbox access to their quotient g/f in polynomial-time. We shall see that with some additional information on f, this is in fact possible. Without this extra information, only exponential deterministic algorithms are known for solving this problem [14]. We note that efficient randomized algorithms for solving this problem are known, see e.g., [8, 25].

1.3 Related Work

The questions considered in this work have been studied extensively in recent years. We organize the prior results around three themes that are central to this paper: sparsity bounds, factorization and reconstruction algorithms, and divisibility and identity testing.

Sparsity of Factors of Sparse Polynomials

Volkovich [26] showed that multilinear factors of s-sparse polynomials are again s-sparse. In [27] he proved that factors of an s-sparse multiquadratic polynomial are s-sparse as well. For general polynomials of bounded individual degree d, Bhargava, Saraf and Volkovich [2], obtained the first non-trivial bound on the sparsity of factors of s-sparse polynomials with bounded individual degree d, namely $s^{O(d^2 \log n)}$. Bisht and Saxena [4] improved this bound to $s^{\mathrm{poly}(d)}$ for the class of *symmetric* s-sparse polynomials of bounded individual degree d. Bisht and Volkovich [5] obtained an $s^{O(d \log s)}$ bound on the sparsity of a quotient of an s-sparse polynomial of bounded individual degree d by a multilinear factor. Finally, the work of Forbes [11] and results in [5] imply that the sparsity of the quotient of $f \in \mathcal{P}(n, s, d)$ by a degree r polynomial is bounded by $(ns)^{O(dr)}$.

Deterministic factorization and reconstruction

Factorization results have largely paralleled the development of sparsity bounds. Volkovich showed in [26] how to efficiently factorize a sparse polynomial whose factors are all multilinear. In [27] he demonstrated how to efficiently compute the irreducible factors of a polynomial from a blackbox access, given that all of them are multiquadratic and sparse. In Theorem 14 we further improve this result by relaxing the condition on the input polynomial – we allow it to be a product of factors of some sparse polynomial of bounded individual degree, and not simply a product of sparse multiquadratic polynomials.

Using their sparsity bound, Bhargava, Saraf, and Volkovich [2] derived a deterministic quasi-polynomial algorithm for factoring sparse polynomials of bounded individual degree. However, their methods do not yield a polynomial-time algorithm for finding the sparse divisors of the input polynomial, so they do not imply Theorem 9. Furthermore, their algorithm does not extend to factorization of products of sparse, bounded degree polynomials, let alone products of factors of such polynomials.

Kumar, Ramanathan, and Saptharishi [19] and Dutta, Sinhababu, and Thierauf [10], gave an algorithm that outputs all the irreducible factors of constant degree of a sparse polynomial in quasi-polynomial-time. As their result applies to general sparse polynomials without restrictions on the individual degree, it does not follow from the methods of this paper.

In [5], an algorithm for deciding if a sparse polynomial of bounded individual degree is a complete power is given; in Section 1.4.1 we describe how to use our stronger methods to obtain this result when the input polynomial is a product of polynomials in $\mathcal{P}_{\text{Factors}}(n, s, d)$, given we have blackbox access to it. However, this generalization does not work when the underlying field is of small characteristic.

Divisibility and identity testing

In [27] it is also shown how to efficiently check if two s-sparse polynomials of bounded individual degree d divide each other; this is improved in Theorem 16, where we show a divisibility testing result for two products of factors of sparse polynomials of bounded individual degree, given blackbox access to them. In the same work, Volkovich also gave a deterministic algorithm for testing a factorization of sparse polynomials, with constant individual degrees, into sparse irreducible factors. That is, testing if $f = \prod g_i$ when f has constant individual degrees and g_i-s are irreducible and sparse. We note that, as a simple corollary of Theorem 9, we are actually able to efficiently find the g_i's and verify that $f = \prod g_i$ in polynomial time.

Forbes [11] gave a quasi-polynomial-time algorithm for checking if a constant degree polynomial divides a sparse polynomial. Note that while our result allows the dividing polynomial to be much more complicated, Forbes' result requires no restriction on the individual degree of the given sparse polynomial. Therefore, our methods fall short in proving his result, or any other result on sparse polynomials with no restriction on their individual degree.

In [5], the authors design an efficient blackbox PIT algorithm for the class $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg d}]}$; while doing so, they obtain some technical results about the class $\mathcal{P}_{\text{Factors}}(n, s, d)$. In Section 1.4.1.1 we show that their methods imply a stronger result than this PIT. Showing PIT for this class is essentially equivalent to giving an algorithm for determining if two products f, g of (n, s, d)-sparse polynomials are equal, given blackbox access to them. We show that the methods of [5] can be used for determining if f divides g. In Section 1.4.1.2 we further generalize and strengthen the technical results of [5], but only for fields of zero or sufficiently large characteristic.

1.4 Proof Overview

We start by describing some technical results about the class of sparse polynomials of bounded individual degree needed for our proofs. We then turn to give an overview of our results regarding factorization of sparse polynomials. We first discuss our result on the recovery of all irreducible sparse multiquadratic factors of a product of (n, s, d)-sparse polynomials, as the proof contains some of the important ideas used in the proofs of our main results. We then move on to describe the recovery of all sparse divisors of sparse polynomials of bounded individual degree and our partial solution to Question 4. Finally, we briefly describe how to use our methods for factoring general (n, s, d)-sparse polynomials and products of them, making use of the bound of [2].

1.4.1 Sparse Polynomials of Bounded Individual Degree and Their Factors

The main tool we use for obtaining our results is a generator $G : \mathbb{F}^6 \to \mathbb{F}^n$ that preserves important properties of polynomials in $\mathcal{P}_{\text{Factors}}(n, s, d)$. Roughly speaking, we want this map G to satisfy the following two properties:

1. Its image $\text{Im}(G)$ is an interpolation set for polynomials in $\mathcal{P}(n, s, d)$.
2. If ϕ and ψ are non-associate[1] irreducible elements of $\mathcal{P}_{\text{Factors}}(n, s, d)$, then $\phi(G_i)$ and $\psi(G_i)$ are coprime as polynomials in x_i, where G_i should be thought of as "G when reviving the i-th variable" (i.e., keeping x_i alive and substituting the remaining variables according to G).

The first property is easily achieved by the basic properties of the Klivans-Spielman generator (KS-generator), introduced in [18]. We show that the second property follows from the first, together with the fact that resultants of sparse polynomials of bounded individual degree are again sparse. However, we are able to do so only when the underlying field $\mathbb{F}$ is of zero characteristic, or if its characteristic is larger than twice the bound on the individual degree. If this is not the case then we are only able to prove something a little weaker – we prove that the second property holds if one replaces the notion of irreducible polynomials with the new notion of *primitive divisors*, which we first define in this paper (we describe this notion below).

Before sketching the proof that the first property of G implies the second, we note that proving that will automatically imply some algorithms for sparse polynomials. For example, it is straightforward to see that by factoring the composition of the input polynomials with G_i, the following two results can be obtained from the (general) second property of G.

1. Given blackbox access to two products of elements of $\mathcal{P}_{\text{Factors}}(n, s, d)$, we can decide if one divides the other in deterministic polynomial-time (Theorem 16).
2. Given blackbox access to a product of elements of $\mathcal{P}_{\text{Factors}}(n, s, d)$, we can decide if it is a complete power in deterministic polynomial-time (Theorem 15).

This already generalizes results from [27] and [5]. As we are able to obtain the general second property of G only for some fields, these results do not apply to arbitrary fields. It turns out, however, that the aforementioned weaker version of the second property of G is strong enough to obtain Theorem 16 over arbitrary fields, in the case we replace $\mathcal{P}_{\text{Factors}}(n, s, d)$ by $\mathcal{P}(n, s, d)$.

In the rest of this subsection we describe our proofs of the second property, both for arbitrary fields and for fields with zero or large enough characteristic.

1.4.1.1 Arbitrary Fields

In [5, Lemmas 4.5,4.6] it is shown that the second property of G holds for each pair of irreducible polynomials ϕ and ψ that satisfy a certain non-degeneracy condition: there exist (n, s, d)-sparse polynomials f, g for which $\phi|f, \psi|g$ and the matrix $\begin{pmatrix} v_\phi(f) & v_\phi(g) \\ v_\psi(f) & v_\psi(g) \end{pmatrix}$ is invertible, where $v_\phi(f)$ is the multiplicity of ϕ as a factor of f, and so are the other entries, respectively. That is, a pair of polynomials $\phi, \psi \in \mathcal{P}_{\text{Factors}}(n, s, d)$ does not satisfy this condition if the multiplicity of ϕ as a factor of an element f of $\mathcal{P}(n, s, d)$ determines the multiplicity of ψ as a factor of f. In this sense, ϕ and ψ are "inseparable" from the point of view of polynomials in $\mathcal{P}(n, s, d)$.

[1] Two polynomials are associate if one is a scalar multiple of the other.

For each irreducible $\phi \in \mathcal{P}_{\text{Factors}}(n,s,d)$, consider all ψs that are inseparable from it, in the sense that ϕ, ψ do not satisfy the aforementioned non-degeneracy condition. For each $f \in \mathcal{P}(n,s,d)$, consider the vector of multiplicities of these ψs as factors of f - $(v_\psi(f))_\psi$. By the degeneracy assumption, these vectors are all proportional. Consider the $\mathbb{Z}$-span of these vectors, and let $(a_\psi)_\psi$ be a generator for this one-dimensional lattice. The polynomial $\prod_\psi \psi^{a_\psi}$ is called a primitive divisor for the class $\mathcal{P}(n,s,d)$. Note that by construction, as the property of being "inseparable" is an equivalence relation on the class of irreducible polynomials in $\mathcal{P}_{\text{Factors}}(n,s,d)$, it must hold that every two primitive divisors are coprime.

The notion of a primitive divisor is a general notion, that applies to a general class of polynomials $\mathcal{P}$, and therefore may be of independent interest. The formal definition of a primitive divisor for a class of polynomials is given in Definition 31.

Primitive divisors for a class of polynomials $\mathcal{P}$ are for the class $\mathcal{P}$ what irreducible polynomials are for the class of all polynomials, in the sense that every element of the class $\mathcal{P}$ can be written as a product of primitive divisors for $\mathcal{P}$ in a unique way. This is not too difficult to see by the construction given in the previous paragraph, and is formulated in the following lemma.

▶ **Lemma 17.** *Let $\mathcal{P}$ be some class of polynomials, and let $f \in \mathcal{P}$. Then, f can be uniquely written as a product of primitive divisors for $\mathcal{P}$.*

Note that Lemma 17 suggests that it makes sense to talk about the multiplicity $v_\phi(f)$ of a primitive divisor ϕ as a factor of $f \in \mathcal{P}$ (see Definition 32).

With the notion of primitive divisors in hand, one can show that every two non-associate primitive divisors ϕ and ψ for a class $\mathcal{P}$ satisfy the non-degeneracy condition of [5]. As a result, we are able to apply the proof of the aforementioned result of [5] to obtain the weaker version of the second property of G, namely, for every two non-associate primitive divisors ϕ, ψ for $\mathcal{P}(n,s,d)$ it holds that $\gcd_{x_i}(\phi(G_i), \psi(G_i)) = 1$.

We further note that for every two products f, g of elements of $\mathcal{P}(n,s,d)$ we have that $f|g$ if and only if $v_\phi(f) \leq v_\phi(g)$ for every primitive divisor ϕ for $\mathcal{P}(n,s,d)$; this is a simple corollary of our unique-factorization theorem. As a result, the weaker second property of G is enough for giving a divisibility testing for f and g: just check if $f(G_i)|g(G_i)$ as polynomials in x_i for every i.

1.4.1.2 Zero or Large Characteristic

In the proof of [5, Lemma 4.5], Bisht and Volkovich consider the Sylvester matrix $M_{x_i}(f^\alpha, g^\beta)$ of some suitably chosen small powers of the polynomials f, g, viewed as polynomials in x_i. They then use the first property of G to prove that the rank of this matrix does not change when composing with the generator G_i, that is, $\text{rank}(M_{x_i}(f^\alpha, g^\beta)) = \text{rank}(M_{x_i}(f(G_i)^\alpha, g(G_i)^\beta))$. From that, they deduce that it must be the case that $\phi(G_i)$ and $\psi(G_i)$ do not share factors, as the existence of such a factor would result in rank decrease.

To obtain our result, instead of considering the Sylvester matrix $M_{x_i}(f^\alpha, g^\beta)$, we consider the matrix defining the discriminant $\Delta(fg)$ of the polynomial fg. That is, we consider the Sylvester matrix $M_{x_i}(fg, (fg)')$ of fg and its formal derivative with respect to x_i, $(fg)'$. We next explain the idea behind it.

Suppose $\phi|f, \psi|g$ are two irreducible elements of $\mathcal{P}_{\text{Factors}}(n,s,d)$. View all polynomials as polynomials in some variable x_i, and consider the Sylvester matrix $M_{x_i}(fg, (fg)')$. It is not hard to prove that its rank essentially counts the number of distinct irreducible factors of fg (when appropriately counted – an irreducible factor of degree d should contribute d distinct irreducible factors). As all the determinants of minors of $M_{x_i}(fg, (fg)')$ are sparse

polynomials of low degree, we are able to show, using the first property of G, that the rank of this matrix does not change when composing with G_i. In particular, this means that $\phi(G_i), \psi(G_i)$ cannot share a common factor, as otherwise the number of unique factors of $fg(G_i)$ would be smaller than that of fg, in contradiction to the fact that the rank remains unchanged. With this result in hand, we obtain Theorem 16 and Theorem 15 by considering the factorization of the inputs composed with the generators G_i.

1.4.2 Finding Multiquadratic Factors of Products of Sparse Polynomials

We next show how to use the results of Section 1.4.1 to obtain Theorem 14. For the sake of presentation we assume here that $\mathrm{char}(\mathbb{F}) = 0$.

We first consider the following general setting. Let f be a product of polynomials in $\mathcal{P}_{\mathrm{Factors}}(n, s, d)$; that is, f is a product of factors of s-sparse polynomials of bounded individual degree d. Consider some irreducible $\phi \in \mathcal{P}_{\mathrm{Factors}}(n, s, d)$. By the second property of G it follows that for every i for which $x_i \in \mathrm{var}(\phi)$, the highest power of $\phi(G_i)$ that divides $f(G_i)$ is precisely the multiplicity of ϕ as a factor of f. In particular, this is true whenever ϕ is an irreducible s-sparse polynomial of bounded individual degree d. With this observation in mind, we are now ready to describe the algorithm for finding multilinear factors. Suppose $f = \prod_{i=1}^{\ell} \phi_i^{e_i} \in \mathbb{F}[x_0, \mathbf{x}]$ where ϕ_i are factors of $(n+1, s, d)$-sparse polynomials. Suppose further that we have blackbox access to f, and we want to recover all of its irreducible multilinear factors. [2] Our algorithm follows the following steps:

1.4.2.1 Step 1 – Normalization and Recursion

In this step, we would like to replace our input polynomial f with another polynomial $\tilde{f}$ that still captures all the information about the factors of f, but is normalized (in the sense that its free term is 1 as a polynomial in x_0). To do that, we use a standard method, employed for example in [5, 2]. Let $f = \sum_{i=0}^{\ell d} f_i x_0^i$ where $f_i \in \mathbb{F}[\mathbf{x}]$ and let $\tilde{f}(y, x_1, \ldots, x_n) = f(f_0 y, x_1, \ldots, x_n)/f_0$.[3] Note that as we have blackbox access to f, we also have blackbox access to f_0 and therefore to $\tilde{f}$. It is straightforward to see that each multilinear factor $h_i(\mathbf{x}) = a_i(\mathbf{x})x_0 + b_i(\mathbf{x})$ of f now corresponds to a multilinear factor $\frac{a_i f_0}{b_i} y + 1$ of $\tilde{f}$. Thus, one sees that:

1. If $a_i \neq 0$, then all the information about a_i and b_i is captured in $\tilde{f}$.
2. If $a_i = 0$, then the factor b_i cannot be studied from $\tilde{f}$. However, note that in this case $b_i | f_0$, and therefore it can be studied from a factorization of f_0.

Note that f_0 has the same structure of f – it is itself a product of factors of s-sparse polynomials of individual degree d. Therefore, by applying one recursive call, we can find all irreducible multilinear factors of f_0. Thus, to understand the factors with $a_i = 0$, we just need to understand, for any multilinear irreducible factor ϕ of f_0, how many times it divides f. We will show a way to do so in the next step. For now, the main benefit from this step is the normalization $\tilde{f}$ and the computation of the multilinear factors ϕ_i of f_0.

---

[2] The general algorithm finds the multiquadratic, sparse, irreducible factors of the input. To keep the presentation here as simple as possible, we consider only the multilinear factors.

[3] It is easy to reduce to the case $f_0 \neq 0$, so to keep the presentation as simple as possible we assume it to be the case.

1.4.2.2 Step 2 – Obtaining the factorization

As we saw in the first step, there are two types of factors – those with $a_i = 0$ and those with $a_i \neq 0$. Thus, we split the algorithm into two parts:

1. Recovering the factors for which $a_i = 0$. In this case, it holds that $b_i | f_0$. Therefore, as we already saw, it suffices to determine which irreducible factors of f_0 divide f (and to what power). To that end, note that for a given sparse irreducible factor ϕ of f_0, it follows from the second property of G and from the fact that f is a product of elements of $\mathcal{P}_{\text{Factors}}(n+1, s, d)$, that to calculate the multiplicity of ϕ as a factor of f it suffices to check how many times $\phi(G_j)$ divides $f(G_j)$, where $x_j \in \text{var}(\phi)$. This can be easily done, for example, by factoring both polynomials, which is possible as they have a constant number of variables and polynomial degree.
2. Recovering the factors for which $a_i \neq 0$. In this case, using the first step, we can get access to $\frac{a_i(G) f_0(G)}{b_i(G)}$, by factoring $\tilde{f}(y, G)$. Recovering a_i from such an expression is the rational interpolation problem that we earlier alluded to. Here we have the additional information that b_i is a multilinear factor of f_0. Thus, to find b_i, it suffices to understand, for each multilinear irreducible factor ϕ of f_0, what is the maximal power of ϕ that divides b_i. For any such factor ϕ, using the fact that a_i and b_i are coprime (as otherwise $a_i x_0 + b_i$ will not be irreducible), we have that $\phi^k | b_i$ if and only if $\phi^{t-k+1} \nmid \frac{a_i f_0}{b_i}$, where t is the multiplicity of ϕ as a factor of f_0. It follows that we just have to understand the multiplicity of ϕ as a factor of $\frac{a_i f_0}{b_i}$. Since a_i is sparse and f_0 is a product of polynomials in $\mathcal{P}_{\text{Factors}}(n, s, d)$, we have that $\frac{a_i f_0}{b_i}$ is a product of polynomials in $\mathcal{P}_{\text{Factors}}(n, s, d)$ as well, so we can finish by computing the multiplicity of $\phi(G_j)$ as a factor of $\frac{a_i(G_j) f_0(G_j)}{b_i(G_j)}$ (where $x_j \in \text{var}(\phi)$). This recovers b_i and subsequently a_i as well, completing the algorithm.

In the above, we actually solved a rational interpolation problem. That is, we had access to $\frac{af}{b}$, where $b | f$, and needed to recover a and b from this information. We claim that in fact, the algorithm described above works in greater generality, as given by Theorem 34.

We conclude with the following remarks. First, we note that it is possible to modify our algorithm to recover all the multiquadratic irreducible factors of f, rather than just the multilinear ones. We further note that the algorithm described above still works when the assumption on the field $\mathbb{F}$ is removed, albeit with a slightly worse running time.

1.4.3 Finding Sparse Divisors of Sparse Polynomials

We now describe the ideas behind our main result (Theorem 9). Recall that our goal is to output all the sparse divisors of a given (n, s, d)-sparse polynomial f. That is, we wish to efficiently find those divisors of f that can be "written down" efficiently.

To obtain such an algorithm, a natural thing to do is to follow the lines of the algorithm presented in Section 1.4.2. The first step of this algorithm is performing a normalization and then applying a recursive call for factoring the free term f_0. But now, a problem arises – this recursive call only returns the irreducible sparse factors of f_0, while the free term of each factor of f (b_i in the language of Section 1.4.2) might be a *reducible* sparse divisor of f_0 whose irreducible factors are not sparse; in this case, the recursive call did not provide us with any valuable information about b_i. In particular, we cannot reconstruct b_i by studying how many times each sparse irreducible factor of f_0 divides it as we did in Section 1.4.2. In fact, the true reason the algorithm presented in Section 1.4.2 works is that in that context, b_i is sparse multilinear (or sparse multiquadratic in the more general case), and therefore so are its irreducible factors; hence, we indeed get information about b_i from the recursive call. In other words, the free terms there belong to a class of sparse polynomials that is closed under factorization.

This motivates considering the problem of finding all the *sparse divisors* of a sparse polynomial. This way, we will still have information about the free terms of the factors of f – the recursive call will actually compute them. Indeed, sparse divisors of sparse divisors of (n, s, d)-sparse polynomials are themselves sparse divisors of (n, s, d)-sparse polynomials, so this class is closed under factorization. This enables us to recover all these sparse divisors in deterministic $\text{poly}(n, s^{\text{poly}(d)})$-time. Indeed, the recursive call returns all the sparse divisors of the free term f_0 rather than just the set of irreducible factors of it, so we can directly brute-force over the list of these factors to find b_i. This yields Theorem 9.

At first glance, this approach may seem surprising, as the existence of such an algorithm suggests that the possible number of outputs is bounded above by $\text{poly}(n, s^{\text{poly}(d)})$. Indeed, Theorem 8 ensures that s-sparse polynomials of bounded individual degree d have no more than $d \log s$ irreducible factors (that are not monomials), and therefore, no more than s^d divisors (not divisible by a monomial) overall. To prove Theorem 8, we show that if $f \in \mathcal{P}(n, s, d)$ is not divisible by any monomial, then one can identify a set of $\log s$ variables that satisfy the property that every irreducible factor of f depends on at least one of them. From this claim it easily follows that f has at most $d \log s$ irreducible factors, and therefore at most $2^{d \log s} = s^d$ divisors overall. To prove that such a set of variables exist, we analyze the Newton polytope of f, and show that if the minimal set of variables satisfying the property is of size k, then the Newton polytope of f has at least 2^k vertices. This is somewhat intuitive, as each additional variable in the set effectively adds a dimension to the polytope and therefore doubles the number of its vertices.

▶ Remark 18. Theorem 8 is easily seen to be tight, for example by considering $f(x_1, \ldots, x_n) = \prod_{i=1}^{\log s}({x_i}^d - 1)$ over $\mathbb{C}$.

We note that Theorem 9 is not implied by the methods of [2]; indeed, there the bound on the sparsity of *all irreducible factors* of f is needed to hit all resultants of different factors; therefore, applying their methods to this problem must result in a running time of at least $s^{O(d^2 \log n)}$, which is their bound on the sparsity of the factors of f.

One main drawback of our algorithm is that it does not tell us what are the *irreducible* sparse divisors of the input polynomial, as we are not familiar with a way of efficiently identifying the irreducible polynomials from the outputs we get. On the other hand, if we are guaranteed that f is a product of irreducible sparse factors then we can easily find the irreducible factors among all the sparse divisors. This solves Question 3 for the special case of (n, s, d)-sparse polynomials.

These ideas are also used in the proof of Theorem 10. Recall that Question 4 asks for a deterministic, polynomial-time algorithm that given blackbox access to a product of (n, s, d)-sparse polynomials returns the factors. While we do not know a polynomial-time algorithm when the number of terms is unbounded, we show an algorithm that runs in quasi-polynomial-time in their number, making the first step towards a resolution of the problem. The algorithm is the natural, straightforward generalization of the algorithm described above. Unfortunately, it requires quasi-polynomial-time as we actually find all the s-sparse divisors of bounded individual degree d of f, and there are (possibly) quasi-polynomially many of them, as can be deduced from Theorem 8 to obtain the following corollary.

▶ **Corollary 19.** *Let f be a product of ℓ elements of $\mathcal{P}_{Factors}(n, s, d)$. The number of (n, s, d)-sparse divisors of f that are not divisible by a monomial is at most*

$$\sum_{i=0}^{d \log s} \binom{\ell d \log s}{i} \leq s^{d(2+\log \ell)}.$$

1.4.4 Factorization for Sparse Polynomials

Lastly, we discuss our algorithms for factoring (n, s, d)-sparse polynomials and their products. These algorithms are essentially the same as the algorithms we already presented; the only difference is that now, when dealing with elements of $\mathcal{P}_{\text{Factors}}(n, s, d)$, we view them as elements of $\mathcal{P}(n, s^{O(d^2 \log n)}, d)$.

Consider for example our algorithm for finding all the sparse divisors of an (n, s, d)-sparse polynomial. With a slight modification, we can make it output all the $(n, s^{O(d^2 \log n)}, d)$-sparse divisors of the input polynomial in $\text{poly}(n, s^{d^2 \log n})$-time. By [2], these are all the possible divisors. Identifying the irreducible factors among these can be easily done. For example, we can check for any two divisors ϕ and ψ if $\psi|\phi$ by trying to sparse-interpolate their quotient – which should be an $s^{O(d^2 \log n)}$-sparse polynomial.

As for our algorithm for factoring products of $\mathcal{P}_{\text{Factors}}(n, s, d)$ elements, it is identical to the algorithm described in Section 1.4.2, now taking into account that all irreducible factors of interest are $s^{O(d^2 \log n)}$-sparse, rather than s-sparse and multiquadratic.

1.5 Organization

In Section 2 we give some basic definitions and known results. Section 2.4 contains the construction of our genrator. Our meta algorithm is given in Section 3. Due to space constraints all the proofs are defered to the full version [6].

2 Preliminaries

We use $\mathbf{x}$ to denote the n-variate vector of variables, $\mathbf{x} = (x_1, \ldots, x_n)$. In general, bold-face letters such as $\mathbf{x}, \mathbf{e}$ will be used to denote n-tuples, where the domain of the n-tuple will always be clear from the context.

We say that two nonzero polynomials f, g are associate if one is a scalar multiple of the other. We also state the following special case of Gauss' lemma.

▶ **Definition 20.** *A polynomial $f \in \mathbb{F}[x_0, \mathbf{x}]$ is said to be reversed-monic with respect to x_0 if its free term as a polynomial in $\mathbb{F}(\mathbf{x})[x_0]$ is 1. Equivalently, f is reversed-monic with respect to x_0 if it is of the form $f = 1 + \sum_{i \geq 1} f_i x_0^i$ where $f_i \in \mathbb{F}[\mathbf{x}]$.*

▷ Claim 21. Let $f \in \mathbb{F}[x_0, \mathbf{x}]$ be a reversed-monic polynomial with respect to x_0. Then, the number of irreducible factors of f is at most $\deg_{x_0}(f)$.

▶ **Definition 22** (Normalization). *Let $f(x_0, \mathbf{x}) = \sum_{i=0}^{d} f_i(\mathbf{x}) x_0^i \in \mathbb{F}[x_0, \mathbf{x}]$, such that $x_0 \nmid f$. The normalization of f with respect to x_0 is the reverse monic polynomial*

$$\tilde{f}(y, \mathbf{x}) = f(y f_0, x_1, \ldots, x_n)/f_0 = 1 + \sum_{i=1}^{d} f_i(\mathbf{x}) f_0^{i-1}(\mathbf{x}) y^i. \tag{1}$$

▷ Claim 23. Let f as in Definition 22. Suppose $f = \prod_{i=0}^{N} h_i \in \mathbb{F}[x_0, \mathbf{x}]$ is the factorization of f into irreducible factors h_i, and denote each factor by $h_i = \sum_{j=0}^{d_i} c_{i,j} {x_0}^j$ (where $c_{i,j} \in \mathbb{F}[\mathbf{x}]$). Then, the factorization of $\tilde{f}$ is given by

$$\tilde{f} = \prod_{i=0}^{N} \tilde{h}_i \text{ , where } \quad \tilde{h}_i = \sum_{j=0}^{d_i} \frac{c_{i,j}}{c_{i,0}} {f_0}^j y^j \ .$$

Moreover, every factor (not necessarily irreducible) of f of the form $\sum_{j=0}^{d} \zeta_j {x_0}^j$ (where $\zeta_j \in \mathbb{F}[\mathbf{x}]$) corresponds to a factor of $\tilde{f}$ of the form $\sum_{j=0}^{d} \frac{\zeta_j}{\zeta_0} {f_0}^j y^j$. In particular, every multiquadratic factor of f of the form $a_2 x_0^2 + a_1 x_0 + b$ corresponds to a factor of $\tilde{f}$ of the form $1 + \frac{a_1 f_0}{b} y + \frac{a_2 f_0^2}{b} y^2$. In addition, we have blackbox access to $\tilde{f}$ from blackbox access to f.

2.1 Factorization and Interpolation of Polynomials

▶ **Theorem 24** (Univariate factorization over $\mathbb{Q}$). *Let $f \in \mathbb{Q}[x]$ have degree d and bit complexity B. Then, there is a deterministic algorithm that outputs all irreducible factors of f whose running time is* $\mathrm{poly}(d, B)$.

For the following theorem see e.g., [29, Section 9].

▶ **Theorem 25** (Univariate factorization over Finite Fields). *Let $\mathbb{F}_q$ be a field of size $q = p^k$ and characteristic $p > 0$. Let $f(x) \in \mathbb{F}_q[x]$ be a polynomial of degree d. Then, there is a deterministic algorithm that outputs all irreducible factors of f in time* $\mathrm{poly}(p, \log q, d)$.

We next state a theorem concerning deterministic multivariate factorization in the multivariate case. See e.g., [20, 17]. These results determine the dependence on the field, its characteristic and the bit complexity of the input polynomial, in all our algorithms that invoke polynomial factorization.

▶ **Theorem 26.** *There exists a deterministic algorithm that given an n-variate polynomial $f \in \mathbb{F}[\mathbf{x}]$ of total degree d, and bit complexity B, outputs its factorization into irreducible polynomials. Its running time is* $\mathrm{poly}(d^n) \cdot T(d, B, \mathbb{F})$, *where $T(d, B, \mathbb{F})$ is the running time of univariate factorization of degree d polynomials, with bit complexity B, over $\mathbb{F}$.*

▶ **Lemma 27.** *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an n-variate polynomial of total degree d. Then, there is a deterministic polynomial-time algorithm that given blackbox access to f returns its monomial representation. Its running time is* $\mathrm{poly}((d+1)^n)$.

▶ **Theorem 28** ([2, Theorem 1]). $\mathcal{P}_{Factors}(n, s, d) \subset \mathcal{P}(n, s^{O(d^2 \log n)}, d)$. *In other words, every factor of an s-sparse polynomial of bounded individual degree d in n-variables is $s^{O(d^2 \log n)}$-sparse.*

▶ **Theorem 29** ([2, Theorem 2]). *There is a deterministic* $\mathrm{poly}(n, d^d, s^{d^7 \log n})$*-time algorithm, that given an (n, s, d)-sparse polynomial, returns its factorization into irreducible factors.*

▶ **Corollary 30** (Finding monomial divisors). *There is a deterministic* $\mathrm{poly}(n, d, s, \ell)$*-time algorithm that, given blackbox access to a product f of ℓ polynomials in $\mathcal{P}_{Factors}(n, s, d) \subset \mathbb{F}[\mathbf{x}]$, returns the multiplicity k_i of x_i as a factor of f. Moreover, if M is the largest monomial divisor of f, such that $f = Mg$, then we can obtain blackbox access to g from blackbox access to f.*

2.2 Primitive Divisors

▶ **Definition 31.** *Let $\mathcal{P} \subset \mathbb{F}[\mathbf{x}]$ be a class of polynomials. We say that a polynomial $g \in \mathbb{F}[\mathbf{x}]$ is a primitive divisor for $\mathcal{P}$ if it satisfies the following two conditions.*

1. *For every $f \in \mathcal{P}$ and every $t \in \mathbb{N}$, the polynomial* $\gcd(f, g^t)$ *is a power of g. Equivalently, for every $f \in \mathcal{P}$, there exists $k \geq 0$ and a polynomial $\tilde{g}$ coprime to g such that $f = g^k \cdot \tilde{g}$.*
2. *No non-scalar multiple of g satisfies the first condition.*

▶ **Definition 32.** *Let $f \in \mathcal{P}$, and let g be a primitive divisor for $\mathcal{P}$. The multiplicity of g as a factor of f, denoted by $v_g(f)$, is the power of g appearing in the factorization of f to primitive divisors.*

▶ **Lemma 33.** *There is a deterministic $\mathrm{poly}(n, (d^2)!, s^{d^3}, \ell)$-time algorithm that, given blackbox access to a product f of ℓ (n, s, d)-sparse polynomials, and blackbox access to a primitive divisor ϕ for $\mathcal{P}(n, s, d)$, returns the multiplicity of ϕ as a factor of f (in the sense of Definition 32). In fact, the algorithm only needs to know the values of f on the image of the generator $G_{(m)}$, for some explicit value $m = \mathrm{poly}(n, (d^2)!, s^{d^3})$. Moreover, if ϕ is an irreducible s-sparse polynomial, this algorithm works if f is a product of polynomials in $\mathcal{P}_{Factors}(n, s, d)$ (rather than $\mathcal{P}(n, s, d)$); if furthermore ϕ is multiquadratic, taking $m = \mathrm{poly}(n, (d^2)!, s^{d^2})$ suffices, and the running time reduces to $\mathrm{poly}(n, (d^2)!, s^{d^2}, \ell)$.*

2.3 Rational Interpolation Theorem

▶ **Theorem 34.** *Suppose we have blackbox access to a set of rational functions, defined over a field $\mathbb{F}$, of the form $\frac{a_j f^j}{b}$, for $1 \le j \le D$, where:*

1. *a_j is an (n, s, d)-sparse polynomial for every $1 \le j \le D$.*
2. *f is a product of ℓ polynomials in $\mathcal{P}_{Factors}(n, s, d)$ to which we have blackbox access.*
3. *We have a set $\mathcal{F}$ of s-sparse irreducible factors of f, call them ϕ_i.*
4. *b is an s-sparse factor of f whose irreducible factors are all elements of $\mathcal{F}$.*
5. *$\gcd(a_1, \ldots, a_D, b) = 1$.*

Then, there is an algorithm that recovers the a_j-s and b from evaluations of $\frac{a_j f^j}{b}$ on the image of the generator $G_{(m)}$, for some explicit m (depending on $\mathrm{char}(\mathbb{F})$), such that the running time is $\mathrm{poly}(m, \ell, D)$, where:

- *If $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > 2d$, one can take $m = \mathrm{poly}(n, d!, s^d, D)$.*
- *For arbitrary fields we have $m = \mathrm{poly}(n, (d^2)!, s^{d^3}, D)$. If we further assume that all elements of $\mathcal{F}$ are multiquadratic, then we can take $m = \mathrm{poly}(n, (d^2)!, s^{d^2}, D)$.*

2.4 Generator Construction

In this section, we construct our generator, which is a map $G : \mathbb{F}^6 \to \mathbb{F}^n$ that satisfies the properties described in Section 1.4.1.

Fix the following sets:

1. A set of integers $\mathcal{I} = \{1, \ldots, m\}$ of size m, that contains m different residues mod q.
2. A set of field elements $\mathcal{A} = \{\alpha_1, \ldots, \alpha_m\} \subset \mathbb{F}$ of size m. For every $1 \le i \le m$ let A_i be the Lagrange interpolating polynomials for $\mathcal{A}$.
3. A set of field elements $\mathcal{B} = \{\beta_1, \ldots, \beta_n\} \subset \mathbb{F}$ of size n. For every $1 \le i \le n$ let B_i be the Lagrange interpolating polynomials for $\mathcal{B}$.
4. A set of field elements $\mathcal{C} = \{\gamma_1, \ldots, \gamma_n\} \subset \mathbb{F}$ of size n. For every $1 \le i \le n$ let C_i be the Lagrange interpolating polynomials for $\mathcal{C}$.
5. A set of field elements $\mathcal{T} = \{\delta_1, \ldots, \delta_{ndq}\} \subset \mathbb{F}$ of size ndq.

To define our generator we first define the generator of Klivans and Spielman and state some of its properties.

▶ **Definition 35** (The KS-generator)**.** *In the notations above, the KS-generator is defined as*

$$G^{KS}_{(m)}(x, y, z, w) = \sum_{i=1}^{m} A_i(y) \left((1 + B_1(z)(w-1))x^{i \bmod q}, \ldots, (1 + B_n(z)(w-1))x^{i^n \bmod q} \right). \quad (2)$$

▶ **Theorem 36** ([18, Theorem 11]). *Let $f \in \mathbb{F}[\mathbf{x}]$ be an (n,s,d)-sparse polynomial. Let $m = (nds)^2$. Then, f can be reconstructed in time* $\mathrm{poly}(n,s,d)$ *from the following set of evaluations:*

$$\left\{ f\left(G^{KS}_{(m)}(x,y,z,w)\right) \mid \ y \in \mathcal{A},\ z \in \mathcal{B},\ x,w \in \mathcal{T} \right\}.$$

Another tool that we will use is the generator defined in [23], which is given by the formula

$$G^{\mathrm{SV}}(u,v) = (C_1(v)u, \ldots, C_n(v)u) \ . \tag{3}$$

The basic property of this generator is that it makes it possible to "revive" a variable of our choice. E.g., by setting $v = \gamma_j$ we get the vector that has zeros in all coordinates except of the j-th coordinate where we have the variable u.

We are now ready to define our generator.

▶ **Definition 37** (The generator $G_{(m)}$). *The generator $G_{(m)} : \mathbb{F}^6 \to \mathbb{F}^n$ is defined as*

$$G_{(m)}(x,y,z,w,u,v) = G^{KS}_{(m)}(x,y,z,w) + G^{SV}(u,v) \ . \tag{4}$$

We further define the generators obtained from reviving a variable. Denote with $G^{KS}_{(m)}(x,y,z,w)_i$ the i-th coordinate of $G^{KS}_{(m)}(x,y,z,w)$.

For any $1 \le i \le n$, let $G_{(m,i)}$ be the generator obtained from $G_{(m)}$ by restricting to $u = u - G^{KS}_{(m)}(x,y,z,w)_i$ and $v = \gamma_i$. That is,

$$G_{(m,i)}(x,y,z,w,u) = \left(G^{KS}_{(m)}(x,y,z,w)_1, \ldots, u, \ldots, G^{KS}_{(m)}(x,y,z,w)_n\right) \ . \tag{5}$$

Let's explain this definition. The first part $G^{\mathrm{KS}}_{(m)}$ enables us to interpolate (n,s,d)-sparse polynomials (Theorem 36).

3 Meta-Algorithm

In this section we present a meta-algorithm that will be used, with some variations, in the proofs of Theorems 9, 10, 11, 12, 13 and 14, which state our algorithmic results. Since Theorem 10 follows as a corollary of Theorem 11, we will not discuss it here.

Each of these algorithms follows the outline of the meta-algorithm, and in their proofs we will refer to steps of the meta-algorithm and describe how they are implemented in the relevant setting.

1. **Setting:**
   We assume that we want to learn properties of a polynomial f such that f is either
   a. (n,s,d)-sparse (Theorem 9, Theorem 12)
   b. a product of polynomials in $\mathcal{P}_{\mathrm{Factors}}(n,s,d)$ (Theorem 11, Theorem 13, Theorem 14)
   and we want to find a list $\mathcal{F}(f)$ of divisors of f having certain properties. For every variable x_i, the polynomial $f|_{x_i=0}$ is also of the same type as f. This is clearly true when f is (n,s,d)-sparse. When f is a product of polynomials in $\mathcal{P}_{\mathrm{Factors}}(n,s,d)$, this follows from Lemma 39, whose formulation and proof are given at the end of this section.
   In what follows, for the sake of convenience, we assume that f is an $n+1$-variate polynomial and is an element of $\mathbb{F}[x_0, \mathbf{x}]$.
2. **Preprocessing:**
   a. Remove from f its largest monomial divisor (using Corollary 30).

b. Normalize f to get $\tilde{f}(y, \mathbf{x}) = f(yf_0, x_1, \ldots, x_n)/f_0$. Set $\hat{f} = f(y, G)$ for an appropriate generator G. Observe that $\tilde{f}(y, \mathbf{x})$ is reverse monic with respect to y, and hence so is $\hat{f}$. By Claim 21 it then follows that $\hat{f}$ has at most $\deg_y(\hat{f}) \leq d$ irreducible factors if f is (n, s, d)-sparse, and at most $\deg_y(\hat{f}) \leq \ell d$ if f is a product of ℓ polynomials in $\mathcal{P}_{\text{Factors}}(n, s, d)$.

c. Use Lemma 27 to interpolate $\hat{f}$ as a polynomial in $O(1)$ many variables. Then apply Theorem 26 to obtain its factorization into irreducible factors.

3. **Recursive call to solve for f_0:** Recursively solve the problem for $f_0 = f|_{x_0=0}$, obtaining a list $\mathcal{F}(f_0)$ of divisors not divisible by a monomial or irreducible factors, according to the problem.
4. **Generating a list of candidate solutions:**
   By Claim 23, every divisor $h = \sum_{i=0}^{d} c_i x_0^i$ of f gives rise to a divisor $\sum_{i=0}^{d} \frac{c_i}{c_0} f_0^i y^i$ of $\tilde{f}$ and hence to a divisor $\hat{h} = \sum_{i=0}^{d} \frac{c_i(G)}{c_0(G)} f_0^i y^i$ of $\hat{f}$. We would like to get access to $c_i(G)$ in order to interpolate them, and through that obtain access to h. This leads us to the rational interpolation problem, where we have some information about c_0 (or even c_0 itself) coming from $\mathcal{F}(f_0)$.

   a. In Theorem 9, Theorem 11 we will have that $c_0 \in \mathcal{F}(f_0)$. In Theorem 12 c_0 is a product of elements of $\mathcal{F}(f_0)$, but we have enough time to enumerate all relevant products so we can pretend that $c_0 \in \mathcal{F}(f_0)$.
   In fact, the above statement is not entirely accurate. The recursive call returns the set $\mathcal{F}(f_0)$ of all divisors of f_0 that are not divisible by a monomial, together with the largest monomial dividing f_0. It might be the case that c_0 is a divisor of f_0 that *is* divisible by a monomial. In this case, c_0 will not belong to $\mathcal{F}(f_0)$. However, c_0 does lie in $\mathcal{F}(f_0)$ *up to a multiplication by a monomial*. As we shall see, this suffices for our purposes.

   b. In Theorem 13 and Theorem 14 the situation is a bit different. Here, c_0 may be a product of elements from $\mathcal{F}(f_0)$, and we cannot go over all possible ways to obtain c_0. However, in this case we will have that $\mathcal{F}(f_0)$ contains all the irreducible factors of c_0, which is what Theorem 34 requires.

   Accordingly, for each factor $\hat{h} = \sum_{i=0}^{d} e_i y^i$ we compose a list $\mathcal{F}_{\hat{h}}$ as follows:

   a. When we have a list containing (up to multiplication by a monomial) the correct c_0 (Theorem 9, Theorem 11, Theorem 12), we iterate over all possible candidates c_0 in that list. For each such candidate, we interpolate the polynomials of the form $\left((x_1 \cdots x_n)^d c_0 / f_0^i\right)(G)\, e_i$ in order to recover the coefficients c_i. We then add to $\mathcal{F}_{\hat{h}}$ those polynomials $\sum_{i=0}^{d} c_i x_0^i$ that are valid candidates for the problem, after dividing by their largest monomial divisor:
      i. In Theorem 9, Theorem 11 we keep all (n, s, d)-sparse polynomials.
      ii. In Theorem 12 we keep all (n, S, d)-sparse polynomials, where $S = s^{O(d^2 \log n)}$ is the upper bound on sparsity of factors of (n, s, d)-sparse polynomials obtained in Theorem 28.

      The term $(x_1 \cdots x_n)^d$ is needed to deal with the fact that c_0 is in $\mathcal{F}(f_0)$ only up to division by a monomial. We shall explain this point in full detail in the proof of Theorem 9.

   b. If we only have a list of irreducible factors of c_0 (Theorem 13 and Theorem 14), apply Theorem 34 on the coefficients of $\hat{h}$ to recover potential c_0 and $c_1, \ldots, c_d$. We put the resulting polynomial $\sum_{i=0}^{d} c_i x_0^i$ in the list $\mathcal{F}_{\hat{h}}$ if:
      i. For Theorem 13 it is (n, S, d)-sparse.
      ii. For Theorem 14, it is multiquadratic and (n, s, d)-sparse.

      We note that, in both cases, we set $\mathcal{F}_{\hat{h}} = \mathcal{F}(f_0)$ for $\hat{h} = 1$.

5. **Pruning the list:**
   Depending on the application we either have to keep divisors or irreducible divisors. We do that in two steps – we first find the divisors in the list of candidates, and then, if needed, identify the irreducible polynomials among them. The implementation of the two steps vary in the different algorithms; we give here a solution that works for all cases, though not always optimal, and provide a better implementation in the proofs of the algorithms.
   In all cases we apply Theorem 16 to remove from the list those polynomials that do not divide f. This already suffices for Theorem 9 and Theorem 11.
   To obtain irreducible divisors as required in Theorem 12, Theorem 13, Theorem 14, we apply Theorem 16 to each two polynomials g, h in the list of divisors to check whether g divides h, and keep only those divisors that are not divisible by any other divisor.
6. **Computing multiplicities:**
   To compute multiplicities, where needed, we apply Lemma 40 or Lemma 33, depending on the characteristic. Here as well there are better solutions in some cases; these are described in the proofs of the algorithms.

The analysis of the running time of the algorithm depends on the exact setting. However, in all settings we perform exactly one recursive call so the overall complexity is determined by the cost of the different steps of the algorithm, and it does not blow up because of the recursive call.

▶ Remark 38. The described meta-algorithm provides the outline that all our algorithms follow. It is, however, *not* an algorithm that *generalizes* all of them. It is possible to formulate two concrete algorithms which all our results follow – one generalizing Theorems 9, 11 and 12, and one generalizing Theorems 13 and 14. However, the precise formulations of these general algorithms are rather complicated and not particularly intuitive; hence, we have chosen not to include them in the present paper.

We close this section by stating the following lemma.

▶ **Lemma 39.** *Let f be a product of ℓ polynomials in $\mathcal{P}_{Factors}(n+1, s, d) \subset \mathbb{F}[x_0, \mathbf{x}]$. Suppose further that $x_0 \nmid f$, and let f_0 be the free term of f as a polynomial in x_0. Then, f_0 is a product of ℓ elements of $\mathcal{P}_{Factors}(n, s, d) \subset \mathbb{F}[\mathbf{x}]$.*

▶ **Lemma 40.** *Let $\mathbb{F}$ be a field of characteristic zero, or of characteristic larger than $2d$. There is a deterministic $\mathrm{poly}(n, d!, s^d, \ell)$-time algorithm that, given blackbox access to a product, f, of ℓ polynomials in $\mathcal{P}_{Factors}(n, s, d)$ and blackbox access to an irreducible polynomial ϕ in $\mathcal{P}_{Factors}(n, s, d)$, both defined over $\mathbb{F}$, returns the multiplicity of ϕ as a factor of f. In fact, the algorithm only needs to know the values of f and ϕ on the image of the generator $G_{(m)}$ for some explicit $m = \mathrm{poly}(n, d!, s^d)$.*

References

1 Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 301–309, 1988.

2 Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Deterministic factorization of sparse polynomials with bounded individual degree. *Journal of the ACM (JACM)*, 67(2):1–28, 2020. doi:10.1145/3365667.

3 Somnath Bhattacharjee, Mrinal Kumar, Shanthanu S Rai, Varun Ramanathan, Ramprasad Saptharishi, and Shubhangi Saraf. Closure under factorization from a result of Furstenberg. *arXiv preprint arXiv:2506.23214*, 2025. doi:10.48550/arXiv.2506.23214.

4 Pranav Bisht and Nitin Saxena. Derandomization via symmetric polytopes: Poly-time factorization of certain sparse polynomials. *ACM Transactions on Computation Theory*, 17(2):1–20, 2025. doi:10.1145/3719022.

5 Pranav Bisht and Ilya Volkovich. On solving sparse polynomial factorization related problems. *Computational Complexity*, 34(1):7, 2025. doi:10.1007/S00037-025-00268-5.

6 Aminadav Chuyoon and Amir Shpilka. On factorization of sparse polynomials of bounded individual degree. *CoRR*, abs/2603.07589, 2026. doi:10.48550/arXiv.2603.07589.

7 Michael Clausen, Andreas Dress, Johannes Grabmeier, and Marek Karpinski. On zero-testing and interpolation of k-sparse multivariate polynomials over finite fields. *Theoretical Computer Science*, 84(2):151–164, 1991. doi:10.1016/0304-3975(91)90157-W.

8 Annie Cuyt and Wen-shin Lee. Sparse interpolation of multivariate rational functions. *Theoretical Computer Science*, 412(16):1445–1456, 2011. doi:10.1016/J.TCS.2010.11.050.

9 Pranjal Dutta and Sumanta Ghosh. Sigact news complexity theory column 121. *SIGACT News*, 55(2):53–88, June 2024. doi:10.1145/3674159.3674165.

10 Pranjal Dutta, Amit Sinhababu, and Thomas Thierauf. Derandomizing multivariate polynomial factoring for low degree factors. In Amit Kumar and Noga Ron-Zewi, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, London School of Economics, London, UK, August 28-30, 2024*, volume 317 of *LIPIcs*, pages 75:1–75:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.APPROX/RANDOM.2024.75.

11 Michael A Forbes. Deterministic divisibility testing via shifted partial derivatives. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 451–465. IEEE, 2015. doi:10.1109/FOCS.2015.35.

12 Michael A Forbes and Amir Shpilka. Complexity theory column 88: Challenges in polynomial factorization1. *ACM SIGACT News*, 46(4):32–49, 2015. doi:10.1145/2852040.2852051.

13 Dima Grigoriev, Marek Karpinski, and Michael F. Singer. Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields. *SIAM J. Comput.*, 19(6):1059–1063, 1990. doi:10.1137/0219073.

14 Dima Grigoriev, Marek Karpinski, and Michael F Singer. Computational complexity of sparse rational interpolation. *SIAM Journal on Computing*, 23(1):1–11, 1994. doi:10.1137/S0097539791194069.

15 Ankit Gupta, Neeraj Kayal, and Satyanarayana V. Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 625–642. ACM, 2012. doi:10.1145/2213977.2214035.

16 Erich Kaltofen. Computing with polynomials given by straight-line programs I: greatest common divisors. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 131–142, 1985. doi:10.1145/22145.22160.

17 Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.*, 14(2):469–489, 1985. doi:10.1137/0214035.

18 Adam R. Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, STOC '01, pages 216–223, New York, NY, USA, 2001. Association for Computing Machinery. doi:10.1145/380752.380801.

19 Mrinal Kumar, Varun Ramanathan, and Ramprasad Saptharishi. Deterministic algorithms for low degree factors of constant depth circuits. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3901–3918. SIAM, 2024. doi:10.1137/1.9781611977912.137.

20 Arjen Klaas Lenstra. Factoring multivariate polynomials over finite fields. *J. Comput. System Sci.*, 30(2):235–248, 1985. doi:10.1016/0022-0000(85)90016-9.

21 Nitin Saxena. Progress on polynomial identity testing. *Bull. EATCS*, 99:49–79, 2009.

22 Nitin Saxena. Progress on polynomial identity testing-ii. In *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, pages 131–146. Springer, 2014.

23 Amir Shpilka and Ilya Volkovich. Improved polynomial identity testing for read-once formulas. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 700–713. Springer, 2009. doi:10.1007/978-3-642-03685-9_52.

24 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010. doi:10.1561/0400000039.

25 Joris van der Hoeven and Grégoire Lecerf. On sparse interpolation of rational functions and gcds. *ACM Commun. Comput. Algebra*, 55(1):1–12, 2021. doi:10.1145/3466895.3466896.

26 Ilya Volkovich. Deterministically factoring sparse polynomials into multilinear factors and sums of univariate polynomials. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, pages 943–958. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.943.

27 Ilya Volkovich. On some computations on sparse polynomials. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*, pages 48–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.48.

28 Joachim Von Zur Gathen and Erich Kaltofen. Factoring sparse multivariate polynomials. *Journal of Computer and System Sciences*, 31(2):265–287, 1985. doi:10.1016/0022-0000(85)90044-3.

29 Joachim von zur Gathen and Victor Shoup. Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, 2(3):187–224, 1992. doi:10.1007/BF01272074.

Wide Replacement Products Meet Gray Codes: Toward Optimal Small-Bias Sets

Gil Cohen
Tel Aviv University, Israel

Itay Cohen
Tel Aviv University, Israel

Abstract

Optimal small-bias sets sit at the crossroads of coding theory and pseudorandomness. Reaching optimal parameters would, in particular, meet the long-standing goal of matching the Gilbert–Varshamov bound for binary codes in the high-distance regime. In a breakthrough, Ta-Shma [18] constructed near-optimal small-bias sets via the Rozenman–Wigderson expander-walk framework, using the wide-replacement product to maintain s "secure" registers and to route the walk through them. Within this framework, two barriers remain en route to optimal small-bias sets: (i) the cost of maintaining registers and (ii) limitations inherited from spectral-gap bounds for expanders.

We overcome the first – arguably the more critical – barrier. Our key technical insight is that registers can be *reused* even after they are exposed. Using a Gray–code–style reuse schedule, we recycle the same s registers exponentially many times in s, thereby reducing the register-maintenance cost exponentially. This yields the first improvement over Ta-Shma's construction in nearly a decade – quantitatively modest but an important first step toward a truly optimal construction. The remaining barrier is fairly standard in isolation; the challenge is to overcome it in concert with our register-reuse framework.

2012 ACM Subject Classification Theory of computation → Error-correcting codes; Theory of computation → Random walks and Markov chains; Theory of computation → Expander graphs and randomness extractors; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases small-bias sets, bias amplification, expansion-amplification, gray codes

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.21

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2025/179/

Funding *Gil Cohen*: Supported by ERC starting grant 949499 and by the Israel Science Foundation grant 2989/24.
Itay Cohen: Supported by ERC starting grant 949499.

1 Introduction

Small-bias sets sit at the crossroads of pseudorandomness and coding theory. From the pseudorandomness viewpoint, a small-bias set can be thought of as a pseudorandom generator against linear (parity) tests. Specifically, a set $S \subseteq \{0,1\}^n$ is called ε-*biased* if for every nonzero $\tau \in \{0,1\}^n$,

$$\left|\mathbb{E}_{s\in S}\left[(-1)^{\langle s,\tau\rangle}\right]\right| \leq \varepsilon.$$

For any nonzero τ, a uniformly random $u \in \{0,1\}^n$ satisfies $\mathbb{E}\left[(-1)^{\langle u,\tau\rangle}\right] = 0$. Thus an ε-biased set behaves like the uniform distribution with respect to parity tests and can be viewed as the output distribution of a PRG that fools such tests within error ε.



© Gil Cohen and Itay Cohen;
licensed under Creative Commons License CC-BY 4.0



COMPUTATIONAL COMPLEXITY CONFERENCE

The main challenge is to explicitly construct, for given n and ε, an ε-biased set $S \subseteq \{0,1\}^n$ whose size is as small as possible. A standard probabilistic argument shows that ε-biased sets of size

$$O\big(n/\varepsilon^2\big)$$

exist. This bound is essentially optimal: up to a multiplicative $\log(1/\varepsilon)$ factor, it matches the lower bound given by the classical linear-programming bound [13].

From the coding-theoretic perspective, an ε-biased set S yields a binary linear code of block length $|S|$, relative distance $\delta = \frac{1}{2} - \varepsilon$, and rate $\rho = \frac{n}{|S|}$. In particular, a construction matching the probabilistic bound would give an explicit code meeting the Gilbert–Varshamov bound [9, 19] – a decades-old open problem in coding theory. In pseudorandomness, small-bias sets are key ingredients in PRG constructions that go beyond linear tests [20, 8, 5] and in the construction of randomness extractors [15, 7]; they also have applications across adjacent areas, including cryptography, combinatorics, and quantum protocols.

Given their importance, small-bias sets have been studied extensively. Naor and Naor gave a construction with linear (optimal) dependence on n but suboptimal dependence on ε, namely $O(n/\varepsilon^c)$ for some constant $c > 2$ [14]. Alon, Goldreich, Håstad, and Peralta provided several constructions with optimal dependence on ε but suboptimal in n, particularly $O\big(n^2/\varepsilon^2\big)$ [1]. From the coding viewpoint, keeping the n-dependence linear is crucial for the rate, while from the pseudorandomness perspective polynomial dependence on n is often acceptable, so the ε-dependence takes center stage. Several other constructions are known. For example, concatenating optimal algebraic–geometric codes with the Hadamard code yields a set of size $O(n/\varepsilon^3)$; using Hermitian codes instead gives an incomparable bound of $O\Big(\big(n/\varepsilon^2\big)^{5/4}\Big)$ [3].

About a decade ago, Ta-Shma [18] achieved a breakthrough with an explicit construction of size very close to optimal:

$$\frac{n}{\varepsilon^{2+\alpha}} \qquad \text{where} \qquad \alpha = \widetilde{O}\left(\left(\frac{1}{\log(1/\varepsilon)}\right)^{1/3}\right). \ [1] \tag{1}$$

1.1 Bias reduction

Ta-Shma's construction is based on *bias reduction*: starting from an ε_0-biased set (where ε_0 can even be a small constant), it transforms it into an ε-biased set for the target parameter ε. Since linear-in-n starting constructions are available, the heart of the matter is how the bias evolves throughout the reduction process.

A simple and natural way to reduce bias is this: given an ε_0-biased set S, consider $S + S \triangleq \{\, s_1 \oplus s_2 : s_1, s_2 \in S \,\}$, where $\oplus$ denotes bitwise XOR. In pseudorandomness terms, the new sample space is obtained by drawing two independent uniform elements from S and outputting their XOR. This approach is, however, expensive: treating $S + S$ as a multiset [1], its size is $|S+S| = |S|^2$, so in particular the construction incurs at least quadratic dependence on n.

---

[1] Throughout, we use the convention that $\widetilde{O}(\cdot)$ suppresses polylogarithmic factors in its argument; in our bounds it hides a factor of $(\log\log(1/\varepsilon))^{O(1)}$.

[1] Despite the name, small-bias "sets" are typically allowed to be multisets. This is natural for two reasons: (i) basic operations on small-bias sets (e.g., $S + S$) can introduce duplicates, and (ii) from the pseudorandomness viewpoint we often treat S as inducing a (possibly nonuniform) distribution over $\{0,1\}^n$.

In unpublished work, Rozenman and Wigderson (credited in Bogdanov's 2012 complexity course notes) proposed derandomizing this idea using expanders. Instead of sampling two independent $s_1, s_2 \in S$, they suggested taking an expander whose vertex set is S, sample a random edge $\{s_1, s_2\}$, and output $s_1 \oplus s_2$. They showed that the resulting bias is at most $\varepsilon_0^2 + \omega$, where ω is the spectral expansion of the expander - the penalty one pays for the derandomization. Iterating this transformation from a constant-bias starting point yields an ε-biased set of size about n/ε^4.

An alternative to repeatedly applying the derandomized "sum-of-two" step is to take a longer walk on the expander G. In particular, for a parameter $t \geq 2$, take a length-t random walk on G and output the bitwise XOR of the t vertices visited. This yields essentially the same sample-space size as the iterated edge-based version.

While the random walk approach did not beat the best known constructions at the time, it was the starting point for Ta-Shma's construction. He observed that, in the random-walk approach, two distinct effects account for the exponent 4 in the sample-space size n/ε^4. The first is the essentially unavoidable *quadratic* relation between the degree d of a d-regular expander and its spectral expansion ω; even for Ramanujan graphs one has $\omega \approx \frac{2}{\sqrt{d}}$. The second factor-of-two loss is the part Ta-Shma managed to eliminate almost entirely.

1.2 Neutralizing adversarial interference

The second factor-of-two arises from the adversary's ability to correlate the walk with a fixed linear test τ. A length-t walk on G is governed by the operator W^t, where W is the random-walk (normalized adjacency) matrix of G. In the presence of a linear test, an additional operator is interleaved. The test τ induces a $\{\pm 1\}$ labeling of the vertices: for each $s \in S$, $\chi_\tau(s) = (-1)^{\langle \tau, s \rangle}$. Let Π be the diagonal matrix with $(\Pi)_{s,s} = \chi_\tau(s)$. Writing $\mathbf{1}$ for the normalized all-ones vector, the bias against τ of the new sample space is

$$\left| \mathbf{1}^\top \Pi \left(W \Pi \right)^{t-1} \mathbf{1} \right|.$$

The interleaving of Π with W captures the "adversarial interference" responsible for the extra factor-of-two loss. Informally, the sign flips introduced by Π can neutralize every other step of the walk, effectively wasting half the steps.

The key insight underlying Ta-Shma's improvement is to restrict, at each step, the set of neighbors a vertex of G may transition to in a manner that is "hidden" from the adversary. In this way, the effect of Π on the walk remains largely under our control. Ta-Shma implements this approach using the *s-wide replacement product*, introduced by Ben-Aroya and Ta-Shma [2] for the purpose of constructing near-Ramanujan graphs.

Informally, the wide–replacement product uses a small "gadget" graph H to subsample the next vertex in a random walk. It extends operators such as the zig–zag product [16] and derandomized squaring [17], and has the advantage of saving randomness: subsampling neighbors uses fewer random bits than choosing a uniformly random neighbor. When H is itself an expander, this subsampling preserves mixing well, so the walk remains well behaved while using much less randomness.

Ta-Shma's key observation is that, beyond its randomness-saving role – which is also crucial for keeping the sample space small in our setting – the wide–replacement product can be used to *hide* information about steps of the random walk, provided it is implemented appropriately. To this end, we will require the gadget H to have a specific structure: think of H as consisting of s *registers* $R_1, \ldots, R_s$, and at step i we use register R_i. This can be achieved by taking H to be the Cayley graph of a group that is a direct product of s groups.

Ta-Shma showed that a length-s walk can be implemented using these s registers, and the net effect is that, instead of losing one in every two steps, we effectively lose only one out of every s steps.

Using more registers reduces the bias, but it also increases the sample-space size – exponentially in s. Balancing these two effects (by choosing s optimally) yields an ε-biased set whose size is as in Equation (1).

1.3 Our results

Ta-Shma's construction is already nearly optimal, but the importance of the problem calls for a truly optimal solution. This is especially compelling from the coding-theoretic viewpoint: matching the Gilbert–Varshamov bound would be a major breakthrough. From the pseudorandomness perspective, linear tests are among the simplest classes, so constructing an optimal PRG against them is a particularly natural goal. Indeed, to the best of our knowledge, no natural test class currently admits a PRG with optimal parameters; linear tests may be the most compelling candidate.

The goal of this paper is to advance this research program. While we do achieve a modest quantitative improvement over Ta-Shma's result, we view this as secondary. Our main technical contribution is to eliminate almost entirely one of the two barriers identified by Ta-Shma to further progress in his framework. We believe this is the harder and more substantial of the two, and we focus our efforts on overcoming it.

This barrier is related to the cost of maintaining s registers. As noted, it inflates the sample–space size by a factor exponential in s, which we then balance against the bias, whose magnitude decreases with the walk length. To obtain a better construction within this framework, one must extract more "juice" from the same register budget – namely, enable substantially longer walks without increasing s.

Our key insight is that, if done carefully, registers can be *reused*. Once a register has been used it becomes "compromised" – an adversary may correlated with it – so it no longer hides the walk by itself. Nevertheless, with a carefully designed reuse schedule, we show that s registers suffice to perform *exponentially* long walks – in particular, of length $2^{s/2}$. This dramatically improves the tradeoff in that aspect.

Using this idea, we obtain an improvement over Ta-Shma's construction of small-bias sets, even without addressing the second barrier.

▶ **Theorem 1.** *For every $n, \varepsilon > 0$ there exists an explicit ε-biased set $S \subseteq \{0,1\}^n$ of size*

$$\frac{n}{\varepsilon^{2+\alpha}} \qquad \textit{where} \qquad \alpha = \widetilde{O}\left(\left(\frac{1}{\log(1/\varepsilon)}\right)^{1/2}\right).$$

The register reuse pattern underlying our construction is a variant of the flip–index sequence of the binary-reflected Gray code on s coordinates. These Gray-style sequences are defined recursively and yield patterns such as 1 2 1 for two registers and 1 2 1 3 1 2 1 for three registers, and so on. We show that, in the absence of the interleaved operator Π, these sequences enable long walks on the wide–replacement product in the standard expander setting. Although the unmodified Gray pattern does not suffice in the context of bias reduction, once Π is present, we introduce a tailored variant that does. For the proof of Theorem 1, we refer the reader to the full version of the paper.

1.3.1 Beyond the current barriers for small-bias sets

The second barrier in Ta-Shma's framework is the inherent factor-two loss in the degree/expansion tradeoff, $\omega \approx \frac{2}{\sqrt{d}}$. To push further, we must bypass this bottleneck. In our setting, the gadget graph required for the bias–reduction step has additional structure that in fact precludes Ramanujan optimality, so instead of the constant 2 we suffer a super-constant loss. At any rate, we would like to avoid this factor altogether.

This may be attainable by switching to a non-backtracking variant of our procedure – or, essentially equivalently, by using a directed gadget. The main challenge is to integrate such a modification with the machinery developed here. An alternative is to employ *rotating expanders* [6], which keep us within the undirected realm. Although each individual expander carries a factor-2 (or higher) penalty, suitable compositions avoid exponential blow-up and incur only a linear loss. Here too, the main challenge is to integrate our framework with rotating expanders; we leave this to future work.

1.4 Related work

Ta-Shma's construction [18] has received significant attention over the past decade. Here we briefly review the works most relevant to ours.

Blanc and Doron [4] proposed an approach to constructing improved small-bias sets based on hypergraphs. They showed that a result quantitatively similar to Theorem 1 would follow from the existence of a hypergraph with certain parameters. While a random hypergraph satisfies these requirements, no explicit construction is currently known. Hypergraphs with even better parameters, if they exist, could lead to further improvements. The authors also obtained the same parameters as ours, although one component of their construction relies on random objects (they also achieve efficient decoding).

As for decoding, no decoding algorithm was provided in [18]. This was later remedied by Jeronimo, Quintana, Srivastava, and Tulsiani [11, 12], who showed that a slight variant of Ta-Shma's codes is efficiently decodable – indeed, in almost linear time in n. Blanc and Doron [4] obtained an improved construction with a better rate. We leave the decodability of our codes to future work.

An interesting extension of [18] was obtained by Jeronimo, Mittal, Roy, and Wigderson, who generalized Ta-Shma's bias-amplification analysis from scalars to matrices of arbitrary dimension [10]. We leave extending our construction to this matrix-valued setting to future work.

1.5 Organization

The wide–replacement product, and in particular its use for bias reduction, is technically involved. Nevertheless, it admits a clean and intuitive interpretation. We therefore begin by developing this picture before introducing the additional machinery required for our results. In Section 2, we give a detailed exposition of Ta-Shma's bias-reduction framework. Along the way, we preview several ideas that are later used in our improved construction and develop a diagrammatic viewpoint that will guide the analysis. Unfortunately, this part – although we believe it substantially clarifies the paper and places our results and techniques in proper context – is necessarily lengthy, and thus prevents us from providing a full proof overview within the first ten pages. Readers already familiar with Ta-Shma's construction may therefore skip directly to Section 3. This section relies largely on standard notation and should be mostly self-contained, though some consultation of the earlier sections may still be required.

Readers who prefer to jump straight to the formal sections may safely skip these two sections. Section 4 formally develops the framework for working with the wide–replacement product. Section 5 introduces the random-walk patterns we employ, and Section 6 analyzes their interaction with the interleaved operators.

2 An Exposition to Ta-Shma's Construction

2.1 Rozenman–Wigderson bias reduction

As discussed in Section 1, Rozenman–Wigderson use expanders to derandomize the $S + S$ construction, where S is an ε_0-biased set. Let $G = (V, E)$ be a d-regular graph on n vertices. Let W denote the normalized adjacency (random-walk) matrix of G, with eigenvalues $1 = \omega_1 \geq \omega_2 \geq \cdots \geq \omega_n$. We say that G has ω-*spectral expansion*, where $\omega \triangleq \max\{|\omega_2|, |\omega_n|\}$. The key property for our purposes is that W fixes the normalized all-ones vector $\mathbf{1}$ (since it is an eigenvector with eigenvalue $\omega_1 = 1$) and contracts every vector in its orthogonal complement $\mathbf{1}^\perp$ by at least a factor ω. That is, for every $v \in \mathbf{1}^\perp$, $\|Wv\| \leq \omega \|v\|$. Moreover, $\mathbf{1}^\perp$ is an invariant subspace of W: $Wv \in \mathbf{1}^\perp$ for all $v \in \mathbf{1}^\perp$.

The Rozenman–Wigderson bias–reduction idea proceeds as follows. Let $S \subseteq \{0,1\}^n$ be an ε_0-biased set, and let $G = (S, E)$ be a d-regular graph on the vertex set S with ω-spectral expansion. That is, we identify elements of S with the vertices of G. In this setup, the naïve reduction considered above takes all pairs of vertices, i.e., sample two independent uniform vertices of G and take their bitwise XOR. Rozenman–Wigderson's idea is to *derandomize* this step by restricting to the edge set, namely, the small-bias set is given by

$$S +_G S \triangleq \{\, s_u + s_v \;:\; uv \in E \,\}.$$

This new set has size

$$m \triangleq |S +_G S| = |E| = \frac{nd}{2}.$$

As a warm-up, let us analyze this construction and prove

▶ **Lemma 2.** *$S +_G S$ is an ε-biased set for $\varepsilon = \varepsilon_0^2 + \omega$.*

Proof. Fix a nonzero $\tau \in \{0,1\}^n$. For each $s \in S$, label the vertex s by $(-1)^{\langle \tau, s\rangle}$. To evaluate the bias, we need to compare the fraction of edges in G whose endpoints have the same label (either $++$ or $--$) with the fraction whose endpoints have different labels ($+-$ or $-+$). To this end, define $\pi : S \to \mathbb{R}$ by $\pi(s) = \frac{1}{\sqrt{|S|}}(-1)^{\langle \tau, s\rangle}$ and view π as a column vector indexed by S. The bias of the new set is

$$\left|\mathbb{E}_{uv\sim E}[\pi(u)\pi(v)]\right| \;=\; \left|\pi^\top W \pi\right|,$$

where the expectation is over a uniformly random edge of G.

Recall that by our convention, $\mathbf{1} \in \mathbb{R}^{|S|}$ denotes the normalized all-ones vector (so $\|\mathbf{1}\| = 1$). Then the (signed) bias of the test τ is

$$\varepsilon_\tau \;\triangleq\; \mathbb{E}_{s\in S}\left[(-1)^{\langle \tau, s\rangle}\right] \;=\; \langle \pi, \mathbf{1}\rangle,$$

which we know satisfies $|\varepsilon_\tau| \leq \varepsilon_0$. Accordingly, we decompose

$$\pi \;=\; \varepsilon_\tau \mathbf{1} \;+\; \pi^\perp, \qquad \text{with } \pi^\perp \in \mathbf{1}^\perp.$$

As $\mathbf{1}$ and $\mathbf{1}^\perp$ are invariant spaces of W, and since $W\mathbf{1} = \mathbf{1}$, we have that

$$\begin{aligned}\pi^\top W\pi &= \left(\varepsilon_\tau \mathbf{1} + \pi^\perp\right)^\top W \left(\varepsilon_\tau \mathbf{1} + \pi^\perp\right) \\ &= \varepsilon_\tau^2 + \left(\pi^\perp\right)^\top W\pi^\perp.\end{aligned}$$

Now, by Cauchy-Schwartz inequality, and since $\|\pi^\perp\| \le 1$,

$$\left|\left(\pi^\perp\right)^\top W\pi^\perp\right| \le \|W\pi^\perp\| \le \omega.$$

Thus, $\left|\pi^\top W\pi\right| \le \varepsilon_\tau^2 + \omega \le \varepsilon_0^2 + \omega$. ◀

Going forward, we fix a nonzero linear test $\tau \in \{0,1\}^n$ once and for all and, for simplicity, write ε_0 (the starting small-bias guarantee) instead of ε_τ. Having fixed τ, it is convenient to introduce the diagonal matrix Π whose diagonal encodes π. Thus, the bias[2] of the resulting set (with respect to τ), $\pi^\top W\pi$, can be written as

$$\mathbf{1}^\top \Pi W \Pi\, \mathbf{1}. \tag{2}$$

In what follows – especially as the constructions become more involved – it will be helpful to reason with diagrams that track the "flow" of a vector, decomposed into several subspaces, as operators act on it. For example, in the proof of Lemma 2, in light of Equation (2), we wish to understand how the component in span $\{\mathbf{1}\}$ – and in particular its norm – evolves as we apply the three operators Π, W, Π, and then project back onto that space.

This behavior is illustrated in Figure 1. The top row, labeled $\|$, represents vectors in span $\{\mathbf{1}\}$; the bottom row, labeled $\perp$, represents vectors in $\mathbf{1}^\perp$. The columns correspond to the operators being applied and should be read right-to-left, matching the order of operator multiplication when acting on column vectors. From each node there are (typically) two outgoing edges, indicating to which subspace the operator maps the vector. For example, if only one horizontal edge appears, the corresponding subspace is invariant under that operator. A number on an edge denotes the contraction factor guaranteed on the projection to the target subspace. An unlabeled edge indicates no quantitative bound; however, all operators we consider have operator norm at most 1, so the norm does not increase.

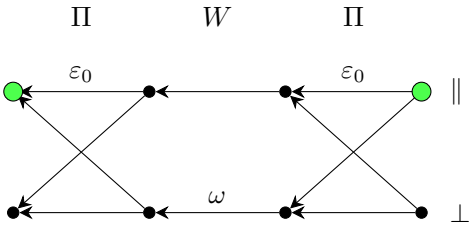


Figure 1 Illustration of the Rozenman–Wigderson bias reduction via expander-edge sampling.

Using this diagram, one can *see* the $\varepsilon_0^2 + \omega$ bound proved in Lemma 2 by summing – over all paths from the right green circle to the left green circle – the product of the edge weights along each path.

Let us instantiate the Rozenman–Wigderson bias reduction procedure with a concrete expander. In particular, take a Ramanujan graph, which satisfies $\omega \approx \frac{2}{\sqrt{d}}$, and – by Lemma 2 – choose the spectral parameter so that $\omega = \varepsilon_0^2$. A quick calculation then shows that, with

[2] In this section we do not distinguish between bias and signed bias.

this choice, the procedure transforms[3] an ε-biased set of size n/ε^c, for some constant c, into an ε-biased set of size

$$O\left(n \cdot \left(\frac{1}{\varepsilon}\right)^{4+c/2}\right).$$

Thus, iterating the procedure yields ε-biased sets with size $n/\varepsilon^{4+o_\varepsilon(1)}$.

2.1.1 Taking longer walks

A perhaps even more natural approach is to take longer walks on the expander graph discussed above. In particular, one might take two steps on G (which corresponds to taking the parity of three vertices rather than two). Extending Equation (2) to this longer walk, one can show that the bias of the resulting set is

$$\mathbf{1}^\top \Pi W \Pi W \Pi \mathbf{1}. \tag{3}$$

Unfortunately, as depicted in Figure 2, the second G step is effectively "wasted". This is because we have very little control over the operator Π (which is induced by the linear test). In particular, Π may map a vector in $\mathbf{1}^\perp$ largely into $\mathbf{1}^\parallel$ (see the bold red edge in the diagram), rendering the subsequent G step moot.

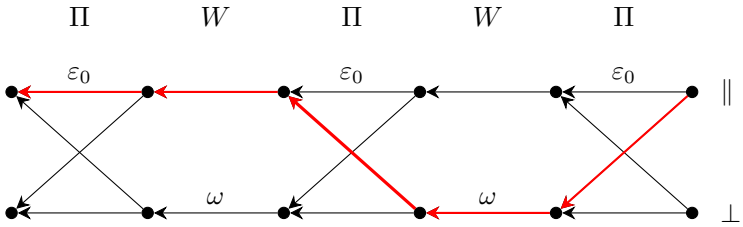


Figure 2 Illustration of how the second step is "wasted" by the interleaved operator Π.

A quick calculation shows that, when we take a long walk, losing every other G-step – together with the unavoidable quadratic dependence of the spectral expansion parameter ω on the degree d of the expander – again yields a sample space of size about n/ε^4.

2.1.2 Ta-Shma's approach

The difficulty with taking long walks, as discussed in Section 2.1.1, is that the "adversarial" operator Π can interfere with the random walk on G in an uncontrolled way – mixing the subspaces span($\mathbf{1}$) and $\mathbf{1}^\perp$ and thereby disrupting the walk on G. As hinted in the introduction, the key insight underlying Ta-Shma's improvement is to restrict, at each step, the set of neighbors a vertex of G may transition to in a manner that is "hidden" from the adversary. In this way, the effect of Π on the walk remains largely under our control.

Ta-Shma implements this approach using the s-wide replacement product, introduced by Ben-Aroya and Ta-Shma [2] for the purpose of constructing near-Ramanujan graphs. Before proceeding, we recall the definitions of the replacement product and the wide-replacement product.

[3] More precisely, as is standard in asymptotic analysis, we apply the procedure to a family of small-bias sets that (ideally) contains a set for every choice of n and ε.

2.2 The s-wide replacement product

2.2.1 The replacement product

Let G be an undirected d-regular graph with spectral expansion ω_G. For each vertex $v \in V(G)$ we arbitrarily label its d neighbors by indices in $[d] = \{1, \ldots, d\}$. Thus, for $i \in [d]$ we may speak of the i-th neighbor of v. If this neighbor is u, then, in general, u may regard v as its j-th neighbor for some $j \in [d]$; the labelings need not be consistent, so j need not equal i. We refer to this labeling as the *rotation map* of G.

Let H be an undirected graph on d vertices with vertex set $V(H)$, which we identify with $[d]$ and use interchangeably. Define the *replacement product* graph $G \textcircled{r} H$ on the vertex set $V(G) \times V(H)$ as follows. For each $v \in V(G)$, place a copy of H on the vertices $\{(v, i) : i \in V(H)\}$; we call this set the *cloud* of v. In addition, connect (v, i) to (u, j) whenever u is the i-th neighbor of v and v is the j-th neighbor of u. Edges within a cloud are called *intra-cloud* edges, and edges connecting distinct clouds are called *inter-cloud* edges.

The replacement product serves as a basic building block for more elaborate constructions, such as the zig–zag product [16], a key tool in constructing expander graphs. To obtain expanders with improved parameters, Ben–Aroya and Ta–Shma introduced a variant of the replacement product, which we now describe.

2.2.2 The s-wide replacement product

Let $s \geq 1$ be a parameter, called the *width*. For the purposes of bias reduction, it is helpful to view the s-wide replacement product not primarily as a graph construction but as a particular random walk on a graph.

Let G be a graph as in Section 2.2.1. The first departure from the standard replacement product is that we now take a graph H on d^s vertices. As in prior applications for constructing expanders we would like both G and H to be good expanders. However, for the purposes of the s-wide replacement product, we also require additional structure on H.[4] More precisely, let Λ be an arbitrary abelian group of order d, and identify $[d]$ with Λ. We then take H to be a Cayley graph on the product group Λ^s with respect to some fixed symmetric generating set C of size c.

The vertex set of the graph on which we are going to take a walk is $V(G) \times V(H)$. As in the discussion of the replacement product, we view the set $\{(v, h) : h \in V(H)\}$ as the *cloud of* v. We now describe length-s walks.

Fix a starting vertex $(v_0, h_0) \in V(G) \times V(H)$ and a sequence of indices $i_1, \ldots, i_s \in [c]$ (each i_t selects a neighbor in H). For $t = 1, \ldots, s$ perform:

1. Let h_t be the i_t-th neighbor of h_{t-1} in H.
2. Write $h_t = (h_t^{(1)}, \ldots, h_t^{(s)}) \in \Lambda^s$, and let v_t be the $h_t^{(t)}$-neighbor of v_{t-1} in G (recall that we identify Λ with $[d]$).

After phase t the state is (v_t, h_t). In particular, phase t uses the t-th coordinate, or "register", of h_t to determine the step on G.

To implement Ta–Shma's idea of "hiding" the random walk from the adversarial operator Π, we label *clouds* – rather than individual vertices – by ± 1, using the signs induced by an ε_0-biased set with respect to a fixed linear test. In other words, we identify the initial small-bias set with the set of clouds $V(G)$ (rather than with the full vertex set $V(G) \times V(H)$). Equivalently, for every $v \in V(G)$ we assign the same label to all vertices (v, h) with $h \in V(H)$.

[4] Interestingly, the original version of [2] analyzed a random H; in a later version, imposing structure on H enabled a cleaner analysis.

The combinatorial idea above takes a particularly clean form in the operator viewpoint. The three operators we now introduce act on vectors indexed by $V(G) \times V(H)$ (equivalently, on $\mathbb{R}^{V(G)\times V(H)}$).

- **Adversarial operator.** Because it is indifferent to the structure *within* each cloud, it factors as

  $$\Pi \otimes I_{|V(H)|},$$

  where we write $\otimes$ for the Kronecker product, and $I_{|V(H)|}$ is the identity on the $|V(H)|$ coordinates inside a cloud. For notational convenience, we overwrite the symbol and denote this lifted operator again by Π.
- **Intra-cloud operator.** The operator H acts identically within each cloud, independent of which cloud we are in. Hence it factors as

  $$\widetilde{H} \triangleq I_{|V(G)|} \otimes H.$$

  We emphasize that Π and $\widetilde{H}$ act on different registers. This is the operator-level manifestation of the combinatorial "hiding" idea: H dictates the within-cloud motion, while the adversary Π can only affect the other register.
- **Inter-cloud operators.** Finally, we have the operators corresponding to G. They encode the rotation map of G and keep track of the step index $i \in [s]$, which determines which register is used. Using the product structure $V(H) = \Lambda^s$, we view the H part as s registers. For each $i \in [s]$, define $\dot{G}_i$ to encode the rotation map while consulting only the i-th register of the H-register. Concretely, $\dot{G}_i$ is a permutation that (i) moves in the $V(G)$ coordinate according to the value stored in the i-th register, and (ii) updates that i-th register to the return label specified by the rotation map; all other H-registers remain unchanged.

2.3 Analysis

We now sketch the analysis of Ta-Shma's construction via the s-wide replacement product, starting with the special case $s = 2$. Our state space consists of labelings of the graph with vertex set $V(G) \times V(H)$, so we consider the function space

$$X \triangleq \mathbb{R}^{V(G)\times V(H)}.$$

Since there are $s = 2$ H-registers, together with the G-register we have three registers in total. Decomposing each register into $\mathrm{Span}(\mathbf{1})$ and its orthogonal complement $\mathbf{1}^\perp$, we obtain a direct-sum decomposition of X into eight subspaces:

$$X \;=\; V_\emptyset^{\|} \oplus V_1^{\|} \oplus V_2^{\|} \oplus V_{12}^{\|} \oplus V_\emptyset^{\perp} \oplus V_1^{\perp} \oplus V_2^{\perp} \oplus V_{12}^{\perp}. \tag{4}$$

By our notation, the superscript indicates whether the G-register lies in $\mathrm{Span}(\mathbf{1})$ ("$\|$") or in $\mathbf{1}^\perp$ ("$\perp$"). The subscript records which of the two H-registers (if any) lie in $\mathbf{1}^\perp$. Here $\mathbf{1}$ denotes the all-ones vector in the relevant register.

In parallel with Equation (3), the bias of the linear test encoded by Π is now

$$\mathbf{1}^\top \Pi \underbrace{\dot{G}_2 \widetilde{H} \Pi}_{M_2} \underbrace{\dot{G}_1 \widetilde{H} \Pi}_{M_1} \mathbf{1}. \tag{5}$$

We trace the effect of the operators from right to left (see Figure 3). First, write $\mathbf{1}$ as $\mathbf{1}\otimes\mathbf{1}\otimes\mathbf{1}$, ordered by the three registers – G first, followed by the two H registers – so $\mathbf{1} \in V_\emptyset^{\|}$. Now:

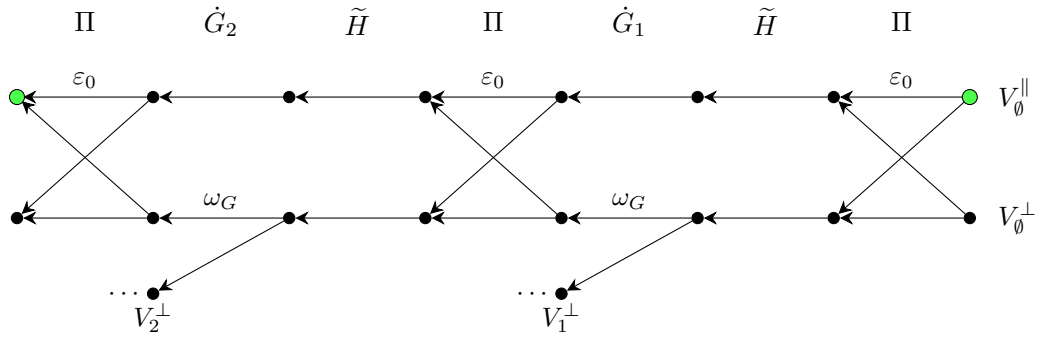


Figure 3 Bias evolution in the $s = 2$ wide–replacement product.

1. The first operator Π acts only on the G register and decomposes the vector into two components: an ε_0 fraction that remains in $V_\emptyset^{\|}$, and a dominant component that moves to $V_\emptyset^{\perp}$ (i.e., the G coordinate lies in $\mathbf{1}^{\perp}$).
2. The next operator $\widetilde{H}$ is agnostic to the G register and acts only within the H registers. Since the H part is $\mathbf{1}$ (as a vector of length $|V(H)|$), $\widetilde{H}$ leaves it unchanged.
3. The third operator applied is $\dot{G}_1$, which leaves the $V_\emptyset^{\|}$ component unchanged. For the $V_\emptyset^{\perp}$ component, only an ω_G fraction of the mass remains in this space; the rest is moved to $V_1^{\perp}$.
   The key point is that register 1 of H will not be used again. Therefore, from this point on in this branch, every $\widetilde{H}$ step acts on a vector in $\mathbf{1}^{\perp}$ (as a vector of length $|V(H)|$), so that component shrinks by a factor of ω_H at each step (in short example, in the remaining second step). We therefore stop tracking it in the diagram, and consider it as a "win".
   We emphasize at this point that our construction departs significantly from Ta-Shma's: we reuse the same register more than once – indeed, *exponentially* many times (in the number of registers). This reuse of "compromised" registers follows a specific pattern and requires a more delicate analysis.
4. Now comes the second action of Π. We've already analyzed what it is doing on the component in $V_\emptyset^{\|}$ (but for an ε_0 mass, it moves to the $V_\emptyset^{\perp}$). However, on the $V_\emptyset^{\perp}$ component we have no control over Π's adversarial behavior in the sense that it can move all mass back to $V_\emptyset^{\|}$. What we can say - and this is the key property - is that Π does not touch the H component at all (and so we are still within the $\emptyset$ subscript).
5. As in the previous application of $\widetilde{H}$, here too $\widetilde{H}$ ignores the G register and acts only on the H registers. Since we consider only branches in which the H part is $\mathbf{1}$, $\widetilde{H}$ leaves it unchanged.
6. As before, $\dot{G}_2$ leaves the $V_\emptyset^{\|}$ component unchanged and shrinks the $V_\emptyset^{\perp}$ component by a factor of ω_G, moving most of it to $V_2^{\|}$.
7. Finally, the operator Π behaves as before: it shrinks the $V_\emptyset^{\|}$ component by ε_0 and may move $V_\emptyset^{\perp}$ back to $V_\emptyset^{\|}$.

Thus, we can bound the bias in Equation (5) by summing over all branches that start and end in $V_\emptyset^{\|}$ (the green nodes in Figure 3), yielding

$$\varepsilon_0^3 + 2\varepsilon_0\,\omega_G + \omega_G^2.$$

Setting $\omega_G = \varepsilon_0$ simplifies this to $O(\varepsilon_0^2)$.

The bound obtained above may not seem like much: indeed, a 2-wide replacement product gives no advantage over plain random walks, as discussed in Section 2.1.1 for bias reduction. The advantage of the s-wide replacement product kicks in only as we increase s. The key feature is that once we move beyond the two subspaces $V_\emptyset^{\|}$ and $V_\emptyset^{\perp}$, we obtain a shrink by ω_H at every $\widetilde{H}$ step. However, to analyze Ta-Shma's construction we also have to consider bilinear forms beyond the specific quadratic form of Equation (5), corresponding to start/end states other than $V_\emptyset^{\|}$. In those cases additional contributions involving ω_H arise: Loosly speaking, once the adversary leaves the $\emptyset$ spaces (that is, $V_\emptyset^{\|}$, $V_\emptyset^{\perp}$) at every step the vector contracts by a factor of ω_H. When returning to the the $\emptyset$ spaces, in every other step a contraction of ω_G or ε_0 occurs.

Formalizing this argument, one can show that there exist universal constants a, b – in particular, independent of s – such that, setting $\omega_G = \varepsilon_0 = \omega_H^a$, we have

$$\left\|M_s M_{s-1} \cdots M_1\right\| \;=\; O(\omega_H)^{s-b}.$$

For ease of exposition we will assume here $a = b = 1$, as the exact constants are unimportant. In particular, we will simply use the bound ω_H^{s-1}.

With this in hand, consider a bias-reduction procedure that takes a length-st random walk, cycling through the s registers periodically, for a parameter t that we will optimize shortly. For simplicity, assume both H and G are Ramanujan graphs. This is, strictly speaking, an idealized assumption – H must satisfy structural constraints that preclude Ramanujan optimality – but it streamlines the exposition and affects the parameters only mildly at this point (though it will become the main remaining challenge for achieving optimal small-bias sets beyond our work). Under this assumption, the bias after the procedure is bounded by

$$\varepsilon \;=\; \omega_H^{(s-1)t}.$$

The size of the resulting ε-biased set is

$$|V(G)| \cdot |V(H)| \cdot c^{st} = O\left(\frac{n}{\varepsilon_0^4} \cdot d^s \cdot c^{st}\right). \tag{6}$$

Substituting ω_H for ε_0 and ω_G, and using the relations between the expander degrees and spectral expansions, we obtain (after a slight oversimplification that does not affect the asymptotics) a set of size

$$\frac{n}{\varepsilon^2} \cdot 2^{st} \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{1}{s}+\frac{1}{t}}.$$

The factor multiplying the target size $\frac{n}{\varepsilon^2}$ is symmetric in s and t, so to minimize it we take $s = t$. The minimum is then attained at $s = \left(\log \frac{1}{\varepsilon}\right)^{1/3}$, resulting in an ε-biased set of size

$$O\left(\frac{n}{\varepsilon^2} \cdot 2^{\left(\log \frac{1}{\varepsilon}\right)^{2/3}}\right).$$

This matches Equation (1) up to a $2^{\mathrm{poly}(\log\log \frac{1}{\varepsilon})}$ factor, stemming from the fact that we cannot take H to be Ramanujan.

3 Proof Overview

As described in Section 2, Ta-Shma's approach to bias reduction follows the Rozenman–Wigderson random-walk framework, but chooses its steps using s registers that are hidden from the adversary (the linear test we seek to fool), encoded by the operator Π. When optimizing the parameters, we found in Section 2.3 that it is best to consider a walk of length s^2, analyze it in blocks of length s, and then apply the submultiplicativity of the spectral norm to bound the entire product. Specifically, we partition the walk into s phases, each of length s; in each phase we "restart" the analysis and treat the s registers as fresh (unused in that phase).

Our main technical contribution is to show that these s registers can be reused – even after they have been "compromised" – provided this is done carefully. To illustrate our approach, we first analyze the simplest setting with no adversary; that is, the operator Π does not appear in the product. While the s-wide replacement product as used for bias reduction is not meaningful under this assumption, it is interesting in its own right to reuse registers in the s-wide replacement product for expander constructions – indeed, this was Ben-Aroya and Ta-Shma's original motivation for introducing the wide product [2]. However, we begin by ignoring Π mainly for expository clarity: it is easier to present some of the core ideas in this simplified setting and then introduce the additional ideas needed to handle the presence of Π. In the next section we first consider this setting in the special case $s = 2$.

3.1 The non-adversarial setting

Since Π is absent, we consider products of the form $M_3M_2M_1$ with $M_i = \dot{G}_i\widetilde{H}$. We wish to understand whether registers can be reused; for example, we may study expressions such as $M_1M_2M_1$, in which the first register is reused. To this end, we take a closer look at how the operators $\widetilde{H}$, $\dot{G}_1$, and $\dot{G}_2$ act on the subspaces $V_S^{\perp}$ and $V_S^{\parallel}$ for all $S \subseteq \{1, 2\}$. Figure 4 summarizes the behavior of $\widetilde{H}$ (we prove this formally in Section 4.2). In particular, whenever $S \neq \emptyset$, we pick up a factor of ω_H. Hence, from the adversary's perspective, the goal is to "revisit" $V_{\emptyset}^{\perp}$ or $V_{\emptyset}^{\parallel}$ as often as possible so as to avoid this contraction to the bias. Moreover, note that $\widetilde{H}$ does not move us between components.

As we prove in Section 4.4, the operator $\dot{G}_1$ acts as follows (see Figure 5). On the parallel subspaces, $\dot{G}_1$ leaves us in place. On the perpendicular subspaces, $\dot{G}_1$ moves only along the axis corresponding to index 1: it maps between $V_{\emptyset}^{\perp} \leftrightarrow V_{\{1\}}^{\perp}$ and $V_{\{2\}}^{\perp} \leftrightarrow V_{\{1,2\}}^{\perp}$. Moreover, if we start in $V_{\emptyset}^{\perp}$ or $V_{\{2\}}^{\perp}$, at most an ω_G fraction of the norm remains in the original subspace. The operator $\dot{G}_2$ acts in the same manner with 1 replaced by 2.

Think of S as the set of *compromised* registers, i.e., registers with which we cannot perform a random-walk step on G. A register in state $\mathbf{1}$ simulates a truly random step on G, while a register in $\mathbf{1}^{\perp}$ yields an invalid step. However, if we are in the G-parallel register $V_S^{\parallel}$, the choice of step is immaterial.

3.1.1 The pebble game

A useful way to think about reusing registers is as a pebble game on the s-dimensional hypercube whose vertices are indexed by subsets $S \subseteq [s] \triangleq \{1, \ldots, s\}$. We will start with the simplest case $s = 2$; the extension to general s is discussed afterwards.

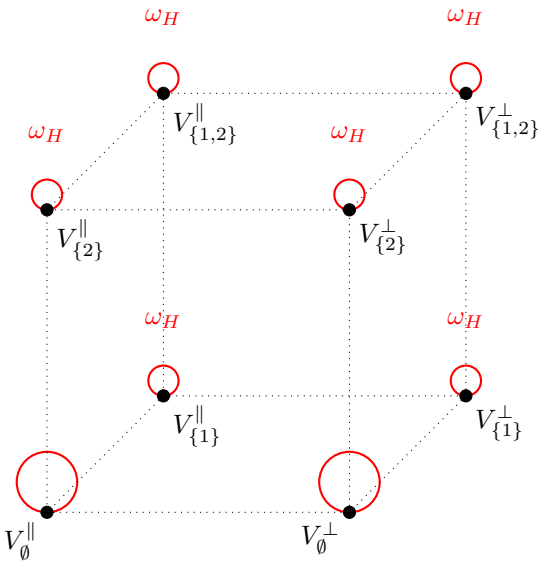


Figure 4 Illustration of the action of $\widetilde{H}$.

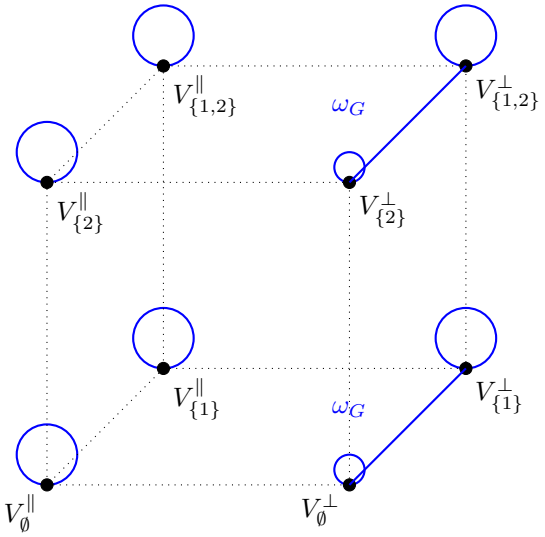


Figure 5 An illustration of the action of $\dot{G}_1$.

At the start of the game, the opponent[5] places a single pebble at $\emptyset$. At each step the adversary moves the pebble according to rules specified below. In some cases the pebble may duplicate; when multiple pebbles are present, all pebbles move simultaneously according to the same rules. Whenever several pebbles are places at the same node, they merge to one. The game lasts t steps, and the adversary's objective is to have a pebble at $\emptyset$ for as many steps as possible.

[5] In this game we call the player we face the *opponent*; we reserve the term *adversary* for the entity that fixes Π.

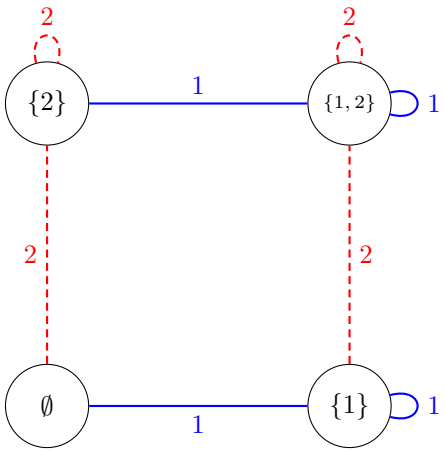
The opponent plays against us – the bias-reduction designer – who chooses a sequence in $\sigma \in [s]^t$ (this corresponds to the order in which registers are used) so as to minimize the opponent's gain. The rules of the game are as follows: Let $\sigma_k \in [s]$ and consider a pebble placed at node S:

- If $\sigma_k \in S$, the pebble stays put but *duplicates*: the copy is placed at $S \setminus \{\sigma_k\}$.
- If $\sigma_k \notin S$, the pebble moves to $S \cup \{\sigma_k\}$.

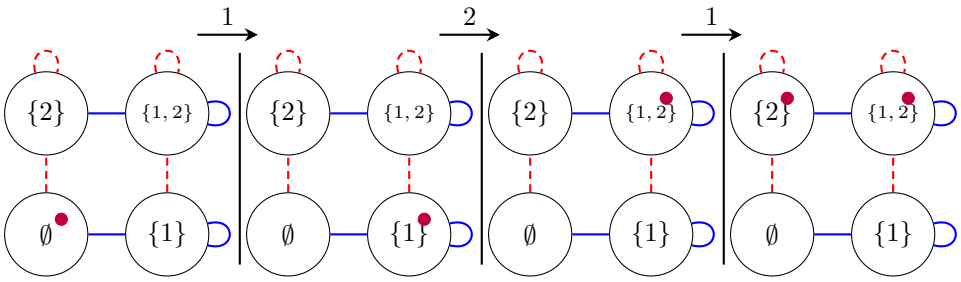
These rules describe the dynamics in the G-perpendicular registers. The parallel registers will come into play only after we reintroduce the adversary, so for now we restrict attention to the G-perpendicular component. We initialize with mass concentrated at $V_{\emptyset}^{\perp}$, thereby giving the opponent a free win in the first step.

The first rule reflects that when we use a compromised register at time k (i.e., $\sigma_k \in S$), we have no control over the corresponding G operator, and the mass may either remain in $V_S^{\perp}$ or move to $V_{S\setminus\{\sigma_k\}}^{\perp}$. The second rule corresponds to using a "good" register; in this case, most of the mass moves to $V_{S\cup\{\sigma_k\}}^{\perp}$. Here we ignore the ω_G-contracted mass that stays in place. In the full analysis we account for it as well.

The following figure captures the rules of the game in a diagram.

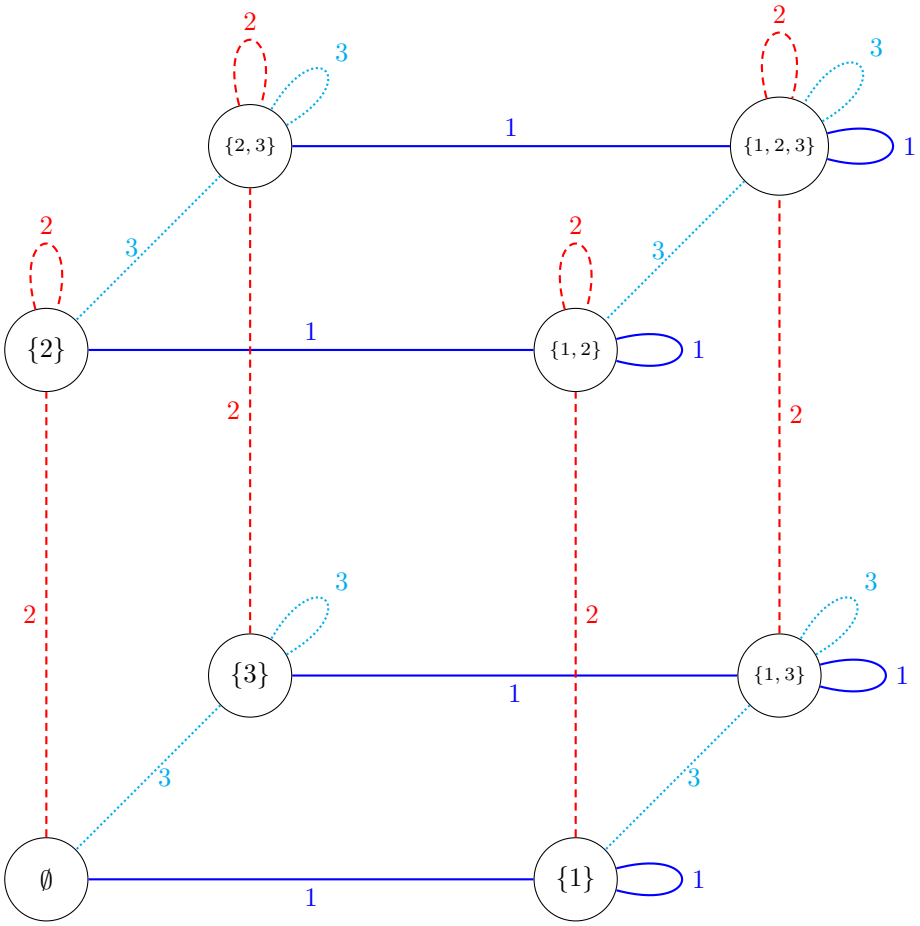


Let us play! In this game we are playing a length 3 game corresponding to the string 1 2 1 capturing the reuse of register 1. As the following diagram depicts, in this game we are able to make this re-usage work. Indeed, the opponent was not able to place a pebble at $\emptyset$ except for the initial position.



The careful reader may notice that the string 1 2 2 2 2 $\cdots$ also works. This is a consequence of an expository simplification: we analyzed only a particular quadratic form, whereas the full argument requires handling additional bilinear forms. In the language of the pebble game, the correct objective is to win on every interval, not just on the entire word.

The following diagram illustrates the extension to width $s = 3$. From the figure we see that playing the register sequence 1 2 1 3 1 2 1 again prevents the opponent from placing a pebble at $\emptyset$ after the first step. In this three-register case, register 1 is used every other step.



This generalizes to s registers via the flip-index sequence of the binary-reflected Gray code on s coordinates, defined recursively by

$$F_1 = (1), \qquad F_s = (F_{s-1},\ s,\ F_{s-1}),$$

where juxtaposition denotes concatenation. Note that the sequence has length $2^s - 1$.

Using this sequence yields an ω_H-factor contraction of the bias at every step except the first. Thus far, we have ignored the possibility that the opponent "cheats," i.e., deviates from the rules by paying a cost ω_G. Nevertheless, we show that the sequence is robust to such strategies. In particular, when ω_G is only modestly smaller than ω_H – say $\omega_G = \omega_H^c$ for some universal constant c independent of s – the opponent has no incentive to cheat. To be somewhat more precise, the opponent may be in a superposition of "cheating" and "not cheating"; the claim above should be understood as conveying the intended intuition.

3.2 The return of the adversary

We now reintroduce the adversary. The game is now played on two s-dimensional hypercubes: the G-perpendicular cube and the G-parallel cube. The simplified game above was restricted to the perpendicular cube – whose node labels we now denote by $S^\perp$ – and its rules remain

unchanged. The rules on the parallel cube are simpler and can be deduced from Figures 4 and 5: every move is a self-loop, i.e., the pebble stays put. The opponent's objective is to maximize the number of time steps with a pebble at $\emptyset^{\perp}$ or $\emptyset^{\parallel}$. As before the game begins with a single pebble at $\emptyset^{\perp}$.

The key new feature is that after each instruction σ_k (the register chosen at time k), a Π-step occurs that may move pebbles between the two cubes, thereby giving the opponent additional freedom. To determine the rules for the Π-steps, we first analyze the Π operator. Figure 6 illustrates its behavior. Starting from $V_S^{\parallel}$, Π keeps only an ε-fraction of the norm in $V_S^{\parallel}$. The self-loops and the horizontal arrow are the only outgoing edges from $V_S^{\parallel}$ and $V_S^{\perp}$. For simplicity of exposition, we ignore the ε-contracting steps (much as we ignored the ω_G-contracting steps earlier). With this, the Π-steps follow these rules:

- A pebble at $S^{\parallel}$ moves to $S^{\perp}$.
- A pebble at $S^{\perp}$ remains in place and duplicates, creating a new pebble at $S^{\parallel}$.

We summarize all rules in Figure 7.

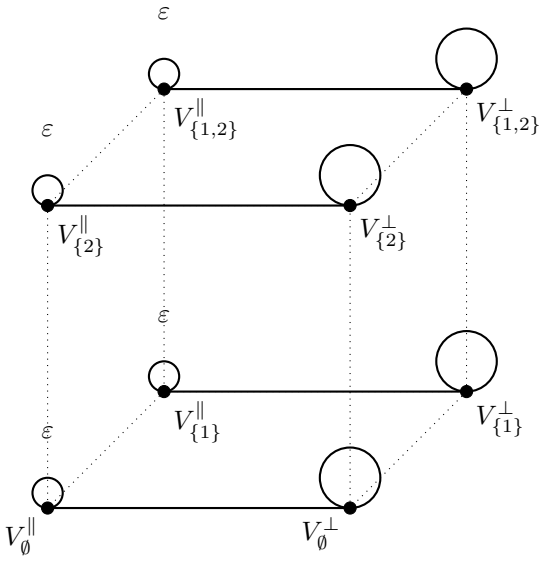


Figure 6 The action of Π on the spaces $V_S^{\perp}$, $V_S^{\parallel}$.

Let us now play the same 1 2 1 string as before, but interleaved with Π-steps: $\Pi\,1\,\Pi\,2\,\Pi\,1$. The reader may verify that, because of the Π-steps, the number of time steps with a pebble in the $\emptyset$-states is now two, compared with only one in the setting without Π-steps. While the difference between one and two may not seem compelling, the failure of the Gray-code strategy becomes much more evident as s increases.

It turns out that with two registers there is no way to reuse registers. The Gray-code string 1 2 1 3 1 2 1 (with interleaved Π-steps, omitted for readability) from before also fails for three registers. However, the following string is a good play: 1 2 3 1 – using it, the opponent cannot return to the $\emptyset$ states. With five registers, one can achieve the same effect with the string

1 2 3 1 4 5 1 2 3 1.

In general, we find a string on s registers of length $\approx 2^{s/2}$ that guarantees the opponent never revisits the $\emptyset$ states. The factor-2 loss in the exponent relative to the no-Π game is immaterial.

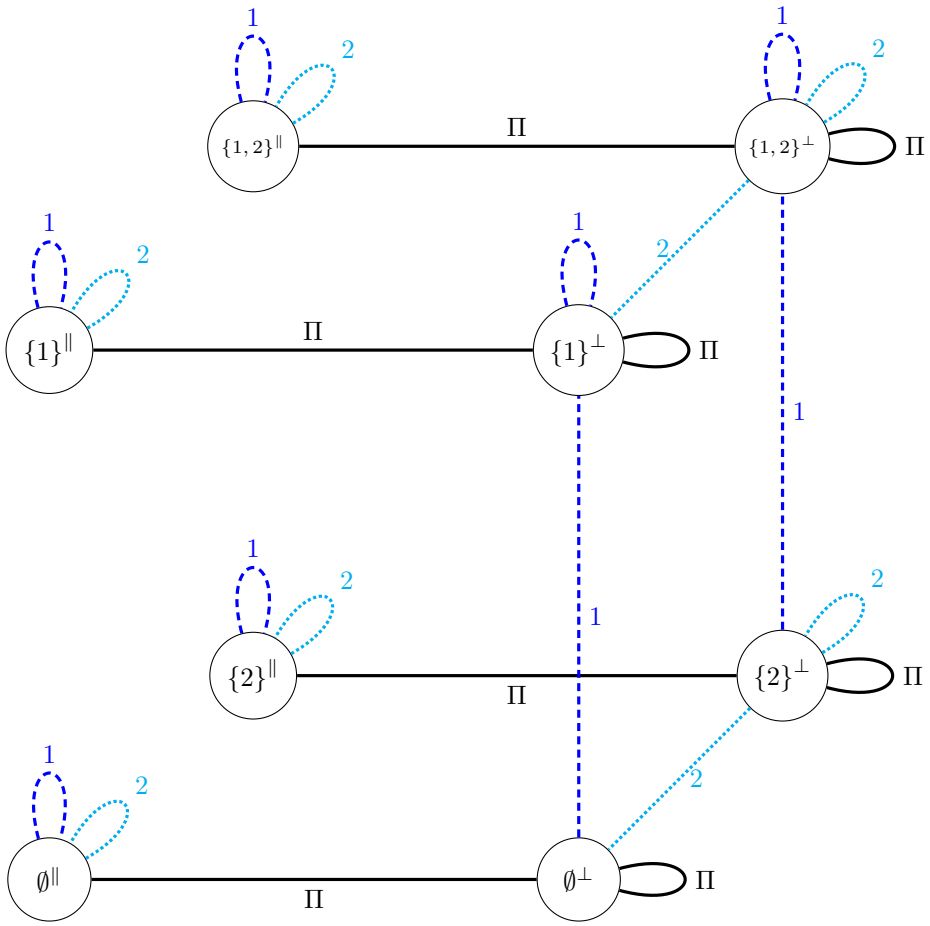


Figure 7 A complete illustration of the two-register pebble game with Π.

We emphasize that matters become more involved once we reintroduce the "cheats" the opponent can perform via ω_G-steps and ε_0-steps (i.e., the Π-steps). Nevertheless, by choosing ε_0 in the same manner as ω_G, we prove the opponent has no incentive to cheat. We summarize the entire two-register game in Section 4.4.

3.3 Setting parameters

We now instantiate our new bias-reduction procedure with s registers (hence walks of length $\ell = 2^{s/2}$), starting from an ε_0-biased set of size $O(n/\varepsilon_0^4)$ constructed in the previous sections. The size of the resulting ε-biased set is

$$|V(G)| \cdot |V(H)| \cdot c^{2^{s/2}} = O\left(\frac{n}{\varepsilon_0^4} \cdot d^s \cdot c^{2^{s/2}}\right). \tag{7}$$

Compare this with Equation (6), where the walk length is exponentially shorter than in our setting.

Substituting ω_H for ε_0 and ω_G for simplicity, as chosen in the previous sections, and using the relations between expander degrees and spectral parameters, we obtain (after a slight oversimplification that does not affect the asymptotics)

$$\frac{n}{\varepsilon^2} \cdot 2^{\ell} \cdot \left(\frac{1}{\varepsilon}\right)^{\frac{\log \ell}{\ell}}. \tag{8}$$

The loss term is minimized at $\ell \approx \sqrt{\log \frac{1}{\varepsilon}}$, yielding an improvement over Ta-Shma's construction – namely, an ε-biased set of size

$$\frac{n}{\varepsilon^{2+\alpha}}, \qquad \alpha = \widetilde{O}\left(\left(\log \tfrac{1}{\varepsilon}\right)^{-1/2}\right),$$

as stated in Theorem 1.

3.4 Toward optimal small-bias sets

The reason we do not obtain optimal small-bias sets is the need to balance the two terms in Equation (8) – namely, 2^{ℓ}, which increases with ℓ, and $(1/\varepsilon)^{\frac{\log \ell}{\ell}}$, which decreases. The first "exponential in ℓ" term does not in fact have base 2 but rather some constant (and in fact, slightly super-constant); we chose to present it as if it is 2 in this informal overview for simplicity. By inspection, this constant comes from two sources:

- The inherent factor of 2 in the relation between spectral expansion and degree, $\omega \approx \frac{2}{\sqrt{d}}$. This was also one of the bottlenecks noted by Ta-Shma in improving his construction.
- A deliberately lossy step in our bias bound: for simplicity we sacrificed absolute constants, which are not currently the bottleneck.

Moving forward, we will need to eliminate both sources of loss. We believe the first is more inherent, while the second is largely technical. As noted, the factor of 2 is not even precise, since the expander H needed for our bias-reduction procedure has additional structure that prevents it from being Ramanujan. Even setting this aside, we still wish to avoid this factor of 2. We believe this may be achievable by switching to a non-backtracking version of our procedure or – essentially equivalently – by using a directed graph H. The main difficulty is to combine such a modification with the machinery developed in this work.

An alternative is to use rotating expanders [6], which keep us in the realm of undirected graphs. Although each individual expander carries a factor-2 penalty, suitable compositions avoid an exponential blow-up and incur only a linear loss. Here too, the challenge is to integrate our framework with rotating expanders.

4 A Formal Framework for the Wide-Replacement Product

In this section, we formalize the mechanism underlying the wide-replacement product. Specifically, we present a variant of the original construction of Ben-Aroya and Ta-Shma [2]. Our presentation is tailored to our needs: it is more abstract, yet it imposes additional constraints. Moreover, we incorporate into the product an auxiliary operator Π. This operator is introduced precisely to enable an analysis of the wide-replacement product in the setting of small-bias sets, following [18].

Ingredients

The components of the wide-replacement product are as follows and will remain fixed throughout this section:

- Let Λ be a finite group of size D.
- For a parameter s, dubbed the *width* parameter, let H be the Cayley graph over the group Λ^s corresponding to a generating set $C \subseteq \Lambda^s$; that is, $H = \mathrm{Cay}(\Lambda^s, C)$. We denote the spectral expansion of H by ω_H.

- Let G be a D-regular graph on n vertices, and write ω_G for its spectral expansion. We represent G via a rotation map

  $$\mathrm{Rot} : V(G) \times \Lambda \to V(G) \times \Lambda.$$

  We assume that the rotation map is locally invertible, i.e., there exists a function $\phi : [D] \to [D]$ such that for every $g \in V(G)$ and $x \in [D]$,

  $$\mathrm{Rot}(g, x) = (h, \phi(x))$$

  for some $h = h(g, x) \in V(G)$. In words, the second component of the output depends only on the second component of the input. We slightly abuse notation by sometimes considering only the projection of the rotation map onto the first register.

The space X

We denote the $\mathbb{R}$-vector space of all real functions on Λ by V, that is,

$$V \triangleq \{f : \Lambda \to \mathbb{R}\}.$$

We similarly define

$$W \triangleq \{f : V(G) \to \mathbb{R}\},$$

where $V(G)$ is the vertex set of G. Define

$$X \triangleq V^{\otimes s} \otimes W. \tag{9}$$

We write $V^{\otimes s}$ as $V_1 \otimes \cdots \otimes V_s$, where each $V_i \cong V$ as an $\mathbb{R}$-vector space. We refer to V_i as the *i-th register* and to the entire factor $V^{\otimes s}$ as the *H-register*. We refer to the factor W in the tensor product $X = V^{\otimes s} \otimes W$ as the *G-register*.

Decomposing the space X

For an $\mathbb{R}$-vector space U, there is a natural decomposition into parallel and perpendicular subspaces:

$$U^{\parallel} \triangleq \mathrm{span}\{\mathbf{1}\}, \qquad U^{\perp} \triangleq \left(U^{\parallel}\right)^{\perp},$$

where $\mathbf{1}$ is the all-ones vector, thus $U = U^{\parallel} \oplus U^{\perp}$. Note that we slightly abuse notation: here $U^{\perp}$ denotes the orthogonal complement of $U^{\parallel}$ *inside* U (with respect to the standard inner product), rather than the orthogonal complement of U in some ambient space; this convention is convenient and will not cause ambiguity.

Recall that the tensor product distributes over direct sums, namely

$$(U_1 \oplus U_2) \otimes U_3 \cong (U_1 \otimes U_3) \oplus (U_2 \otimes U_3).$$

Applying this repeatedly yields a decomposition of the space X from Equation (9) into 2^{s+1} subspaces: in each coordinate we choose either the parallel or the perpendicular component (for each of $V_1, \ldots, V_s$ and for the G-register). For a set $S \subseteq [s]$, we denote

$$V_S \triangleq \bigotimes_{i=1}^{s} U_i, \qquad \text{where} \qquad U_i \triangleq \begin{cases} V_i^{\perp}, & i \in S, \\ V_i^{\parallel}, & i \notin S. \end{cases}$$

Equivalently (as a shorthand), we may write

$$V_S = \bigotimes_{i \in S} V_i^{\perp} \otimes \bigotimes_{i \notin S} V_i^{\parallel},$$

with the tacit convention that tensor factors are ordered by increasing index. Thus, for example, if $S = \{1, 3\} \subseteq [3]$, then the display above yields $V_S = V_1^{\perp} \otimes V_3^{\perp} \otimes V_2^{\parallel}$, which we identify with $V_1^{\perp} \otimes V_2^{\parallel} \otimes V_3^{\perp}$ by reordering the factors.

Note that V_S itself is not a subspace of X, as it does not include the W factor. To obtain subspaces of X, define

$$V_S^{\parallel} \triangleq V_S \otimes W^{\parallel}, \qquad V_S^{\perp} \triangleq V_S \otimes W^{\perp}, \qquad V_S^{\perp\!\!\!\perp} \triangleq V_S \otimes W. \tag{10}$$

Thus, the superscript in $V_S^{\parallel}$ and $V_S^{\perp}$ indicates whether we take the parallel or the perpendicular component of the G-register (equivalently, of the W-factor). We also abuse notation again and denote

$$V^{\parallel} \triangleq V^{\otimes s} \otimes W^{\parallel}, \qquad V^{\perp} \triangleq V^{\otimes s} \otimes W^{\perp}, \tag{11}$$

so that, with our notation,

$$X = V^{\otimes s} \otimes W = V^{\parallel} \oplus V^{\perp}.$$

For each one of the subspaces of X defined above, there is a corresponding orthogonal projection operator, which we denote by P, with the same subscript and superscript (e.g. $P_S^{\parallel}$ projects to $V_S^{\parallel}$.)

4.1 The Linear Operators $\dot{G}$, $\widetilde{H}$ and Π

We now describe the linear operators acting on the space X defined in Equation (9) that are used in the s-wide product. Specifically, we consider the operators $\widetilde{H}, \dot{G}_1, \ldots, \dot{G}_s$ which induce walks over the vertices $V(G) \times V(H)$, and the noise operator Π.

Informally, the operator $\widetilde{H}$ corresponds to taking a local H-step within each g-cloud. Formally,

▶ **Definition 3.** *The operator $\widetilde{H}$ is the linear operator on X defined as follows. For $f \in X$, $x \in \Lambda^s$ and $g \in V(G)$,*

$$\left(\widetilde{H} f\right)_g(x) = \mathbb{E}_{c \in C}\left[\, f_g(x + c)\,\right],$$

where the expectation is uniform over the generating set C of H.

▶ **Definition 4.** *For $i \in [s]$, we define the linear operator $\dot{G}_i$ on X as follows. For $f \in X$, $x = (x_1, \ldots, x_s) \in \Lambda^s$ and $g \in V(G)$, let*

$$\mathrm{Rot}\,(g, x_i) = (h, \phi(x_i)).$$

Then

$$\left(\dot{G}_i f\right)_g(x) \triangleq f_h\big(\phi(x_i),\, x_{-i}\big).$$

In words: take a G-step from g according to the label x_i, updating the G-register and the i-th register, while keeping the other registers unchanged.

Last is Π which we refer to as the *noise operator* which acts on entire clouds.

▶ **Definition 5.** *Let $\pi : V(G) \to \mathbb{R}$ be a function. Define the linear operator Π on X as follows. For $f \in X$, $x \in \Lambda^s$ and $g \in V(G)$,*

$$(\Pi f)_g(x) = \pi(g)\, f_g(x).$$

In words, Π multiplies each g-cloud by the scalar $\pi(g)$.

Throughout this section, we denote the *bias* of π by

$$\varepsilon \triangleq \left|\mathbb{E}_{g \in V(G)}\left[\pi(g)\right]\right| \tag{12}$$

As the notation suggests, we will typically assume that this bias is small, and certainly smaller than 1.

4.2 Action of $\widetilde{H}$

▶ **Lemma 6.** *For every $S \subseteq [s]$ the following hold*

1. *$\widetilde{H}$ preserves V_S locally. That is,*

$$\widetilde{H} V_S^{\perp\!\!\!\perp} \subseteq V_S^{\perp\!\!\!\perp},$$

Moreover,

   a. *$\widetilde{H}$ acts trivially on $V_\emptyset^{\perp\!\!\!\perp}$.*
   b. *For $S \neq \emptyset$, for every $f \in V_S^{\perp\!\!\!\perp}$,*

$$\left\|\widetilde{H} f\right\| \leq \omega_H \left\|f\right\|. \tag{13}$$

2. *$\widetilde{H}$ preserves $V^\perp$ and $V^\parallel$ globally. That is,*

$$\widetilde{H} V^\parallel \subseteq V^\parallel, \qquad \widetilde{H} V^\perp \subseteq V^\perp.$$

▶ **Corollary 7.** *For every $S \subseteq [s]$, $\widetilde{H}$ preserves $V_S^\parallel$ and $V_S^\perp$.*

We refer the reader to Figure 4 for an illustration of the action of $\widetilde{H}$, and to the full version of the paper for the proof of the lemma.

4.3 Action of Π

▶ **Lemma 8.** *The operator Π satisfies the following properties:*

1. *For every $f \in V^\parallel$,*

$$\left\|P^\parallel \Pi f\right\| \leq \varepsilon \left\|f\right\|.$$

   In particular, when ε is small, Π maps $V^\parallel$ mostly into $V^\perp$.
2. *For every $S \subseteq [s]$, Π locally preserves V_S, i.e.,*

$$\Pi V_S^{\perp\!\!\!\perp} \subseteq V_S^{\perp\!\!\!\perp}.$$

We refer the reader to Figure 6 for an illustration of the action of Π, and to the full version of the paper for the proof of the lemma.

4.4 Action of the $\dot{G}_i$-s

Informally, for each $i \in [s]$, the operator $\dot{G}_i$ may shift us along the i-th coordinate. It is therefore convenient to define the combined subspace

$$V_{S\pm i} \triangleq V_{S\setminus\{i\}} \oplus V_{S\cup\{i\}} \subseteq V^{\otimes s}.$$

We extend the subspace $V_{S\pm i}$ to subspaces of X.

$$V_{S\pm i}^{\shortparallel} \triangleq V_{S\pm i}\otimes W, \qquad V_{S\pm i}^{\perp} \triangleq V_{S\pm i}\otimes W^{\perp} \subseteq X.$$

▶ **Lemma 9.** *The operator $\dot{G}_i$ satisfies the following invariance properties:*

1. *For every $S \subseteq [s]$, $V_S^{\shortparallel}$ is $\dot{G}_i$ invariant; equivalently,*

$$\dot{G}_i V_S^{\shortparallel} \subseteq V_S^{\shortparallel}.$$

2. *For every $S \subseteq [s]$, $\dot{G}_i$ leaves $V_{S\pm i}^{\perp}$ invariant; equivalently,*

$$\dot{G}_i V_{S\pm i}^{\perp} \subseteq V_{S\pm i}^{\perp}.$$

3. *For every $S \subseteq [s]$ with $i \notin S$, $\dot{G}_i$ leaves only an ω_G-fraction of the norm in $V_S^{\perp}$. That is, for every $f \in V_S^{\perp}$,*

$$\left\| P_S^{\perp} \dot{G}_i f \right\| \leq \omega_G \left\| f \right\|.$$

We refer the reader to Figure 5 for an illustration of the action of the $\dot{G}_i$-s, and to the full version of the paper for the proof of the lemma.

We end this section with a Section 4.4 that illustrates the action of all the operators we discussed in this section.

4.5 Walks based on the wide product

Given $s \in \mathbb{N}$ and two graphs G and H with suitable parameters, we defined and analyzed the operators $\widetilde{H}$ and $\dot{G}_i$, for $i \in [s]$. These operators naturally refer to random walks on the set of vertices $V(G) \times V(H)$. Generally, we take a zig-zag walk. First an $\widetilde{H}$-step, then a $\dot{G}$-step, then an $\widetilde{H}$-step again. However, we still have the freedom to choose *which* $\dot{G}_i$ to use, as we have s distinct options in every step. For a word $W = W_{|W|} \cdots W_1$ over the alphabet $[s]$, we define the W-walk of H and G to be the graph with the random walk matrix

$$G \textcircled{w} H \triangleq \dot{G}_{W_{|W|}} \widetilde{H} \cdots \dot{G}_{W_1} \widetilde{H}$$

To construct small biased sets out of wide random walks, we identify the vertices of G with vectors from a small biased set, $S_{\texttt{outer}}$, so that each vertex $g \in V(G)$ correspond to a vector in $S_{\texttt{outer}}$. Similarly to the work of [18], the elements of our new biased set $S_{\texttt{new}}$ are defined to be sums of elements from $S_{\texttt{outer}}$, which correspond to the W-walks over the wide product of G and H. For each such a walk, we add a new vector to our new biased set $S_{\texttt{new}}$. The vector is the sum of the vectors from $S_{\texttt{outer}}$ that we encountered on the way. Explicitly, for a walk

$$(g_0, h_0) \to_{\widetilde{H}} (g_0, h_1) \to_{\dot{G}} (g_1, h_2) \to_{\widetilde{H}} \cdots \to_{\dot{G}} \left(g_{|W|}, h_{2|W|}\right),$$

we add the vector $\sum_{i=1}^{|W|} g_i$ to $S_{\texttt{new}}$. Given a linear test $\pi : \mathbb{Z}_2^k \to \{\pm 1\}$, the vertex g gets the value $\pi(g) = (-1)^{\langle g,\pi\rangle}$. This induces the operator Π.

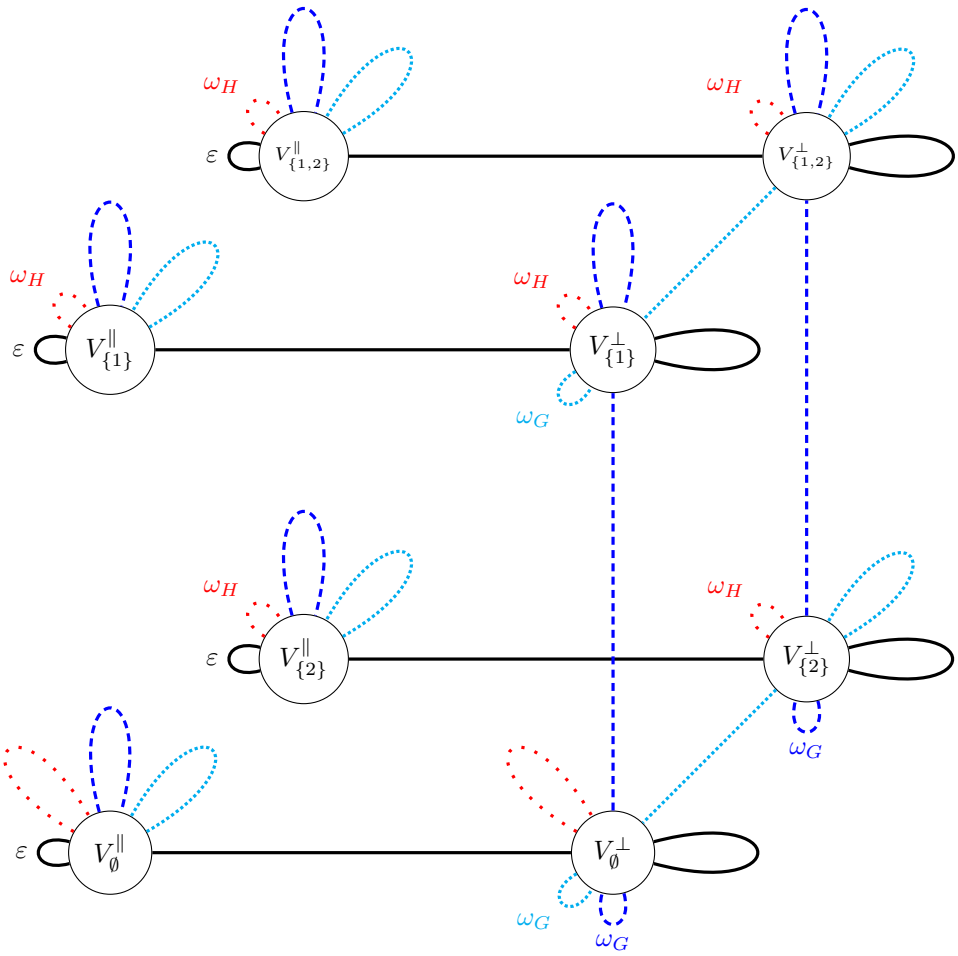


Figure 8 Illustration of the action of all operators: Π (black solid), $\widetilde{H}$ (red loosely dotted), $\dot{G}_1$ (blue dashed), and $\dot{G}_2$ (light-blue densely dotted).

Throughout, we will have a correspondence between words and linear operators. For a letter $\sigma \in [s]$, we define the corresponding operator that helps us track the bias as

$$\widehat{\sigma} = \Pi \dot{G}_\sigma \widetilde{H}.$$

We extend this definition to words. For a word $W = \sigma_{|W|} \cdots \sigma_1$,

$$\widehat{W} = \widehat{\sigma}_{|W|} \cdots \widehat{\sigma}_1,$$

As proved in [18], the bias of the new set on the test π can be expressed algebraically:

▷ Claim 10. For every word W over the alphabet $[s]$, the bias of the new set $S_{\texttt{new}}$ is algebraically expressed as

$$\mathsf{Bias}(S_{\texttt{new}}) = \mathbf{1}^t \Pi \dot{G}_{W_{|W|}} \widetilde{H} \cdots \Pi \dot{G}_{W_2} \widetilde{H} \Pi \dot{G}_{W_1} \widetilde{H} \mathbf{1} = \mathbf{1}^t \widehat{W} \mathbf{1}. \tag{14}$$

5 Conflict-Free Words

In this brief section, we present a construction of the Gray–code–style string used in our register-reuse pattern.

▶ **Definition 11.** *A nonempty word $w \in \Sigma^\star$* has a conflict *if every letter that appears in w appears at least twice (equivalently, no letter appears exactly once). A word w is* conflict-free *if there exists a letter $\sigma \in \Sigma$ that appears in w exactly once, or if it is the empty word.*

▶ **Definition 12.** *A word $w \in \Sigma^\star$ is* double conflict-free *if either $|w| < 2$, or there exist two consecutive positions whose letters each appear exactly once in w.*

▶ **Definition 13.** *A word $w \in \Sigma^\star$ is* (double) conflict-free on intervals *if every contiguous subword (interval) of w is (double) conflict-free.*

▶ **Lemma 14.** *For every $s \in \mathbb{N}$ there is an explicit word w_s of length $2^s - 1$ over the alphabet $\{\sigma_1, \ldots, \sigma_s\}$ that is conflict-free on intervals.*

▶ Remark 15. The length $2^s - 1$ in Lemma 14 is optimal: every word of length 2^s over an alphabet of size s has a conflict.

▶ **Lemma 16.** *For every $s \in \mathbb{N}$ there exists an explicit word w_{2s-1}, over the alphabet $\{\sigma_1, \ldots, \sigma_{2s-1}\}$, of length $3 \cdot 2^{s-1} - 2$ that is double conflict-free on intervals.*

▶ Remark 17. The bound $3 \cdot 2^{s-1} - 2$ in Lemma 16 is tight: every word of length $3 \cdot 2^{s-1} - 1$ over an alphabet of size $2s - 1$ fails to be double conflict-free on intervals.

The explicit constructions of these words and the proofs of the lemmas in this section are straightforward and are deferred to the full version of the paper.

6 Tracking the Norm

The following lemma exploits the key property of *double-conflict-free* words: every word contains a pair of consecutive letters that appear *exactly once* in the word. To analyze such a unique consecutive pair, we "zoom in" the corresponding pair of operators. Let σ and σ' such letters, and consider the product of their operators,

$$\widehat{\sigma}\widehat{\sigma'} = \Pi \dot{G}_\sigma \widetilde{H} \cdot \Pi \dot{G}_{\sigma'} \widetilde{H}.$$

Informally, because the letters σ and σ' appear only once, we can assume that before and after this block the process acts on subspaces that are independent of these coordinates. This observation motivates the definition of the space of functions that are independent of the registers σ and σ'.

$$V^{\perp\!\!\!\perp}_{-\{\sigma,\sigma'\}} = \bigoplus_{S:\sigma,\sigma' \notin S} V^{\perp\!\!\!\perp}_S, \tag{15}$$

and of the corresponding projection operator to this space, $P^{\perp\!\!\!\perp}_{-\{\sigma,\sigma'\}}$.

▶ **Lemma 18.** *For every $\sigma, \sigma' \in [s]$,*

$$\left\| P^{\perp\!\!\!\perp}_{-\{\sigma,\sigma'\}} \dot{G}_\sigma \widetilde{H} \Pi \dot{G}_{\sigma'} P^{\perp\!\!\!\perp}_{-\{\sigma,\sigma'\}} \right\| \leq 2\omega_G + \varepsilon.$$

Lemma 18 is the main technical ingredient in the following theorem, which is the key to proving our main result.

▶ **Theorem 19.** *Assume that* $2\omega_G + \varepsilon \leq \frac{32}{20}\omega_H^5$. *Let* w *be a word over the alphabet* $[s]$, *which is double-conflict-free on intervals. Then, the norm of its corresponding operator* $W = \widehat{w}$ *is bounded by*

$$\left\|\widehat{W}\right\| \leq 10 \cdot (2\omega_H)^{|W|-2}$$

Using Theorem 19 and Claim 10 we can easily bound the bias of the resulting small-bias set that we construct. We refer the reader to the full version of the paper for the proof of the lemma and the theorem, as well as the way to combine them in order to prove our main theorem.

References

1 N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Addendum to: "Simple constructions of almost k-wise independent random variables". *Random Structures Algorithms*, 4(1):119–120, 1993. doi:10.1002/rsa.3240040109.

2 Avraham Ben-Aroya and Amnon Ta-Shma. A combinatorial construction of almost-Ramanujan graphs using the zig-zag product. *SIAM J. Comput.*, 40(2):267–290, 2011. doi:10.1137/080732651.

3 Avraham Ben-Aroya and Amnon Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. *Theory Comput.*, 9:253–272, 2013. doi:10.4086/toc.2013.v009a005.

4 Guy Blanc and Dean Doron. New near-linear time decodable codes closer to the GV bound. In *37th Computational Complexity Conference*, volume 234 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 10, 40. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/lipics.ccc.2022.10.

5 Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. *Theory Comput.*, 15:Paper No. 10, 26, 2019. doi:10.4086/toc.2019.v015a010.

6 Gil Cohen and Gal Maor. Random walks on rotating expanders. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 971–984, 2023. doi:10.1145/3564246.3585133.

7 Gil Cohen, Ran Raz, and Gil Segev. Non-malleable extractors with short seeds and applications to privacy amplification. In *2012 IEEE 27th Conference on Computational Complexity—CCC 2012*, pages 298–308. IEEE Computer Soc., Los Alamitos, CA, 2012. doi:10.1109/CCC.2012.21.

8 Michael A. Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In *59th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2018*, pages 946–955. IEEE Computer Soc., Los Alamitos, CA, 2018. doi:10.1109/FOCS.2018.00093.

9 Edgar N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31(3):504–522, May 1952. URL: https://vtda.org/pubs/BSTJ/vol31-1952/articles/bstj31-3-504.pdf.

10 Fernando Granha Jeronimo, Tushant Mittal, Sourya Roy, and Avi Wigderson. Almost-Ramanujan Expanders From Arbitrary Expanders via Operator Amplification. *SIAM J. Comput.*, 54(5):FOCS22–120–FOCS22–158, 2025. doi:10.1137/22M1538739.

11 Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. Unique decoding of explicit ε-balanced codes near the Gilbert-Varshamov bound. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pages 434–445. IEEE Computer Soc., Los Alamitos, CA, [2020] ©2020. doi:10.1109/FOCS46700.2020.00048.

12 Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. Near-linear time decoding of Ta-Shma's codes via splittable regularity. In *STOC '21—Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1527–1536. ACM, New York, [2021] ©2021.

13 Robert J. McEliece, Eugene R. Rodemich, Jr. Rumsey, Howard, and Lloyd R. Welch. New upper bounds on the rate of a code via the delsarte–macwilliams inequalities. *IEEE Transactions on Information Theory*, 23(2):157–166, March 1977. doi:10.1109/TIT.1977.1055688.

14 Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. doi:10.1137/0222053.

15 Ran Raz. Extractors with weak random seeds. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20. ACM, New York, 2005. doi:10.1145/1060590.1060593.

16 Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 3–13. IEEE, 2000. doi:10.1109/SFCS.2000.892006.

17 Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Approximation, randomization and combinatorial optimization*, volume 3624 of *Lecture Notes in Comput. Sci.*, pages 436–447. Springer, Berlin, 2005. doi:10.1007/11538462_37.

18 Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 238–251, 2017. doi:10.1145/3055399.3055408.

19 R. R. Varshamov. Estimate of the number of signals in error-correcting codes. *Doklady Akademii Nauk SSSR*, 117:739–741, 1957. English translation reprinted in I. F. Blake (ed.), *Algebraic Coding Theory: History and Development*, Dowden, Hutchinson & Ross, 1973, pp. 68–71.

20 Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. *Comput. Complexity*, 18(2):209–217, 2009. doi:10.1007/s00037-009-0273-5.

Multilinear Algebraic Branching Programs and the Min-Partition Rank Method

Théo Borém Fabris
University of Copenhagen, Denmark

Nutan Limaye
IT University of Copenhagen, Denmark

Srikanth Srinivasan
University of Copenhagen, Denmark

Amir Yehudayoff
University of Copenhagen, Denmark
Technion-ITT, Haifa, Israel

Abstract

It is a long-standing open problem in algebraic complexity to prove lower bounds against multilinear algebraic branching programs (mlABPs), however the best lower bounds are still quadratic (Alon, Kumar and Volk (*Combinatorica* 2020)). At the same time, it remains a possibility that the "min-partition rank" method introduced by Raz (*Theory Comput.* 2006), which is used to prove all known multilinear lower bounds, can also be used to prove superpolynomial lower bounds on the size of mlABPs. In this paper, we analyze the potential of the min-partition rank method to prove lower bounds on the size of mlABPs, and show the following results:

1. We relate this method to a purely combinatorial question regarding the minimum size of set systems whose chains satisfy a discrepancy condition. In the case of set-multilinear ABPs, this combinatorial measure *characterizes* the best lower bound that can be achieved via the min-partition rank method.
2. We prove a non-trivial upper bound on the size of a set system satisfying this combinatorial property. Together with our construction of full-rank mlABPs from set systems, this recovers a superpolynomial separation between mlABPs and multilinear formulas (Dvir, Malod, Perifel and Yehudayoff (*STOC 2012*)) via a conceptually different proof.
3. The property we study extends combinatorial notions of "balancing sets" considered in previous works, for which near-tight bounds are known via intervals families. We show that any intervals set system is very far from satisfying our property. This showcases how our methods capture combinatorial structures that evade previous techniques, and also allows us to improve and generalize known lower bounds for sum of ordered set-multilinear ABPs (Chatterjee, Kush, Saraf, Shpilka (*CCC 2024*)).

These results build a bridge between algebraic complexity theory and the behavior of random walks. Our upper bound uses the fact that, with noticeable probability, a random walk of length n on the integers returns to its starting point at least once every $n/\log n$ steps (Csáki, Erdős, and Révész (*PTRF 1985*)), while, for our lower bound, we prove that two independent random walks are "far" from each other in *discrete Fréchet distance*.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Algebraic branching programs, Multilinear computations, Rank methods

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.22

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2026/001/`

Funding *Théo Borém Fabris*: Supported by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA).
Nutan Limaye: Supported by Independent Research Fund Denmark (grant agreement No. 10.46540/3103-00116B) and Basic Algorithms Research Copenhagen (BARC), funded by VILLUM Foundation Grant 54451.



© Théo Borém Fabris, Nutan Limaye, Srikanth Srinivasan, and Amir Yehudayoff;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 22; pp. 22:1–22:20



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Srikanth Srinivasan: Supported by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA).
Amir Yehudayoff: Supported by DNRF Chair grant no. 20231101-28697.

1 Introduction

Basic background. Algebraic complexity theory is the study of algebraic problems in models such as algebraic formulas, branching programs and circuits. Analogous to the Boolean circuit classes $\mathbf{NC}^1 \subseteq \mathbf{NL} \subseteq \mathbf{NC}^2$, in algebraic complexity we study complexity classes $\mathbf{VF} \subseteq \mathbf{VBP} \subseteq \mathbf{VP}$, and a central goal is to prove lower bounds against all classes in this chain.

Notable success has been achieved in the *multilinear* setting where we restrict each of the algebraic models to compute only multilinear polynomials (in a syntactic[1] way). In particular, this gives us a multilinear hierarchy $\mathbf{mlVF} \subseteq \mathbf{mlVBP} \subseteq \mathbf{mlVP}$. Well-known results of Raz [15, 14] showed superpolynomial lower bounds on the size of multilinear formulas computing some explicit polynomials in $\mathbf{mlVP}$, thus separating the first and third levels of this hierarchy. Subsequently, Dvir, Malod, Perifel and Yehudayoff [7] proved a stronger separation between the first and second levels of this hierarchy. There has been a large volume of follow-up work in algebraic complexity on extending the known techniques to different algebraic models (e.g. [16, 18, 19, 5, 2, 10, 11, 4]), and they also found applications in quantum computation [1, 19] and proof complexity [17, 9]. Despite this interest, we have not yet been able to extend our current lower bounds techniques to prove superpolynomial lower bounds for higher classes in the multilinear hierarchy such as $\mathbf{mlVBP}$. The main motivation for this work is to understand whether it is possible to extend known lower bound techniques from the multilinear setting to obtain superpolynomial lower bounds against $\mathbf{mlVBP}$.

Multilinear algebraic branching programs (mlABPs). The mlABP computational model is an algebraic analogue of Boolean branching programs, which model space-eficient (Boolean) computation. An mlABP is defined to be a directed acyclic graph G with a source a and sink b. The edges of G are labelled with affine linear polynomials (e.g. $1 - 2x_1 + x_2$). The polynomial computed is the sum, over all a-to-b paths, of the products of the edge-labels occurring along the path. To keep the polynomial multilinear, we impose the condition that the same variable cannot appear in more than one edge label along a path. Multilinear formulas and circuits are defined similarly using algebraic analogues of Boolean formulas and Boolean circuits, but we omit their formal definitions here as they are not crucial in what follows. The central open question we care about is the following:

▶ **Question 1.** Are there explicit n-variate multilinear polynomials P_n that cannot be computed by mlABPs of size $\mathrm{poly}(n)$?

As mentioned above, this question has been resolved positively for multilinear formulas by the work of Raz [15, 14]. For mlABPs and the stronger model of syntactially multilinear circuits, the best lower bounds are nearly quadratic [16, 2].

[1] We can also define a *semantic* notion of multilinearity, which coincides with the syntactic notion for formulas in terms of computational power, but is potentially more powerful in the setting of branching programs and circuits. In this paper, the word "multilinear" will only refer to the syntactic variants.

The lower bound method. Nearly all known lower bounds for multilinear models of computation (including all those mentioned above) can be proved using the "min-partition rank" technique developed by Raz [15][2], which relies on ideas of Nisan [12] and Nisan and Wigderson [13].

Given a multilinear polynomial $P(x_1, \ldots, x_n)$ with n even and a partition $\Pi = (Y, Z)$ of the underlying variable set into two parts, we define the *coefficient matrix* $M_\Pi(P)$ to be a $2^{|Y|} \times 2^{|Z|}$ matrix whose rows and columns are labelled by multilinear monomials m_Y and m_Z respectively. The (m_Y, m_Z)-th entry of $M_\Pi(P)$ is the coefficient of $m_Y m_Z$ in P. We define the *min-partition rank* of P by minimizing the rank of $M_\Pi(P)$ over all *equipartitions* (i.e. $|Y| = |Z| = n/2$):

$$\Gamma(P) := \min_{\substack{\Pi \\ \text{an equipartition}}} \operatorname{rank}(M_\Pi(P)).$$

We call the polynomial P *full-rank* if $\Gamma(P) = 2^{n/2}$. In other words, M_Π has full rank for all Π.

Using this notion of the complexity of a polynomial, Raz established the following.

▶ **Theorem 1** (Raz [15, 14]). *Let $\mathbb{F}$ be any field. Any multilinear formula computing a full-rank multilinear polynomial $P(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ must have size at least $n^{\Omega(\log n)}$.*

Further, if $\mathbb{F}$ is infinite, then there exist multilinear circuits of size $\operatorname{poly}(n)$, and multilinear ABPs and formulas of size at most $n^{O(\log n)}$ computing full-rank polynomials.[3]

Using the above theorem, Raz proved the existence of explicit sequences of n-variate polynomials that have small multilinear circuits but no multilinear formulas of size $\leq n^{o(\log n)}$. A follow-up result of Dvir, Malod, Perifel and Yehudayoff [7] exhibited a similar separation between mlABPs and multilinear formulas, but this was based on the fact that small mlABPs can compute a polynomial that is full-rank with respect to a suitable restricted subfamily of all equipartitions, while small multilinear formulas cannot.

Known lower bounds on mlABP size. While Theorem 1 shows that full-rankness is not sufficient to prove a superpolynomial multilinear circuit lower bound, it could potentially prove such a lower bound on mlABP size. However, the best mlABP lower bound we have (which is also based on this method) is only $\tilde{\Omega}(n^2)$. We review this result [16, 2] below, as it naturally leads to our new approach to mlABP lower bounds.

Assume that a full-rank polynomial $P(x_1, \ldots, x_n)$ has an mlABP of size s and assume for simplicity that the mlABP is "layered" in the sense that vertices are divided into layers and edges go from one layer to the next, and for all vertices v at layer i, the set X_v of variables seen on the paths from a to v is a set of size i. To prove a lower bound for such an mlABP, we note that for any $i \in [n]$, the polynomial P can be decomposed as

$$P = \sum_{v \text{ in layer } i} L_v(X_v) \cdot R_v(X \setminus X_v), \tag{1}$$

[2] It should be noted that *separations* between different multilinear models require us to sometimes use specializations of this technique that yield a more fine-grained understanding. But the basic technique suffices to just prove a lower bound.

[3] Raz's work actually proves this result for a suitable transcendental extension of any field but this can be circumvented over an infinite field by replacing the transcendental elements by random elements from (a large subset of) $\mathbb{F}$. For example, as we did in the proof of Theorem 26.

where L_v corresponds to the sum of all paths from a to v and R_v the sum of all paths from v to b. Proving a lower bound on the number of terms in any such decomposition yields a lower bound on the number of vertices at layer i.

To lower bound the number of terms in the decomposition, we turn to *discrepancy*. We show that if the number of summands on the right hand side of (1) is small, then we can find an equipartition $\Pi = (Y, Z)$ such that each summand has low-rank for this choice of Π. This can be guaranteed by ensuring that, for each v, the induced partition $\Pi' = (Y \cap X_v, Z \cap X_v)$ of X_v is sufficiently "imbalanced". Informally, this ensures that the coefficient matrices of L_v and R_v are both "far from square" matrices and hence that $M_\Pi(f_v \cdot g_v)$ is far from full-rank.

This leads to the combinatorial question of understanding the smallest set system where we have such a discrepancy bound [16]. More precisely, fix an $i \in [n]$. Call a set system $\mathcal{Y} \subseteq \binom{[n]}{i}$ *k-balancing* if for any balanced function $f : [n] \to \{\pm 1\}$ (representing the equipartition), there is a $Y \in \mathcal{Y}$ such that $|\sum_{y \in Y} f(y)| \leq k$. Then, the size of the smallest such k-balancing set system represents the best lower bound we can prove for the number of vertices in layer i of the mlABP. Note that the answer to this is at most n, since the set of all intervals in $\{1, \ldots, n\}$ of size i yields a set system with discrepancy bound 1. This simple example is nearly tight for $i \in [n/4, 3n/4]$ and small k: Alon, Kumar and Volk [2] proved a tight lower bound of $\Omega(n/k)$ on the size of such "balancing set systems" for all k, which leads to a near-linear bound on the number of vertices in each layer and hence a near-quadratic lower bound on mlABP size.[4]

Our new technique. In order to prove a stronger lower bound, we suggest a modification of the decomposition in (1). We observe that such a decomposition of P can be obtained by looking at *any* vertex cut in the mlABP separating the source a from the sink b. As long as we can find a partition Π and a suitable cut C such that for each vertex $v \in C$, the set X_v is imbalanced with respect to Π, we obtain a lower bound. This gives us potentially exponentially many possibilities to work with and hence a much better chance at proving a lower bound against mlABPs.

By graph-theoretic arguments, the non-existence of such a cut is equivalent to the existence of an a-b path such that all the vertices v along the path are balanced. These vertices correspond to a *chain* of subsets of the variable set, which naturally brings us the following definition.

▶ **Definition 2** (Balanced-chain set systems). *Let X be a finite set with n elements and let $\mathcal{X} \subseteq \mathcal{P}(X)$ be a family of subsets of X. Denote by $\mathcal{C}(\mathcal{X})$ the set of* maximal chains *of $\mathcal{P}(X)$ contained in $\mathcal{X}$; i.e., a chain $(C_0, \ldots, C_l)$ is in $\mathcal{C}(\mathcal{X})$ iff $l = n$ and $C_i \in \mathcal{X}$ and $|C_i| = i$ for each $i \in \{0, \ldots, l\}$.*

For a function $f \colon X \to \{-1, 1\}$ which we will call a partition*, and $S \subseteq X$, let $f(S) := \sum_{x \in S} f(x)$. We say that f is* balanced *(or that f is a* balanced partition *of X) if $f(X) = 0$. We define the* chain-balance *of $\mathcal{X}$ with respect to f as*

$$\mathrm{cbal}_{\mathcal{X}}(f) := \min_{(C_0, \ldots, C_n) \in \mathcal{C}(\mathcal{X})} \max_{i \in [n]} |f(C_i)|.$$

We define the chain-balance of $\mathcal{X}$ to be

$$\mathrm{cbal}(\mathcal{X}) := \max_{\text{balanced } f \colon X \to \{-1, 1\}} \mathrm{cbal}_{\mathcal{X}}(f).$$

We say that $\mathcal{X}$ is a k-balanced-chain set system *if* $\mathrm{cbal}(\mathcal{X}) \leq k$.

[4] The general "unlayered" case is more complicated.

An example: intervals. Let $X := [n]$ and consider the set system $\mathcal{I}$ of all intervals contained in $[n]$:

$$\mathcal{I} := \{[i,j] \mid 1 \le i,j \le n\} \subseteq \mathcal{P}(X).$$

As noted above, this set system provides minimum-sized examples of 1-balanced set systems contained in $\binom{[n]}{i}$ for any i. It is therefore natural to understand if they yield balanced-*chain* set systems (the key new word is "chain"). The following balanced partition is an example of a partition such that $\mathcal{I}$ has large chain-balance: let $f : [n] \to \{\pm 1\}$ given by

$$f(j) := \begin{cases} 1 & \text{if } j \in [n/4] \cup [(3n/4)+1, n], \\ -1 & \text{otherwise.} \end{cases}$$

Pictorially, the graph of $i \mapsto f([i])$ comprises a mountain of linear height and a valley of linear depth. It can be shown that any chain in $\mathcal{C}(\mathcal{I})$ contains a set S that is very much not balanced; i.e., $|f(S)| \ge \Omega(n)$. Note that balance $n/2$ is the worst possible. This partition shows that the intervals set system $\mathcal{I}$ is not a balanced-chain set system, but it could be the case that this partition is an untypical example and $\mathcal{I}$ has small balance for most balanced partition. Our Theorem 8 shows that this is not the case: for most balanced partitions, $\mathcal{I}$ has chain-balance at least $n^{\Omega(1)}$.

1.1 Our results

Balanced-chain set systems and lower bounds. Our first result relates balanced-chain set systems to the problem of proving mlABP lower bounds via the min-partition rank method. We show the following two-directional equivalence.

▶ **Theorem 3.** *Let $s := s(n)$ be any growing function of n with $s \ge n$. Then the following hold:*

- *If any $(\log s)$-balanced-chain set system has size at least s, then any mlABP computing a full-rank polynomial has size at least $s^{\Omega(1)}$.*
- *Conversely, if there is an $O(1)$-balanced-chain set system of size s, then there is an mlABP computing a full-rank polynomial of size at most $s \cdot \mathrm{poly}(n)$.*

The reader may find the forward direction fairly intuitive, given the discussion above. In the other direction, it is possible to turn this purely combinatorial construction back into a polynomial that satisfies the algebraic property of full-rankness.

▶ **Observation 4.** *The above theorem provides a near-tight characterization of the size of the smallest mlABP in terms of the size of the smallest balanced-chain set system. The two directions differ only in the balance parameter: e.g., an mlABP lower bound of the form $s \ge n^c$ requires us to show that every $\Omega(c \log n)$-balanced-chain set system needs size n^c, but an upper bound of $s \le n^c$ can only be ensured by an $O(c)$-balanced-chain set system of size n^c. We can close this gap completely for the more restrictive setting of* set-multilinear *ABPs (defined below) where both directions use a balance parameter that is $\Theta(c)$. So at least in the set-multilinear setting, where the lower bound question is still open [4], we have a perfect characterization of the ABP complexity of full-rank polynomials in terms of the combinatorial question.*

The next question is to understand how to analyze and prove lower and upper bounds on the minimal size of a balanced-chain set system. A priori, this problem could be difficult, since finding a specific partition $f : X \to \{\pm 1\}$ with respect to which $\mathrm{cbal}_{\mathcal{X}}(f)$ is large could be

hard. However, we show, via a simple covering argument, that a *uniformly random* partition is as hard as a worst-case partition in the following sense: if there is a small set system that contains a balanced chain for most partitions, then there is also a small balanced-chain set system. We make this precise below.

▶ **Definition 5.** *For $\varepsilon \in [0,1]$ and $k \in \mathbb{R}_{\geq 0}$, we call an mlABP set system $\mathcal{X}$ an (ε, k)-*balanced-chain set system *if, for a uniformly random balanced partition $f : X \to \{\pm 1\}$, we have*

$$\mathbb{P}_f \left[\mathrm{cbal}_{\mathcal{X}}(f) \leq k\right] \geq \varepsilon.$$

Note that a k-balanced-chain set system is also a $(1,k)$-balanced-chain set system.

▶ **Lemma 6** (Worst-case to average-case reduction)**.** *If there is an (ε, k)-balanced-chain set system, over the set $[n]$, of size s, then there is a k-balanced-chain set system, over the set $[n]$, of size at most $O(sn/\varepsilon)$.*

We use both directions of Theorem 3 and Lemma 6 in what follows.

An improved upper bound on the size of balanced-chain set systems. We note that Theorem 3 in combination with the results of Raz (Theorem 1) immediately implies that there is an $O((\log n)^2)$-balanced-chain set system of size $n^{O(\log n)}$. In fact, it is not hard to give a direct proof of the slightly stronger fact that there is a 1-balanced-chain set system of size $n^{O(\log n)}$. This can be proved using an inductive construction (for completeness, see Theorem 12), using the fact (already mentioned above) that the family of intervals contained in $[n]$ are tight for the "balancing set systems" problem. The above lemma, however, is still consistent with the fact that the mlABP complexity of any full-rank polynomial could be $n^{\Omega(\log n)}$, which is no better than the upper bound in Theorem 1. Using a more sophisticated inductive argument, we are able to improve the easy upper bound superpolynomially, marking the first such improvement.

▶ **Theorem 7.** *There is a 1-balanced-chain set system of size at most $n^{O(\log n/\log\log n)}$.*

Using Theorem 3, this implies that there are full-rank mlABPs of size $n^{O(\log n/\log\log n)}$. An immediate consequence (using Theorem 1) is that mlABPs are superpolynomially more powerful than multilinear formulas.

This recovers a superpolynomial separation between these models that was proved in [7]. Our result is weaker than this work both quantitatively (the superpolynomial lower bound) and qualitatively (the construction of the set system and hence the mlABP are not uniform). However, we note that our proof is very different from that of [7]. While the proof of [7] reveals more about the structure of multilinear formulas by showing that they are not full-rank with respect to a much more restricted family of equipartitions (called arc-partitions), our proof yields a better understanding of the computational power of mlABPs, which is what we need to prove lower bounds against the latter model.

From a broader computational complexity perspective, understanding the size of ABPs is closely related to studying the computational power of dynamic programming. The construction above shows that dynamic programs can be quite powerful. More concretely, consider the following non-deterministic read-once branching program model. The devices in this model are directed acyclic graphs with a designated source a and a designated sink b. The input is $x \in \{0,1\}^n$ for even n. Each edge in the graph is labeled by two variables (x_i, x_j). On every directed path every variable appears at most once. Assume that all a-to-b

paths are of length exactly $n/2$. On input $x \in \{0,1\}^n$, an edge labeled by (x_i, x_j) is open iff $x_i + x_j = 1$ (in other words, it is balanced). The program accepts x iff there is an open path from a to b. We care about such programs that accept x iff the Hamming weight of x is exactly $n/2$. A program that achieves this functionality can actually "prove" its correctness. If it accepts an input x, then the open path that verifies acceptance shows that half of the entries of x are 1s and half are 0s.

What is the smallest program with this functionality? It is not hard to inductively construct such a program of size $n^{O(\log n)}$. The above theorem gives a non-trivial *superpolynomial* improvement. The exact complexity remains an open question.

Intervals and random partitions. As outlined above, set systems based on intervals have played an important role in previous investigations on multilinear lower bounds. Such families are optimal for constructions of balanced set systems and furthermore, results that separate multilinear circuits and ABPs from formulas [14, 7] are also based on combinatorial constructions that use intervals.

It is therefore natural to ask if intervals can be used to construct k-balanced-chain set systems for small k. We have already seen before that this family does not yield a good balanced-chain set system for a worst-case f. However, Lemma 6 shows that if this set system yields a balanced chain for a random partition, then that would allow us also to devise a somewhat larger set system that has balanced chains for all partitions. We are able to rule out this possibility with a strong negative result. This provides evidence that our technique may be able to prove superpolynomial lower bounds against general mlABPs.

▶ **Theorem 8** (Intervals vs. random partitions). *The intervals set system $\mathcal{I}$ defined before is not an (ε, k)-balanced-chain set system for any $\varepsilon > 2^{-\Omega(n^{1/5})}$ and $k < n^{1/5}$.*

In particular, the above is a considerable generalization of a recent result of Chatterjee, Kush, Saraf and Shpilka [4], whose results imply such a bound for the subset $\mathcal{I}_0$ of $\mathcal{I}$ that consists only of intervals of the form $[i]$ for $i \in \{0, \ldots, n\}$. The set system $\mathcal{I}_0$ contains just one maximal chain, while the set system $\mathcal{I}$ contains *exponentially* many chains, which makes the arguments from [4] inapplicable for $\mathcal{I}$. Using our results, we can prove the following lower bound for a generalization (formally defined in Section 5.4) of the sum of ordered set-multilinear ABPs model considered by [4].

▶ **Corollary 9** (Lower bounds for sum of interval-mlABPs). *Let $\mathbb{F}$ be any field. Any $\Sigma_\pi$mlABP program computing a full-rank polynomial $P(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ must have size at least $2^{\Omega(n^{1/5})}$.*

The main technical part of the proof of Theorem 8 analyzes the discrete Fréchet distance between two one-dimensional random walks. This is a basic problem of independent interest. We have two random walks, and we want to understand what is the best time parametrization that makes them as close as possible. Here is a slightly more formal description (but still not fully formal; for details see Section 4). A random walk defines a random function $f : \{0, 1, \ldots, n\} \to \mathbb{Z}$ where $f(t)$ is the position of the walk at time t. The Fréchet distance between two functions f and g is

$$d_F(f, g) := \min_{\alpha,\beta} \max_t |f(\alpha(t)) - g(\beta(t))|,$$

where α, β are time parameterizations (that may depend on both f and g). We have two independent (1-Lipschitz) random functions, we want to prove that we cannot hope to parametrize time to make them close (even when we know the future). The property that makes this problem challenging is the dependency on the future.

The set-multilinear setting. Two recent works have conducted investigations into the power of *set-multilinear* ABPs (smlABPs). It is not hard to show that proving lower bounds against mlABPs is at least as hard as the analogous problem for smlABPs, and proving lower bounds against the latter is still an open question. The work of Bhargava, Dwivedi and Saxena [3] showed that proving lower bounds on smlABPs for low-degree polynomials implies general ABP lower bounds. A follow-up result of [4] showed restricted lower bounds against smlABPs in both the low and high-degree settings. As mentioned above, this latter paper implicitly proves a result that is a special case of Theorem 8 above.

1.2 Outline of the proofs

From ABPs to balanced-chain set systems and vice versa. We sketch the forward direction of Theorem 3 by showing if there is an mlABP of size s computing a full-rank polynomial P, then there is an $O(\log s)$-balanced-chain set system of size at most s. Let X denote the set of n variables. We associate with each vertex v in the mlABP a set $X_v \subseteq X$ of variables that appear on paths from the source vertex to v. The set system $\mathcal{X}$ we consider is the set of all such X_v (there are some hidden technicalities here). If there is a balanced $f : X \to \{\pm 1\}$ such that there is no $O(\log s)$-balanced-chain in $\mathcal{X}$, then we can find a cut C separating source from sink in the mlABP containing only vertices v such that X_v is, for example, $10 \log s$-imbalanced with respect to f. Using the decomposition

$$P = \sum_{v \in C} L_v(X_v) \cdot R_v(X \setminus X_v),$$

we can show that P is not full-rank with respect to f, which is a contradiction.

In the setting of *set-multilinear* ABPs, we can get a tighter result by showing that there is an $O(\log_m s)$-balanced-chain set system of size at most s where m is the number of variables in each part of the set-multilinear variable-set.

For the other direction, we start with a k-balanced-chain set system $\mathcal{X}$ of size s and show that there is a combinatorial transformation that produces a 1-balanced-chain set system of size $s \cdot n^k$. In the 1-balanced-chain setting, we use an idea that was first used by Raz [14] to construct full-rank polynomials. The observation is that, for a known equipartition $\Pi = (\{y_1, \ldots, y_{n/2}\}, \{z_1, \ldots, z_{n/2}\})$ of the n variables, the simple "gadget" polynomial

$$Q_\Pi = \prod_{i=1}^{n/2} (y_i + z_i)$$

is full-rank with respect to Π. To handle an unknown partition, we construct a polynomial that is a sum of such polynomials for all possible equipartitions. A 1-balanced-chain set system gives us a recipe for constructing such a polynomial efficiently. Given any equipartition $\Pi = (Y, Z)$, we know that there is a chain $C = (C_0, \ldots, C_n)$ of sets in the set system such that each set has chain-balance either 0 or 1 with respect to Π. In particular, the even-sized sets $C_0, C_2, C_4, \ldots, C_n$ all are perfectly balanced and $C_{i+2} = C_i \cup \{y_{i/2+1}, z_{i/2+1}\}$ for each even $i < n$. A natural choice to construct a full-rank Q_Π is to pair up $y_{i/2+1}$ with $z_{i/2+1}$. It is possible to design an mlABP that computes the sum of all such Q_Π for all chains in a given set system, as in [7]. To ensure that the resulting polynomial is full-rank with respect to all equipartitions, we need to take a suitable linear combination of all these Q_Π. Raz [14] observed that this can be done over a transcendental extension of the base field $\mathbb{F}$ by using these additional transcendental field elements in the linear combination (the above gadget polynomial is actually from [16]). We note that if the field $\mathbb{F}$ is infinite, these transcendental

elements can be replaced by random elements of $\mathbb{F}$, giving us a full-rank polynomial over the base field. A similar construction (with a different gadget polynomial) also works in the set-multilinear case.

Worst-case to average-case reduction. Assume that we have an (ε, k)-balanced-chain set system $\mathcal{X}$. The main observation is that, given a worst-case balanced partition $f : X \to \{\pm 1\}$, and a uniformly random permutation $\pi : X \to X$, the function $f \circ \pi$ is a uniformly random partition and hence, the set system $\mathcal{X}$ contains a k-balanced-chain with respect to $f \circ \pi$ with probability at least ε. This is the same as saying that the random set system $\mathcal{X}_\pi$ (obtained by permuting the elements of each set in $\mathcal{X}$ according to π) contains a k-balanced-chain with respect to f with probability at least ε. Choosing $O(n/\varepsilon)$ many such permutations π and taking the union of the corresponding set systems allows us to increase this probability to $1 - 2^{-n}$, at which point the probabilistic method guarantees the existence of a suitable set system that works for all partitions.

Constructing a 1-balanced-chain set system of size $n^{O(\log n/\log\log n)}$. By the preceding argument, it suffices to construct a 1-balanced-chain set system of this size that contains a 1-balanced-chain for a uniformly random balanced partition f with good probability (e.g. $1/\operatorname{poly}(n)$). To do this, we identify the set X with $[n]$ and view $f : X \to \{\pm 1\}$ as a uniformly random walk of length n on the integers that starts and ends at the origin 0 (also known as a random bridge). We note that a maximal chain that is balanced with respect to f contains $n/2$ sets C that are balanced with respect to f. A random walk that returns to 0 many times would give us many *intervals* that satisfy this criterion. A result of Csáki, Erdős and Révész [6] tells us that a uniformly random walk f, with probability $\geq 1/\operatorname{poly}(n)$, returns to 0 at least once every $m \approx n/\log n$ steps. This implies that the set of all intervals, while not balanced, does contain a *non-maximal* chain of balanced intervals

$$\varnothing \subseteq [i_1] \subseteq [i_2] \subseteq \cdots \subseteq [i_r] = [n]$$

where each subsequent interval in the chain adds at most m elements to the previous one. To finish the argument, we only need to add sets to "fill in" the missing sets in the chain to make it maximal. This can be formulated as the same problem for the smaller set size m. Implementing a recursive construction leads to a set system of the given size.

▶ **Observation 10.** *While we only used the family of intervals in the above construction, it should be noted that the overall family constructed above consists of sets that are very different from intervals. This is because we need to carry out the aforementioned average-case to worst-case argument at each stage, which is done by randomly permuting the elements in the sets. This does not preserve the property of being an interval. It seems interesting to derandomize this "construction".*

Intervals and random partitions. Let $\mathcal{I}$ denote the intervals set system over $[n]$ as defined before. Assume that $f : [n] \to \{\pm 1\}$ is chosen uniformly at random (for the sake of exposition, we drop for now the assumption that f is balanced). The claim is that the chance that there is a maximal chain $(C_0, \ldots, C_n)$ of intervals that make f balanced is exponentially small.

For the sake of exposition, we also assume that the chain starts at $C_1 = \{n/2\}$. The random walk f can be partitioned into two independent random walks: g that goes from $n/2$ to n and h that goes from $n/2$ to 1. The random walk f is chain-balanced only if there are time parameterizations under which g, h are close at all times (this is a discrete version of

the Fréchet distance). We thus have two independent random walks (each of length $n/2$) and the claim is that it is very unlikely that there is a way to parametrize time so the distance between g, h is say $k < n^{1/3}$. The main idea is to first sample the random walk g and identify a pattern in it that h can not possibly follow. Fix the position of the random walk g at times that are multiples of some ℓ. Let p_t be the position of g at time $t\ell$ for $t \leq T := n/(2\ell)$. Typically, the distance between p_t and p_{t+1} is roughly $\sqrt{\ell}$. For concreteness, think of the case that $p_t = 0$ for even t and $p_t = \sqrt{\ell}$ for odd t. It remains to show that there is no sequence of (strictly increasing) time steps $\tau_1 < \tau_2 < \ldots < \tau_T$ such that $h(\tau_t)$ is close to p_t for all t.

Let $(C_0, \ldots, C_n)$ denote any maximal chain in $\mathcal{I}$ and assume that $C_1 = \{i\}$. We can decompose f into two random functions $g : [i] \to \{\pm 1\}$ and $h : [i+1, \ldots, n] \to \{\pm 1\}$. Each set C_j in the chain can be written as the disjoint union of two intervals $I'_j = [i - i' + 1, i]$ and $I''_j = [i+1, i+j-i']$ contained in the domains of g and h respectively. The quantity $f(C_j)$ is $g(I'_j) + h(I''_j)$ and thus the balance of this chain with respect to f is equal to

$$k := \max_j |g(I'_j) + h(I''_j)| = \max_j |g(I'_j) - (-h(I''_j))|.$$

Thinking of g and $-h$ as random walks on the integers, the chain can be interpreted as an "alignment" of these two random walks that ensures that they stay within distance k at all time steps, or equivalently, these two random walks are at distance at most k from each other in *discrete Fréchet distance* (defined formally below). The main technical part of the proof is to show that the probability that $k \leq n^{O(1)}$ is exponentially small.

This is done by finding some pattern in one of the two walks, say g, that does not appear in the other except with small probability. To simplify further, assume that g and h are random walks of the same length $n/2$. We divide the time steps of g into epochs of some suitable length ℓ and consider the positions of the random walk g at these various time steps. I.e.

$$p_1 = g([(n/2) - \ell, n/2]), p_2 = g([(n/2) - 2\ell, n/2]), \ldots, p_{n/2\ell} = g([n/2]).$$

As long as ℓ is large enough, there is enough randomness between any pair of time steps to ensure that most pairs of consecutive positions are quite far from one another.[5] Conditioning on these positions, we show that with high probability, there is no subsequence of (strictly increasing) timesteps when the positions of $-h$ are $p'_1, p'_2, \ldots, p'_{n/2\ell}$ such that $|p_t - p'_t| \leq k$ for each t. This implies that the two walks cannot be aligned as described above.

To see that $-h$ cannot come close to the positions achieved by g, we think of the case when $p_1 = p_3 = p_5 = \cdots = 0$ and $p_2 = p_4 = p_6 = \cdots = \sqrt{\ell}$. Intuitively, this is a bad case as 0 is the most probable location of a random walk at any (even) time step and we expect a random walk to travel a distance of $\sqrt{\ell}$ every ℓ steps. Now, the expected number of times that a random walk of length $n/2$ returns to 0 is $\Theta(\sqrt{n})$ and we can also prove a concentration bound on the upper tail of this random variable. Thus, if we choose ℓ significantly smaller than $\sqrt{n}$, it is very unlikely that the random walk $-h$ can return to (a position close to) 0 at least $\Omega(n/\ell)$ many times.

1.3 Conclusion and further questions

In this work, we study the power and limitations of the min-partition rank method to prove lower bounds for mlABPs. We first introduce a combinatorial complexity measure, the chain-balance, for set systems that generalizes previous combinatorial notions used to

[5] In the actual argument, we vary the lengths of these epochs to ensure that *every* consecutive pair of positions are indeed far from one another.

prove lower bounds for mlABPs. We prove general properties of this measure, together with an $n^{O(\ln n/\ln\ln n)}$-upper bound for the size of a set system with chain-balance 1, and an average-case $\Omega(n^{1/5})$-lower bound for the chain-balance of restricted set systems obtained from intervals. Then we translate those combinatorial results into the algebraic complexity world. We obtain both upper and lower bounds for mlABP and its variants. We show that no full rank polynomial can be computed by a $\sum_\pi$ mlABP program of size $2^{o(n^{1/5})}$, and also recover a superpolynomial separation between multilinear formulas and mlABPs. Additionally, our $n^{O(\log n/\log\log n)}$ upper bound on the size of mlABP computing a full rank polynomial implies that the min-partition rank method cannot be used to prove the potentially optimal $n^{\Omega(\ln n)}$-size lower bound against mlABPs for a (full rank) polynomial in **mlVP**. The main open questions related to our work are the following:

- Can we close the gap between the lower bound $\Omega(n^2/k)$ and upper bound $n^{O(\ln n/\ln\ln n)}$ for the size of k-balanced-chain set systems? Both directions are interesting since Theorem 3 shows that sufficiently high lower bounds on the size of balanced-chain set systems imply new lower bounds for the size of mlABPs computing full rank polynomials, and a poly(n) upper bound implies that the min-partition rank method cannot be used to prove superpolynomial lower bounds for mlABPs.
- Can we prove $n^{\Omega(1)}$ lower bounds for the chain-balance of other families of restricted set systems? In Theorem 8, we showed an $\Omega(n^{1/5})$-lower bound for the chain-balance of intervals set systems, and applied it almost as a black box to obtain an exponential lower bound against the $\sum_\pi$ mlABP model. Thus, it is natural to ask whether we can reduce the problem of showing lower bounds for other interesting restricted versions of mlABPs to the question of showing lower bounds for the chain-balance of restricted set systems.
- Can we obtain an explicit construction of full rank mlABPs of size $n^{o(\ln n)}$? A possible approach for this question is a derandomization of our construction. In particular, if we are allowed to use field extensions (as the constructions in [15, 7]), it is sufficient to derandomize our proof of the worst-case to average-case reduction, which is a covering argument via random permutations of a given set system (Lemma 14).

1.4 Organization of the paper

We start with some preliminaries and then prove some combinatorial properties of the chain-balance of set systems in Section 2, including the worst-case to average-case reduction (Lemma 6). We will then show our main combinatorial theorems, starting with the construction of a 1-balanced-chain set system of size $n^{O(\log n/\log\log n)}$ (Theorem 7) in Section 3, followed by the analysis of intervals with respect to random partitions (Theorem 8) in Section 4. Finally, in Section 5, we prove the formal connection between the combinatorial constructions and the mlABP complexity of full-rank polynomials (Theorem 3) and derive the consequences of our combinatorial theorems to algebraic complexity, that is, a separation between multilinear formulas and mlABPs, and an exponential lower bound for the $\sum_\pi$ mlABP model (Corollary 9).

The omitted proofs can be found in the full version of this paper (`https://eccc.weizmann.ac.il/report/2026/001/`).

2 Properties of balanced-chain set systems and their chain-balance

In this section, we prove some properties of balanced-chain set systems related to bounds for the size of these systems and conversions between different notions of chain-balance.

2.1 Upper and lower bounds for the size of balanced-chain set systems

Now let us define a natural class of set systems.

▶ **Definition 11.** *For every $n, m \in \mathbb{N}$, we denote by $\mathcal{I}_{n,m}$ the* m-interval set system *over $[n]$ defined as*

$$\mathcal{I}_{n,m} := \{\, I_1 \cup \ldots \cup I_l \mid I_1, \ldots, I_l \text{ are intervals of } [n] \text{ and } l \leq m \}.$$

Assume that $m \leq n/2$. Since any element of $\mathcal{I}_{n,m}$ is specified by the endpoints of the intervals, using the usual bounds on the binomial numbers, we get that the size of the m-interval set system is at most

$$|\mathcal{I}_{n,m}| \leq (n/m)^{O(m)}.$$

We note below that using a simple inductive analysis, we can obtain a 1-balanced-chain set system from intervals.

▶ **Theorem 12.** *For every even number $n \in \mathbb{N}$, the chain-balance of the $(2\lceil \lg n \rceil)$-interval set system is* 1.

Proof. We omit this proof but it can be found in the full version of this paper. Actually, in Section 3, we analyse the probabilistic behaviour random balanced partitions, instead of the structural result used in the proof of this theorem, and construct a 1-balanced-chain set system of size $n^{O(\lg n / \lg \lg n)}$. ◀

▶ **Observation 13.** *We also note that using previous results about balancing systems, we can prove a $\Omega(n^2/k)$ lower bound for the size of any k-balanced-chain set system for $k \leq n/5$. More specifically, a result by Alon, Kumar, and Volk [2] can be rephrased as follows: for every $k \in \mathbb{N}$ and sufficiently large $n \in \mathbb{N}$, if a collection $S_1, \ldots, S_m \subseteq [n]$ of sets satisfies that $2k \leq |S_i| \leq n - 2k$ for every $i \in [m]$ and, for every balanced partition f of n, there is an $i \in [m]$ such that $|f(S_i)| \leq 2k$, then*

$$m \geq \Omega(n/k).$$

Hence, for any $2k$-balanced-chain set system $\mathcal{X}$ over $[n]$, we can consider, for every $l \in \{2k, \ldots, n-2k\}$, the collection

$$\mathcal{X}_l := \{\, S \in \mathcal{X} \mid |S| = l \}.$$

As $\mathrm{cbal}(\mathcal{X}) \leq 2k$, every balanced partition f has a maximal chain $(C_0, \ldots, C_n)$ of $\mathcal{X}$ such that, for every $i \in [n]$, the set C_i satisfies $|C_i| = i$ and

$$|f(C_i)| \leq 2k.$$

By the lower bound above, we obtain that $|\mathcal{X}_l| \geq \Omega(n/k)$, which implies that

$$|\mathcal{X}| \geq \Omega((n-4k)n/k).$$

2.2 Conversion of (p, l)-balanced-chain systems into l-balanced-chain systems

As mentioned in the introduction, we can prove a worst-case to average-case reduction for balanced-chain set systems.

▶ **Lemma 14** (Lemma 6). *Let $n \in \mathbb{N}$ be an even number, and X be a set of n elements. Let $l \in [n]$ and $p \in (0, 1]$. If $\mathcal{X}$ is a (p, l)-balanced-chain set system, over X, of size s, then there is an l-balanced-chain set system $\mathcal{Y}$, over X, of size $O(sn/p)$.*

2.3 Conversion of l-balancedness to 1-balancedness

Let us now prove a combinatorial lemma that shows how to convert an l-balanced-chain set system into a 1-balanced-chain set system with some size blow-up.

▶ **Lemma 15.** *Let $n \in \mathbb{N}$ be an even number, and let X be a finite set with n elements. If $\mathcal{X}$ is an l-balanced-chain set system, over X, with size s, then there is a set system $\mathcal{Y}$ over X such that*

- *$\mathcal{Y}$ has chain-balance 1; and*
- *the size of $\mathcal{Y}$ is at most $s\binom{|X|}{\leq l}$.*

3 Construction of a 1-balanced-chain set system of size $n^{O(\ln n/\ln\ln n)}$

In this section, our goal is to prove Theorem 7. Our proof uses concepts from one-dimensional random walk theory (Definition 16) and a result about the probability distribution of the length of the longest excursion in a random bridge (Lemma 17). In Section 5.2, the main result of this section is used to obtain a separation between multilinear formulas and mlABPs.

▶ **Definition 16.** *Let $n \in \mathbb{N}$ and let $f\colon [n] \to \{-1,1\}$. We associate a function $W_f\colon \{0,\dots,n\} \to \mathbb{Z}$ to f as follows: let $W_f(0) := 0$ and, for every $i \in [n]$,*

$$W_f(i) := f([i]),$$

where $f([i]) = \sum_{j\in[i]} f(j)$ as defined in Definition 2. We call W_f the walk defined from f, *and n the* length *of W_f. Note that the definition above is a bijection between f and W_f. Thus, for a given walk W, we denote by f_W the function such that $W = W_{f_W}$. Sometimes we also consider W_f to be the sequence $(W_f(0),\dots,W_f(n))$.*

We say that W_f is a bridge *if f satisfies $f([n]) = 0$. We say that W_f is a* excursion *if W_f is a bridge and, for every $i \in [n-1]$,*

$$W_f(i) \neq 0.$$

For f sampled uniformly at random from the set of all functions from $[n]$ to $\{-1,1\}$, we say $W := W_f$ is a random walk *of length n, that is,*

$$W \sim \mathrm{Unif}(\{\, W_f \mid f\colon [n] \to \{-1,1\}\}.$$

Simarly, for even n, we define a random bridge *B of length n as*

$$B \sim \mathrm{Unif}(\{\, W_f \mid f\colon [n] \to \{-1,1\}, f([n]) = 0\}).$$

Let B be a random bridge of length n and $f := f_B$. We say that a point $t \in \{0,\dots,n\}$ is a zero *of B if*

$$f([t]) = 0,$$

or, equivalently, $B(t) = 0$. Let $Z(B)$ be the (random) set of zeros of B. Note that $\{0,n\} \subseteq Z(B)$ by the definition of bridge. Let $\{z_0,\dots,z_l\} := Z(B)$ be the elements of $Z(B)$, where $z_0 < z_1 < \ldots < z_l$. Let $\lambda(B)$ be the random variable defined as follows:

$$\lambda(B) := \max_{i\in[l]} z_i - z_{i-1}.$$

We call $\lambda(B)$ the length of the longest excursion *in B.*

▶ **Lemma 17** (Lemma 10 from [6]). *Let $n \in \mathbb{N}$ and let B be a random bridge of length $2n$. There are universal real constants $0 < C_1 \leq C_2$ and $\beta > 0$ such that, for every $a \geq n^{2/3}$, we have*

$$\mathbb{P}_B[\lambda(B) \leq 2a] = C(n,a) \min\{(n+1)^{-1/2}, a^{-1/2}\} \exp(-\beta n/a),$$

for $C_1 \leq C(n,a) \leq C_2$.

Let us now prove our main theorem.

▶ **Theorem 18** (Theorem 7). *For every sufficiently large even number $n \in \mathbb{N}$, there is a 1-balanced-chain set system $\mathcal{X}_n$ over $[n]$ of size $n^{O(\ln n / \ln \ln n)}$.*

Proof. For every $n \in \mathbb{N}$, let $N(n)$ be the minimum size of a 1-balanced-chain set system over $[n]$, and let A_n be a fixed 1-balanced-chain set system of size $N(n)$. Our goal is to obtain a recursive upper bound for $N(n)$.

For any $m \in [n]$, let B_m be the set system defined as follows:

$$B_m := \{\, [1,i] \odot A_t \mid t \in [m], 0 \leq i \leq n-t \},$$

where we define $[1,i] \odot A_t$ as

$$[1,i] \odot A_t := \{\, [1,i] \cup \{\, i+j \mid j \in R\} \mid R \in A_t \},$$

which can be interpreted as the translation of the set system A_t to the end of the interval $[1,i]$. Note that B_m is a set system over $[n]$ of size at most $n^2 N(m)$. Let f be a uniformly random balanced partition of $[n]$. We say that a point $t \in [n]$ is a *zero* of f if $\sum_{i \in [t]} f(i) = 0$, and we say that a set system A is *f-balanced* if $\mathrm{cbal}_A(f) \leq 1$. Let E_m be the event that the maximum gap between two consecutive zeros of f is at most m, and let

$$\delta_m := \mathbb{P}_f[E_m].$$

Consider the following auxiliary lemma:

▶ **Lemma 19.** *For any balanced partition f of $[n]$, if $f \in E_m$, then B_m is f-balanced.*

Proof. This follows from the fact that if $f \in E_m$, then the maximum gap between two consecutive zeros z_i and z_{i+1} of f is at most m. Consequently, we can construct a f-balanced chain of B_m in the following inductive way:

1. We know that $z_1 \leq m$, and A_{z_1} has a $f\restriction_{[1,z_1]}$-balanced chain C_1, and $A_{z_1} = [1,0] \odot A_{z_1}$, so C_1 is also a subchain of B_m.
2. For every $i \geq 1$, suppose that we already have a subchain C_i of B_m such that C_i is $f\restriction_{[1,z_i]}$-balanced. Note that $f\restriction_{[z_i+1,z_{i+1}]}$ is a balanced partition of $[z_i+1, z_{i+1}]$, and, by defining a balanced partition g of $[z_{i+1} - z_i]$ as

   $$g(j) := f(z_i + j)$$

   for every $j \in [z_{i+1} - z_i]$, we can obtain a g-balanced chain C from the set system $A_{z_{i+1}-z_i}$. Furthermore, we have that $z_i + S \subseteq [z_i+1, z_{i+1}]$ for every $S \in C$, which implies that

   $$C_{i+1} := C_i \cdot ([1, z_i] \cup (z_i + S) \mid S \in C)$$

   is a $f\restriction_{[1,z_{i+1}]}$-balanced subchain of B_m, where $\cdot$ denotes the concatenation of chains (which can be interpreted as strings). By induction on i, we can obtain f-balanced chain of B_m. ◀

By Lemma 17 and interpreting its result about random bridges B in the language of random balanced partitions f, there are positive universal constants $b, c \in \mathbb{R}$ such that, for every even $m \in [n]$ such that $(n/2)^{2/3} \leq m/2$, we have

$$\delta_m = \mathbb{P}_B[\lambda(B) \leq m] \geq cn^{-1/2} \exp(-bn/m).$$

Now fix the value of m to $m := 2\lceil n/(2 \ln n)\rceil$, which implies that

$$\delta := \delta_m \geq c/n^{b+1/2}.$$

Hence, for this choice of m, we have that B_m is a $(\delta, 1)$-balanced-chain set system over $[n]$ by Lemma 19. By Lemma 14, there is a 1-balanced-chain set system A, over $[n]$, of size

$$|A| \leq O(|B_m|n/\delta) \leq (n^2 N(n/\ln n))n^{b+2} \leq n^d N(n/\ln n),$$

for a positive constant d. By solving the recurrence relation $N(n) \leq n^d N(n/\ln n)$, we obtain that

$$N(n) \leq n^{O(\ln n/\ln\ln n)}.$$ ◀

4 The chain-balance of interval set systems and the Fréchet distance of random walks

In this section, our main goal is to prove Theorem 8. In order to prove this theorem, we establish a connection between the chain-balance of interval set systems and the Fréchet distance between two random walks (Lemma 21), and prove that, with very high probability, the Fréchet distance of two random walks of length n is at least $n^{\Omega(1)}$ (Lemma 22). In Section 5.4, the main result of this section is used to obtain lower bounds on the size of sum of interval-mlABPs computing full rank polynomials.

4.1 Fréchet distance of random walks

Let us start by defining the Fréchet distance of two discrete functions [8].

▶ **Definition 20.** *Let $n \in \mathbb{N}$, and let $l, r \in [n]$ such that $l + r = n$. Let $X\colon \{0, \ldots, l\} \to \mathbb{R}$ and $Y\colon \{0, \ldots, r\} \to \mathbb{R}$ be two functions. We define the* (discrete) Fréchet distance $d_F(X, Y)$ *between X and Y as*

$$d_F(X, Y) := \min_{\alpha,\beta} \max_{t\in[n]} |X(\alpha(t)) - Y(\beta(t))|,$$

where the minimum ranges over all non-decreasing functions

$$\alpha\colon \{0, \ldots, n\} \to \{0, \ldots, l\} \text{ and } \beta\colon \{0, \ldots, n\} \to \{0, \ldots, r\}$$

satisfying the following properties:

- $\alpha(0) = 0 = \beta(0)$;
- $\alpha(n) = l$ *and* $\beta(n) = r$;
- *For every $t \in [n]$, we have $\alpha(t) \leq \alpha(t-1) + 1$, and $\beta(t) \leq \beta(t-1) + 1$, and*

$$\alpha(t) - \alpha(t-1) + \beta(t) - \beta(t-1) = 1.$$

Now let us state the main technical lemmas of this section.

▶ **Lemma 21** (Discrepancy and Fréchet distance). *Let $n \in \mathbb{N}$ be an even number, and let $\mathcal{I} := \mathcal{I}_{n,1}$ be the 1-interval set system (Definition 11). Let $l, r \in [n]$ such that $l + r = n$, and let $W_{l,r} := (W_l, W_r)$ be a pair of two independent random walks W_l and W_r (Definition 16) of length l and r, respectively. For f being a uniformly random balanced partition, and $W := (X, Y) := W_{n,n}$, and $X_l := X\restriction_{\{0,\ldots,l\}}$ and $Y_r := Y\restriction_{\{0,\ldots,r\}}$, we get that, for every $k \in \mathbb{R}$,*

$$\mathbb{P}_f[\mathrm{cbal}_{\mathcal{I}}(f) \leq k] \leq O(n^{5/2}) \max_{r\in[n]} \mathbb{P}_W[d_F(X_r, Y_{n-r}) \leq k].$$

▶ **Lemma 22** (Fréchet distance of two random walks). *There is a positive constant $c_1 \in \mathbb{R}$ such that, for every sufficiently large $n \in \mathbb{N}$, and every $l := l(n) \in [n]$, and every $\varepsilon := \varepsilon(n) \in (0, 1/4)$ such that $n^{1/4-\varepsilon} \geq \frac{2}{3} \ln n$, we have, for $(X, Y) := W := W_{l,n-l}$,*

$$\mathbb{P}_W[d_F(X, Y) < n^{1/4-\varepsilon}] \leq \exp(-c_1 \cdot \min\{n^{1/4-\varepsilon}, n^{4\varepsilon}\}).$$

Proof. This lemma is proved in Section 4.2. ◀

Let us now prove Theorem 8.

▶ **Theorem 23** (Theorem 8). *There is a universal constant $c > 0$ such that, for every sufficiently large even number $n \in \mathbb{N}$, the 1-interval set system $\mathcal{I}_{n,1}$ is not an (ε, k)-balanced-chain set system for $\varepsilon > 2^{-cn^{1/5}}$ and $k < n^{1/5}$.*

4.2 Proof of Lemma 22

We omit the proof of this lemma, but we give an informal description of its proof. We consider two random walks g and h of total length n. Assume w.l.o.g. that g has length at least $n/2$. Let Δ be a parameter that we will choose later. We observe that w.h.p. there are approximately $k \approx n/\Delta^3$ points in g such that every consecutive pair of points is Δ-far from each other. We then analyze the probability that h can reach close to each of these points (which is required for the Fréchet distance between them to be small). The number of steps to go from one of these points to the next is captured by a first passage random variable F_δ with distance parameter $\delta = \Theta(\Delta)$. For the Fréchet distance to be small, the sum of k independent copies of these random variables must be at most n. However, this probability is exponentially small as long as $k^2\delta^2 \gg n$ which is true for small enough Δ.

5 Full rank multilinear ABPs and balanced-chain set systems

In this section, our goals are to show a relationship between constructions of mlABPs computing full rank polynomials and balanced-chain set systems, and use this relationship together with our results from Sections 3 and 4 to, respectively, obtain a separation between the power of multilinear formulas and mlABPs to compute full rank polynomials, and prove lower bounds for the size of sums of *interval-mlABPs* computing full rank polynomials. We first prove our conversion from balanced-chain set systems to full-rank mlABPs (Section 5.1) and its consequences (Section 5.2), and then our conversion from full-rank mlABPs to balanced-chain set systems (Section 5.3) and its consequences (Section 5.4).

5.1 Construction of full rank mlABPs from balanced-chain set systems

In this section, our goals are to prove our main construction of full rank mlABPs from 1-balanced-chain set systems (Theorem 24) and generalize it to the case of l-balanced-chain set systems (Theorem 25). In order to obtain our generalization, we use a combinatorial lemma (Lemma 15) that shows how to convert an l-balanced-chain set system into a 1-balanced-chain set system with some size blow-up.

We start with the 1-balanced case. As is standard [14], we will start by proving it by enlarging the field by adding transcendental elements. Later on, we will show how to remove the transcendental elements at the expense of making the polynomial less "explicit" (by replacing these elements by random elements from the field).

▶ **Theorem 24.** *Let $n \in \mathbb{N}$ be an even number, and let X be a set of n elements. Let $W_X := \{ w_t \mid t := (i,u) \in [n] \times X \}$ and $V_X := \{ x_i \mid i \in X \}$ be two disjoint sets of variables with, respectively, n^2 and n elements. If $\mathcal{X}$ is a 1-balanced-chain set system over X, then there is a polynomial $P_{\mathcal{X}}$ over the variables $V_X \cup W_X$ such that the following hold:*

- *All coefficients of $P_{\mathcal{X}}$ are 0 or 1;*
- *For every field $\mathbb{F}$, the polynomial $P_{\mathcal{X}}$ is full rank over $\mathbb{F}(W_X)[V_X]$ and can be computed by an mlABP of size at most $|\mathcal{X}|$ over $\mathbb{F}(W_X)$.*

More generally, we have the following stronger version of Theorem 24 for l-balanced-chain set systems.

▶ **Theorem 25.** *Let $n \in \mathbb{N}$ be an even number, let $l \in [n]$ and X finite with n elements. Let $W_X := \{ w_t \mid t := (i,u) \in [n] \times X \}$ and $V_X := \{ x_i \mid i \in X \}$ be two disjoint sets of variables with, respectively, n^2 and n elements. If $\mathcal{X}$ is a l-balanced-chain set system over X, then there is a polynomial $P_{\mathcal{X}}$ over the variables $V_X \cup W_X$ such that*

- *All coefficients of $P_{\mathcal{X}}$ are 0 or 1;*
- *For every field $\mathbb{F}$, the polynomial $P_{\mathcal{X}}$ is full rank over $\mathbb{F}(W_X)[V_X]$ and can be computed by an mlABP of size at most $|\mathcal{X}|\binom{n}{\leq l}$ over $\mathbb{F}(W_X)$.*

5.2 A separation between multilinear formulas and mlABPs

As mentioned in the introduction, we can use Theorem 3 together with Theorem 18 to obtain a superpolynomial separation between multilinear formulas and mlABPs. We show this separation (Corollary 27) in this section, after we prove the following formal version of the second part of Theorem 3.

▶ **Theorem 26** (Second part of Theorem 3). *Let $n \in \mathbb{N}$ be an even number, let $l \in [n]$ and X finite with n elements. If $\mathcal{X}$ is a l-balanced-chain set system over X, then, for every infinite field $\mathbb{F}$, there is an n-variate full rank polynomial $P_{\mathcal{X}}$ over $\mathbb{F}$ that can be computed by an mlABP over $\mathbb{F}$ with size at most $|\mathcal{X}|\binom{n}{\leq l}$.*

Now let us use Theorem 26 and Theorem 18 to prove a separation between multilinear formulas and mlABPs.

▶ **Corollary 27.** *Let $n \in \mathbb{N}$ be a sufficiently large even number, and $\mathbb{F}$ be an infinite field. Then there is a n-variate polynomial P_n over $\mathbb{F}$ computed by a mlABP of size $n^{O(\ln n / \ln \ln n)}$, but any multilinear formula computing P_n has size at least $n^{\Omega(\ln n)}$.*

5.3 Construction of balanced-chain set systems from full-rank mlABPs

In this section, our goal is to prove our formal version of the first part of Theorem 3. In particular, we are going to prove a more general theorem (Theorem 28), which allows to relax the hypothesis from the first part of Theorem 3 to average-case almost-full-rank guarantees.

▶ **Theorem 28** (Balanced-chain set systems from layered mlABPs (first part of Theorem 3))**.** *Let $n \in \mathbb{N}$ be an even number, and X be a set of n variables. Let $p := p(n) \in [0,1], \varepsilon := \varepsilon(n) \in (0,1]$. Fix any mlABP $\mathcal{A}$ of size $s \geq n$ computing a (p,ε)-almost full-rank polynomial P over the variables X, Then there exists a $(p, O(\log(sn/\varepsilon))$-balanced-chain set-system $\mathcal{X}$ of size at most $O(s^2 n)$.*

5.4 Lower bounds against $\sum_\pi$ mlABPs

We show in this section how to use the combinatorial results in Section 4 and the connection between multilinear ABPs and balanced-chain set systems to obtain a stronger version of a result of [4]. While we state the result for multilinear ABPs, the same idea also works in the set-multilinear setting.

Let $X := \{x_1, \ldots, x_n\}$ be a set of n variables. Given a bijection $\pi : [n] \to [n]$, we denote by $\mathcal{I}_{X,\pi}$ *the set system consisting of intervals with respect to* π contained in the set X. Formally, we have

$$\mathcal{I}_{X,\pi} := \{\{x_{\pi(i)}, x_{\pi(i+1)}, \ldots, x_{\pi(j)}\} \mid 1 \leq i, j \leq n\}.$$

A π-interval mlABP is defined to be an ABP whose vertices are divided into $n+1$ layers. Each vertex v is labelled with a *π-interval* $Y_v \in \mathcal{I}_{X,\pi}$, and all vertices from the i-th layer are labelled with sets of size i. All edges go from vertices in layer i, for some i, to vertices in layer $i+1$. An edge (u,v) from vertex u in layer i to vertex v in layer $i+1$ is labelled by either a linear polynomial in the unique variable $x \in Y_v \setminus Y_u$, or a field constant. We say that an ABP $\mathcal{A}$ is an *interval mlABP* if there is a bijection π such that $\mathcal{A}$ is a π-interval mlABP.

We denote by $\Sigma_\pi$mlABP the computational model that is a sum of interval mlABPs (possibly with respect to different orderings π).[6] The size of an instance is the sum of all the sizes of the individual mlABPs that make up the instance.

This model is a generalization of the model considered in [4] who prove a lower bound for the sum of *ordered* (set-)multilinear ABPs. The difference between the ordered and interval settings is that in the former, all the X_v are intervals of the form $[i]$ for some $i \in \{0, \ldots, n\}$. Note that the underlying set system has only one maximal chain in the ordered case, but exponentially many chains in the interval setting. This makes our result below more challenging.

We show the following strong lower bound against $\Sigma_\pi$mlABP.

▶ **Theorem 29** (Corollary 9)**.** *Let $n \in \mathbb{N}$ be sufficiently large even number, and X be a set of n variables, and $\mathbb{F}$ be any field. Let P be a full rank polynomial in $\mathbb{F}[X]$. If $\mathcal{A}$ is a Σ_π*mlABP *of size s computing P, then*

$$s \geq 2^{\Omega(n^{1/5})}.$$

[6] The notation here is from [4] but we add an additional subscript "π" to recall that the mlABPs are all interval mlABPs.

5.5 The set-multilinear case

Both parts of Theorem 3 work also in the set-multilinear setting with nearly the same proofs. We sketch the ideas here, noting only the points of departure in the proofs. We assume that we are working with set-multilinear polynomials w.r.t. an equipartition $\mathcal{P} := \{X_1, \ldots, X_n\}$ into sets of size N.

For the second part of Theorem 3, we note that the proofs of Theorem 24, Theorem 25 and Theorem 26 above work almost without change in the set-multilinear case. For the first part of Theorem 3, we note that we can get something *stronger* in the set-multilinear setting by showing that if a set-multilinear ABP of size s computes a (p, ε)-almost full-rank polynomial, then there is a $(p, O(\log_N(sn/\varepsilon))$-balanced-chain set-system of size $O(s^2 n)$. The improvement is in the chain-balance parameter in the set-system where the base of the logarithm is now N instead of 2.

In the interesting case that $N = n^{\Theta(1)}$, the above paragraphs show that we get a tight connection between the smallest set-multilinear ABP computing a full-rank polynomial. For any c, there is a set-multilinear ABP of size $n^{O(c)}$ computing a full-rank polynomial if and only if there is an $O(c)$-balanced-chain set-system of size $n^{O(c)}$.

Finally, these results can be used to argue the lower bound for sums of interval-set-multilinear ABPs just as for the multilinear case.

References

1 Scott Aaronson. Multilinear formulas and skepticism of quantum computing. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 118–127. ACM, 2004. doi:10.1145/1007352.1007378.

2 Noga Alon, Mrinal Kumar, and Ben Lee Volk. Unbalancing sets and an almost quadratic lower bound for syntactically multilinear arithmetic circuits. *Comb.*, 40(2):149–178, 2020. doi:10.1007/S00493-019-4009-0.

3 C. S. Bhargav, Prateek Dwivedi, and Nitin Saxena. Lower bounds for the sum of small-size algebraic branching programs. *Theor. Comput. Sci.*, 1041:115214, 2025. doi:10.1016/J.TCS.2025.115214.

4 Prerona Chatterjee, Deepanshu Kush, Shubhangi Saraf, and Amir Shpilka. Lower bounds for set-multilinear branching programs. In Rahul Santhanam, editor, *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*, volume 300 of *LIPIcs*, pages 20:1–20:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CCC.2024.20.

5 Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. Small-depth multilinear formula lower bounds for iterated matrix multiplication with applications. *SIAM J. Comput.*, 48(1):70–92, 2019. doi:10.1137/18M1191567.

6 E Csáki, P Erdős, and P Révész. On the length of the longest excursion. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 68(3):365–382, 1985.

7 Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 615–624. ACM, 2012. doi:10.1145/2213977.2214034.

8 Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. techreport cd-tr 94/64, 1994.

9 Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. *Theory Comput.*, 17:1–88, 2021. doi:10.4086/TOC.2021.V017A010.

10 Deepanshu Kush and Shubhangi Saraf. Improved low-depth set-multilinear circuit lower bounds. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 38:1–38:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.38.

11 Deepanshu Kush and Shubhangi Saraf. Near-optimal set-multilinear formula lower bounds. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK*, volume 264 of *LIPIcs*, pages 15:1–15:33. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.CCC.2023.15.

12 Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418. ACM, 1991. doi:10.1145/103418.103462.

13 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complex.*, 6(3):217–234, 1997. doi:10.1007/BF01294256.

14 Ran Raz. Separation of multilinear circuit and formula size. *Theory Comput.*, 2(6):121–135, 2006. doi:10.4086/TOC.2006.V002A006.

15 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):8:1–8:17, 2009. doi:10.1145/1502793.1502797.

16 Ran Raz, Amir Shpilka, and Amir Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM J. Comput.*, 38(4):1624–1647, 2008. doi:10.1137/070707932.

17 Ran Raz and Iddo Tzameret. The strength of multilinear proofs. *Comput. Complex.*, 17(3):407–457, 2008. doi:10.1007/S00037-008-0246-0.

18 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Comput. Complex.*, 18(2):171–207, 2009. doi:10.1007/S00037-009-0270-8.

19 Ran Raz and Amir Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *J. Comput. Syst. Sci.*, 77(1):167–190, 2011. doi:10.1016/j.jcss.2010.06.013.

Quantum–Classical Equivalence for AND-Functions

Sreejata Kishor Bhattacharya
TIFR, Mumbai, India

Farzan Byramji
UC San Diego, La Jolla, CA, USA

Arkadev Chattopadhyay
TIFR, Mumbai, India

Yogesh Dahiya
UC San Diego, La Jolla, CA, USA

Shachar Lovett
UC San Diego, La Jolla, CA, USA

Abstract

A major open problem in quantum communication complexity is whether quantum protocols can be exponentially more efficient than classical protocols for computing total Boolean functions; the prevailing conjecture is that they cannot be so. In a seminal work, Razborov (2002) resolved this question for AND-functions of the form

$$F(x, y) = f(x_1 \wedge y_1, \ldots, x_n \wedge y_n),$$

when the outer function f is symmetric, by proving that their bounded-error quantum and classical communication complexities are polynomially related. Since then, extending this result to *all* AND-functions has remained open and has been posed by several authors.

In this work, we settle this problem in a strong way. We show that for every Boolean function f, the bounded-error quantum and classical deterministic communication complexities of the function $f \circ \text{AND}_2$ are polynomially related, up to polylogarithmic factors in n. We prove this by showing that both are characterized – up to polynomial loss – by the logarithm of the De Morgan sparsity of f.

Our results build on the recent work of Chattopadhyay, Dahiya, and Lovett [8] on structural characterizations of non-sparse Boolean functions, which we extend to resolve the conjecture for general AND-functions.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Quantum communication complexity

Keywords and phrases Communication complexity, quantum communication complexity, De Morgan sparsity, approximate gamma two norm, And-functions

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.23

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2026/013/

Funding *Sreejata Kishor Bhattacharya*: Supported by the Department of Atomic Energy, Govt. of India, under project #RTI4014 and by a Google PhD Fellowship.
Farzan Byramji: Supported by Simons Investigator Award #929894, and NSF awards CCF-2425349 and AF: Medium 2212136.
Arkadev Chattopadhyay: Supported by the Department of Atomic Energy, Govt. of India, under project #RTI4014 and by a Google India Faculty Award.
Yogesh Dahiya: Supported by Simons Investigator Award #929894 and NSF award CCF-2425349.
Shachar Lovett: Supported by Simons Investigator Award #929894 and NSF award CCF-2425349.



© Sreejata Kishor Bhattacharya, Farzan Byramji, Arkadev Chattopadhyay, Yogesh Dahiya, and Shachar Lovett; licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 23; pp. 23:1–23:24



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Understanding when/if the laws of quantum mechanics can be exploited to provide significant computational advantage is a major research theme. Two classical results show that indeed quantum algorithms can have surprising power. Shor's polynomial time algorithm on quantum computers for integer factoring beats all *known* classical factoring algorithms by an *exponential* speed-up. Grover search in the quantum query model beats all *possible* classical algorithms by a *quadratic* factor. The first result on exponential speed-up is *conditional* on the hardness of factoring by classical computers. The second result is provable advantage but only quadratic and we know no better is possible in that query model. In this work, we're concerned with provable exponential advantage for quantum models when the task is to compute a total Boolean function. It is well known that both the factoring problem and the search problem for a marked item that Grover's algorithm solves, can readily be re-phrased in terms of total Boolean functions.

Communication complexity, introduced by Yao [41], studies the amount of communication required to compute a function whose input is distributed among multiple parties. Since its inception, communication complexity has become a central tool in theoretical computer science, with applications ranging from streaming algorithms and time–space tradeoffs to data structure lower bounds and circuit complexity. See the textbooks [22, 28] for excellent introductions to the area and its applications.

In the most standard setting – the two-party model – two players, Alice and Bob, wish to compute a Boolean function $F : X \times Y \to \{0, 1\}$, where Alice receives $x \in X$ and Bob receives $y \in Y$. They exchange messages according to a pre-agreed protocol in order to compute $F(x, y)$, with the goal of minimizing the total number of bits communicated in the worst case.

Several variants of communication complexity arise depending on the type of interaction allowed and whether the protocol is permitted to err with small probability. In this work, we focus on three such models: two classical models and one quantum model.

In the classical deterministic model, the *deterministic communication complexity* of F, denoted $\mathrm{D}^{cc}(F)$, is the minimum number of bits that must be exchanged by a protocol that computes $F(x, y)$ correctly on all inputs. In the public-coin randomized model, Alice and Bob have access to shared public randomness and are required to compute $F(x, y)$ with error probability at most $1/3$ on every input; the corresponding complexity measure is the *randomized communication complexity*, denoted $\mathrm{R}^{cc}(F)$.

The quantum variant of communication complexity, also introduced by Yao [42], allows the messages exchanged between the parties to be quantum states (qubits). At the end of the protocol, one of the parties performs a measurement on its quantum state to produce the output. In addition, the parties may share an arbitrary entangled state prior to the start of the protocol, at no communication cost [10]. The number of qubits exchanged by the best such protocol that computes $F(x, y)$ with error probability at most $1/3$, for the worst input, is called the *bounded-error quantum communication complexity* of F, and is denoted $\mathrm{Q}^{cc}(F)$.

A central question in quantum computation is to understand when quantum models enable efficient solutions to problems that are believed to be hard for classical models. In the context of communication complexity, this question asks when quantum protocols can be super-polynomially more efficient than classical (randomized) protocols, and has been a major driving force behind research in the area.

Exponential quantum advantages were discovered early on for structured problems arising from partial functions, sampling problems and relations, see [29, 4, 2, 14, 32]. This understanding has been refined over a number of papers, see [13, 12, 15], with latest work

exhibiting exponential quantum advantage over such promised problems even when the quantum protocol is extremely restricted and no restrictions are imposed on competing classical protocols. In a very recent work, exponential quantum advantage was shown for a total search problem in TFNP by Goos et. al. [16]. However, as re-iterated by [16], no such separation is known for *total* Boolean functions which remains the holy grail of the search for quantum advantage. The largest gap currently known between bounded-error quantum and randomized communication complexity for a total function is only polynomial: a cubic separation (up to polylogarithmic factors), obtained by lifting cubic separations between quantum and randomized query complexity [3, 39].

These results have led to the prevailing belief that, in the absence of special structure – most notably for total Boolean functions – quantum and randomized communication complexities are always polynomially related. This belief was explicitly formulated by Shi and Zhu [40] as the *Log-Equivalence Conjecture* (LEC).

▶ **Conjecture 1.1** (Log-Equivalence Conjecture (LEC) [40]). *For every total Boolean function, the bounded-error quantum and randomized communication complexities are polynomially related in the two-party communication model.*

Despite intensive research efforts, the problem of proving or disproving the LEC remains wide open. In light of the lack of progress on general total functions, several authors [5, 31, 18, 40] have proposed studying *composed* communication problems of the form $F = f \circ g$, where $g : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$ is a small, preferably constant-size, gadget. Among such restricted classes, AND-functions of the form

$$F(x,y) = f(x_1 \wedge y_1, \ldots, x_n \wedge y_n)$$

have received particular attention. This is partly motivated by the fact that some of the most studied problems in communication complexity, such as *Set Disjointness* and *Inner Product*, are AND-functions.

To the best of our knowledge, AND-functions, especially in the context of quantum communication complexity, were first systematically studied by Buhrman and de Wolf [5]. They showed that for this class, deterministic communication complexity and zero-error quantum communication complexity are polynomially related when the outer function f is symmetric or monotone. However, the relationship between bounded-error quantum and randomized communication complexity for AND-functions remained poorly understood at the time. Indeed, very few lower bounds were known against bounded-error quantum protocols, essentially limited to those – such as Inner Product – obtained via the discrepancy method [21]. In particular, no polynomial lower bound on the quantum communication complexity of Set Disjointness was known.

In a major breakthrough, Razborov [31] established the optimal $\Omega(\sqrt{n})$ lower bound for the bounded-error quantum communication complexity of Set Disjointness over a universe of size n. More generally, his method yielded that for AND-functions $f \circ \text{AND}_2$ with symmetric outer function f, the bounded-error quantum communication complexity is polynomially equivalent even to the deterministic communication complexity. Extending this result to *all* AND-functions remained open since then and this has been investigated by several authors [5, 31, 18, 35, 40, 37]. Our main result resolves this problem.

▶ **Theorem 1.2.** *Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function. Then:*

1. $D^{cc}(f \circ \text{AND}_2) = O\big(Q^{cc}(f \circ \text{AND}_2)^7 \cdot (\log n)^2\big)$.
2. $D^{cc}(f \circ \text{AND}_2) = O\big(R^{cc}(f \circ \text{AND}_2)^5 \cdot (\log n)^2\big)$.

Ignoring polylogarithmic factors in n, this shows that for all AND-functions, deterministic communication complexity is polynomially equivalent to bounded-error quantum communication complexity. Notably, prior to our work it was not even known whether randomized communication complexity is polynomially equivalent to deterministic complexity for this class of functions. Before describing our results, we briefly review prior developments related to this question.

While not stated explicitly in Razborov's work, his quantum lower bound for Set Disjointness can be used to obtain lower bounds for a broader class of composed functions. In particular, consider any constant-size gadget $g : \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$ whose communication matrix contains both AND_2 and OR_2 as submatrices. For such gadgets, the bounded-error quantum communication complexity of $f \circ g$ is $\Omega(\sqrt{\text{bs}(f)})$ [43], where $\text{bs}(f)$ denotes the block sensitivity of f. This follows from the fact that a promised Set Disjointness instance of size $\text{bs}(f)$ can be embedded into $f \circ g$.

Combining this lower bound with Nisan's classical result [27], which upper bounds the deterministic query complexity of f by a polynomial in $\text{bs}(f)$, yields a quantum–classical equivalence for composed functions $f \circ g$ whenever g embeds both AND_2 and OR_2. We refer to this class as AND-OR-*functions*.

Subsequently, using different techniques, Sherstov [35] gave an independent proof of quantum–classical equivalence for AND-OR-functions via his pattern matrix method. In related and independent work, Shi and Zhu [40] proved quantum–classical equivalence for composed functions with gadgets satisfying certain pseudorandomness properties, although their gadgets were required to have size $\Omega(\log n)$.

Another important line of work concerns XOR-functions, i.e., functions composed with the XOR_2 gadget. Shi and Zhang [44] showed that, up to polylogarithmic factors, the Log-Equivalence Conjecture holds for XOR-functions when the outer function f is symmetric. Subsequently, Montanaro and Osborne [26] proved a polynomial equivalence between deterministic and zero-error quantum communication complexity for XOR-functions with monotone outer functions. With the AND-function case resolved in this work, extending the Log-Equivalence Conjecture to XOR-functions in full generality emerges as a natural next milestone toward a complete understanding of the conjecture.

More recently, Chattopadhyay, Dahiya, and Lovett [8] revisited the Log-Equivalence Conjecture. In all previously known classes of functions satisfying the conjecture, an even stronger statement holds: deterministic communication complexity is polynomially related to bounded-error quantum communication complexity. In [8], the authors studied composed functions of the form $f \circ \text{EQ}_4$, where EQ_4 denotes equality on four bits. Since $\text{AND} \circ \text{EQ}_4$ (which is just the equality function) belongs to this class, deterministic communication complexity is exponentially separated from randomized communication complexity; nevertheless, they showed that the Log-Equivalence Conjecture continues to hold (up to polylogarithmic factors). Our work builds on and extends the ideas developed in [8], which we elaborate on in subsequent sections.

1.1 Our Results

For any Boolean function $f : \{0,1\}^n \to \{0,1\}$, our main result shows that the bounded-error quantum and deterministic communication complexities of the AND-function $f \circ \text{AND}_2$ are polynomially related, up to polylogarithmic factors in n. That is,

$$\text{D}^{cc}(f \circ \text{AND}_2) = \text{Q}^{cc}(f \circ \text{AND}_2)^{O(1)} \cdot (\log n)^{O(1)}.$$

To establish such a relationship, it is helpful to characterize $\mathrm{D}^{cc}(f \circ \textsc{And}_2)$ in terms of structural properties of the outer function f. This was achieved by Knop, Lovett, McGuire, and Yuan [19], who showed that the deterministic communication complexity of $f \circ \textsc{And}_2$ is characterized – up to polynomial loss and ignoring poly-$\log(n)$ factors – by the logarithm of the *De Morgan sparsity* of f.

Recall that every Boolean function f admits a unique multilinear polynomial representation over the reals,

$$f(x) = \sum_{S \subseteq [n]} a_S \prod_{i \in S} x_i.$$

The (De Morgan) *sparsity* of f, denoted $\mathrm{spar}(f)$, is the number of nonzero coefficients a_S.

Knop et al. [19] showed that the deterministic communication complexity of an $\textsc{And}$-function satisfies

$$\mathrm{D}^{cc}(f \circ \textsc{And}_2) = O\big((\log \mathrm{spar}(f))^5 \cdot \log n\big).$$

This characterization has an immediate and important consequence. Since the sparsity of f coincides with the rank of the communication matrix of $f \circ \textsc{And}_2$, this bound yields a resolution of the log-rank conjecture for $\textsc{And}$-functions, up to a $\log n$ factor.

Given this characterization of $\mathrm{D}^{cc}(f \circ \textsc{And}_2)$, our task reduces to showing that large sparsity of f forces large bounded-error quantum communication complexity. Concretely, we show that

$$\mathrm{Q}^{cc}(f \circ \textsc{And}_2) = (\log \mathrm{spar}(f))^{\Omega(1)},$$

ignoring polylogarithmic factors in n. In fact, we prove a stronger statement: large sparsity of f implies a large *approximate* γ_2 norm of the communication matrix $M_{f \circ \textsc{And}_2}$. This is a strengthening of a direct lower bound on quantum communication complexity, since the approximate γ_2 norm is a known lower bound on bounded-error quantum communication complexity.

The approximate γ_2 norm of a two-party Boolean function $F : X \times Y \to \{0,1\}$, denoted $\widetilde{\gamma}_2(F)$, is defined as the minimum total weight of a rectangle decomposition that approximates the communication matrix M_F within constant error:

$$\widetilde{\gamma}_2(F) = \min\Big\{ \sum_i |\alpha_i| \;\Big|\; \big\| M_F - \sum_i \alpha_i R_i \big\|_\infty \le 1/3 \Big\},$$

where each $R_i(x,y) = g_i(x)h_i(y)$ is a combinatorial rectangle.

Any bounded-error quantum protocol for F using c qubits of communication induces a pointwise approximation of M_F of the form

$$M_F \approx \sum_i \alpha_i R_i,$$

with total weight $\sum_i |\alpha_i| \le 2^{O(c)}$. While such decompositions are immediate for deterministic and randomized protocols, they also hold in the quantum setting by unpacking the definition of quantum communication protocols (see [25]).

Our main technical contribution is to show that large sparsity of f lifts to a lower bound on the approximate γ_2 norm of $f \circ \textsc{And}_2$.

▶ **Theorem 1.3.** *For every total Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\log \widetilde{\gamma}_2(f \circ \textsc{And}_2) = \Omega\left(\left(\frac{\log \mathit{spar}(f)}{\log n} \right)^{1/3} \right).$$

We now record several immediate consequences of Theorem 1.3. Recall that for a communication problem $F : X \times Y \to \{0,1\}$, the *rank* of F is the rank (over the reals) of its communication matrix M_F, and the *approximate rank*, denoted $\widetilde{\mathrm{rank}}(F)$, is the minimum rank of a real matrix that approximates M_F entrywise within a small error $1/3$.

The approximate γ_2 norm can be viewed as a convex relaxation of the approximate rank and is known to be essentially equivalent to it. It is shown in [23] that, on the logarithmic scale, the two measures coincide up to an additive $O(\log\log|X||Y|)$ term: for every communication problem $F : X \times Y \to \{0,1\}$,

$$\Omega\big(\log\widetilde{\gamma}_2(F)\big) \;\le\; \log\widetilde{\mathrm{rank}}(F) \;\le\; O\big(\log\widetilde{\gamma}_2(F) + \log\log|X||Y|\big).$$

Combining this observation with our result and the result of Knop et al. [19], we obtain

$$\begin{aligned}\mathrm{R}^{cc}(f\circ\textsc{And}_2) \le \mathrm{D}^{cc}(f\circ\textsc{And}_2) &= O\big((\log\mathrm{spar}(f))^5\cdot\log n\big)\\ &= O\Big((\log\widetilde{\mathrm{rank}}(f\circ\textsc{And}_2))^{15}\cdot(\log n)^6\Big).\end{aligned}$$

While the exponents above are not optimized and can be improved (see Section 4.2), this implies that the *log-approximate-rank conjecture* holds for AND-functions, up to poly-log(n) factors.

The Log-Approximate-Rank Conjecture (LARC), a term first coined by Lee and Shraibman [24], is a natural approximate analogue of the classical Log-Rank conjecture. It asserts that, just as the logarithm of the rank of the communication matrix is believed to characterize deterministic communication complexity up to polynomial loss, the logarithm of the approximate rank should similarly characterize randomized communication complexity up to polynomial loss.

Somewhat surprisingly, the LARC was recently shown to be false by Chattopadhyay, Mande, and Sherif [9], by exhibiting an XOR-function as a counterexample. In contrast, our results show that LARC *does* hold for AND-functions, up to polylogarithmic factors in n.

In fact, our results give the following chain of inequalities:

$$\begin{aligned}&\Omega\big(\log\widetilde{\gamma}_2(f\circ\textsc{And}_2)\big) \;\le\; \mathrm{Q}^{cc}(f\circ\textsc{And}_2) \;\le\; \mathrm{R}^{cc}(f\circ\textsc{And}_2) \;\le\; \mathrm{D}^{cc}(f\circ\textsc{And}_2),\\ &\qquad \mathrm{D}^{cc}(f\circ\textsc{And}_2) \;\le\; (\log\mathrm{spar}(f))^{O(1)}(\log n)^{O(1)} \;\le\; \log\widetilde{\gamma}_2(f\circ\textsc{And}_2)^{O(1)}(\log n)^{O(1)}.\end{aligned}$$

As a result, for an AND-function $F := f\circ\textsc{And}_2$, up to polylogarithmic factors in n, the quantities

$$\mathrm{D}^{cc}(F),\ \mathrm{R}^{cc}(F),\ \mathrm{Q}^{cc}(F),\ \log\widetilde{\gamma}_2(F),\ \log\widetilde{\mathrm{rank}}(F),\ \log\mathrm{rank}(F),\ \text{and}\ \log\mathrm{spar}(f)$$

are all polynomially equivalent. In other words, for AND-functions, diverse notions of complexity – from communication-theoretic to algebraic/analytic – coincide. Interestingly, this striking equivalence of various measures was suspected in the early work of Buhrman and de Wolf [5] on AND-functions more than two decades ago. Our results finally confirm their intuition.

1.2 Proof overview

As discussed above, our main technical contribution is Theorem 1.3, which lifts the sparsity of a Boolean function f to a lower bound on the approximate γ_2-norm of the lifted function $f\circ\textsc{And}_2$. We now give a high-level overview of the proof.

We begin with a concrete example that will later help illustrate the general argument. The canonical Boolean function with large sparsity is the n-bit $\textsc{Or}_n$ function, whose sparsity equals 2^n-1. When composed with the $\textsc{And}_2$ gadget, the function $\textsc{Or}_n\circ\textsc{And}_2$ corresponds to the *Set Intersection* problem, the negation of the well-known Set Disjointness function.

Lower bound for Set Intersection. We illustrate our approach using the Set Intersection function

$$(\mathrm{OR}_n \circ \mathrm{AND}_2)(x, y) \;=\; \mathrm{OR}_n(x_1 \wedge y_1, \ldots, x_n \wedge y_n),$$

where we write $z_i := x_i \wedge y_i$.

The key structural property of OR_n is that it remains an OR under every restriction $\rho_z \in \{0, *\}^n$: the restricted function retains full degree on the free variables. This hardness under $\{0, *\}$-restrictions is precisely what gives OR_n its large sparsity. More importantly, as we show later, the existence of many such *max-degree* restrictions is a general consequence of large sparsity, allowing the argument to extend beyond Set Intersection to arbitrary AND-functions.

To lift this hardness to the communication setting, we map restrictions on the z-variables to restrictions on the input pairs (x_i, y_i). Given $\rho_z \in \{0, *\}^n$, we define a lifted restriction ρ by

$$(\rho(x_i), \rho(y_i)) = \begin{cases} (\Delta, 0), & \text{if } \rho_z(z_i) = 0, \\ (*, 1), & \text{if } \rho_z(z_i) = *, \end{cases}$$

so that $(x_i \wedge y_i)|_\rho = \rho_z(z_i)$. Here Δ denotes a free but *masked* variable: although syntactically free, the restricted function does not depend on it.

These lifted restrictions have three crucial properties: (i) all y-variables are fixed, (ii) the restricted function $(\mathrm{OR}_n \circ \mathrm{AND}_2)|_\rho$ computes an OR on exactly the $*$-variables, and (iii) it is independent of the masked variables. As a result, if ρ_z has exactly d stars, then

$$\deg\big(\mathbb{E}_{x_{M(\rho)}}[(\mathrm{OR}_n \circ \mathrm{AND}_2)|_\rho]\big) = d,$$

where $M(\rho)$ denotes the masked variables and $\mathbb{E}_{x_{M(\rho)}}[\cdot]$ denotes expectation over independent uniform assignments to them.

This motivates the following *restriction-and-averaging procedure*. We sample an unlifted restriction $\rho_z \in \{0, *\}^n$ uniformly at random subject to having $d = \Theta(n^{2/3})$ stars (we will explain shortly why this specific parameter is chosen), lift it to obtain ρ, and then take expectation over the masked variables.

Under this restriction-and-averaging procedure, $\mathrm{OR}_n \circ \mathrm{AND}_2$ retains its hardness – in terms of degree – with exact degree $\Theta(n^{2/3})$. On the other hand, we show that the same procedure simplifies any small-weight rectangle decomposition arising from a small approximate γ_2 norm. Concretely, it converts such a decomposition into a low-degree polynomial (more precisely, one whose total Fourier mass on high-degree monomials is small).

This leads to a contradiction. Indeed, from a small approximate γ_2 norm of size $2^{O(n^{1/3})}$ for $\mathrm{OR}_n \circ \mathrm{AND}_2$, we would obtain a polynomial of degree $O(n^{1/3})$ that approximates $\mathbb{E}_{x_{M(\rho)}}[(\mathrm{OR}_n \circ \mathrm{AND}_2)|_\rho]$, even though the latter has exact degree $\Theta(n^{2/3})$. This contradicts the known quadratic relationship between degree and approximate degree, forcing the approximate γ_2 norm of $\mathrm{OR}_n \circ \mathrm{AND}_2$ to be $2^{\Omega(n^{1/3})}$. Extracting low-degree approximating polynomials from a decomposition exhibiting low approximate γ_2 norm of a matrix, in the context of communication complexity, was initiated in the work of Razborov [31], followed by other works, including that of [6, 38]. Our particular method is inspired by the work of [8] and the work of Krause and Pudlák [20] who used it in the context of proving lower bounds on the Fourier sparsity of lifted functions.

We now briefly explain why our restriction-and-averaging procedure simplifies a small-weight rectangle decomposition into a low-degree polynomial. Let

$$\Pi(x,y) \;=\; \sum_{i=1}^{m} b_i\, g_i(x)\, h_i(y)$$

be an approximator for Set Intersection with small total weight $\sum_i |b_i|$. Under any lifted restriction, all y-variables are fixed, so each rectangle collapses to a function of the x-variables alone. Moreover, for any Boolean function $g : \{0,1\}^n \to \{0,1\}$ on x-variables, we show that the expected Fourier mass of g above level $k \geq n^{1/3}$ after restriction and averaging decays as $2^{-\Omega(k)}$ (this is where our choice of $d = \Theta(n^{2/3})$ plays a part). Intuitively, this happens because any Fourier character involving a masked variable vanishes upon averaging. An averaging argument then yields a restriction ρ such that the Fourier mass of $\mathbb{E}_{x_{M(\rho)}}[\Pi|_\rho]$ above level $k = O(n^{1/3})$ is negligible, provided $\sum_i |b_i| \leq 2^{O(n^{1/3})}$. Discarding higher-degree monomials produces a low-degree $O(n^{1/3})$ approximator for $\mathbb{E}_{x_{M(\rho)}}[(\mathrm{OR}_n \circ \mathrm{AND}_2)|_\rho]$, contradicting its degree $\Theta(n^{2/3})$.

This shows a $2^{\Omega(n^{1/3})}$ lower bound on the approximate γ_2 norm of Set Intersection. Moreover, if you observe closely, the same argument also yields a $2^{\Omega(n^{1/3})}$ lower bound for the approximate γ_2 norm of Inner Product function $\mathrm{XOR}_n \circ \mathrm{AND}_2$, another central problem in communication complexity.

Generalization to arbitrary functions with large sparsity. In the discussion above for Set Intersection and Inner Product, we relied crucially on a structural property of the outer functions OR_n and XOR_n: under every restriction in $\{0,*\}^n$, the restricted function retains full degree on the surviving free variables. As hinted earlier, the existence of many such *max-degree* restrictions is not specific to these functions. Rather, it is a general consequence of large sparsity – and this is the key insight of our work that allows us to extend the argument to arbitrary AND-functions.

More concretely, we show that any Boolean function $f : \{0,1\}^n \to \{0,1\}$ of large sparsity admits a fixed set of variables $V \subseteq [n]$, which we call the *core variables*, with the following property. For every assignment $\alpha \in \{0,*\}^V$, the remaining variables in $[n] \setminus V$ can be fixed – possibly depending on α – so that the resulting restriction of f has full degree on the free variables. We refer to the resulting collection of restrictions as a *semi-adaptive max-degree restriction tree*. For OR_n and XOR_n, the core set is simply $V = [n]$, but in general this need not be the case. For a function of sparsity s, we show that one can always find V of size $\Omega(\log s / \log n)$, which we call the *depth* of the restriction tree.

Having obtained such a collection of restrictions $\mathcal{D}$ for f, we lift them to restrictions for the two-party function $f \circ \mathrm{AND}_2$, extending the construction used for Set Intersection. Given a restriction $\rho_z \in \mathcal{D}$, we define a lifted restriction ρ on the variables (x_i, y_i) by

$$(\rho(x_i), \rho(y_i)) = \begin{cases} (\Delta, 0), & \text{if } \rho_z(z_i) = 0 \text{ and } i \in V, \\ (*, 1), & \text{if } \rho_z(z_i) = * \text{ and } i \in V, \\ (1, 1), & \text{if } \rho_z(z_i) = 1 \text{ and } i \notin V, \\ (1, 0), & \text{if } \rho_z(z_i) = 0 \text{ and } i \notin V, \end{cases}$$

ensuring that $(x_i \wedge y_i)|_\rho = \rho_z(z_i)$. As in the Set Intersection case, this construction guarantees that $(f \circ \mathrm{AND}_2)|_\rho$ coincides with $f|_{\rho_z}$ (up to renaming of variables) and is independent of the masked variables $M(\rho)$. As a result, for every max-degree restriction ρ_z,

$$\deg\big(\mathbb{E}_{x_{M(\rho)}}\big[(f \circ \mathrm{AND}_2)|_\rho\big]\big) = \deg(f|_{\rho_z}) = |\rho_z^{-1}(*)| = |\rho^{-1}(*)|.$$

Thus, the lifted function $f \circ \textsc{And}_2$ retains its hardness under the same restriction-and-averaging procedure used for Set Intersection. Moreover, the lifting has an additional convenient feature: all non-core x-variables are fixed to 1. This allows the same Fourier-analytic argument to go through, showing that the restriction-and-averaging procedure simplifies any small-weight rectangle decomposition into a low-degree polynomial.

The only quantitative change from the Set Intersection analysis is that the effective parameter n is replaced by the number of core variables $|V| = \Omega(\log s / \log n)$. Carrying out the argument yields

$$\log \widetilde{\gamma}_2(f \circ \textsc{And}_2) = \Omega\left(\left(\frac{\log s}{\log n}\right)^{1/3}\right),$$

which is exactly the bound stated in Theorem 1.3.

Finally, we place our work in the context of the results of Chattopadhyay, Dahiya, and Lovett [8]. Among other results, that work studies the sparsity of Boolean functions and provides a structural characterization in terms of *max-degree restriction trees*. Their restriction trees are fully adaptive: the choice of which variable to restrict next depends on the outcomes of previous restrictions, and there is no fixed set of core variables, as in our setting. Moreover, their framework applies to a broader class of complexity measures, which they call one-sided and two-sided nice measures.

While this adaptive viewpoint is powerful, we do not know how to use such fully adaptive restriction trees to control analytic quantities such as the approximate γ_2 norm. In contrast, our work constructs a more structured, *semi-adaptive* form of max-degree restriction trees directly from sparsity, with a fixed core set of variables on which all $\{0, *\}$-restrictions occur. This additional structure is crucial for our analysis and enables us to lift sparsity to lower bounds on the approximate γ_2 norm.

Organization

In Section 2, we introduce the necessary preliminaries and notation. In Section 3, we formally define semi-adaptive max-degree restriction trees and show how to construct them for Boolean functions with large sparsity. Finally, in Section 4, we prove our main technical contribution: a lifting theorem that lifts the sparsity of a Boolean function f into a lower bound on the approximate γ_2 norm of its lifted function $f \circ \textsc{And}_2$, and we discuss the resulting consequences for And-functions.

2 Preliminaries

All functions considered are defined on the Boolean hypercube $\{0,1\}^n$ and unless stated otherwise, all polynomials are real and multilinear.

Multilinear representations

Over the Boolean domain, Boolean functions admit a canonical polynomial representation.

▶ **Definition 2.1** (Multilinear polynomial representation). *A polynomial $Q \in \mathbb{R}[x_1, \ldots, x_n]$ is* multilinear *if each variable appears with degree at most one in every monomial. Every function $f : \{0,1\}^n \to \mathbb{R}$ admits a unique multilinear polynomial representation; that is, there exists a unique multilinear polynomial $Q \in \mathbb{R}[x_1, \ldots, x_n]$ such that $Q(x) = f(x)$ for all $x \in \{0,1\}^n$.*

We freely identify Boolean functions with their unique multilinear polynomial representations, and use the two interchangeably.

Polynomial complexity measures

Let $Q \in \mathbb{R}[x_1, \ldots, x_n]$ be a multilinear polynomial written as

$$Q(x) = \sum_{S \subseteq [n]} a_S \prod_{i \in S} x_i.$$

The *degree* of Q is $\deg(Q) := \max\{|S| : a_S \neq 0\}$, and the *sparsity* of Q, denoted $\mathrm{spar}(Q)$, is the number of nonzero coefficients a_S.

For a function $f : \{0,1\}^n \to \mathbb{R}$, let $\mathcal{P}(f)$ denote its unique multilinear polynomial representation. We define

$$\deg(f) := \deg(\mathcal{P}(f)), \qquad \mathrm{spar}(f) := \mathrm{spar}(\mathcal{P}(f)).$$

Approximate degree

Let $f : \{0,1\}^n \to \mathbb{R}$ and $\varepsilon > 0$. The *ε-approximate degree* of f is defined as

$$\widetilde{\deg}_\varepsilon(f) := \min\big\{\deg(Q) : |Q(x) - f(x)| \leq \varepsilon \text{ for all } x \in \{0,1\}^n\big\}.$$

When $\varepsilon = 1/3$, we write $\widetilde{\deg}(f) := \widetilde{\deg}_{1/3}(f)$.

▶ **Theorem 2.2** ([7, Theorem 10, Section 3.4]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. For any $0 < \varepsilon < 1/2$,*

$$\widetilde{deg}_\varepsilon(f) = O\big(\widetilde{deg}(f) \cdot \log(1/\varepsilon)\big).$$

▶ **Theorem 2.3** ([1, Theorem 4]). *For every Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$deg(f) = O\big(\widetilde{deg}(f)^2\big).$$

Fourier basis

Another fundamental representation of Boolean functions is given by the *Fourier basis*, where each monomial $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$ represents the ± 1-valued parity function on the subset $S \subseteq [n]$ of variables.

▶ **Definition 2.4** (Fourier complexity measures). *Let $f : \{0,1\}^n \to \mathbb{R}$ have the Fourier expansion*

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\, \chi_S(x).$$

The Fourier degree *of f is the maximum size of a set $S \subseteq [n]$ with $\widehat{f}(S) \neq 0$. Since $\chi_S(x) = \prod_{i \in S}(1 - 2x_i)$, the Fourier degree coincides with the ordinary degree.*

The Fourier sparsity *of f, denoted $\left\|\widehat{f}\right\|_0$, is the number of nonzero Fourier coefficients $\widehat{f}(S)$. The* Fourier ℓ_1-norm *of f is*

$$\left\|\widehat{f}\right\|_1 := \sum_{S \subseteq [n]} |\widehat{f}(S)|.$$

For $t \geq 0$, the Fourier ℓ_1-mass above level t *is*

$$\left\|\widehat{f}\right\|_1^{\geq t} := \sum_{\substack{S \subseteq [n] \\ |S| \geq t}} |\widehat{f}(S)|.$$

Communication complexity

We assume familiarity with the standard model of communication complexity and refer the reader to [22] for background. In this model, two parties – Alice and Bob – aim to compute a Boolean function $F : X \times Y \to \{0,1\}$, where Alice receives $x \in X$ and Bob receives $y \in Y$. The *deterministic communication complexity* of F, denoted $\mathrm{D}^{cc}(F)$, is the minimum number of bits exchanged by any deterministic protocol that always outputs $F(x,y)$. In the public-coin randomized model, Alice and Bob have access to shared randomness and must compute $F(x,y)$ with error at most $1/3$; the corresponding measure is the *randomized communication complexity* $\mathrm{R}^{cc}(F)$.

We also assume familiarity with quantum communication complexity [11]. We use $\mathrm{Q}^{cc}(F)$ to denote the bounded-error (error at most $1/3$) quantum communication complexity of F in the model with unlimited shared entanglement.

The γ_2 and approximate γ_2 norms

▶ **Definition 2.5** (γ_2 and approximate γ_2 norms). *Let $F : X \times Y \to \{0,1\}$ be a Boolean function, and let $M_F \in \{0,1\}^{X \times Y}$ denote its communication matrix, defined by $M_F(x,y) = F(x,y)$.*

The γ_2 norm *of F is*

$$\gamma_2(F) = \min\Big\{\sum_i |\alpha_i| : M_F = \sum_i \alpha_i R_i\Big\},$$

where each R_i is a combinatorial rectangle*, i.e., $R_i(x,y) = g_i(x)h_i(y)$ for Boolean functions $g_i : X \to \{0,1\}$ and $h_i : Y \to \{0,1\}$.*

For $0 < \varepsilon < 1/2$, the ε-approximate γ_2 norm *is*

$$\gamma_2^{\varepsilon}(F) = \min\Big\{\gamma_2(A) : A \in \mathbb{R}^{X \times Y},\ \|A - M_F\|_\infty \le \varepsilon\Big\}.$$

We write $\widetilde{\gamma}_2(F) := \gamma_2^{1/3}(F)$.

▶ Remark 2.6. The γ_2 norm is often defined via a factorization-based formulation. For a real matrix M, one may equivalently define

$$\gamma_2(M) = \min_{X,Y : XY^\top = M} r(X)\, r(Y),$$

where $r(X)$ denotes the maximum ℓ_2-norm of a row of X. This formulation, sometimes called the *μ-norm*, is equivalent to the rectangle-based definition up to constant factors; see, e.g., Chapter 2 of [24]. We use the rectangle-based definition throughout.

▶ **Theorem 2.7** ([25, Theorem 1]). *For every Boolean function $F : X \times Y \to \{0,1\}$,*

$$R^{cc}(F) = \Omega(\log \widetilde{\gamma}_2(F)) \qquad \text{and} \qquad Q^{cc}(F) = \Omega(\log \widetilde{\gamma}_2(F)).$$

3 Semi-Adaptive Max-Degree Restriction Trees from Sparsity

▶ **Definition 3.1** (Restrictions). *A* restriction *ρ on a set of variables $V \subseteq \{x_1, \dots, x_n\}$ is a partial assignment $\rho : V \to \{0,1,*\}$, where for $x_i \in V$, $\rho(x_i) \in \{0,1\}$ indicates that x_i is fixed, and $\rho(x_i) = *$ means x_i is left free. For a polynomial $Q \in \mathbb{R}[x_1, \dots, x_n]$, we write $Q|_\rho$ for the polynomial obtained by substituting $x_i = \rho(x_i)$ for all fixed variables x_i.*

A central tool in our work is to extract structural consequences of a function having large polynomial sparsity. In particular, we seek restrictions under which a function remains maximally hard, in the sense of retaining full degree.

At a high level, large sparsity guarantees the existence of many restrictions under which the function retains full degree. More concretely, if a multilinear polynomial has sparsity s, then there exists a set of variables V of size $\Omega(\log s/\log n)$ such that, for every assignment of the variables in V to values in $\{0,*\}$, one can fix the remaining variables so that the restricted polynomial has full degree in the variables left free.

We capture this collection of restrictions using what we call *semi-adaptive max-degree restriction trees*. The term *semi-adaptive* reflects the following structure. There is a fixed set of variables V such that every assignment in $\{0,*\}^V$ appears as a restriction, rather than variables being chosen adaptively based on previous assignments, as in fully adaptive restriction trees (e.g., in the work of [8]). However, the restrictions are not fully non-adaptive: although the set V is fixed, for each assignment $\rho \in \{0,*\}^V$, the fixing of the remaining variables in $[n] \setminus V$ that ensures full degree may depend on ρ. This intermediate structure motivates the term *semi-adaptive*.

The qualifier *max-degree* indicates that under every restriction in the tree, the polynomial retains full degree on the variables that remain free. This notion compactly encodes the key structural consequence of large sparsity that we exploit later.

We now formalize this notion.

▶ **Definition 3.2** (Max-degree restriction). *Let $Q \in \mathbb{R}[x_1,\dots,x_n]$ be a nonzero multilinear polynomial and let $\rho : \{x_1,\dots,x_n\} \to \{0,1,*\}$ be a restriction. We say that ρ is a* max-degree restriction *of Q if $deg(Q|_\rho) = |\rho^{-1}(*)|$, that is, the restricted polynomial has full degree in its free variables.*

▶ **Definition 3.3** (Semi-adaptive restriction tree). *A* semi-adaptive restriction tree of depth *d on n variables is a collection $\mathcal{D}$ of 2^d restrictions $\rho : \{x_1,\dots,x_n\} \to \{0,1,*\}$ for which there exists a fixed set of variables $V \subseteq \{x_1,\dots,x_n\}$ with $|V| = d$, called the* core variables*, such that:*

- *For every $\rho \in \mathcal{D}$ and every $x_i \notin V$, $\rho(x_i) \in \{0,1\}$.*
- *For every assignment $\alpha \in \{0,*\}^V$, there exists a unique $\rho \in \mathcal{D}$ such that $\rho(x_i) = \alpha(x_i)$ for all $x_i \in V$.*

Equivalently, $\mathcal{D}$ consists of all restrictions obtained by assigning each variable in V either 0 or $*$, while fixing all variables outside V as a function of this assignment.

▶ **Definition 3.4** (Semi-adaptive max-degree restriction tree). *Let $Q \in \mathbb{R}[x_1,\dots,x_n]$ be a multilinear polynomial and let $\mathcal{D}$ be a semi-adaptive restriction tree. We say that $\mathcal{D}$ is a* max-degree restriction tree *for Q if every $\rho \in \mathcal{D}$ is a max-degree restriction of Q, i.e., $deg(Q|_\rho) = |\rho^{-1}(*)|$.*

Examples

To illustrate the definitions, we give two informative examples.

▶ **Example 3.5.** OR_n. The function $\mathrm{OR}_n(x_1,\dots,x_n)$ has sparsity $2^n - 1$. It admits a semi-adaptive max-degree restriction tree of depth n with core variables $V = \{x_1,\dots,x_n\}$. Let $\mathcal{D} = \{\, \rho : \{x_1,\dots,x_n\} \to \{0,*\} \,\}$. For any $\rho \in \mathcal{D}$, the restricted function $\mathrm{OR}_n|_\rho$ is an OR over the free variables $\rho^{-1}(*)$, and hence $\deg(\mathrm{OR}_n|_\rho) = |\rho^{-1}(*)|$. Thus $\mathcal{D}$ is a semi-adaptive max-degree restriction tree for OR_n.

▶ **Example 3.6.** $\text{AND}_n \circ \text{OR}_2$. Consider

$$(\text{AND}_n \circ \text{OR}_2)(x_1, \ldots, x_n, y_1, \ldots, y_n) = \text{AND}_n(\text{OR}_2(x_1, y_1), \ldots, \text{OR}_2(x_n, y_n)),$$

which has sparsity 3^n. We construct a semi-adaptive max-degree restriction tree of depth n with core variables $V = \{x_1, \ldots, x_n\}$. For a restriction $\rho_x \in \{0, *\}^{\{x_1, \ldots, x_n\}}$, define a restriction $\rho_y^{\rho_x}$ on $\{y_1, \ldots, y_n\}$ by

$$\rho_y^{\rho_x}(y_i) = \begin{cases} 0, & \text{if } \rho_x(x_i) = *, \\ 1, & \text{if } \rho_x(x_i) = 0. \end{cases}$$

Set $\mathcal{D} = \left\{ \rho_x \cup \rho_y^{\rho_x} \;\middle|\; \rho_x \in \{0, *\}^{\{x_1, \ldots, x_n\}} \right\}$. For any $\rho \in \mathcal{D}$, each gate $\text{OR}_2(x_i, y_i)$ evaluates to a free variable when $\rho(x_i) = *$, and to the constant 1 when $\rho(x_i) = 0$. As a result, $(\text{AND}_n \circ \text{OR}_2)|_\rho$ computes an AND over exactly the free core variables $\rho^{-1}(*)$, and therefore $\deg\big((\text{AND}_n \circ \text{OR}_2)|_\rho\big) = |\rho^{-1}(*)|$. Thus, $\mathcal{D}$ is a semi-adaptive max-degree restriction tree of depth n for $\text{AND}_n \circ \text{OR}_2$.

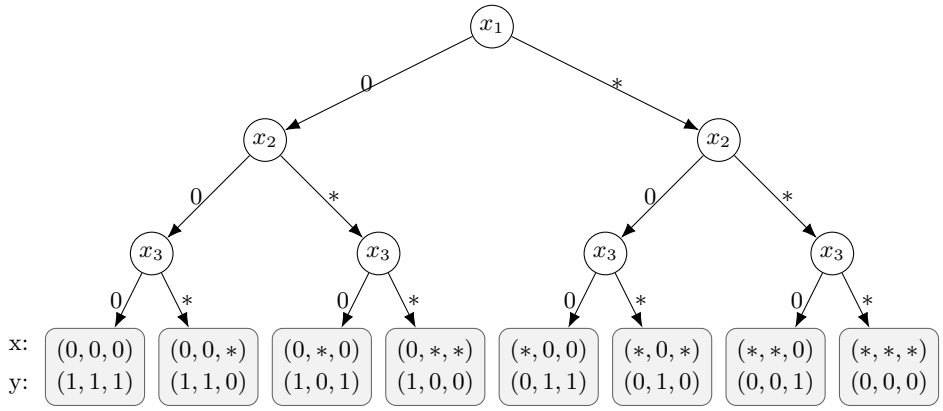


Figure 1 A semi-adaptive max-degree restriction tree for $\text{AND}_3 \circ \text{OR}_2$ with core variables $V = \{x_1, x_2, x_3\}$. Leaves correspond to the $2^{|V|}$ restrictions in the tree. For each leaf restriction, the restricted function $(\text{AND}_3 \circ \text{OR}_2)|_\rho$ computes an AND over exactly the variables left free.

The following lemma shows that large sparsity guarantees the existence of deep semi-adaptive max-degree restriction trees.

▶ **Lemma 3.7.** *Let $Q : \{0,1\}^n \to \mathbb{R}$ be a nonzero multilinear polynomial of sparsity s. Then Q admits a semi-adaptive max-degree restriction tree of depth $\Omega(\log s / \log n)$.*

Proof. Let $\mathcal{M}_Q \subseteq 2^{\{x_1, \ldots, x_n\}}$ denote the family of supports of monomials appearing in Q. Let $V \subseteq \{x_1, \ldots, x_n\}$ be a largest set shattered by $\mathcal{M}_Q$. Recall that V being shattered by $\mathcal{M}_Q$ means that for every $S \subseteq V$, there exists $T \in \mathcal{M}_Q$ such that $T \cap V = S$. The Sauer–Shelah–Perles lemma [33, 34] states that that for a set system $\mathcal{F}$ containing subsets of $[n]$, if d is the maximum size of a set shattered by $\mathcal{F}$, then $|\mathcal{F}| \leq O(n^d)$. Using this lemma, we obtain $|V| = \Omega(\log s / \log n)$; write $|V| = d$.

For each subset $S \subseteq V$, we construct a restriction ρ_S. The resulting family $\{\rho_S : S \subseteq V\}$ will form a semi-adaptive max-degree restriction tree for Q.

Fix an arbitrary $S \subseteq V$. Define the *free restriction* $\rho_S^{\text{free}} : V \to \{0, *\}$ by setting $\rho_S^{\text{free}}(x) = *$ for all $x \in S$ and $\rho_S^{\text{free}}(x) = 0$ for all $x \in V \setminus S$. Under this restriction, every monomial of Q containing a variable from $V \setminus S$ vanishes. As a result, the restricted polynomial can be written as $Q|_{\rho_S^{\text{free}}} = \sum_{T \subseteq S} \left(\prod_{x \in T} x \right) \cdot R_T$, where each R_T is a multilinear polynomial over the variables $\{x_i : x_i \notin V\}$.

Since V is shattered by $\mathcal{M}_Q$, there exists a monomial of Q whose support intersects V exactly in S. Equivalently, the coefficient polynomial R_S is nonzero. We now fix the remaining variables to witness this nonzeroness. Choose a *fixing restriction* $\rho_S^{\text{fix}} : \{x_1, \ldots, x_n\} \setminus V \to \{0,1\}$ such that $R_S|_{\rho_S^{\text{fix}}} \neq 0$.

Let $\rho_S := \rho_S^{\text{free}} \cup \rho_S^{\text{fix}}$. Then $Q|_{\rho_S}$ has degree exactly $|S|$, and its set of free variables is $\rho_S^{-1}(*) = S$. Hence, ρ_S is a max-degree restriction of Q.

Therefore, the family $\{\rho_S : S \subseteq V\}$ forms a semi-adaptive max-degree restriction tree of depth $|V| = \Omega(\log s / \log n)$. ◀

4 Lifting with the AND_2 Gadget

In this section, we present our main technical contribution: a lifting theorem that lifts the sparsity of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ into a lower bound on the approximate γ_2 norm of the lifted function $F := f \circ \text{AND}_2$. We begin with a high-level overview of the proof.

Proof overview

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function of sparsity s. Our goal is to lower bound the approximate γ_2 norm of the lifted function $f \circ \text{AND}_2$ in terms of $\log s$. Suppose, toward a contradiction, that $f \circ \text{AND}_2$ admits a small approximate γ_2 norm. Then there exists an approximator

$$\Pi(x, y) = \sum_{i=1}^{m} b_i \, g_i(x) h_i(y)$$

with $\sum_i |b_i|$ small. Our goal is to show that such an approximator cannot exist. The proof proceeds via a carefully designed random restriction argument that preserves the hardness of $f \circ \text{AND}_2$, in terms of degree, while simplifying any such approximator.

Step 1: Structure from sparsity. A key consequence of large sparsity is that f admits a *semi-adaptive max-degree restriction tree* $\mathcal{D}$ of depth $d = \Omega(\log s / \log n)$. Equivalently, there exists a fixed set of core variables $V \subseteq \{z_1, \ldots, z_n\}$, with $|V| = d$, such that for every assignment in $\{0, *\}^V$, the remaining variables can be fixed so that the restricted function has full degree in the surviving free variables. This provides a large family of restrictions under which f remains maximally hard in terms of degree.

Step 2: Lifting restrictions through the AND_2 gadget. We lift the restriction tree $\mathcal{D}$ for f to a collection of restrictions $\mathcal{D} \circ \text{AND}_2$ for the lifted function

$$(f \circ \text{AND}_2)(x_1, \ldots, x_n, y_1, \ldots, y_n) = f(\text{AND}_2(x_1, y_1), \ldots, \text{AND}_2(x_n, y_n)).$$

Each restriction $\rho_f \in \mathcal{D}$ is mapped to a restriction ρ on the variables (x_i, y_i) so that $\text{AND}_2(x_i, y_i)|_\rho = \rho_f(z_i)$ for every i. Concretely, the lifted restriction is given by

$$(\rho(x_i), \rho(y_i)) = \begin{cases} (\Delta, 0), & \text{if } \rho_f(z_i) = 0 \text{ and } z_i \in V, \\ (*, 1), & \text{if } \rho_f(z_i) = * \text{ and } z_i \in V, \\ (1, 1), & \text{if } \rho_f(z_i) = 1 \text{ and } z_i \notin V, \\ (1, 0), & \text{if } \rho_f(z_i) = 0 \text{ and } z_i \notin V. \end{cases}$$

Here both $*$ and Δ denote free variables. Algorithm 1 formalizes this construction.

The lifted restrictions satisfy two immediate properties. First, all y-variables are fixed under every lifted restriction. Second, the lifting introduces a special type of free variable, called a *masked variable*: these are free x_i-variables assigned Δ, whose corresponding y_i is fixed to 0; consequently, the restricted function is independent of them.

As a result, for every lifted restriction ρ, averaging over the masked variables yields a function that coincides (up to renaming variables) with $f|_{\rho_f}$. For a restriction ρ, we denote the set of masked variables by

$$M(\rho) := \{\, x_i \mid \rho(x_i) = \Delta \,\},$$

and write $\mathbb{E}_{x_{M(\rho)}}[\cdot]$ for expectation over independent, uniform assignments to the variables in $M(\rho)$. We will use this notation throughout.

Since $\mathcal{D}$ is a max-degree restriction tree, it follows that

$$\deg\big(\mathbb{E}_{x_{M(\rho)}}[(f \circ \textsc{And}_2)|_\rho]\big) = |\rho^{-1}(*)|.$$

Step 3: Random restrictions and Fourier decay. We place a uniform distribution over lifted restrictions $\rho \in \mathcal{D} \circ \textsc{And}_2$ having exactly pd many $*$-variables. We analyze the effect of sampling such a restriction and then taking expectation over the masked variables on the following:

1. *The target function $f \circ \textsc{And}_2$.* This operation preserves hardness: by construction, the resulting function has exact degree pd.
2. *An arbitrary Boolean function g on the x-variables.* We show that the same operation causes the Fourier ℓ_1-mass above level k to decay exponentially, provided $k \gtrsim p^2 d$.

To derive a contradiction later, we need the Fourier mass above level $\sqrt{pd}$ to be exponentially small, since this would imply an approximating polynomial of degree $O(\sqrt{pd})$ for the restricted target function. Since exponential decay is guaranteed only above level $p^2 d$, we choose parameters so that $p^2 d \approx \sqrt{pd}$. Solving this relation yields $p = \Theta(d^{-1/3})$, for which $pd = \Theta(d^{2/3})$ and $\sqrt{pd} = \Theta(d^{1/3})$. Thus the Fourier tail decays exponentially above the precise level needed for the contradiction.

Step 4: Deriving a contradiction. Applying this random restriction-and-averaging procedure to the approximator Π, each function $h_i(y)$ collapses to a constant, while the Fourier tails of the corresponding $g_i(x)$ terms decay rapidly. Since $\sum_i |b_i|$ is small, we conclude that the expected Fourier mass of $\mathbb{E}_{x_{M(\rho)}}[\Pi|_\rho]$ above level $k = \Theta(\sqrt{pd}) = \Theta(d^{1/3})$ is negligible.

Discarding this high-degree mass yields a polynomial of degree $O(\sqrt{pd})$ that still approximates $\mathbb{E}_{x_{M(\rho)}}[(f \circ \textsc{And}_2)|_\rho]$. However, by the construction above, this target function has exact degree pd, contradicting the general fact that the exact degree of a Boolean function is at most quadratic in its approximate degree. Choosing parameters appropriately, this contradiction implies

$$\log \widetilde{\gamma}_2(f \circ \textsc{And}_2) = \Omega\left(\left(\frac{\log s}{\log n}\right)^{1/3}\right),$$

completing the proof.

4.1 Lifting De Morgan Sparsity to the Approximate γ_2-Norm via the $\textsc{And}_2$ Gadget

We now formalize the proof strategy outlined in the proof overview. The first step is to lift max-degree restrictions for f to a collection of restrictions for the lifted function $f \circ \textsc{And}_2$.Algorithm 1 describes this lifting procedure.

Algorithm 1 LIFTEDRESTRICTION.

Input: A semi-adaptive max-degree restriction tree $\mathcal{D}$ for $f : \{0,1\}^n \to \{0,1\}$
Output: A collection of lifted restrictions $\mathcal{D} \circ \text{AND}_2$, where each $\rho : \{x_i, y_i\}_{i=1}^n \to \{0, 1, *, \Delta\}$ is a restriction for $f \circ \text{AND}_2$
Let V be the set of *core variables* of the semi-adaptive restriction tree $\mathcal{D}$.
for each restriction $\rho_f \in \mathcal{D}$ **do**
    Define the lifted restriction $\rho = \text{Lift}_{\mathcal{D}}(\rho_f)$ as follows:
    for each $i \in [n]$ **do**
        Set $(\rho(x_i), \rho(y_i)) = \begin{cases} (\Delta, 0), & \text{if } \rho_f(z_i) = 0 \text{ and } z_i \in V, \\ (*, 1), & \text{if } \rho_f(z_i) = * \text{ and } z_i \in V, \\ (1, 1), & \text{if } \rho_f(z_i) = 1 \text{ and } z_i \notin V, \\ (1, 0), & \text{if } \rho_f(z_i) = 0 \text{ and } z_i \notin V. \end{cases}$
    end for
end for
return $\mathcal{D} \circ \text{AND}_2 := \{\text{Lift}(\rho_f) : \rho_f \in \mathcal{D}\}$

A lifted restriction assigns each variable a value in $\{0, 1, *, \Delta\}$. When applying such a restriction to $f \circ \text{AND}_2$, both symbols $*$ and Δ are treated as free variables. Variables marked by Δ are called *masked variables*; although syntactically free, the restricted function does not depend on them. Tracking masked variables explicitly will be convenient for the subsequent analysis.

Basic structure of lifted restrictions

Let $V \subseteq \{z_1, \ldots, z_n\}$ denote the set of core variables of the semi-adaptive restriction tree $\mathcal{D}$, and let $V_x := \{x_i : z_i \in V\}$ be the corresponding set of x-variables, which we refer to as the *core x-variables*. We will use this notation throughout.

By construction, every lifted restriction $\rho \in \mathcal{D} \circ \text{AND}_2$ fixes all y-variables and fixes all x-variables outside V_x to 1. Moreover, the assignment on V_x uniquely determines the entire restriction: for every $\alpha \in \{\Delta, *\}^{V_x}$, there exists a unique $\rho \in \mathcal{D} \circ \text{AND}_2$ such that $\rho(x_i) = \alpha(x_i)$ for all $x_i \in V_x$. This follows directly from the defining property of $\mathcal{D}$, which guarantees that for every $\beta \in \{0, *\}^V$ there is a unique $\rho_f \in \mathcal{D}$ satisfying $\rho_f(z_i) = \beta(z_i)$ for all $z_i \in V$. As a result, the family $\mathcal{D} \circ \text{AND}_2$ is parametrized by assignments to the core x-variables.

The construction of the lifted restrictions, together with the fact that $\mathcal{D}$ is a max-degree restriction tree, implies the following structural properties. In particular, these properties show that $f \circ \text{AND}_2$ retains its hardness (in terms of degree) under the lifted restrictions.

▷ Claim 4.1. Let $\mathcal{D} \circ \text{AND}_2$ be the collection of lifted restrictions for $f \circ \text{AND}_2$ obtained from a semi-adaptive max-degree restriction tree $\mathcal{D}$ for f via Algorithm 1. Then the following properties hold:

1. For every $\rho \in \mathcal{D} \circ \text{AND}_2$, the restricted function $(f \circ \text{AND}_2)|_\rho$ does not depend on the masked variables $M(\rho)$.
2. For every $\rho \in \mathcal{D} \circ \text{AND}_2$, $\deg\big(\mathbb{E}_{x_{M(\rho)}}[(f \circ \text{AND}_2)|_\rho]\big) = \deg\big((f \circ \text{AND}_2)|_\rho\big) = |\rho^{-1}(*)|$.

Proof. For (1), if x_i is masked in ρ, then by construction $\rho(y_i) = 0$. Hence $\text{AND}_2(x_i, y_i)|_\rho = 0$ regardless of the value of x_i, and therefore $(f \circ \text{AND}_2)|_\rho$ is independent of all masked variables.

For (2), fix $\rho \in \mathcal{D} \circ \text{AND}_2$, and let $\rho_f \in \mathcal{D}$ be the restriction used to generate ρ, i.e., $\rho = \text{Lift}(\rho_f)$. By construction, for every $i \in [n]$ we have $\text{AND}_2(x_i, y_i)|_\rho = \rho_f(z_i)$. Thus, $(f \circ \text{AND}_2)|_\rho$ depends on the variables $\{x_i, y_i\}$ only through the tuple $(\rho_f(z_1), \ldots, \rho_f(z_n))$.

After ignoring the masked variables (which $(f \circ \text{AND}_2)|_\rho$ does not depend on by part (1)), the resulting function coincides with $f|_{\rho_f}$. Since ρ_f is a max-degree restriction of f, we obtain $\deg((f \circ \text{AND}_2)|_\rho) = \deg(f|_{\rho_f}) = |\rho_f^{-1}(*)| = |\rho^{-1}(*)|$.

Finally, since $(f \circ \text{AND}_2)|_\rho$ does not depend on the variables in $M(\rho)$, the function $\mathbb{E}_{x_{M(\rho)}}[(f \circ \text{AND}_2)|_\rho]$ is obtained by simply viewing $(f \circ \text{AND}_2)|_\rho$ as a function on the remaining free variables. In particular, taking expectation over the masked variables leaves the function unchanged as a polynomial in the remaining free variables. Hence, $\deg\big(\mathbb{E}_{x_{M(\rho)}}[(f \circ \text{AND}_2)|_\rho]\big) = \deg\big((f \circ \text{AND}_2)|_\rho\big)$. ◁

Next, we introduce a probability distribution on the restrictions in $\mathcal{D} \circ \text{AND}_2$ for use in a random restriction argument. Fix a parameter $p \in (0,1)$, and recall that every restriction $\rho \in \mathcal{D} \circ \text{AND}_2$ satisfies $|\rho^{-1}(*)| + |\rho^{-1}(\Delta)| = d$, where d is the depth of the semi-adaptive restriction tree $\mathcal{D}$.

We consider the uniform distribution over all restrictions in $\mathcal{D} \circ \text{AND}_2$ that leave exactly pd variables marked by $*$. Formally, let $U := \{\, \rho \in \mathcal{D} \circ \text{AND}_2 \mid |\rho^{-1}(*)| = pd \,\}$. By construction, $|U| = \binom{d}{pd}$. We sample a restriction uniformly at random from U, i.e., $\Pr[\rho] = 1/\binom{d}{pd}$ for all $\rho \in U$. We denote this distribution by $\mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)$.

Next, we analyze the effect of sampling a restriction $\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)$, applying it to $f \circ \text{AND}_2$, and then taking expectation over the masked variables $M(\rho)$. By Claim 4.1, the resulting function retains degree pd. On the other hand, we show that applying the same process – sampling $\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)$ and taking expectation over $M(\rho)$ – to an arbitrary Boolean function g over the x-variables causes its high-degree Fourier mass to decay exponentially above level k, provided $k \gtrsim p^2 d$. Optimizing this tradeoff leads to the choice $p = \Theta(d^{-1/3})$, for which $pd = \Theta(d^{2/3})$ and the Fourier tail decays exponentially above level $k = \Theta(\sqrt{pd}) = \Theta(d^{1/3})$. Combining these two observations yields our main result: a lower bound on the approximate γ_2 norm of $f \circ \text{AND}_2$ in terms of the sparsity of f.

▷ Claim 4.2. Let $\mathcal{D} \circ \text{AND}_2$ be the collection of lifted restrictions for $f \circ \text{AND}_2$ obtained from a semi-adaptive max-degree restriction tree $\mathcal{D}$ of depth d via Algorithm 1. Let $g : \{0,1\}^n \to \{0,1\}$ be an arbitrary Boolean function on the x-variables $\{x_1, \ldots, x_n\}$. Then for every integer $k \geq 0$,

$$\mathbb{E}_{\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)}\left[\left\|\widehat{\mathbb{E}_{x_{M(\rho)}}[g|_\rho]}\right\|_1^{\geq k}\right] \leq \sum_{t=k}^{pd} \left(p\sqrt{\frac{ed}{t}}\right)^t.$$

In particular, if $k \geq 4ep^2 d$, then

$$\mathbb{E}_{\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)}\left[\left\|\widehat{\mathbb{E}_{x_{M(\rho)}}[g|_\rho]}\right\|_1^{\geq k}\right] \leq 2^{-k+1}.$$

Proof. By construction, every restriction $\rho \in \mathcal{D} \circ \text{AND}_2$ fixes all x-variables outside the set V_x to 1. Let $h : \{0,1\}^{|V_x|} \to \{0,1\}$ be the Boolean function obtained from g by fixing all variables outside V_x to 1. Then for every $\rho \in \mathcal{D} \circ \text{AND}_2$, we have $g|_\rho = h$. Without loss of generality, assume $V_x = \{x_1, \ldots, x_d\}$.

Taking expectation over any variable kills all Fourier monomials containing it. Therefore,

$$\begin{aligned}\mathbb{E}_{\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)}\left[\left\|\widehat{\mathbb{E}_{x_{M(\rho)}}[g|_\rho]}\right\|_1^{\geq k}\right] &= \mathbb{E}_{\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)}\left[\left\|\widehat{\mathbb{E}_{x_{M(\rho)}}[h]}\right\|_1^{\geq k}\right] \\ &= \sum_{\substack{S \subseteq [d] \\ |S| \geq k}} |\widehat{h}(S)| \cdot \Pr_\rho\big[S \subseteq \rho^{-1}(*)\big].\end{aligned}$$

Under the uniform distribution over restrictions with exactly pd variables marked by $*$, the probability $\Pr_\rho[S \subseteq \rho^{-1}(*)]$ is zero when $|S| > pd$, and for $|S| \le pd$ we have

$$\Pr_\rho\big[S \subseteq \rho^{-1}(*)\big] = \frac{\binom{d-|S|}{pd-|S|}}{\binom{d}{pd}} \;\le\; p^{|S|}.$$

Therefore,

$$\sum_{\substack{S\subseteq[d]\\ k\le|S|\le pd}} |\widehat{h}(S)| \cdot \Pr_\rho\big[S \subseteq \rho^{-1}(*)\big] \;\le\; \sum_{t=k}^{pd}\sum_{|S|=t} |\widehat{h}(S)|\, p^t.$$

Applying the Cauchy–Schwarz inequality, together with Parseval's identity (which implies $\sum_S \widehat{h}(S)^2 \le 1$ for the Boolean function h), we obtain $\sum_{|S|=t} |\widehat{h}(S)| \le \sqrt{\binom{d}{t}}$. Hence,

$$\sum_{t=k}^{pd}\sum_{|S|=t} |\widehat{h}(S)|\, p^t \;\le\; \sum_{t=k}^{pd}\sqrt{\binom{d}{t}}\, p^t \;\le\; \sum_{t=k}^{pd}\left(p\sqrt{\frac{ed}{t}}\right)^t.$$

Finally, suppose $k \ge 4ep^2 d$. Then for every $t \ge k$,

$$p\sqrt{\frac{ed}{t}} \le p\sqrt{\frac{ed}{k}} \le \frac{1}{2}.$$

Therefore,

$$\sum_{t=k}^{pd}\left(p\sqrt{\frac{ed}{t}}\right)^t \le \sum_{t=k}^{\infty} 2^{-t} \le 2^{-k+1}.$$

This completes the proof. ◁

▶ **Theorem 1.3** (Restated). *For every total Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\log \widetilde{\gamma_2}(f \circ \mathrm{AND}_2) = \Omega\left(\left(\frac{\log spar(f)}{\log n}\right)^{1/3}\right).$$

Proof. Let f have sparsity s. By Lemma 3.7, there exists a semi-adaptive max-degree restriction tree $\mathcal{D}$ for f of depth $d = c_1 \log s/\log n$, for a suitable absolute constant $c_1 > 0$.

Let $c_2, c_3 > 0$ be absolute constants such that Theorem 2.2 converts a 0.44-approximator of degree r into a 1/3-approximator of degree at most $c_2 r$, and such that every Boolean function g satisfies $\deg(g) \le c_3 \cdot (\widetilde{\deg}(g))^2$. Choose an absolute constant $c > 0$ sufficiently small so that $c^{3/2} \le \frac{1}{4ec_2\sqrt{2c_3}}$. Set

$$p := cd^{-1/3} \qquad \text{and} \qquad k := \frac{c^{1/2}d^{1/3}}{c_2\sqrt{2c_3}}.$$

Observe that $pd = cd^{2/3}$, while $k = \Theta(d^{1/3}) = \Theta(\sqrt{pd})$.

Now suppose, for the sake of contradiction, that there exists a decomposition of the communication matrix $M_{f\circ \mathrm{AND}_2}$ of the form

$$\Pi(x,y) = \sum_{i=1}^{m} b_i\, g_i(x) h_i(y),$$

where each g_i, h_i is Boolean, such that $\|\Pi - M_{f\circ\mathrm{AND}_2}\|_\infty \le 1/3$ and $\sum_{i=1}^m |b_i| \le \frac{1}{20}\, 2^k$. We derive a contradiction, thereby proving the theorem.

Let $\mathcal{D} \circ \text{AND}_2$ be the collection of lifted restrictions for $f \circ \text{AND}_2$ obtained from $\mathcal{D}$ via Algorithm 1. Sample a restriction $\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)$. We study the effect of applying ρ and then taking expectation over the masked variables $M(\rho)$. For notational convenience, define

$$F_\rho := \mathbb{E}_{x_{M(\rho)}}\left[(f \circ \text{AND}_2)|_\rho\right], \qquad G_{i,\rho} := \mathbb{E}_{x_{M(\rho)}}\left[g_i|_\rho\right].$$

Both are functions of the starred x-variables under ρ.

1. **Hardness of the restricted function.** By Claim 4.1, $\deg(F_\rho) = |\rho^{-1}(*)|$. Since every restriction in the support of $\mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)$ satisfies $|\rho^{-1}(*)| = pd$, we have $\deg(F_\rho) = cd^{2/3}$.
2. **Simplification of the approximator Π.** For every restriction $\rho \in \mathcal{D} \circ \text{AND}_2$, all y-variables are fixed. As a result, for each term in the decomposition $\Pi(x, y) = \sum_{i=1}^m b_i\, g_i(x) h_i(y)$, the restricted function $h_i|_\rho$ becomes a constant, which we denote by $a_{\rho,i} \in \{0, 1\}$. Thus, $\Pi|_\rho = \sum_{i=1}^m b_i\, a_{\rho,i}\, g_i|_\rho$, which is a function only of the core x-variables, since all non-core x-variables are fixed to 1 in every $\rho \in \mathcal{D} \circ \text{AND}_2$.
   Fix a restriction ρ. The ℓ_1-mass of the Fourier spectrum of $\mathbb{E}_{x_{M(\rho)}}\left[\Pi|_\rho\right]$ above level k can be bounded as follows:

$$\begin{aligned}
\left\|\widehat{\mathbb{E}_{x_{M(\rho)}}[\Pi|_\rho]}\right\|_1^{\geq k} &= \sum_{\substack{S \subseteq [n] \\ |S| \geq k}} \left|\sum_{i=1}^m b_i\, a_{\rho,i}\, \widehat{G_{i,\rho}}(S)\right| \\
&\leq \sum_{\substack{S \subseteq [n] \\ |S| \geq k}} \sum_{i=1}^m |b_i| |\widehat{G_{i,\rho}}(S)| \\
&= \sum_{i=1}^m |b_i| \sum_{\substack{S \subseteq [n] \\ |S| \geq k}} |\widehat{G_{i,\rho}}(S)| \\
&= \sum_{i=1}^m |b_i| \left\|\widehat{G_{i,\rho}}\right\|_1^{\geq k}.
\end{aligned}$$

   Moreover,

$$4ep^2 d = 4ec^2 d^{1/3} \leq \frac{c^{1/2} d^{1/3}}{c_2 \sqrt{2c_3}} = k,$$

   where the inequality follows from the choice of c. Taking expectation over ρ and applying Claim 4.2, we obtain

$$\begin{aligned}
\mathbb{E}_{\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)}\left[\left\|\widehat{\mathbb{E}_{x_{M(\rho)}}[\Pi|_\rho]}\right\|_1^{\geq k}\right] &\leq \sum_{i=1}^m |b_i|\, \mathbb{E}_{\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)}\left[\left\|\widehat{G_{i,\rho}}\right\|_1^{\geq k}\right] \\
&\leq \sum_{i=1}^m |b_i| \cdot 2^{-k+1} \;\leq\; \frac{1}{10},
\end{aligned}$$

   where the final inequality follows from the assumed bound on $\sum_i |b_i|$.
   Thus, there exists a restriction $\rho \sim \mathcal{U}_p(\mathcal{D} \circ \text{AND}_2)$ such that the ℓ_1-mass of the Fourier spectrum of $\Pi|_\rho$ above level k, after averaging over the masked variables $M(\rho)$, is at most 0.1.

Combining the two items, there exists a restriction ρ such that $\deg(F_\rho) = pd = cd^{2/3}$, while the ℓ_1-mass of the Fourier spectrum of $\mathbb{E}_{x_{M(\rho)}}[\Pi|_\rho]$ above level k is at most 0.1.

Let $\tilde{\Pi}$ be the polynomial obtained from $\mathbb{E}_{x_{M(\rho)}}[\Pi|_\rho]$ by deleting all Fourier monomials of degree at least k. Since the discarded Fourier mass is at most 0.1 and Π is a 1/3-approximator of $f \circ \text{AND}_2$, the polynomial $\tilde{\Pi}$ has degree $< k$ and 0.44-approximates F_ρ. By standard error reduction (Theorem 2.2), $\tilde{\Pi}$ can be converted into a 1/3-approximator of degree at most $c_2 k$. As a result,

$$\widetilde{\deg}(F_\rho) < c_2 k = \frac{c^{1/2} d^{1/3}}{\sqrt{2c_3}}.$$

On the other hand, $\deg(F_\rho) = cd^{2/3}$ which contradicts the general inequality $\deg(g) \leq c_3 \cdot (\widetilde{\deg}(g))^2$ for Boolean functions g (Theorem 2.3). This contradiction completes the proof. ◀

4.2 Consequences

Knop et al. [19] showed that $\log \text{spar}(f)$ characterizes the deterministic communication complexity of AND-functions ($f \circ \text{AND}_2$), up to polynomial loss and polylogarithmic factors in n. In particular, they proved that for every Boolean function f,

$$\text{D}^{cc}(f \circ \text{AND}_2) = O\big((\log \text{spar}(f))^5 \cdot \log n\big).$$

Combining this bound with Theorem 1.3, together with the fact that the logarithm of the approximate γ_2 norm lower bounds bounded-error quantum communication complexity, we immediately obtain that for every Boolean function f,

$$\text{D}^{cc}(f \circ \text{AND}_2) = O\big(\text{Q}^{cc}(f \circ \text{AND}_2)^{15} \cdot (\log n)^6\big).$$

A tighter relationship can be obtained using a more refined structural result of Knop et al., which relates deterministic AND-query complexity to sparsity and a combinatorial measure known as *monotone block sensitivity*.

▶ **Definition 4.3** (Monotone Block Sensitivity)**.** *The* monotone block sensitivity *of a Boolean function $f : \{0,1\}^n \to \{0,1\}$, denoted MBS($f$), is a variant of block sensitivity that only considers flipping 0's to 1's. A subset $B \subseteq [n]$ is called a* sensitive 0-block of f at input x *if $x_i = 0$ for all $i \in B$, and $f(x) \neq f(x \oplus 1_B)$, where $x \oplus 1_B$ denotes the input obtained by flipping all bits in B from 0 to 1. For an input $x \in \{0,1\}^n$, let MBS(f, x) denote the maximum number of pairwise disjoint sensitive 0-blocks of f at x. Then, MBS(f) $= \max_{x \in \{0,1\}^n}$ MBS(f, x).*

Knop et al. [19] showed that deterministic communication complexity can be bounded in terms of both sparsity and monotone block sensitivity.

▷ Claim 4.4 ([19, Lemma 3.2, Claim 4.4, Lemma 4.6, Theorem 1.2]). For every Boolean function $f : \{0,1\}^n \to \{0,1\}$,

$$\text{D}^{cc}(f \circ \text{AND}_2) = O\big(\text{MBS}(f)^2 \cdot \log \text{spar}(f) \cdot \log n\big).$$

Intuitively, a large value of MBS(f) indicates that a large-arity PROMISE-OR function can be embedded into f via suitable restrictions and identifications of variables. When such a function f is lifted via composition with AND_2, this structure gives rise to an embedded instance of the *unique set disjointness* problem.

The *unique set disjointness* function UDISJ_k is a partial Boolean function on inputs $x, y \in \{0,1\}^k$, defined as

$$\mathrm{UDISJ}_k(x,y) = \begin{cases} 0, & \text{if } |x \wedge y| = 0, \\ 1, & \text{if } |x \wedge y| = 1, \\ \text{undefined}, & \text{otherwise}, \end{cases}$$

where $x \wedge y$ denotes the bitwise AND and $|\cdot|$ the Hamming weight. That is, under the promise that the inputs are either bitwise disjoint or intersect in exactly one coordinate, the function distinguishes between these two cases.

The following result of Knop et al. shows that large monotone block sensitivity forces large embedded instances of unique set disjointness.

▷ Claim 4.5 ([19, Claim 4.7]). Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function with $\mathrm{MBS}(f) = k$. Then the communication matrix of $f \circ \textsc{And}_2$ contains, as a submatrix (up to flipping output bits), the communication matrix of UDISJ_k.

Using known lower bounds for unique set disjointness, we obtain the following.

▶ **Theorem 4.6** ([17, 30, 31, 36]). *$R^{cc}(\mathrm{UDISJ}_k) = \Omega(k)$ and $Q^{cc}(\mathrm{UDISJ}_k) = \Omega(\sqrt{k})$.*

▷ Claim 4.7. If $f : \{0,1\}^n \to \{0,1\}$ satisfies $\mathrm{MBS}(f) = k$, then

$$\mathrm{R}^{cc}(f \circ \textsc{And}_2) = \Omega(k) \qquad \text{and} \qquad \mathrm{Q}^{cc}(f \circ \textsc{And}_2) = \Omega(\sqrt{k}).$$

Proof. This follows immediately from Claim 4.5 and Theorem 4.6. ◁

Combining Claim 4.7 with Claim 4.4, we obtain the following relationships between deterministic, randomized, and quantum communication complexity for And-functions.

▶ **Theorem 1.2** (Restated). *Let $f : \{0,1\}^n \to \{0,1\}$ be any Boolean function. Then:*

1. $D^{cc}(f \circ \textsc{And}_2) = O\big(Q^{cc}(f \circ \textsc{And}_2)^7 \cdot (\log n)^2\big)$.
2. $D^{cc}(f \circ \textsc{And}_2) = O\big(R^{cc}(f \circ \textsc{And}_2)^5 \cdot (\log n)^2\big)$.

Proof. Combine Theorem 1.3, Theorem 2.7, Claim 4.4 and Claim 4.7. ◀

References

1 Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of huang's sensitivity theorem. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1330–1342, 2021. doi:10.1145/3406325.3451047.

2 Andris Ambainis, Leonard J Schulman, Amnon Ta-Shma, Umesh Vazirani, and Avi Wigderson. The quantum communication complexity of sampling. *SIAM Journal on Computing*, 32(6):1570–1585, 2003. doi:10.1137/S009753979935476.

3 Nikhil Bansal and Makrand Sinha. K-forrelation optimally separates quantum and classical query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1303–1316, 2021. doi:10.1145/3406325.3451040.

4 Harry Buhrman, Richard Cleve, John Watrous, and Ronald de Wolf. Quantum fingerprinting. *Physical review letters*, 87(16):167902, 2001.

5 Harry Buhrman and Ronald de Wolf. Communication complexity lower bounds by polynomials. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity*, pages 120–130, 2001. doi:10.1109/CCC.2001.933879.

6 Harry Buhrman, Nikolai K. Vereshchagin, and Ronald de Wolf. On computation and communication with small bias. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 24–32. IEEE Computer Society, 2007. doi:10.1109/CCC.2007.18.

7 Mark Bun and Justin Thaler. Approximate degree in classical and quantum computing. *Found. Trends Theor. Comput. Sci.*, 15(3-4):229–423, 2022. doi:10.1561/0400000107.

8 Arkadev Chattopadhyay, Yogesh Dahiya, and Shachar Lovett. Restriction trees for sparsity and applications. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC 2026)*, 2026. URL: https://eccc.weizmann.ac.il/report/2025/172/.

9 Arkadev Chattopadhyay, Nikhil S. Mande, and Suhail Sherif. The log-approximate-rank conjecture is false. *J.ACM*, 67(4):1–28, 2020. doi:10.1145/3396695.

10 Richard Cleve and Harry Buhrman. Substituting quantum entanglement for communication. *Physical Review A*, 56(2):1201, 1997.

11 Ronald de Wolf. Quantum communication and complexity. *Theoretical computer science*, 287(1):337–353, 2002. doi:10.1016/S0304-3975(02)00377-8.

12 Dmitry Gavinsky. Bare quantum simultaneity versus classical interactivity in communication complexity. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 401–411. ACM, 2020. doi:10.1145/3357713.3384243.

13 Dmitry Gavinsky. Entangled simultaneity versus classical interactivity in communication complexity. *IEEE Trans. Inf. Theory*, 66(7):4641–4651, 2020. doi:10.1109/TIT.2020.2976074.

14 Dmitry Gavinsky, Julia Kempe, Oded Regev, and Ronald de Wolf. Bounded-error quantum state identification and exponential separations in communication complexity. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of Computing*, pages 594–603, 2006. doi:10.1145/1132516.1132602.

15 Uma Girish, Ran Raz, and Avishay Tal. Quantum versus randomized communication complexity, with efficient players. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, Virtual Conference, January 6-8, 2021*, LIPIcs, pages 54:1–54:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.54.

16 Mika Göös, Tom Gur, Siddhartha Jain, and Jiawei Li. Quantum communication advantage in TFNP. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 1465–1475. ACM, 2025. doi:10.1145/3717823.3718155.

17 Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992. doi:10.1137/0405044.

18 Hartmut Klauck. Lower bounds for quantum communication complexity. *SIAM J. Comput.*, 37(1):20–46, 2007. doi:10.1137/S0097539702405620.

19 Alexander Knop, Shachar Lovett, Sam McGuire, and Weiqiang Yuan. Log-rank and lifting for and-functions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 197–208, 2021. doi:10.1145/3406325.3450999.

20 Matthias Krause and Pavel Pudlák. On computing boolean functions by sparse real polynomials. In *36th Annual Symposium on Foundations of Computer Science, FOCS 1995, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 682–691. IEEE Computer Society, 1995. doi:10.1109/SFCS.1995.492670.

21 Ilan Kremer. Quantum communication. *Master's Thesis, Hebrew University, Computer Science Department*, 1995.

22 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

23 Troy Lee and Adi Shraibman. An approximation algorithm for approximation rank. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 351–357. IEEE, 2009. doi:10.1109/CCC.2009.25.

24 Troy Lee and Adi Shraibman. Lower bounds in communication complexity. *Foundations and Trends® in Theoretical Computer Science*, 3(4):263–399, 2009. doi:10.1561/0400000040.

25 Nati Linial and Adi Shraibman. Lower bounds in communication complexity based on factorization norms. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 699–708, 2007. doi:10.1145/1250790.1250892.

26 Ashley Montanaro and Tobias Osborne. On the communication complexity of XOR functions. *arXiv preprint arXiv:0909.3392*, 2009. arXiv:0909.3392.

27 Noam Nisan. CREW prams and decision trees. *SIAM J. Comput.*, 20(6):999–1007, 1991. doi:10.1137/0220062.

28 Anup Rao and Amir Yehudayoff. *Communication complexity: and applications*. Cambridge University Press, 2020.

29 Ran Raz. Exponential separation of quantum and classical communication complexity. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 358–367, 1999. doi:10.1145/301250.301343.

30 Alexander A Razborov. On the distributional complexity of disjointness. In *International Colloquium on Automata, Languages, and Programming*, pages 249–253. Springer, 1990. doi:10.1007/BFB0032036.

31 Alexander A Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya: Mathematics*, 67(1):145, 2003.

32 Oded Regev and Bo'az Klartag. Quantum one-way communication can be exponentially stronger than classical communication. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 31–40. ACM, 2011. doi:10.1145/1993636.1993642.

33 Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972. doi:10.1016/0097-3165(72)90019-2.

34 Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

35 Alexander A Sherstov. The pattern matrix method for lower bounds on quantum communication. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 85–94, 2008. doi:10.1145/1374376.1374392.

36 Alexander A Sherstov. The pattern matrix method (journal version). *arXiv preprint arXiv:0906.4291*, 2009. arXiv:0906.4291.

37 Alexander A. Sherstov. On quantum-classical equivalence for composed communication problems. *Quantum Inf. Comput.*, 10(5&6):435–455, 2010. doi:10.26421/QIC10.5-6-5.

38 Alexander A. Sherstov. The multiparty communication complexity of set disjointness. *SIAM J. Comput.*, 45(4):1450–1489, 2016. doi:10.1137/120891587.

39 Alexander A Sherstov, Andrey A Storozhenko, and Pei Wu. An optimal separation of randomized and quantum query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1289–1302, 2021. doi:10.1145/3406325.3451019.

40 Yaoyun Shi and Yufan Zhu. Quantum communication complexity of block-composed functions. *Quantum Inf. Comput.*, 9(5&6):444–460, 2009. doi:10.26421/QIC9.5-6-7.

41 Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 209–213, 1979. doi:10.1145/800135.804414.

42 Andrew Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*, pages 352–361. IEEE, 1993. doi:10.1109/SFCS.1993.366852.

43 Shengyu Zhang. On the tightness of the buhrman-cleve-wigderson simulation. In Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra, editors, *Algorithms and Computation*, pages 434–440, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. doi:10.1007/978-3-642-10631-6_45.

44 Zhiqiang Zhang and Yaoyun Shi. Communication complexities of symmetric XOR functions. *Quantum information and computation*, 9(3&4):255–263, 2009. doi:10.26421/QIC9.3-4-5.

All Polynomial Generators Preserve Distance with Mutual Correlated Agreement

Sarah Bordage
EPFL, Lausanne, Switzerland

Alessandro Chiesa
EPFL, Lausanne, Switzerland

Ziyi Guan
EPFL, Lausanne, Switzerland

Ignacio Manzur
EPFL, Lausanne, Switzerland

Abstract

A generator is a function that maps a random seed to a list of coefficients. We study generators that preserve distance to a linear code: the linear combination of any list of vectors using coefficients sampled by the generator has distance to the code no smaller than that of the original vectors, except for a small error. Distance preservation plays a central role in modern probabilistic proofs, and has been formalized in several ways. We study *mutual correlated agreement*, the strongest known form of distance preservation.

We initiate a systematic study of mutual correlated agreement, aiming to characterize the class of generators with this property. Towards this, we study polynomial generators, a rich class that includes all examples of generators considered in the distance preservation literature. Our main result is that *all polynomial generators guarantee mutual correlated agreement for every linear code*. This improves on prior work both in generality (the class of generators covered) and in parameters (the error bounds).

We additionally provide new results for the case where the linear code is a Reed–Solomon code, which is of particular interest in applications. We prove that all polynomial generators satisfy mutual correlated agreement for Reed–Solomon codes up to the Johnson bound. In particular, we improve upon the state-of-the-art by Ben-Sasson, Carmon, Ishai, Kopparty, and Saraf (FOCS 2020) and answer a question posed by Arnon, Chiesa, Fenzi, and Yogev (Eurocrypt 2025).

Along the way we develop a flexible and general toolbox for mutual correlated agreement, and are the first to establish distance preservation for generators that lie beyond polynomial generators.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases proximity testing, distance preservation, mutual correlated agreement

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.24

Related Version *Full Version*: `https://eprint.iacr.org/2025/2051`

Funding The authors are supported in part by the Ethereum Foundation, the Sui Foundation and Polygon Labs.

Acknowledgements We thank Giacomo Fenzi for pointing out relevant references.

1 Introduction

We study probabilistic reductions that preserve the distance of a set of vectors to a linear code. The reductions are captured via generators that, given a random seed, output coefficients used to linearly combine the vectors. We recall notions of distance preservation and then introduce the coding theory problem that we study.



© Sarah Bordage, Alessandro Chiesa, Ziyi Guan, and Ignacio Manzur;
licensed under Creative Commons License CC-BY 4.0





Distance preservation. A *generator* G over a field $\mathbb{F}$ with output size ℓ is a function that maps a seed $\mathbf{x}$ in a set S to a coefficient vector in $\mathbb{F}^\ell$. In its simplest form, distance preservation for G states the following: for every $\mathbf{u}_1, \ldots, \mathbf{u}_\ell \in \mathbb{F}^n$ where at least one $\mathbf{u}_i$ is far (in Hamming distance) from a linear code $\mathcal{C} \subseteq \mathbb{F}^n$, the linear combination $\mathbf{u}_\mathbf{x} := \sum_{j\in[\ell]} G(\mathbf{x})_j \cdot \mathbf{u}_j$ is far from $\mathcal{C}$ with high probability over a random seed $\mathbf{x} \in S$. For example, [43] prove for the identity generator $G\colon \mathbb{F}^\ell \to \mathbb{F}^\ell$ where $G(\mathbf{x}) := \mathbf{x}$ (the ℓ coefficients are random elements in $\mathbb{F}$) that the probability that the maximum distance drops by more than half is at most $\frac{1}{|\mathbb{F}|}$; they use this to design interactive proofs of proximity. Statements about distance preservation such as these are fundamental coding-theoretic properties that play an essential role in various *batch proximity testing problems*, which involve protocols for testing whether $\mathbf{u}_1, \ldots, \mathbf{u}_\ell$ are all close to $\mathcal{C}$. Applications, however, often demand forms of distance preservation that are *more powerful and flexible* than the result in [43].

Improved distance preservation. Statements about distance preservation can be superior in several ways.

- *No distance drop:* [3, 12, 11] show that for the identity generator $G(\mathbf{x}) := \mathbf{x}$ studied in [43] the maximum distance *does not drop at all* (except for a small error over the choice of seed $\mathbf{x}$).
- *More generators:* [10] proves distance preservation for the univariate powers generator $G\colon \mathbb{F} \to \mathbb{F}^{d+1}$ where $G(x) := (1, x, \ldots, x^d)$. Several works [15, 14, 26, 41, 8] establish distance preservation properties for specific generators based on low-degree polynomials. These other specific generators are chosen to best fit the application and setting studied in those papers.
- *Quality of distance preservation:* [10] introduces distance preservation with *correlated agreement* (CA), which states that if $\mathbf{u}_\mathbf{x}$ is close to $\mathcal{C} \subseteq \mathbb{F}^n$ for many $\mathbf{x} \in S$ then there is a large subset of indices $T \subseteq [n]$ over which $\mathbf{u}_1, \ldots, \mathbf{u}_\ell$ agree with corresponding codewords in $\mathcal{C}$. [10] shows that the identity generator and the univariate powers generator have correlated agreement if $\mathcal{C}$ is a Reed–Solomon code. [5] introduces distance preservation with *mutual correlated agreement* (MCA), a strengthening that we study in this paper and discuss below. These stronger forms of distance preservation are useful in applications because they provide information on *how* the vectors agree with the code $\mathcal{C}$.

Mutual correlated agreement. We study distance preservation with *mutual correlated agreement* (MCA) [5]; informally, for many seeds $\mathbf{x} \in S$, if $\mathbf{u}_\mathbf{x} := \sum_{j\in[\ell]} G(\mathbf{x})_j \cdot \mathbf{u}_j$ agrees with an element in $\mathcal{C}$ over a large subset $T \subseteq [n]$ of indices, then each $\mathbf{u}_j$ agrees with a corresponding codeword in $\mathcal{C}$ *over the same set of indices* T. The fraction of $\mathbf{x} \in S$ for which this does not hold is the *MCA error of* G *for the code* $\mathcal{C}$.

▶ **Definition 1.** *$G\colon S \to \mathbb{F}^\ell$ has* **mutual correlated agreement** *for a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with error ϵ_{MCA} if*

$$\forall\, \mathbf{U} = \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_\ell \end{pmatrix} \in \mathbb{F}^{\ell\times n},\ \forall\, \gamma \in [0,1], \quad \Pr_{\mathbf{x}\in S}\left[\exists\, T \subseteq [n] : \begin{array}{l} |T| \geq n\cdot(1-\gamma) \\ \wedge\, (G(\mathbf{x})\cdot\mathbf{U})|_T \in \mathcal{C}|_T \\ \wedge\, \vee_{j\in[\ell]}\, \mathbf{u}_j|_T \notin \mathcal{C}|_T \end{array}\right] \leq \epsilon_{\mathsf{MCA}}(\gamma)\ .$$

Of course, any generator has MCA for any code with the trivial error bound $\epsilon_{\mathsf{MCA}}(\gamma) := 1$; hence when we say that a generator "has MCA" we mean that the error bound $\epsilon_{\mathsf{MCA}}(\gamma)$ is not trivial.

MCA is related to mathematically elegant properties of linear codes. For example, MCA implies a natural *list-decoding commutativity*: if G has MCA for the code $\mathcal{C}$ then, except for the MCA error over a random seed $\mathbf{x}$, the list of codewords close to the combined word $\mathbf{u}_{\mathbf{x}} := \sum_{j\in[\ell]} G(\mathbf{x})_j \cdot \mathbf{u}_j$ equals the list obtained by taking the linear combination according to $G(\mathbf{x})$ of the codewords in the lists of each word $\mathbf{u}_j$; that is, "linearly combine and then list decode" equals "list decode and then linearly combine".

Properties such as list-decoding commutativity are at the heart of the power of MCA for the design and analysis of testing protocols like interactive oracle proofs (IOPs) and interactive oracle reductions (IORs). MCA has thus rapidly become the "gold standard" for distance preservation, by enabling highly efficient protocols and facilitating simpler security analyses (see, e.g., [5, 18, 8, 31]).

Unfortunately, MCA remains *poorly understood*. The MCA error of the affine space generator $G\colon \mathbb{F}^s \to \mathbb{F}^{s+1}$ where $G(x_1,\ldots,x_s) := (1, x_1, \ldots, x_s)$ for every linear code is studied in [30, 29]. However, which generators G have small MCA error and how this error depends on properties of the code $\mathcal{C}$ remains wide open. In fact, the study of distance preservation for several specific generators has led to a medley of different and suboptimal analyses (see [15, 14, 26, 41, 8]).

The prominent practical applications of MCA motivate a deeper mathematical understanding of MCA. For example, all known analyses that establish distance preservation for generators (including those in this paper) are limited by the list-decoding bound of the given code $\mathcal{C}$. Yet, establishing MCA beyond this bound (if at all possible) for codes such as Reed–Solomon codes is closely related to desirable efficiency improvements in practical hash-based succinct arguments (e.g., see [5]), so much so that Ethereum Foundation has allocated over $1M towards resolving certain conjectures about MCA [27].

Which generators have MCA? We initiate a systematic study of generators with MCA: what are the necessary and sufficient properties for a generator $G\colon S \to \mathbb{F}^\ell$ to have MCA? An initial observation is that G must be *zero-evading*, namely, the zero-evading error $\epsilon_{\mathsf{ZE}} := \max_{\mathbf{v}\in\mathbb{F}^\ell\setminus\{\mathbf{0}\}} \Pr_{\mathbf{x}\in S}[G(\mathbf{x})\cdot\mathbf{v}^\intercal = 0]$ (maximum probability of obtaining a zero for a non-zero vector) must be small because $\epsilon_{\mathsf{ZE}} \le \min_{\gamma\in[0,1]} \epsilon_{\mathsf{MCA}}(\gamma)$. We wish to understand the reverse direction:

Do all zero-evading generators have mutual correlated agreement?

In other words, can one upper bound the MCA error $\epsilon_{\mathsf{MCA}}(\gamma)$ of a generator G in terms of its zero-evading error ϵ_{ZE} (perhaps up to some loss beyond ϵ_{ZE})? We seek a fundamental understanding of why and how a zero-evading generator G has a non-trivial MCA error $\epsilon_{\mathsf{MCA}}(\gamma)$.

In this paper we focus on the distance preservation properties of *polynomial generators*, a large and flexible class of zero-evading generators that includes all examples of generators previously studied in the distance preservation literature.

▶ **Definition 2.** *For $S_1,\ldots,S_s \subseteq \mathbb{F}$, $G\colon S_1\times\cdots\times S_s \to \mathbb{F}^\ell$ is a* **polynomial generator** *if $G(\mathbf{x}) = (P_j(\mathbf{x}))_{j\in[\ell]}$ for every $\mathbf{x}\in S_1\times\cdots\times S_s$, where $P_1,\ldots,P_\ell$ are linearly independent polynomials in $\mathbb{F}[\mathtt{X}_1,\ldots,\mathtt{X}_s]$.*

1.1 Our results

Our main result is that **all polynomial generators have MCA for every linear code**. Moreover, for every polynomial generator we establish **better MCA errors for Reed–Solomon codes**, linear codes whose MCA properties have significant practical implications.

Tables 1–3 compare our results and prior art in the distance preservation literature; our error bounds **match or improve all bounds for every generator** that was previously studied. We obtain our results thanks to a flexible and general framework of tools for studying MCA that we deem of independent interest. Next we discuss our results in more detail.

MCA for arbitrary linear codes. The polynomial generator $G\colon S_1 \times \cdots \times S_s \to \mathbb{F}^\ell$ where $G(\mathbf{x}) = (P_j(\mathbf{x}))_{j\in[\ell]}$ has zero-evading error $\epsilon_{\mathsf{ZE}} \leq \sum_{i\in[s]} \frac{d_i}{|S_i|}$ and $d_i := \max_{j\in[\ell]}\{\deg_{\mathtt{X}_i}(P_j)\}$ is the maximum degree of $\mathtt{X}_i$ in $(P_j)_{j\in[\ell]}$. Hence for G we expect that $\min_{\gamma\in[0,1]} \epsilon_{\mathsf{MCA}}(\gamma) \geq \sum_{i\in[s]} \frac{d_i}{|S_i|}$. The theorem below bounds the MCA error ϵ_{MCA} of this polynomial generator *for every linear code* $\mathcal{C}$, with a bound corresponding to a unique-decoding analysis and another corresponding to a list-decoding analysis. (While in introductory sections we use big-O notation, all technical sections include explicit constants in the upper bounds.)

▶ **Theorem 3** (informal). *Let $G\colon S_1 \times S_2 \times \cdots \times S_s \to \mathbb{F}^\ell$ be a polynomial generator where $G(\mathbf{x}) = (P_j(\mathbf{x}))_{j\in[\ell]}$; let $d_i := \max_{j\in[\ell]}\{\deg_{\mathtt{X}_i}(P_j)\}$ for every $i \in [s]$. Then for every linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with relative distance $\delta_{\mathcal{C}}$ and tradeoff parameter $\eta \in (0,1)$, G has MCA for $\mathcal{C}$ with error*

$$\epsilon_{\mathsf{MCA}}(\gamma) := \begin{cases} \max\{n\cdot\gamma, 1\} \cdot \sum_{i\in[s]} \frac{d_i}{|S_i|} & \text{if } \gamma \leq \delta_{\mathcal{C}}/(\max_{i\in[s]} d_i + 2) \\ O\left(\frac{n}{\eta}\right) \cdot \sum_{i\in[s]} \frac{d_i}{|S_i|} & \text{if } \gamma \leq 1 - (1-\delta_{\mathcal{C}}+\eta)^{\frac{1}{\max_{i\in[s]} d_i + 2}} \end{cases}.$$

Both upper bounds are the zero-evading error of G with a multiplicative overhead; the two bounds are incomparable, both in applicability (whether the bound holds based on γ) and the achieved error bound. Theorem 3 *matches or improves all bounds previously established for specific generators*; see Tables 1 and 2 for a comparison. We prove that the error bound in the unique-decoding setting (the upper one) is *tight* for the univariate powers generator $G(x) = (1, x, \ldots, x^s)$ (see the discussion in Section 2.3).

The affine space generator $G\colon \mathbb{F}^s \to \mathbb{F}^{s+1}$ where $G(x_1, \ldots, x_s) := (1, x_1, \ldots, x_s)$ is a notable special case; its zero-evading error is $\epsilon_{\mathsf{ZE}} = \frac{1}{|\mathbb{F}|}$. Numerous works obtain error bounds on its distance preservation properties [43, 3, 11, 10, 30, 29]. For any given linear code $\mathcal{C}$, directly applying Theorem 3 to this generator recovers the best bounds in prior work [30]:

$$\epsilon_{\mathsf{MCA}}(\gamma) := \begin{cases} \max\{n\cdot\gamma, 1\} \cdot \frac{s}{|\mathbb{F}|} & \text{if } \gamma \leq \delta_{\mathcal{C}}/3 \\ O\left(\frac{n}{\eta}\right) \cdot \frac{s}{|\mathbb{F}|} & \text{if } \gamma \leq 1 - (1-\delta_{\mathcal{C}}+\eta)^{\frac{1}{3}} \end{cases}.$$

In fact, for any linear code $\mathcal{C} \subseteq \mathbb{F}^n$, we show that the MCA error of the affine space generator is equal to the MCA error of the affine line generator $\mathbb{F} \to \mathbb{F}^2$; $x \mapsto (1, x)$ up to a small, constant multiplicative overhead.

▶ **Lemma 4** (informal). *Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code. Then the generator $G\colon \mathbb{F}^s \to \mathbb{F}^{s+1}$ such that $G(x_1, \ldots, x_s) := (1, x_1, \ldots, x_s)$ has MCA for $\mathcal{C}$ with error*

$$\epsilon_{\mathsf{MCA}}(\gamma) := \begin{cases} O(\max\{n\cdot\gamma, 1\}) \cdot \frac{1}{|\mathbb{F}|} & \text{if } \gamma \leq \delta_{\mathcal{C}}/3 \\ O\left(\frac{n}{\eta}\right) \cdot \frac{1}{|\mathbb{F}|} & \text{if } \gamma \leq 1 - (1-\delta_{\mathcal{C}}+\eta)^{\frac{1}{3}} \end{cases}.$$

This gives an MCA error for the affine space generator that is *independent of the number of variables* s, and leads to the tightest bounds on the MCA error for this generator. This agrees with the fact that $\epsilon_{\mathsf{ZE}} = \frac{1}{|\mathbb{F}|}$ is independent of s. Moreover, our reduction does not impose a lower bound on the field size, contrary to prior analyses for CA [33].

[11] constructs a linear code $\mathcal{C}$ and words $\mathbf{u}_1, \mathbf{u}_2$ that are far from $\mathcal{C}$ such that their linear combination via the affine line generator $x \mapsto (1, x)$ drops with probability almost one. This gives intuition for why a result that holds for every linear code, such as the above lemma, holds only up to some distance γ and incurs errors that get close to 1 in some parameter settings (e.g., when the blocklength n of $\mathcal{C}$ approaches $|\mathbb{F}|$).

MCA for Reed–Solomon codes. Numerous highly efficient probabilistic proofs rely on distance preservation properties of Reed–Solomon codes (e.g., [3, 9, 11, 13, 26, 4, 48, 5, 17, 46, 24, 41, 19, 20, 18, 7, 8, 42, 31]). For Reed–Solomon codes, [10] proves *correlated agreement* for the univariate powers generator and the affine space generator, achieving better error bounds than what is known for arbitrary linear codes. We strengthen their result to achieve MCA and use our framework to extend it to hold for all polynomial generators. Below, $\mathsf{RS}[\mathbb{F}, D, k]$ denotes the Reed–Solomon code consisting of evaluations of polynomials over $\mathbb{F}$ of degree less than k over $D \subseteq \mathbb{F}$; we let $n := |D|$ and $\rho := k/n$.

▶ **Theorem 5** (informal). *Let $G\colon \mathbb{F}^s \to \mathbb{F}^\ell$ be a polynomial generator where $G(\mathbf{x}) = (P_j(\mathbf{x}))_{j\in[\ell]}$; let $d_i := \max_{j\in[\ell]}\{\deg_{\mathbf{X}_i}(P_j)\}$ for every $i \in [s]$. Then for every tradeoff parameter $m \geq 3$, G has MCA for $\mathsf{RS}[\mathbb{F}, D, k]$ with error*

$$\epsilon_{\mathsf{MCA}}(\gamma) := O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{\sum_{i\in[s]} d_i}{|\mathbb{F}|} \quad \text{if } \gamma \leq 1 - \left(1 + \frac{1}{2m}\right) \cdot \sqrt{\rho}\ .$$

The error bound in Theorem 5 matches the one for correlated agreement in [10] (when fixing the specific generators studied therein). Table 3 gives a comparison of our bound on Reed–Solomon codes with prior art (and similarly for Reed–Muller codes). As a special case, Theorem 5 resolves a question asked in [5] on the MCA error for the univariate powers generator $x \mapsto (1, x, \ldots, x^d)$ for Reed–Solomon codes; prior to this work [38] resolves this question for the affine line generator $x \mapsto (1, x)$ $(d = 1)$.

List decoding and MCA. If a linear code $\mathcal{C}$ is list-decodable with list size L up to distance δ, then the affine space generator has a small MCA error $\epsilon_{\mathsf{MCA}}(\gamma)$ for $\mathcal{C}$ provided $\gamma < 1 - \sqrt{1-\delta}$ [29]. This connection is particularly interesting for code families that are list-decodable up to capacity, e.g., random linear codes [37, 36, 22, 40], random Reed–Solomon codes [44, 45, 34, 28, 32, 16, 35, 2, 39], folded Reed–Solomon codes, and univariate multiplicity codes [21]. However, for codes in this paper, namely arbitrary Reed–Solomon codes and arbitrary linear codes, the best known list-decoding bound is the Johnson bound, which is $\approx 1-(1-\delta_{\mathcal{C}})^{1/2}$ if the code $\mathcal{C}$ has distance $\delta_{\mathcal{C}}$. Applying [29]'s result would yield a non-trivial MCA error for the affine space generator only for γ within distance $\approx 1-(1-\delta_{\mathcal{C}})^{1/4}$ (the "double Johnson bound"). Our results for the affine space generator (implied by Lemma 4 and Theorem 5) subsume the application of [29] to these codes.

Operations on codes. Most of this work investigates which generators have MCA for general linear codes (and sometimes specific ones like Reed–Solomon codes). An orthogonal question is what happens to MCA when we fix an arbitrary generator and apply various operations on codes. We prove for example that if any generator G has MCA for some linear code $\mathcal{C}$ with error ϵ_{MCA} then G has MCA for the k-interleaving $\mathcal{C}^k$ with error at most $k \cdot \epsilon_{\mathsf{MCA}}$. Similarly we obtain upper bounds on the MCA error of arbitrary generators when we take the direct product of codes, or when we concatenate them.

Open questions. Theorem 3 makes progress towards a characterization of generators with MCA by providing a large class of generators with MCA. Yet we already know that more can be said: we prove that a zero-evading generator in [1], which is *not* a polynomial generator, has MCA (see Section 2.6). This tells us that MCA holds beyond polynomial generators and a characterization of MCA remains open.

The behavior of MCA under natural operations on codes remains to be understood. Of particular interest are tensor products of codes, and distance amplification procedures on codes.

For the special case of Reed–Solomon codes, Theorem 5 states that any polynomial generator has MCA for γ close to the code's list-decoding bound $1 - \sqrt{\rho}$. In probabilistic proofs, distance preservation at a larger distance typically implies more efficient verification. Can one bound the MCA error for distances γ beyond $1 - \sqrt{\rho}$? [5] conjectures MCA for the univariate powers generator for γ close to the code's capacity $1 - \rho$, and similar conjectures on Reed–Solomon codes are sometimes used in practice for efficiency. Yet [25, 23] shows that this is too much to hope for: there is a family of Reed–Solomon codes for which distance preservation fails for the affine line generator $x \mapsto (1, x)$ when γ is close to the code's capacity $1 - \rho$. Understanding whether MCA holds between the list-decoding and capacity bounds remains open.

1.2 Comparison with prior work

We give a comparison with prior work that studies distance preservation with CA and with MCA.

Table 1: arbitrary linear codes in the unique decoding setting [3] proves that the identity generator has CA error $n \cdot \delta_{\mathcal{C}} \cdot \frac{1}{|\mathbb{F}|}$ within distance $\frac{\delta_{\mathcal{C}}}{3}$. Our results imply that the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$ has MCA (hence, CA) with the improved error $\frac{n \cdot \delta_{\mathcal{C}}}{3} \cdot \frac{1}{|\mathbb{F}|}$ within distance $\frac{\delta_{\mathcal{C}}}{3}$.

Table 2: arbitrary linear codes in the list-decoding setting [11] shows that the affine line generator $x \mapsto (1, x)$ has CA for every linear code almost up to distance $1 - (1 - \delta_{\mathcal{C}})^{1/3}$, known as the "1.5 Johnson bound". [30] obtains the stronger MCA for the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$ for every linear code almost up to the 1.5 Johnson bound. [47] improves upon the constant in the error of [30] in the case of the affine line generator $x \mapsto (1, x)$. [8] uses [47] to prove MCA for the multilinear equality generator $\mathbf{x} \mapsto (\widehat{\mathsf{eq}}(\mathbf{v}, \mathbf{x}))_{\mathbf{v} \in \{0,1\}^s}$ almost up to the 1.5 Johnson bound. We recover all of these specific results via our Theorem 3 (MCA for all polynomial generators). Moreover, for the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$, we improve over prior bounds by removing the dependency on s.

Table 3: Reed–Solomon and Reed–Muller [10] shows that both the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$ and the univariate powers generator $x \mapsto (1, x, \ldots, x^d)$ have CA for Reed–Solomon codes up to the Johnson bound (i.e., beyond the 1.5 Johnson bound known for arbitrary linear codes). We show that the same bound holds for MCA and for every polynomial generator.

[41] uses [10] to show that the univariate powers generator has CA for Reed–Muller codes up to some distance parameter well below the list-decoding bound. Our results imply a tighter bound on the MCA error for the univariate powers generator for Reed–Muller codes.

Table 1 Results on correlated agreement (CA) and mutual correlated agreement (MCA) based on unique-decoding analyses, for linear/affine space generators and arbitrary linear codes $\mathcal{C} \subseteq \mathbb{F}^n$ of relative distance $\delta_{\mathcal{C}}$.

	Generator	(M)CA	Distance	Error
[3]	$(\mathtt{X}_1, \ldots, \mathtt{X}_s)$	CA	$\delta_{\mathcal{C}}/3$	$n \cdot \delta_{\mathcal{C}} \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
This work	$(1, \mathtt{X}_1, \ldots, \mathtt{X}_s)$	CA	$\delta_{\mathcal{C}}/3$	$\frac{n \cdot \delta_{\mathcal{C}}}{3} \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
This work	$(1, \mathtt{X}_1, \ldots, \mathtt{X}_s)$	MCA	$\delta_{\mathcal{C}}/3$	$\frac{n \cdot \delta_{\mathcal{C}}}{3} \cdot \frac{1}{\lvert\mathbb{F}\rvert}$

Table 2 Comparison of results on correlated agreement (CA) and mutual correlated agreement (MCA) for several generators and arbitrary linear codes $\mathcal{C} \subseteq \mathbb{F}^n$ of relative distance $\delta_{\mathcal{C}}$. The parameter $\eta \in (0, 1)$ is a tradeoff between distance and error. In the table we assume that $d_i \leq n$ for every i. The polynomial $\widehat{\mathsf{eq}}$ is the multilinear equality polynomial.

	Generator	(M)CA	Distance	Error
[11]	$(1, \mathtt{X})$	CA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{1}{\eta^2}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[6]	$(\vec{\mathtt{X}}^{\mathbf{v}})_{\mathbf{v} \in \{0,1\}^s}$	CA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{2s}{\eta^2}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[30]	$(1, \mathtt{X}_1, \ldots, \mathtt{X}_s)$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{n \cdot s}{\eta}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[47]	$(1, \mathtt{X})$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{n}{\eta}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[8]	$(\widehat{\mathsf{eq}}(\mathbf{v}, \vec{\mathtt{X}}))_{\mathbf{v} \in \{0,1\}^s}$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3} - \eta$	$O\left(\frac{3^s}{s \cdot \eta^2}\right) \cdot \frac{s}{\lvert\mathbb{F}\rvert}$
This work	$(P_j(\vec{\mathtt{X}}))_{j \in [\ell]}$, $\max_{j \in [\ell]} \deg_{\mathtt{X}_i}(P_j) \leq d_i$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{\frac{1}{\max_{i \in [s]} d_i + 2}}$	$O\left(\frac{n}{\eta}\right) \cdot \frac{\sum_i d_i}{\lvert\mathbb{F}\rvert}$
	$(1, \mathtt{X}_1, \ldots, \mathtt{X}_s)$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{n}{\eta}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
	$(\widehat{\mathsf{eq}}(\mathbf{v}, \vec{\mathtt{X}}))_{\mathbf{v} \in \{0,1\}^s}$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{n \cdot s}{\eta}\right) \cdot \frac{s}{\lvert\mathbb{F}\rvert}$

Table 3 Comparison of results on correlated agreement (CA) and mutual correlated agreement (MCA) for some generators and specific linear codes. $\mathsf{RS}[\mathbb{F}, n, \rho]$ is the Reed–Solomon code of evaluations of univariate polynomials over $\mathbb{F}$ of degree at most $\rho \cdot n - 1$ over a domain of size n. For an extension $\mathbb{B} \subseteq \mathbb{F}$, $\mathsf{RM}[\mathbb{F}, \mathbb{B}^2, k]$ consists of bivariate polynomials $f \colon \mathbb{B}^2 \to \mathbb{F}$ of total degree at most k. The integer $m \geq 3$ is a tradeoff between distance and error. In the table we assume that $d \leq \lvert\mathbb{B}\rvert^2$.

	Code	Generator	(M)CA	Distance	Error
[10]	$\mathsf{RS}[\mathbb{F}, n, \rho]$	$(\mathtt{X}_1, \ldots, \mathtt{X}_s)$	CA	$1 - \left(1 + \frac{1}{2m}\right)\sqrt{\rho}$ for $m \geq 3$	$O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[10]		$(1, \mathtt{X}, \ldots, \mathtt{X}^d)$	CA		$O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{d}{\lvert\mathbb{F}\rvert}$
[38]		$(1, \mathtt{X})$	MCA		$O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
This work		$(P_j(\vec{\mathtt{X}}))_{j \in [\ell]}$, $\max_j \deg_{\mathtt{X}_i}(P_j) \leq d_i$	MCA		$O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{\sum_i d_i}{\lvert\mathbb{F}\rvert}$
[41]	$\mathsf{RM}[\mathbb{F}, \mathbb{B}^2, k]$	$(1, \mathtt{X}, \ldots, \mathtt{X}^d)$	CA	$1 - O\left(\frac{d}{\lvert\mathbb{B}\rvert^{\frac{1}{14d+2}}}\right)$	$\frac{\lvert\mathbb{B}\rvert^4}{k^2 \cdot d} \cdot \frac{d}{\lvert\mathbb{F}\rvert}$
This work		$(1, \mathtt{X}, \ldots, \mathtt{X}^d)$	MCA	$1 - \left(\frac{k}{\lvert\mathbb{B}\rvert} + \eta\right)^{\frac{1}{d+2}}$	$O\left(\frac{\lvert\mathbb{B}\rvert^2}{\eta}\right) \cdot \frac{d}{\lvert\mathbb{F}\rvert}$

2 Technical overview

Prior work with bounds for MCA does not offer suitable starting points towards Theorem 3 and Theorem 5, as existing techniques typically target either specific generators or specific codes. To overcome these limitations, we develop a toolbox for MCA for our setting and explain how it helps us prove Theorem 3 and Theorem 5.

In Section 2.1 we compare the MCA and zero-evading errors. In Section 2.2 we reduce proving Theorem 3 to establishing MCA for the univariate powers generator. In Section 2.3 we sketch how to obtain MCA for the univariate powers generator for all linear codes. In Section 2.4 we sketch the proof of Lemma 4. In Section 2.5 we explain the main idea underlying Theorem 5. Lastly, in Section 2.6 we obtain a bound on the MCA error of a non-polynomial generator for all linear codes and summarize our toolbox for MCA.

2.1 On mutual correlated agreement vs. zero-evading error

We outline why the MCA error is lower bounded by the zero-evading error. This justifies our focus on the class of zero-evading generators, and serves as an opportunity to build intuition on the definition of MCA. A simple initial observation is that the MCA error of $G\colon S \to \mathbb{F}^\ell$ is monotonically non-decreasing, almost by definition. Therefore, it suffices to show that the MCA error at 0 is lower bounded by the zero-evading error:

$$\epsilon_{\mathsf{MCA}}(0) \geq \Pr_{\mathbf{x}\in S}\left[\begin{array}{l} G(\mathbf{x})\cdot\mathbf{U}\in\mathcal{C} \\ \wedge\, \exists j\in[\ell],\ \mathbf{u}_j\notin\mathcal{C}\end{array}\right] \geq \max_{\mathbf{v}\in\mathbb{F}^\ell\setminus\{\mathbf{0}\}} \Pr_{\mathbf{x}\in S}\left[G(\mathbf{x})\cdot\mathbf{v}^\intercal=0\right] \ .$$

The right-hand side is mainly about cancellations: it measures the maximum probability that a non-trivial linear combination of $G(\mathbf{x})$ equals zero. Proving the lower bound corresponds to the intuition that if the combined word $G(\mathbf{x})\cdot\mathbf{U}$ falls into the code while some of the individual words $\mathbf{u}_j$ do not, this must be due to similar cancellations. This can be concretized as follows. Without loss of generality the code is not the entire space, so there exists a word $\mathbf{u}\in\mathbb{F}^n\setminus\mathcal{C}$. Fix a non-zero vector $\mathbf{v}\in\mathbb{F}^\ell\setminus\{\mathbf{0}\}$ and define words $\mathbf{u}_1,\ldots,\mathbf{u}_\ell$ such that, for every $j\in[\ell]$, $\mathbf{u}_j := \mathbf{v}_j\cdot\mathbf{u}$. Since $\mathbf{v}$ is not the zero vector, there is j such that $\mathbf{u}_j\notin\mathcal{C}$. Then, for every seed $\mathbf{x}\in S$, the linear combination $G(\mathbf{x})\cdot\mathbf{U}$ is equal to $(G(\mathbf{x})\cdot\mathbf{v}^\intercal)\cdot\mathbf{u}$ and thus belongs to the code if and only if $G(\mathbf{x})\cdot\mathbf{v}^\intercal=0$. Since $\mathbf{v}$ was arbitrary, this establishes the desired lower bound.

With a bit more effort, we can show that the MCA error at 0 in fact *equals* the zero-evading error. This amounts to showing that there exists, for every $\mathbf{u}_1,\ldots,\mathbf{u}_\ell\in\mathbb{F}^n$, a non-zero vector $\mathbf{v}\in\mathbb{F}^\ell\setminus\{\mathbf{0}\}$ such that

$$\Pr_{\mathbf{x}\in S}\left[\begin{array}{l} G(\mathbf{x})\cdot\mathbf{U}\in\mathcal{C} \\ \wedge\, \exists j\in[\ell],\ \mathbf{u}_j\notin\mathcal{C}\end{array}\right] \leq \Pr_{\mathbf{x}\in S}\left[G(\mathbf{x})\cdot\mathbf{v}^\intercal=0\right] \ .$$

This is neatly achieved by introducing parity checks for the code $\mathcal{C}$ and considering the linear combinations of these parity checks that are violated by the words $\mathbf{u}_1,\ldots,\mathbf{u}_\ell$.

Towards upper bounds on ϵ_{MCA}. The previous result gives hope that one might, in general, be able to upper bound the MCA error ϵ_{MCA} of a generator $G\colon S\to\mathbb{F}^\ell$ via (some function of) its zero-evading error ϵ_{ZE}. We do not know this to be true. In this paper we establish results for the MCA error ϵ_{MCA} when G is a polynomial generator (or some other special cases). For these generators, we show that the zero-evading error *does* play a useful role in upper bounding ϵ_{MCA}. We explain this connection in Section 2.3.

2.2 All polynomials generators have MCA for all linear codes

Our approach to prove Theorem 3 consists of two main steps:

(i) we develop tools to reduce MCA of polynomial generators to MCA of the univariate powers generator, with minimum parameter loss; and

(ii) we prove that the univariate powers generator has MCA for all linear codes.

Let $G(\mathbf{x}) := (P_j(\mathbf{x}))_{j\in[s]}$ be a polynomial generator. Each polynomial P_j is a linear combination of monomials in the variables $\mathtt{X}_1, \dots, \mathtt{X}_s$, where the individual degree of $\mathtt{X}_i$ is at most $\deg_{\mathtt{X}_i}(P_j)$. Define $d_i := \max_{j\in[\ell]}\{\deg_{\mathtt{X}_i}(P_j)\}$ and $k := \prod_{i\in[s]}(1+d_i)$. There exists a matrix $\mathbf{A} \in \mathbb{F}^{k\times\ell}$ such that

$$(P_j(\mathtt{X}_1, \dots, \mathtt{X}_s))_{j\in[\ell]} = \left(\bigotimes_{i\in[s]}(1, \mathtt{X}_i, \dots, \mathtt{X}_i^{d_i})\right) \cdot \mathbf{A} \ .$$

The expression $\bigotimes_{i\in[s]}(1, \mathtt{X}_i, \dots, \mathtt{X}_i^{d_i})$ represents the tensor product of the univariate powers generators. The polynomials $(P_j)_{j\in[\ell]}$ are linearly independent, so $\mathbf{A}$ has a left inverse (any non-trivial linear dependence on the rows of $\mathbf{A}$ yields a non-trivial linear dependence on $(P_j)_{j\in[\ell]}$). In sum, every polynomial generator can be viewed as a linear transformation of the tensor products of univariate powers generators. In light of this, we prove lemmas to track MCA across these two operations.

▶ **Lemma 6** (tensor products preserve MCA)**.** *Let $G\colon S \to \mathbb{F}^\ell$, $G'\colon S' \to \mathbb{F}^{\ell'}$ be generators that have MCA for a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with respective errors $\epsilon_{\mathsf{MCA}}, \epsilon'_{\mathsf{MCA}}$. Then the tensor product generator $G \otimes G'$ has MCA for $\mathcal{C}$ with error $\epsilon_{\mathsf{MCA}} + \epsilon'_{\mathsf{MCA}}$.*

▶ **Lemma 7** (linear transformations preserve MCA)**.** *Let $G\colon S \to \mathbb{F}^\ell$ be a generator that has MCA for a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with error ϵ_{MCA}. For every $\tilde{\ell} \in \mathbb{N}$ and $\mathbf{A} \in \mathbb{F}^{\ell\times\tilde{\ell}}$ with a left inverse, let $\tilde{G}\colon S \to \mathbb{F}^{\tilde{\ell}}$ be the generator defined as $\tilde{G}(\mathbf{x}) := G(\mathbf{x}) \cdot \mathbf{A}$. Then $\tilde{G}$ has MCA for $\mathcal{C}$ with error ϵ_{MCA}.*

From Lemmas 6 and 7 we deduce that the MCA error $\epsilon_{\mathsf{MCA}}(\gamma)$ for $\mathcal{C}$ of the polynomial generator G is

$$\epsilon_{\mathsf{MCA}}(\gamma) = \sum_{i\in[s]} \xi_{\mathcal{C},d_i,|S_i|}(\gamma)$$

where $\xi_{\mathcal{C},d,|S|}$ is the MCA error for $\mathcal{C}$ of the univariate powers generator $G\colon S \to \mathbb{F}^{d+1}$ where $G(x) := (1, x, \dots, x^d)$. Next, we turn our attention to establishing the MCA error of the univariate powers generator.

▶ Remark 8 (MDS generators). The univariate powers generator is a special case of an *MDS generator*. One can associate to each generator G a corresponding linear code $\mathcal{C}_G$; we say G is MDS if $\mathcal{C}_G$ meets the Singleton bound. In fact, we prove a generalization of Theorem 9: every *MDS generator has MCA for arbitrary linear codes*. We believe this generalization is of independent interest as MDS generators are not necessarily polynomial generators.

2.3 The univariate powers generator

Our theorem below establishes MCA of the univariate powers generator for every linear code.

▶ **Theorem 9.** *Let $G\colon S \subseteq \mathbb{F} \to \mathbb{F}^{d+1}$ be the univariate powers generator where $G(x) := (1, x, \ldots, x^d)$. Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code with relative distance $\delta_{\mathcal{C}}$. Then, for every tradeoff parameter $\eta \in (0,1)$, G has MCA for $\mathcal{C}$ with error $\epsilon_{\mathsf{MCA}}(\gamma) := \xi_{\mathcal{C},d,|S|}(\gamma)$ where*

$$\xi_{\mathcal{C},d,|S|}(\gamma) := \begin{cases} n \cdot \gamma \cdot \frac{d}{|S|} & \text{if } \gamma \leq \delta_{\mathcal{C}}/(d+2) \\ O\left(\frac{n}{\eta}\right) \cdot \frac{d}{|S|} & \text{if } \gamma \leq 1 - (1 - \delta_{\mathcal{C}} + \eta)^{\frac{1}{d+2}} \end{cases} .$$

Moreover, we prove that Theorem 9 is *tight* for the regime $\gamma \leq \delta_{\mathcal{C}}/(d+2)$: we show that for every linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with relative minimum distance $\delta_{\mathcal{C}}$ and $\gamma < \delta_{\mathcal{C}}/2$, the MCA error of the univariate powers generator (when $S = \mathbb{F}$) is lower bounded by $n \cdot \gamma \cdot \frac{d}{|\mathbb{F}|}$. In fact, we prove the lower bound for the weakest form of distance-preservation instead of MCA.

Next, we outline the main ideas behind the result: in Section 2.3.1 we discuss the upper bound arising in the unique-decoding setting, and in Section 2.3.2 we discuss the upper bound arising in the list-decoding setting. Our analysis in the list-decoding setting is directly inspired by the combinatorial techniques in [30, 47].

2.3.1 Unique-decoding setting

Fix $\mathbf{u}_1, \ldots, \mathbf{u}_{d+1} \in \mathbb{F}^n$. We define the set of "bad seeds":

$$\mathsf{B} := \left\{ x \in S \;\middle|\; \exists\, T_x \subseteq [n] : \begin{array}{l} |T_x| \geq n \cdot (1-\gamma) \\ \wedge\ (\sum_{j \in [d+1]} G(x)_j \cdot \mathbf{u}_j)|_{T_x} \in \mathcal{C}|_{T_x} \\ \wedge\ \vee_{j \in [d+1]}\, \mathbf{u}_j|_{T_x} \notin \mathcal{C}|_{T_x} \end{array} \right\} .$$

The goal is to show $|\mathsf{B}| \leq n \cdot \gamma \cdot d$. Assume without loss of generality that $|\mathsf{B}| \geq d+1$ (else the claim holds trivially). For every $x \in \mathsf{B}$, define $\mathbf{u}_x := \sum_{j \in [d+1]} G(x)_j \cdot \mathbf{u}_j$ and let T_x be a set specified in the definition of B (break ties arbitrarily); as we are within the unique decoding bound, there is a unique codeword $\mathbf{c}_x \in \mathcal{C}$ such that $\mathbf{u}_x|_{T_x} = \mathbf{c}_x|_{T_x}$. We first argue that it suffices to find codewords $\mathbf{c}_1^*, \ldots, \mathbf{c}_{d+1}^* \in \mathcal{C}$ such that

$$\forall\, x \in \mathsf{B},\ \mathbf{c}_x = \sum_{j \in [d+1]} G(x)_j \cdot \mathbf{c}_j^* \ . \tag{1}$$

Consider the agreement set T defined as

$$T := \{ i \in [n] : \forall\, j \in [d+1], \mathbf{u}_j[i] = \mathbf{c}_j^*[i] \} \ .$$

By definition of B, for every $x \in \mathsf{B}$ there exists $i \in T_x$ with $i \notin T$. Moreover, note that $T \subseteq T_x$ because

$$\mathbf{u}_x|_T = \sum_{j \in [d+1]} G(x)_j \cdot \mathbf{u}_j|_T = \sum_{j \in [d+1]} G(x)_j \cdot \mathbf{c}_j^*|_T = \mathbf{c}_x|_T \ .$$

We deduce that, $T \subsetneq T_x$.

Set $\mathbf{v}_i := (\mathbf{u}_1[i] - \mathbf{c}_1^*[i], \ldots, \mathbf{u}_{d+1}[i] - \mathbf{c}_{d+1}^*[i])$ for every $i \in [n]$. The vector $\mathbf{v}_i$ is non-zero precisely when $i \notin T$. For every $i \in T_x$, $\mathbf{u}_x[i] = \mathbf{c}_x[i]$ by definition, so $G(x) \cdot \mathbf{v}_i^{\mathsf{T}} = \mathbf{u}_x[i] - \mathbf{c}_x[i] = 0$. Hence, for every $i \in T_x \setminus T$, it holds that

(i) $\mathbf{v}_i$ is non-zero and

(ii) $G(x) \cdot \mathbf{v}_i^{\mathsf{T}} = 0$.

This yields the following equivalent characterization of the set of bad seeds:

$$\mathsf{B} = \{x \in \mathsf{B} : \exists\, i \in T_x \setminus T,\ G(x) \cdot \mathbf{v}_i^\intercal = 0\} \ .$$

We use this characterization to bound $|\mathsf{B}|$ as follows:

$$\begin{aligned} |\mathsf{B}| &= |\{x \in \mathsf{B} : \exists\, i \in T_x \setminus T,\ G(x) \cdot \mathbf{v}_i^\intercal = 0\}| \\ &\leq \sum_{i \in [n] \setminus T} |\{x \in S : G(x) \cdot \mathbf{v}_i^\intercal = 0\}| \\ &\leq n \cdot d \ , \end{aligned} \tag{2}$$

where the last inequality follows from the zero-evading error of G. In fact, with more effort in the technical sections we can lower bound the size of T by $n \cdot (1-\gamma)$, which leads to the claimed upper bound $|\mathsf{B}| \leq n \cdot \gamma \cdot d$.

In sum, we reduce the problem of bounding the size of B to bounding the size of $T_x \setminus T$ and the zero-evading error of G, once there is an agreement set T such that $T \subsetneq T_x$ for all $x \in \mathsf{B}$. We use this idea also to bound the MCA error in the list-decoding regime in Section 2.3.2 and to show Theorem 5 in Section 2.5.

We are left to show the existence of $\mathbf{c}_1^*, \ldots, \mathbf{c}_{d+1}^*$ satisfying Equation (1). Fix distinct $x_1, \ldots, x_{d+1} \in B$. Let $\mathbf{M} \in \mathbb{F}^{(d+1)\times(d+1)}$ be the matrix whose i-th row is $(1, x_i, \ldots, x_i^d)$; $\mathbf{M}$ is invertible because it is a Vandermonde matrix. For every $i \in [d+1]$, let $\mathbf{c}_i$ be the (unique) closest codeword to $\mathbf{u}_{x_i}$. We construct $\mathbf{c}_1^*, \ldots, \mathbf{c}_{d+1}^*$ as follows:

$$\begin{pmatrix} \mathbf{c}_1^* \\ \vdots \\ \mathbf{c}_{d+1}^* \end{pmatrix} := \mathbf{M}^{-1} \cdot \begin{pmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{d+1} \end{pmatrix} \ .$$

For every $x \in \mathsf{B}$, a repeated application of the triangle inequality shows that

$$\delta\left(\mathbf{c}_x, G(x) \cdot \begin{pmatrix} \mathbf{c}_1^* \\ \vdots \\ \mathbf{c}_{d+1}^* \end{pmatrix}\right) < (d+2) \cdot \gamma \leq \delta_{\mathcal{C}} \ .$$

The last inequality uses the fact that $\gamma \leq \delta_{\mathcal{C}}/(d+2)$. Since $\delta_{\mathcal{C}}$ is $\mathcal{C}$'s minimum distance, we deduce the desired property:

$$\mathbf{c}_x = G(x) \cdot \begin{pmatrix} \mathbf{c}_1^* \\ \vdots \\ \mathbf{c}_{d+1}^* \end{pmatrix} \ .$$

2.3.2 List-decoding setting

We sketch our proof that, when $\gamma \leq 1 - (1 - \delta_{\mathcal{C}} + \eta)^{\frac{1}{d+2}}$, the set of bad seeds can be upper bounded as

$$|\mathsf{B}| \leq O\left(\frac{n}{\eta}\right) \cdot d \ . \tag{3}$$

The main challenge is that, as we are not guaranteed to be in the unique decoding setting, there is no notion of a unique closest codeword. We cannot find fixed $\mathbf{c}_1^*, \ldots, \mathbf{c}_{d+1}^* \in \mathcal{C}$ such that $\mathbf{c}_x = \sum_{j \in [d+1]} G(x) \cdot \mathbf{c}_j^*$ for every $x \in \mathsf{B}$. Therefore, we cannot define an agreement set T that is included in T_x for every bad seed x in B.

Our solution relies on a "cutoff" parameter $\gamma' := 1 - (1 - \delta_{\mathcal{C}} + \eta)^{\frac{1}{d+1}}$ $(\delta_{\mathcal{C}} > \gamma' > \gamma)$, adapted from [30]. We partition B into two disjoint sets B_1 and B_2:

(1) B_1 contains all seeds $x \in \mathsf{B}$ for which there exist $\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{d+1} \in \mathcal{C}$ such that their agreement set $\tilde{T} := \{i \in [n] : \wedge_{j\in[d+1]} \mathbf{u}_j[i] = \tilde{\mathbf{c}}_j[i]\}$ satisfies $\tilde{T} \subsetneq T_x$ and $|\tilde{T}| \geq n \cdot (1-\gamma')$; and
(2) $\mathsf{B}_2 := \mathsf{B} \setminus \mathsf{B}_1$.

We then separately upper bound the sizes of B_1 and B_2.

Bounding the size of B_1. The definition of B_1 allows us to apply our counting argument similarly as in Section 2.3.1. For every $\tilde{T}$, define $\mathsf{B}_{1,\tilde{T}} := \{x \in \mathsf{B}_1 : \tilde{T} \subsetneq T_x\}$. Note that $\mathsf{B}_1 \subseteq \bigcup_{\tilde{T}} \mathsf{B}_{1,\tilde{T}}$, which implies that

$$|\mathsf{B}_1| \leq |\{\tilde{T} \subseteq [n] : \exists\, \tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{d+1} \in \mathcal{C},\ \forall\, j \in [d+1],\ \mathbf{u}_j|_{\tilde{T}} = \tilde{\mathbf{c}}_j|_{\tilde{T}}\}| \cdot \max_{\tilde{T}}\{|\mathsf{B}_{1,\tilde{T}}|\} \ .$$

The number of $\tilde{T}$ is upper bounded by $1/\eta$ by a combinatorial argument based on the Corrádi lemma. For every fixed $\tilde{T}$, we bound the size of $\mathsf{B}_{1,\tilde{T}}$ via a similar argument as in Equation (2), namely, to bound the size of $T_x \setminus \tilde{T}$ and the zero-evading error of G. Together these yield

$$|\mathsf{B}_1| \leq \frac{1}{\eta} \cdot n \cdot d \ .$$

In fact, because we have a lower bound $|\tilde{T}| \geq n \cdot (1-\gamma')$, the upper bound on $|\mathsf{B}_1|$ can be improved to $\frac{1}{\eta} \cdot n \cdot \gamma' \cdot d$ as in Section 2.3.1.

Bounding the size of B_2. Since $\mathsf{B}_2 = \mathsf{B} \setminus \mathsf{B}_1$, we do not have large agreement sets $\tilde{T}$. Nevertheless, we can bound the size of B_2 using a similar counting argument as before (Equation (2)).

Assume for contradiction that B_2 is large. Using a probabilistic argument, one can show that there exists a set $\hat{T}$ where

(i) $n \cdot (1-\delta_{\mathcal{C}}) < |\hat{T}| < n \cdot (1-\gamma')$;
(ii) there exist $\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_{d+1} \in \mathcal{C}$ with $\mathbf{u}_j|_{\hat{T}} = \hat{\mathbf{c}}_j|_{\hat{T}}$; and
(iii) $|\{x \in \mathsf{B}_2 : \hat{T} \subsetneq T_x\}| \geq C(d, \eta, \delta_{\mathcal{C}})$ for some quantity $C(d, \eta, \delta_{\mathcal{C}})$.

Note that properties of $\hat{T}$ described in Items i and ii allow us to adapt our usual counting argument (Equation (2)) to obtain an upper bound on the possible number of T_x such that $\hat{T} \subsetneq T_x$. Specifically,

$$|\{x \in \mathsf{B}_2 : \hat{T} \subsetneq T_x\}| < C(d, \eta, \delta_{\mathcal{C}}) \ ,$$

which contradicts Item iii above. Hence, B_2 cannot be large.

2.4 The affine space generator

The affine space generator is $G\colon \mathbb{F}^s \to \mathbb{F}^{s+1}$ where $G(x_1, \ldots, x_s) := (1, x_1, \ldots, x_s)$. We show that, for any linear code $\mathcal{C} \subseteq \mathbb{F}^n$, the affine space generator has the same MCA error as the affine line generator $x \mapsto (1, x)$, up to a small constant multiplicative overhead. This implies our Lemma 4.

Let $\mathbf{u}_0, \ldots, \mathbf{u}_s \in \mathbb{F}^n$ be words, and define $U := \mathsf{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_s\}$ and $V := \mathbf{u}_0 + U$. The intuition for the proof is fairly simple: if the "bad seed" set for the affine space generator is large, then we can find a line of the form $\mathbf{u}' + t \cdot \tilde{\mathbf{u}}$, with $\mathbf{u}' \in V$ and $\tilde{\mathbf{u}} \in U$ that contradicts the MCA error of the affine line generator. More precisely, there are many $t \in \mathbb{F}$ such that there exists $T \subseteq [n]$ with $(\mathbf{u}' + t \cdot \tilde{\mathbf{u}})|_T \in \mathcal{C}|_T$ and $\tilde{\mathbf{u}}|_T \notin \mathcal{C}|_T$.

Here we outline this reduction. The set of bad seeds for the affine space generator is

$$\mathsf{B} := \left\{ \mathbf{x} = (x_1, \ldots, x_s) \in \mathbb{F}^s \ : \begin{array}{l} \exists\, T_{\mathbf{x}} \subseteq [n] : \\ \quad |T_{\mathbf{x}}| \geq n \cdot (1-\gamma) \\ \quad \wedge\, (\mathbf{u}_0 + \sum_{j\in[s]} x_j \cdot \mathbf{u}_j)|_{T_{\mathbf{x}}} \in \mathcal{C}|_{T_{\mathbf{x}}} \\ \quad \wedge\, \exists j \in \{0,\ldots,s\},\ \mathbf{u}_j|_{T_{\mathbf{x}}} \notin \mathcal{C}|_{T_{\mathbf{x}}} \end{array} \right\} .$$

For any $\mathbf{x} \in \mathsf{B}$, fix $T_{\mathbf{x}}$ as in the definition of B. The first step is to find $(\lambda_1^\star, \ldots, \lambda_s^\star) \in \mathbb{F}^s$ such that if $\tilde{\mathbf{u}} := \sum_{j\in[s]} \lambda_j^\star \cdot \mathbf{u}_j \in U$ then $\mathsf{B}' := \{\mathbf{x} \in \mathsf{B} : \tilde{\mathbf{u}}|_{T_{\mathbf{x}}} \notin \mathcal{C}|_{T_{\mathbf{x}}}\} \subseteq \mathsf{B}$ is large, which we do via a linear algebraic observation. Consider the quotient linear space $\mathbb{F}^{|T_{\mathbf{x}}|}/\mathcal{C}|_{T_{\mathbf{x}}}$ and the linear map

$$\mathcal{L} : \mathbb{F}^s \to \mathbb{F}^{|T_{\mathbf{x}}|}/\mathcal{C}|_{T_{\mathbf{x}}}, \quad (\lambda_1, \ldots, \lambda_s) \mapsto \sum_{j=1}^{s} \lambda_j \cdot [\mathbf{u}_j] \ ,$$

where $[\mathbf{u}]$ denotes the class of $\mathbf{u}|_{T_{\mathbf{x}}}$ in $\mathbb{F}^{|T_{\mathbf{x}}|}/\mathcal{C}|_{T_{\mathbf{x}}}$. For every $\mathbf{x} \in \mathsf{B}$, the kernel of $\mathcal{L}$ is a proper subspace of $\mathbb{F}^s$. By an averaging argument over the choice of $(\lambda_1, \ldots, \lambda_s)$, we deduce that there exists $(\lambda_1^\star, \ldots, \lambda_s^\star) \in \mathbb{F}^s$ such that B' has size at least $|\mathsf{B}|\,(1 - 1/|\mathbb{F}|)$.

We are left to construct $\mathbf{u}' \in V$ such that for many $t \in \mathbb{F}$, $\mathbf{u}' + t \cdot \tilde{\mathbf{u}}$ breaks the MCA condition. The key idea is to consider the following line *in seed space* for $\mathbf{v} \in \mathbb{F}^s$:

$$\mathbf{x}_{\mathbf{v}}(t) := \mathbf{v} + t \cdot (\lambda_1^\star, \ldots, \lambda_s^\star) \ .$$

While there is no guarantee that $\mathbf{x}_{\mathbf{v}}(t) \in \mathsf{B}'$, by using another averaging argument, there exists $\mathbf{v}^\star \in \mathbb{F}^s$ such that:

$$\Pr_{t\in\mathbb{F}}[\mathbf{x}_{\mathbf{v}^\star}(t) \in \mathsf{B}'] \geq \frac{|\mathsf{B}'|}{|\mathbb{F}^s|} \ .$$

In other words, the density of $t \in \mathbb{F}$ such that $\mathbf{x}_{\mathbf{v}^\star}(t) \in \mathsf{B}'$ is at least the density of B' in $\mathbb{F}^s$. Set $\mathbf{u}' := \mathbf{u}_0 + \sum_{j\in[s]} \mathbf{v}_j^\star \cdot \mathbf{u}_j$. By the definition of B', for every $t \in \mathbb{F}$ such that $\mathbf{x}_{\mathbf{v}^\star}(t) = \mathbf{x} \in \mathsf{B}'$, we have:

$$(\mathbf{u}_0 + \sum_{j\in[s]} x_j \cdot \mathbf{u}_j)|_{T_{\mathbf{x}}} \in \mathcal{C}|_{T_{\mathbf{x}}} \quad \text{and} \quad \tilde{\mathbf{u}}|_{T_{\mathbf{x}}} \notin \mathcal{C}|_{T_{\mathbf{x}}} \ .$$

But here $\mathbf{u}_0 + \sum_{j\in[s]} x_j \cdot \mathbf{u}_j = \mathbf{u}' + t \cdot \tilde{\mathbf{u}}$. Let ϵ_{MCA} be the error of the affine line generator for the code $\mathcal{C}$. By our discussion,

$$\epsilon_{\mathsf{MCA}}(\gamma) \geq \Pr_{t\in\mathbb{F}} \left[\begin{array}{l} \exists\, T_t \subseteq [n] : \\ \quad |T_t| \geq n \cdot (1-\gamma) \\ \quad \wedge\, (\mathbf{u}' + t \cdot \tilde{\mathbf{u}})|_{T_t} \in \mathcal{C}|_{T_t} \\ \quad \wedge\, \tilde{\mathbf{u}}|_{T_t} \notin \mathcal{C}|_{T_t} \end{array} \right] \geq \Pr_{t\in\mathbb{F}}[\mathbf{x}_{\mathbf{v}^\star}(t) \in \mathsf{B}'] \geq \frac{|\mathsf{B}'|}{|\mathbb{F}^s|} \geq \frac{|\mathsf{B}|\,(1 - 1/|\mathbb{F}|)}{|\mathbb{F}^s|} \ .$$

In turn, the size of B is at most $\epsilon_{\mathsf{MCA}}(\gamma) \cdot |\mathbb{F}|^s \cdot (1 - 1/|\mathbb{F}|)^{-1}$, which concludes the proof.

2.5 Polynomial generators and Reed–Solomon codes

We outline our proof of Theorem 5. Similarly to Theorem 3, first we establish that the univariate powers generator has MCA for Reed–Solomon codes up to the list-decoding bound and thereafter rely on our MCA-preserving transformations to establish MCA for all polynomial generators. We discussed the second (generic) step in Section 2.2 (see Lemmas 6 and 7); below we focus on the first step.

We prove that the univariate powers generator $G(x) := (1, x, \ldots, x^d)$ has MCA for $\mathcal{C} := \mathsf{RS}[\mathbb{F}, D, k]$ with error

$$\epsilon_{\mathsf{MCA}}(\gamma) := O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{d}{|\mathbb{F}|} \quad \text{if } \gamma \leq 1 - \left(1 + \frac{1}{2m}\right) \cdot \sqrt{\rho} \ .$$

We bound the size of the set of bad seeds B defined as follows:

$$\mathsf{B} := \left\{ x \in \mathbb{F} \;\middle|\; \exists\, T_x \subseteq D : \begin{array}{l} |T_x| \geq n \cdot (1 + 1/2m) \cdot \sqrt{\rho} \\ \wedge\ \left(\sum_{i=0}^{d} z^i \cdot \mathbf{u}_i\right)\Big|_{T_x} \in \mathcal{C}|_{T_x} \\ \wedge\, \exists\, \ell \in \{0, \ldots, d\}, \mathbf{u}_\ell|_{T_x} \notin \mathcal{C}|_{T_x} \end{array} \right\} \ .$$

Adapting techniques from [10, 38], B can be partitioned into sets $(\mathsf{B}_i)_{i \in [\frac{m+1/2}{\sqrt{\rho}}]}$ where, for each i, if $|\mathsf{B}_i| > \frac{(m+1/2)^6 \cdot d \cdot n^2}{3\rho}$ then there exist codewords $\mathbf{c}_1, \ldots, \mathbf{c}_{d+1}$ such that

$$\mathsf{B}_i = \left\{ x \in \mathbb{F} \;\middle|\; \exists\, T_x \subseteq D : \begin{array}{l} |T_x| \geq n \cdot (1 + 1/2m) \cdot \sqrt{\rho} \\ \wedge\ \left(\sum_{j=0}^{d} x^j \cdot \mathbf{u}_j\right)\Big|_{T_x} = \left(\sum_{j=0}^{d} x^j \cdot \mathbf{c}_j\right)\Big|_{T_x} \\ \wedge\, \exists\, \ell \in \{0, \ldots, d\}, \mathbf{u}_\ell|_{T_x} \notin \mathcal{C}|_{T_x} \end{array} \right\} \ .$$

Letting $\tilde{T} := \{i \in [n] : \forall j \in [d+1],\ \mathbf{u}_j[i] = \mathbf{c}_j[i]\}$, we deduce that $\tilde{T} \subsetneq T_x$ for every $x \in \mathsf{B}_i$. Hence, using the idea explained in Section 2.3 (specifically the counting argument captured in Equation (2)) we show that the number of $x \in \mathbb{F}$ with $\tilde{T} \subsetneq T_x$ is at most $n \cdot d$, which implies that $|\mathsf{B}_i| \leq n \cdot d$. Since $|\mathsf{B}_i| > \frac{(m+1/2)^6 \cdot d \cdot n^2}{3\rho} > n \cdot d$, we reach a contradiction. Therefore,

$$|\mathsf{B}| = \sum_{i \in [\frac{m+1/2}{\sqrt{\rho}}]} |\mathsf{B}_i| \leq \frac{m + 1/2}{\sqrt{\rho}} \cdot \frac{(m+1/2)^6 \cdot d \cdot n^2}{3\rho} = \frac{(m+1/2)^7 \cdot d \cdot n^2}{3\rho^{3/2}} \ .$$

2.6 An MCA toolbox, and going beyond polynomial generators

Our results rely, in particular, on several transformations that we prove preserve MCA. In fact, via such transformations, we prove that a zero-evading generator in [1], which is *not* a polynomial generator, has MCA for arbitrary linear codes; no prior work shows any distance preservation property for generators that are not polynomial generators. This shows that the "MCA toolbox" that we develop enables going beyond polynomial generators. Below we briefly discuss the generator in [1] and our MCA toolbox.

MCA for a non-polynomial generator. Fix a finite field $\mathbb{F}_q$ with q elements and let $\mathbb{F}_{q^k}$ be its extension of degree k. Fix an isomorphism ϕ of $\mathbb{F}$-vector spaces that maps $\mathbb{F}_{q^k}$ to $\mathbb{F}_q^k$. The generator $G \colon \mathbb{F}_{q^k} \times \mathbb{F}_q^k \to \mathbb{F}_q^\ell$ in [1] is defined as

$$G(x, \mathbf{v}) := (\phi(x^j) \cdot \mathbf{v}^\intercal)_{j \in [\ell]} \ .$$

The zero-evading error of this generator is $\epsilon_{\mathsf{ZE}} = \frac{\ell - 1}{q^k} + \frac{1}{q}$.

While G is not a polynomial generator, it can be viewed as the *matrix-vector composition* of two polynomial generators in the following sense. Consider $G_{\mathsf{in}} \colon \mathbb{F}_q^k \to \mathbb{F}_q^k$ and $G_{\mathsf{out}} \colon \mathbb{F}_{q^k} \to (\mathbb{F}_{q^k})^\ell$ defined as

$$\forall\, \mathbf{v} \in \mathbb{F}_q^k, G_{\mathsf{in}}(\mathbf{v}) := \mathbf{v} \ ;$$
$$\forall\, x \in \mathbb{F}_{q^k}, G_{\mathsf{out}}(x) := (x, x^2, \ldots, x^\ell) \ .$$

Then

$$\forall\, j \in [\ell], G(x, \mathbf{v})[j] = \langle \phi(G_{\mathsf{out}}(x)[j]), G_{\mathsf{in}}(\mathbf{v}) \rangle \ .$$

In other words, $G(x, \mathbf{v}) = \mathbf{A}_x \cdot G_{\mathsf{in}}(\mathbf{v})^\intercal$ where $\mathbf{A}_x \in \mathbb{F}^{\ell \times k}$ is the matrix whose j-th row equals $\phi(G_{\mathsf{out}}(x)[j])$.

We show that the matrix-vector composition of two generators as above results in a generator whose MCA error is the sum of their two MCA errors. Moreover, we show that removing some coordinates in the output of a generator (to construct a sub-generator) does not change its MCA error. Since G_{in} is a sub-generator of the affine space generator and G_{out} is a sub-generator of the univariate powers generator, we can bound the MCA error of G using Theorem 3.

Toolbox for MCA. We summarize the transformations that we prove preserve MCA. Formal statements and proofs for these are provided in the full version.

- A sub-generator of a generator has the same MCA error for any code as the generator.
- The linear transformation of a generator has the same MCA error for any code as the generator.
- The tensor product of two generators has MCA error that is the sum of the MCA errors of the original generators.
- If the affine line generator $x \mapsto (1, x)$ has MCA error for a code, then the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$ has the same MCA error for the code.
- The matrix-vector composition of two generators has MCA error that is the sum of the MCA errors of the original generators.

These transformations enable us to capture all polynomial generators, and even beyond that as [1]'s generator illustrates. The MCA properties of these transformations are likely to be useful beyond this paper.

References

1 Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992. doi:10.1002/RSA.3240030308.

2 Omar Alrabiah, Venkatesan Guruswami, and Ray Li. Randomly punctured Reed–Solomon codes achieve list-decoding capacity over linear-sized fields. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC '24, pages 1458–1469, 2024. doi:10.1145/3618260.3649634.

3 Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 24th ACM Conference on Computer and Communications Security*, CCS '17, pages 2087–2104, 2017. doi:10.1145/3133956.3134104.

4 Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed–solomon proximity testing with fewer queries. In *Proceedings of the 44th Annual International Cryptology Conference*, CRYPTO '24, pages 380–413, 2024. doi:10.1007/978-3-031-68403-6_12.

5 Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. WHIR: Reed–solomon proximity testing with super-fast verification. In *Proceedings of the 44th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '25, pages 214–243, 2025. doi:10.1007/978-3-031-91134-7_8.

6 Daniel Augot, Sarah Bordage, and Jade Nardi. Efficient multivariate low-degree tests via interactive oracle proofs of proximity for polynomial codes. *Des. Codes Cryptography*, 91(3):1111–1151, November 2022. doi:10.1007/S10623-022-01134-Z.

7 Anubhav Baweja, Pratyush Mishra, Tushar Mopuri, and Matan Shtepel. FICS and FACS: Fast iopps and accumulation via code-switching. Cryptology ePrint Archive, Report 2025/737, 2025.

8 Anubhav Baweja, Pratyush Mishra, Tushar Mopuri, and Matan Shtepel. Query-optimal IOPPs for linear-time encodable codes. IACR Cryptology ePrint Archive, Report 2025/1588, 2025.

9 Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed–Solomon interactive oracle proofs of proximity. In *Proceedings of the 45th International Colloquium on Automata, Languages and Programming*, ICALP '18, pages 14:1–14:17, 2018. doi:10.4230/LIPIcs.ICALP.2018.14.

10 Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for Reed–Solomon codes. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '20, pages 900–909, 2020. doi:10.1109/FOCS46700.2020.00088.

11 Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: Sampling outside the box improves soundness. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference*, ITCS '20, pages 5:1–5:32, 2020. doi:10.4230/LIPIcs.ITCS.2020.5.

12 Eli Ben-Sasson, Swastik Kopparty, and Shubhangi Saraf. Worst-case to average case reductions for the distance to a code. In *Proceedings of the 33rd ACM Conference on Computer and Communications Security*, CCS '18, pages 24:1–24:23, 2018. doi:10.4230/LIPIcs.CCC.2018.24.

13 Rishabh Bhadauria, Zhiyong Fang, Carmit Hazay, Muthuramakrishnan Venkitasubramaniam, Tiancheng Xie, and Yupeng Zhang. Ligero++: A new optimized sublinear iop. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, pages 2025–2038, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3372297.3417893.

14 Jonathan Bootle, Alessandro Chiesa, Ziyi Guan, and Siqi Liu. Linear-time probabilistic proofs with sublinear verification for algebraic automata over every field. Cryptology ePrint Archive, Paper 2022/1056, 2022.

15 Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-knowledge succinct arguments with a linear-time prover. In *Proceedings of the 42nd Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '22, 2022.

16 Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. Generic Reed–Solomon codes achieve list-decoding capacity. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC '23, pages 1488–1501, 2023. doi:10.1145/3564246.3585128.

17 Martijn Brehm, Binyi Chen, Ben Fisch, Nicolas Resch, Ron D. Rothblum, and Hadas Zeilberger. Blaze: Fast SNARKs from interleaved RAA codes. In *Proceedings of the 44th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '25, 2025.

18 Benedikt Bünz, Alessandro Chiesa, Giacomo Fenzi, and William Wang. Linear-Time Accumulation Schemes. In *Proceedings of the 16th Theory of Cryptography Conference*, TCC '25, 2025.

19 Benedikt Bünz, Pratyush Mishra, Wilson Nguyen, and William Wang. Accumulation without homomorphism. In *Proceedings of the 16th Innovations in Theoretical Computer Science Conference*, ITCS '25, pages 23:1–23:25, 2025. doi:10.4230/LIPIcs.ITCS.2025.23.

20 Benedikt Bünz, Pratyush Mishra, Wilson Nguyen, and William Wang. Arc: Accumulation for Reed–Solomon codes. In *Proceedings of the 45th Annual International Cryptology Conference*, CRYPTO '25, 2025.

21 Yeyuan Chen and Zihan Zhang. Explicit folded Reed–Solomon and multiplicity codes achieve relaxed generalized singleton bounds. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 1–12, 2025. doi:10.1145/3717823.3718114.

22 Mahdi Cheraghchi, Venkatesan Guruswami, and Ameya Velingker. Restricted isometry of fourier matrices and list decodability of random linear codes. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1764–1778, 2014.

23 Elizabeth Crites and Alistair Stewart. On reed–solomon proximity gaps conjectures. Cryptology ePrint Archive, Paper 2025/2046, 2025.

24 Benjamin Diamond and Jim Posen. Polylogarithmic proofs for multilinears over binary towers. Cryptology ePrint Archive, Report 2024/504, 2024.

25 Benjamin E. Diamond and Angus Gruen. On the distribution of the distances of random words. Cryptology ePrint Archive, Paper 2025/2010, 2025.

26 Benjamin E. Diamond and Jim Posen. Proximity Testing with Logarithmic Randomness. *IACR Communications in Cryptology*, 1(1), 2024. doi:10.62056/AKSDKP10.

27 Ethereum Foundation. The Proximity Prize, 2025. URL: https://proximityprize.org/.

28 Asaf Ferber, Matthew Kwan, and Lisa Sauermann. List-decodability with large radius for Reed–Solomon codes. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science*, FOCS '22, pages 720–726, 2022.

29 Yiwen Gao, Dongliang Cai, Yang Xu, and Haibin Kan. From list-decodability to proximity gaps. Cryptology ePrint Archive, Paper 2025/870, 2025.

30 Yiwen Gao, Haibin Kan, and Yuan Li. Linear proximity gap for linear codes within the 1.5 johnson bound. IACR Cryptology ePrint Archive, Report 2024/1810, 2024.

31 Albert Garreta, Nicolas Mohnblatt, and Benedikt Wagner. A simplified round-by-round soundness proof of FRI. Cryptology ePrint Archive, Paper 2025/1993, 2025.

32 Eitan Goldberg, Chong Shangguan, and Itzhak Tamo. List-decoding and list-recovery of Reed–Solomon codes beyond the Johnson radius for every rate. *IEEE Transactions on Information Theory*, 69(4):2261–2268, 2023. doi:10.1109/TIT.2022.3222877.

33 Rohan Goyal and Venkatesan Guruswami. Optimal proximity gaps for subspace-design codes and (random) Reed–Solomon codes. Cryptology ePrint Archive, Paper 2025/2054, 2025.

34 Zeyu Guo, Ray Li, Chong Shangguan, Itzhak Tamo, and Mary Wootters. Improved list-decodability and list-recoverability of Reed–Solomon codes via tree packings. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science*, FOCS '21, pages 708–719, 2021.

35 Zeyu Guo and Zihan Zhang. Randomly punctured Reed–Solomon codes achieve the list decoding capacity over polynomial-size alphabets. In *Proceedings of the 64th IEEE Symposium on Foundations of Computer Science*, FOCS '23, pages 164–176, 2023. doi:10.1109/FOCS57990.2023.00019.

36 Venkatesan Guruswami, Johan Håstad, and Swastik Kopparty. On the list-decodability of random linear codes. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 176–186, 2011.

37 Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5):1021–1034, 2002. doi:10.1109/18.995539.

38 Ulrich Haböck. A note on mutual correlated agreement for Reed–Solomon codes. Cryptology ePrint Archive, Report 2025/2110, 2025.

39 Matan Levi, Jonathan Mosheiff, and Nikhil Shagrithaya. Random Reed–Solomon codes and random linear codes are locally equivalent. In *Proceedings of the 66th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '25, 2025.

40 Ray Li and Mary Wootters. Improved list-decodability of random linear binary codes. In *Proceedings of the 22nd International Conference on Randomization and Computation*, RANDOM '18, pages 50:1–50:19, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.50.

41 Dor Minzer and Kai Zhe Zheng. Improved Round-by-round Soundness IOPs via Reed-Muller Codes. In *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '25, 2025.

42 Andrija Novakovic and Guillermo Angeris. Ligerito: A small and concretely fast polynomial commitment scheme. Cryptology ePrint Archive, Paper 2025/1187, 2025.

43 Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: Delegating computation in sublinear time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, STOC '13, pages 793–802, 2013. doi:10.1145/2488608.2488709.

44 Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 764–773, 2014. doi:10.1145/2591796.2591797.

45 Chong Shangguan and Itzhak Tamo. Combinatorial list-decoding of Reed–Solomon codes beyond the Johnson radius. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC '20, pages 538–551, 2020. doi:10.1145/3357713.3384295.

46 Ruihan Wang, Carmit Hazay, and Muthuramakrishnan Venkitasubramaniam. Ligetron: Lightweight scalable end-to-end zero-knowledge proofs post-quantum zk-snarks on a browser. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 1760–1776, 2024. doi:10.1109/SP54263.2024.00086.

47 Hadas Zeilberger. Khatam: Reducing the communication complexity of code-based SNARKs. IACR Cryptology ePrint Archive, Paper 2024/1843, 2025.

48 Hadas Zeilberger, Binyi Chen, and Ben Fisch. Basefold: Efficient field-agnostic polynomial commitment schemes from foldable codes. In *Proceedings of the 44th Annual International Cryptology Conference*, volume 14929 of *CRYPTO '24*, pages 138–169, 2024. doi:10.1007/978-3-031-68403-6_5.

Polynomial Identity Testing for Read-4 Arithmetic Formulas

Nimrod Kaplan
Blavatnik School of Computer Science and AI, Tel Aviv University, Israel

Amir Shpilka
Blavatnik School of Computer Science and AI, Tel Aviv University, Israel

Abstract

We present the first algorithms for polynomial identity testing (PIT) of read-4 arithmetic formulas in the non-multilinear setting. Specifically, we give a polynomial-time PIT algorithm in the whitebox model and a quasi-polynomial-time algorithm in the blackbox model. Since our techniques are based on proving *hardness of representation* results, we extend our algorithms to *orbits* of read-4 formulas under the action of the affine linear group. Prior to our work, no subexponential white- or blackbox algorithms were known for this class of formulas. All our results hold over any field $\mathbb{F}$ with $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) \geq 5$.

Prior work addressed only restricted cases. Anderson, van Melkebeek, and Volkovich (*Computational Complexity*, 2015) studied *multilinear* read-k formulas, giving a polynomial-time whitebox PIT algorithm and a quasi-polynomial-time blackbox algorithm. Without the multilinearity restriction, Mahajan, Rao, and Sreenivasaiah (*TCS*, 2014) gave polynomial-time whitebox algorithms for read-2 and read-3 formulas, Prakriya (*Doctoral Thesis*, 2019) obtained quasi-polynomial-time blackbox PIT algorithm for both read-2 and read-3 formulas. Independently, Shamir (*Master's Thesis*, 2022) obtained a quasi-polynomial-time blackbox PIT algorithm for read-2 formulas. For *bounded-depth* read-k formulas, Agrawal, Saha, Saptharishi, and Saxena (*SICOMP*, 2016) obtained a polynomial-time blackbox algorithm in the non-multilinear case. The running time of their algorithm is $n^{k^{2^{\Delta}}}$ for read-k, depth-Δ formulas, and hence it is applicable only to constant depth.

Partial derivatives are a central tool in the study of deterministic PIT for bounded-read formulas. However, for non-multilinear RkF, differentiation may increase the number of reads. To address this, we develop new structural results that ensure "nice behavior" of derivatives. Specifically, we introduce a new *Fragmentation Lemma* that reduces the PIT problem for general RkFs to simpler models via differentiation. In addition, we define the notion of *dominating degree patterns* and show that, in certain cases, taking partial derivatives with respect to these patterns preserves the read count.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases algebraic complexity theory, polynomial identity testing, PIT, bounded read formulas

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.25

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2026/076/ [27]

Funding This research was funded by the European Union (ERC, EACTP, 101142020). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgements We thank the anonymous referee who brought [39] to our attention.



© Nimrod Kaplan and Amir Shpilka;
licensed under Creative Commons License CC-BY 4.0



COMPUTATIONAL COMPLEXITY CONFERENCE

1 Introduction

Arithmetic circuits are directed acyclic graphs that compute multivariate polynomials using the arithmetic operations $+$ and $\times$. Similarly, arithmetic formulas are circuits whose underlying graph is a tree. These are the most common models for studying the complexity of computing polynomials.

Polynomial Identity Testing (PIT) is a central problem in algebraic complexity theory that asks, given an arithmetic circuit or formula, whether it computes the identically zero polynomial. The problem is studied in two main settings. In the *whitebox* model, the circuit is explicitly given to the algorithm. In the *blackbox* model the algorithm can only make query access to the circuit. That is, it can only evaluate the polynomial at chosen inputs. Solving PIT in the blackbox model is equivalent to constructing a *hitting set* for the class of polynomials under consideration. A hitting set consists of a set of inputs such that any nonzero polynomial in the class evaluates to a nonzero value on at least one point in the set.

If randomness is allowed, the polynomial identity lemma (also known as the Ore-DeMillo-Lipton-Schwartz-Zippel lemma [38, 12, 45, 54]) guarantees a randomly chosen input from a sufficiently large grid will, with high probability, provide a nonzero evaluation point for the circuit. The main challenge in the field, and indeed, one of the most fundamental open problems in theoretical computer science, is to find a *deterministic* algorithm. PIT is the most general and well-studied algebraic derandomization problem, with wide range of applications, including primality testing [2, 3], parallel algorithms for perfect matchings [31, 15, 52], linear matroid intersection [23], derandomization of Noether normalization lemma [17, 37] and even the celebrated proof that IP equals PSPACE [32, 46].

Another reason for the importance of the PIT problem is its deep connection to circuit lower bounds. A series of works [24, 26, 1, 14, 11, 29, 20, 30, 7] established that derandomizing PIT is essentially equivalent to proving strong lower bounds for arithmetic circuits, in various settings. Furthermore, depth reduction results show that solving PIT even for highly restricted models such as depth-3 and depth-4 circuits [4, 28, 53, 21] would have major consequences.

Given its importance and the absence of deterministic algorithms in the general case, much research has focused on restricted models of computation. There is a rich body of work on PIT for small-depth circuits, read-once algebraic branching programs, bounded-read formulas, read-once determinants, and other restricted settings. For broader surveys on PIT and its connections, see [43, 51, 42, 44, 13].

One natural approach to derandomizing PIT for arithmetic formulas is to restrict the number of times each variable is read at the leaves. This line of work began with Shpilka and Volkovich [50], who provided a quasi-polynomial-size hitting set and a polynomial-time whitebox algorithm for read-once formulas, i.e., formulas in which each variable appears at most once. They also showed how to reconstruct (learn) such formulas deterministically [49]. Minahan and Volkovich completely settled the case of read-once formulas by providing a polynomial-sized hitting set. The works [49, 50] also introduced what is now known as the SV-generator, a k-wise independent polynomial map.

Anderson, van Melkebeek, and Volkovich significantly extended this line of work by studying *read-k multilinear formulas* (multilinear RkF for short),[1] for constant k. These are formulas in which each variable appears in at most k-leaves, and each gate computes

[1] We later refer to their model as *structurally-multilinear* to distinguish it from cases where multilinearity arises only through cancellations of nonmultilinear terms. In the literature this is often called *syntactically-multilinear*, but in our setting we wish to capture this "syntactic" behavior variable-wise and therefore use a different term.

a multilinear polynomial. They presented a polynomial-time whitebox algorithm and a quasi-polynomial-size hitting set. Notably, their result also implies a $2^{(1-\varepsilon)n}$-size hitting set for general multilinear formulas (without any restriction on the number of reads) of size $O(n)$.

As the results of [6] only hold for multilinear formulas, attempts were made to remove this restriction. Mahajan, Rao, and Sreenivasaiah [34] studied read-2 and read-3 formulas, and gave polynomial-time whitebox PIT algorithms for them. Prakriya [39] gave quasi-polynomial-size hitting sets for read-2 and read-3 formulas. Independently, Shamir [47] gave quasi-polynomial-size hitting sets for read-2 formulas.

In a different direction, Anderson et al. [5] studied read-k algebraic branching programs (Rk-ABPs, for short). An ABP is a model that subsumes formulas and is believed to be weaker than circuits but stronger than formulas. Any RkF can be represented as an Rk-ABP, meaning that each variable labels at most k edges (for more on ABPs see [51, 42]). In general though, in an ROABP each variable can appear on multiple edges, as long as they all appear in the *same layer*. Thus, it is a much stronger model than read-once formulas. Anderson et al. gave a subexponential-time whitebox[2] PIT algorithm for Rk-ABPs. Specifically, their algorithm runs in time $2^{\tilde{O}(n^{1-1/2^{k-1}})}$, and yields an algorithm with similar running time for RkFs.

Other than these results, the problem of derandomizing PIT for RkFs remains widely open, both in the whitebox and blackbox settings.

In this work, we design a deterministic quasi-polynomial-time blackbox PIT algorithm for R4Fs. Before our result, no nontrivial deterministic blackbox or whitebox PIT algorithms were known for R4Fs, apart from the slightly subexponential Rk-ABP algorithm of [5].

1.1 Our Results

All our results are stated in terms of *independent polynomial maps*, a key tool that allows us to reduce the number of variables while preserving nonzeroness. We denote with $\mathcal{G}_m$ a polynomial map with seed length m.[3]

The main technical contribution of this work is a quasi-polynomial-time blackbox algorithm for R4Fs. The class R4F is of particular interest, as four is the minimum number of reads required for a non-structural variable to appear.

▶ **Definition 1** (Structural variable, Structural formula)**.** *For some algebraic formula F, a variable $x_i \in var(F)$ is said to be structural if for each pair of gates $v, u \in F$, such that v is an ancestor of u, it holds that $\deg_{x_i}(v) \geq \deg_{x_i}(u)$.*

A formula F is structural if every $x_i \in var(F)$ is structural. ⌟

Non-structural variables may cause "massive cancellation" which is the core difficulty in verifying polynomial identities.

For a formula F over the variables $\mathbf{x}$ we denote with $p_F(\mathbf{x})$ the polynomial computed at the root of the formula.

All our results hold over any field $\mathbb{F}$ such that $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) \geq 5$. This restriction arises because our argument uses partial derivatives of order at most four, which may vanish identically over fields of smaller characteristic. Henceforth, we omit explicit mention of the underlying field. Note, however, that constructing a hitting set may require passing to an extension field if $|\mathbb{F}|$ is too small.

[2] The algorithm uses its whitebox access only to determine the order in which the variables are read. Such a model is sometimes referred to in the PIT literature as *grey-box*.

[3] Explicit constructions such as the SV-generator [50] and the RFE-generator [25] are known.

▶ **Theorem 2** (Main blackbox result). *Let $p_F(\mathbf{x})$ be an n-variate polynomial computable by an* R4F *F, over a field $\mathbb{F}$. There exists an absolute constant c_2, such that $p_F(\mathbf{x}) \not\equiv 0$ if and only if*

$$p_F \circ \mathcal{G}_{9\log n + c_2} \not\equiv 0\,.$$

We also present a polynomial-time whitebox algorithm for the same model.

▶ **Theorem 3** (Main whitebox result). *Let F be an n-variate* R4F *formula over a field $\mathbb{F}$. There exists an algorithm that, given whitebox access to F, checks in time poly(n) whether $p_F \equiv 0$.*

For R3Fs, we obtain a constant-factor improvement over the seed length used for the hitting set of [39] (and also compared to Theorem 2)

▶ **Theorem 4.** *Let $F \in$* R3F *be nonzero. Then, there exists a constant c_4*

$$p_F \circ \mathcal{G}_{2\log n + c_4} \not\equiv 0\,.$$

We next give a constant-factor improvement in the seed length over the result of [47], and an additive constant improvement in the seed length over the result of [39] (and also compared to Theorem 4).

▶ **Theorem 5.** *Let $F \in$* R2F *be nonzero. Then*

$$p_F \circ \mathcal{G}_{\log n + 4} \not\equiv 0\,.$$

Having established deterministic hitting sets for bounded-read formulas, we next extend these results to orbit classes. I.e., we compose polynomials with *full-rank affine transformations*. The key observation is that hardness of representation results for the base class carry over to the corresponding orbit class. This allows us to "lift" results from the backbone class (the class whose being composed with the affine transformation) to the orbit class. In all results below, $N \le n$ denotes the number of variables of the backbone formulas.

We begin by generalizing the results for sums of ROFs from [50] and [9].

▶ **Theorem 6.** *Let $F_1(\mathbf{y}), F_2(\mathbf{y}), \ldots, F_k(\mathbf{y}) \in$* ROF*, and let $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$. Then*

$$\sum_{i=1}^{k} p_{F_i}(\mathbf{y}) \not\equiv 0 \quad \Rightarrow \quad \sum_{i=1}^{k} p_{F_i}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{\log N + 3k} \not\equiv 0\,.$$

If $k = 2$ then we can compose with $\mathcal{G}_{\log N + 3}$ instead of $\mathcal{G}_{\log N + 6}$.

We next prove the result for orbits of structurally multilinear polynomials. Our proof closely follows the argument of [6], except that we use our version of the Fragmentation Lemma rather than the version introduced in that work.

▶ **Theorem 7.** *Let $F(\mathbf{y}) \in$* RkF *be a structurally multilinear formula, and let $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$. Then, for $r_k \coloneqq r_{2,2k} = k^{o(k)}$,*

$$p_F(\mathbf{y}) \not\equiv 0 \quad \Rightarrow \quad p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{r_k + k\log N} \not\equiv 0\,.$$

Finally, we lift our results for general read-2, read-3, and read-4 formulas to their corresponding orbit classes.

▶ **Theorem 8.** *Let* $F(\mathbf{y}) \in \mathsf{R2F}$ *and let* $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$*. Then*

$$p_F(\mathbf{y}) \not\equiv 0 \quad \Rightarrow \quad p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{3\log N+5} \not\equiv 0.$$

▶ **Theorem 9.** *Let* $F(\mathbf{y}) \in \mathsf{R3F}$ *and let* $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$*. Then, there exists a constant* c_9

$$p_F(\mathbf{y}) \not\equiv 0 \quad \Rightarrow \quad p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{6\log N+c_9} \not\equiv 0\,.$$

▶ **Theorem 10.** *Let* $F(\mathbf{y}) \in \mathsf{R4F}$ *and let* $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$*. Then, there exists a constant* c_{10}

$$p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \not\equiv 0 \quad \Rightarrow \quad p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{12\log N+c_{10}} \not\equiv 0\,.$$

To the best of our knowledge, prior to this work, no nontrivial blackbox PIT algorithm was known for any of these classes.

An important tool in our proofs, and indeed in most previous work on PIT for bounded-read formulas, is the use of partial derivatives and the analysis of the resulting simplified formula. While this approach is effective for *multilinear* formulas, this is no longer the case for formulas with non-multilinear structure; in fact, differentiation may result in a more complex formula than the original one.

In this work, we introduce several structural tools that enable the identification of partial derivatives that do achieve the desired simplification. First, we define two key sets of gates: the **R(k − 1)F-frontier** , which identifies "maximal" R(k − 1)F subformulas, and the set $\boldsymbol{\mathcal{F}^{(2)}}$, which captures "minimal" RkF subformulas.

▶ **Definition 11** (R(k − 1)F Frontier)**.** *Let* F *be an* RkF*. The* R(k − 1)F *Frontier of* F*, denoted* $\mathcal{F}_F$*, is the set of the topmost gates that compute an* $R(k-1)P$ *polynomial. That is, there is no other gate in the path from these gates to the root that computes an* $R(k-1)P$ *polynomial. Whenever* F *is clear from context, we write* $\mathcal{F}$ *instead of* $\mathcal{F}_F$*.* ⌟

▶ **Definition 12** ($\mathcal{F}^{(2)}$)**.** *Let* F *in* RkF*. Denote by* $\mathcal{F}_F^{(2)}$ *the set of all the gates in* F *which have two children in* $\mathcal{F}_F$*. Whenever* F *is clear from context, we write* $\mathcal{F}^{(2)}$ *instead of* $\mathcal{F}_F^{(2)}$*.* ⌟

We denote, for $d \in \mathbb{N}$

$$\partial_{x_i^d}(p) \coloneqq \frac{\partial^d p}{\partial x_i^d} \quad \text{and} \quad \partial_{x_i^{\deg}}(p) \coloneqq \partial_{x_i^{\deg_{x_i}(p)}}(p)$$

to mean the derivative of p taken with respect to x_i as many times as the degree of x_i in p.

Additionally, we define the notion of **stable variables**:

▶ **Definition 13** (Stable set of a gate in an RkF, $U^{(s)}(v)$)**.** *Let* $F \in \mathsf{RkF}$ *and let* $v \in F$*. We denote by* $Unv_F(v)$ *the set of unvisited children of multiplication gates along the path from* v *to the root of* F*. The* stable set *of* v*, denoted* $U_F^{(s)}(v) \subseteq \mathbf{x}$*, is the set of variables* $x_i \in var(v)$ *such that*

$$\partial_{x_i^{\deg}} p_F = \partial_{x_i^{\deg}} p_v \cdot f \quad \text{or} \quad \partial_{x_i^{\deg}} p_F = f\,,$$

where $f = \prod_{i=1}^{m} f_i$ *is an RkP and the* f_i*s are computed by distinct subformulas of gates in* $Unv_F(v)$*. When* F *is clear from context, we write* $U^{(s)}(v)$*.* ⌟

Differentiating with respect to a stable variable of a gate guarantees that the resulting structure above that gate is simplified.

These tools are combined and applied in our **Fragmentation Lemma**, which provides a method for simplifying the structure of a general RkF via differentiation. This lemma generalizes previous variants from earlier work and, in comparison, requires fewer partial derivatives to achieve the desired simplification.

Moreover, we introduce the concept of **dominating degree patterns**.

▶ **Definition 14** (Dominating Degree Patterns, Informal). *A degree pattern is a partial exponent vector $\mathfrak{e} \in \mathbb{N}^J$ defined over a domain $J \subseteq [n]$. We say $\mathfrak{e}$ is* dominating *in a polynomial f if f contains a monomial whose exponent vector coincides with $\mathfrak{e}$ on J, while any monomial in f with a strictly higher degree in one variable of J must necessarily have a strictly lower degree in another.*

We then demonstrate that taking partial derivatives with respect to such dominating degree patterns preserves the read-count of all other variables, provided each variable in the pattern is read at most twice and that $|J| \leq 3$.

▶ **Lemma 15.** *Let $S \subseteq [n]$ be of size $|S| \leq 3$. Let $F \in$ RkF be such that for every $i \in S$, $Read_F(x_i) \leq 2$. Let $\mathfrak{e} \in \{1,2\}^S$ be a dominating degree pattern of F with domain S. Then, $\partial_{\mathfrak{e}} F \in$ RkF, and moreover it is composed entirely of disjoint subformulas of F.*

1.2 Comparison to previous work

There are several works concerning bounded-read formulas with more than one read per variable.

Shpilka and Volkovich [50] studied sums of k ROFs. They gave an $n^{O(k)}$ whitebox algorithm and a hitting set of size $n^{O(\log n+k)}$. Their result is actually more flexible and allows to substitute univariable polynomials $f_i(x_i)$ for the x_i-s. This work also introduced the SV-generator and the hardness of representation technique (HoR for short). In a beautiful work, Minahan and Volkovich proved that a 1-independent map hits ROFs. Using that they obtained an $n^{O(k)}$ blackbox PIT algorithm for sum of k ROFs.

We note that a sum of k ROFs is naturally a *multilinear read-k formula*, but of a very restricted structure. Anderson, van Melkebeek and Volkovich presented a polynomial-time whitebox algorithm and a quasi-polynomial-size hitting set for the general model of multilinear read-k formulas. We later explain what the difficulty is in extending their techniques to the non-multilinear case.

Going beyond multilinearity, Mahajan, Rao, and Sreenivasaiah [34] studied read-2 and read-3 formulas, without multilinearity restrictions, and gave polynomial-time whitebox PIT algorithms for them. In his Ph.D. thesis, Prakriya [39] gave quasi-polynomial-size hitting sets for read-2 and read-3 formulas. A central component of his approach is a reduction from blackbox PIT for general RkFs to blackbox PIT for $\sum^4 \mathsf{R(k-1)F}$. This tool serves as an alternative to our fragmentation lemma. The main difference is the target model, while Prakriya's reduction targets $\sum^4 \mathsf{R(k-1)F}$, our lemma reduces to certain derivatives of $\sum^2 \mathsf{R(k-1)F}$. Furthermore, Prakriya's also introduced a generalization to the "shattering" lemma from [6]. Independently, Shamir, in his M.Sc. thesis [47], obtained quasi-polynomial-size hitting sets for read-2 formulas.

A different multi-read restricted model was considered by Mahajan, Rao and Sreenivasaiah [33]. Specifically, they considered what we term as $\sum^2 \prod \mathsf{ROF}$. That is, a sum of two terms, each is a product of ROFs. For this model they gave a whitebox PIT algorithm.

We note that while this model can allow unbounded number of reads, it is a much easier model, from the PIT perspective. Indeed, in the whitebox model it is quite simple to find irreducible factors of ROFs and then all we have to do is normalize each factor and compare multiplicities and evaluation at a nonzero point common to all the ROFs.

Bisht, Gupta and Volkovich [9] gave a polynomial-time blackbox PIT algorithm for $\sum^2 \prod \mathsf{ROF}$. In the same work, they also developed polynomial-time and quasi-polynomial-time blackbox algorithms for the classes of sums of three powers of ROFs and sums of three powers of multilinear RkFs, respectively. Thus, these models allow more than 2 reads, and they don't compute multilinear polynomials, yet there is a very strong restriction on their structure.

To summarize, prior work either dealt with multilinear bounded-read formulas, or bounded-read formulas with a very restricted structure. In particular, no whitebox or blackbox PIT was known for sums of two R2Fs, which is a very restricted form of R4Fs.

Algebraic branching programs (ABPs) are layered graphs with a source and sink nodes, in which variables label edges that go between layers and the polynomial computed by the ABP is the sum over all source-sink paths of the product of the labels of the edges in the path. ABPs form an important model of computation that is (believed to be) stronger than formulas and weaker than circuits. In this model, a read-once ABPs is an ABP in which each variable can appear on edges between two specific adjacent layers. This model is considerably stronger than ROFs as an ROF can be computed by an ABP in which a variable labels only a single edge.

Raz and Shpilka [40] provided a polynomial time whitebox PIT algorithm for ROABPs. This immediately implies such an algorithm for read-once formulas (ROFs), although in the case of formulas the problem is much easier. In the blackbox setting, Forbes and Shpilka [16] gave the first quasi-polynomial time blackbox algorithm for ROABPs. This started a long line of work on read-once models of ABPs – the usual ROABP model, commutative-ROABP, any-order ROABPs, bounded-width ROABPs, sums of ROABPs etc. See [13] for a recent survey on PIT that contains many of the state of the art results on ROABPs.

We note that according to the basic definition, ROABPs compute multilinear polynomials. However, similar to [50], one can allow univariate polynomials to label edges instead of just variables. This results in non-multilinear computations, that are still as structured as ROABPs.

ABPs that read each variable more than once (i.e., in more than one layer), were first considered by [22] who studied PIT for *sum* of ROABPs. They gave a a whitebox PIT algorithm for the sum of k ROABPs that runs in time $w^{O(2^k)} \cdot (nd)^{O(k)}$, where d is an upper bound on the univariates degree, k is the number of summands, and w is the maximal width (i.e. size of layer) of any of the ROABPs in the sum. In the blackbox setting they gave a hitting set of size $(ndw)^{k2^k \log(ndw)}$ for the model. This model allows reading variables in different layers, and allows non-multilinearity by enabling univariates on edges, yet its underlying structure still comes form that of a multilinear sum of ROABPs. In that respect, this model can be seen as the ABP analog of the sum-of-ROFs results of [50, 36] .

Other restricted models allowing multiple reads considered bounded depth formulas. There is a vast literature on PIT for bounded depth models, but as we are concerned with formulas without any depth restriction we refer the readers to the surveys [43, 51, 44, 13] as well as to Forbes' thesis [19].

Finally, the aforementioned work of [5] considered unrestricted read-k ABPs, which is a considerably stronger model than read-k formulas. They constructed a gray-box PIT algorithm running in time $2^{\tilde{O}(n^{1-1/2^{k-1}})}$ for this model. This is a stronger model than what

we consider, but we achieve a polynomial time whitebox algorithm (compared to theirs slightly sub-exponential algorithm) and a quasi-polynomial-size hitting set, whereas their algorithm needs to know the order of the reads so it does not work in a blackbox model, and of course, it is also slightly sub-exponential, compared to our quasi-polynomial-size hitting set.

To conclude, besides the slightly sub-exponential PIT algorithm of [5], PIT algorithms for multi-read ABPs dealt with substitutions of univariates to models based on multilinear polynomials.

Another line of research, motivated by the fact that simple polynomials, when composed with full-rank affine transformations, form dense subsets in broader and more interesting classes. This line of work started with the work of Bringmann, Ikenmeyer and Zuiddam [10] who proved that orbits of width-2 ROABPs are dense in the class of poly-size formulas. Subsequent work, considered orbits of classes such as ROFs [35], sparse polynomials [35, 41] and ROABPs [41, 8]. Of particular relevance is the work by Medini and Shpilka [35] who proved that orbits of ROFs are dense within poly-size arithmetic formulas. They also gave hitting sets and reconstruction algorithms for polynomials in such orbits. As Forbes and Shpilka [18] noted, if these hitting sets could be made *robust*[4] then they would imply hitting set for *general arithmetic formulas*. Thus, exploring hitting sets for such simple classes is an extremely interesting question, with the goal being developing that would eventually lead to robust hitting sets. We note that in the aforementioned work [6], the authors allowed variables to be replaced with arbitrary sparse polynomial (of polynomial degree), but required that the children of every multiplication node in the substituted multilinear read-k formula are variable disjoint. Thus, this allows more flexibility in the sense that more general polynomials than linear functions are being substituted, however there is the strong restriction of remaining multilinear. In contrast, when we compose, even multilinear read-k formulas with a linear transformation, the result is unlikely to remain multilinear.

2 Proof overview and techniques

As mentioned earlier, the main difficulty in analyzing read-4 formulas is that variables cease to behave *structurally* (recall Definition 1). For read-3 or fewer read formulas, every variable occurs in a "consistent" way: if a node v depends on a variable x and u is a descendant of v, then the degree of x in the subformula at u cannot exceed its degree at v. For read-4 formulas this is no longer the case, and this loss of structural behavior is the source of most of the technical difficulties we face.

One reason this loss of structure is problematic is that previous algorithms crucially relied on taking partial derivatives of the formula. In the multilinear setting, differentiating with respect to a variable only simplifies the formula, while not increasing the number of reads. This is captured by the *fragmentation lemma* of [6]. However, once variables can appear with higher degrees, differentiation may destroy the formula's bounded-read property. For example, if a subformula contains terms such as $(Ax + B)(Cx + D)$ and the quadratic term ACx^2 cancels out, then the formula depends on x with degree 2 at some gates and degree 1 at some of their parents. Differentiating twice with respect to x yields zero, while differentiating once produces a term like $AD + BC$, which may involve many more reads than the original subformula. Thus, the derivative can increase the read count beyond 4, breaking the inductive structure that underlies previous analyses.

[4] We say that a hitting set $\mathcal{H}$ is *robust* for a class $\mathcal{C}$ if there exists a constant $c > 0$ and a fixed norm $\|\cdot\|$ on $\mathbb{C}[\boldsymbol{x}]$ such that for every $f \in \mathcal{C}$ there exists $\boldsymbol{\alpha} \in \mathcal{H}$ satisfying $|f(\boldsymbol{\alpha})| \geq c \cdot \|f\|$.

It is not surprising that cancellations lie at the heart of the difficulty, since they are precisely what make PIT challenging in general. Many efficient computations for example, the Determinant polynomial, rely on intricate patterns of cancellation that obscure the underlying algebraic structure.

At a high level, and omitting many technical details, our approach is to isolate and control these cancellations. We first show that if a formula has only few structural variables, then we can reduce to the extreme case in which *no* variable is structural. Furthermore, we can assume that all cancellations occur at the top gate. This reduction yields formulas of the form $\sum^2 \mathsf{R2F}$, where the resulting polynomial is multilinear even though each R2F has degree 2 in every variable. We refer to such formulas as *totally non-structural*. Analyzing the patterns of cancellation that arise in this setting constitutes the main technical component of our work.

We shall now expand more on the ideas that we use to handle these obstacles. Due to space constraints, we provide only a very high-level overview here; full details appear in the full version [27].

The first tool that we need is a *fragmentation lemma*. Fragmentation lemmas, as termed in [6], aim to reduce the problem of hitting the class RkF to that of hitting the class $\sum^2 \mathsf{R(k-1)F}$. Such lemmas were instrumental to the works of [6] and [47]. In our case, by concentrating on $\mathcal{F}^{(2)}$ (recall Definition 12), i.e., gates whose children are read-3 but they themselves are read-4, we prove that by taking an appropriate partial derivative with respect to a stable variable of one of these first read-4 gates, simplifies the formula, thus achieving *fragmentation*.

▶ **Lemma 16** (Fragmentation Lemma). *Let F be a nonzero* RkF *and set $t \coloneqq |\mathcal{F}_F^{(2)}|$. Write $\mathcal{F}_F^{(2)} = \{v_j\}_{j=1}^t$, and let $I = \{i_j\}_{j=1}^t$ with $x_{i_j} \in U^{(s)}(v_j)$. Denote*

$$\mathcal{D}_I \coloneqq \left\{ \partial_{x_{i_j}^{\deg}} p_{v_j} : j = 1, \ldots, t \right\}.$$

If $\mathcal{H}$ is a hitting set for $\mathcal{D}_I \cup \mathsf{R(k-1)F}$, then[5]

$$p_F \circ (\mathcal{H} + \mathcal{G}_{\log t+1}) \not\equiv 0.$$

Applying this procedure to R2Fs reduces the formula to a $\sum^2$ ROFs, which is hit by $\mathcal{G}_3$, which is how we obtain Theorem 5. In a similar vain, to prove Theorem 4, we again use our fragmentation lemma to reduce the PIT problem for R3Fs to PIT for an R3Fs of the form $\sum^2 \mathsf{R2F}$. If the sum is multilinear, then by the following claim it is structurally multilinear, and the result of [6] applies directly.

▷ Claim 17. Let $F \in \mathsf{R3F}$. Then, there exists a structural $F' \in \mathsf{RkF}$ such that $p_F = p_{F'}$ and for every $x_i \in \boldsymbol{x} : \mathrm{Read}_{x_i}(F') \leq \mathrm{Read}_{x_i}(F)$.

Otherwise, some variable has degree 2, implying that its degrees in the two summands are unbalanced. In this case, taking the second-order partial derivative with respect to that variable reduces the problem to hitting a nonzero R2F.

2.1 Proof Overview of the Blackbox Algorithm (Theorem 2)

The following sketch (Figure 1) outlines the structure of the proof and the dependencies among its key components.

[5] Whenever we write a sum of generators, we mean they are defined over disjoint sets of variables.

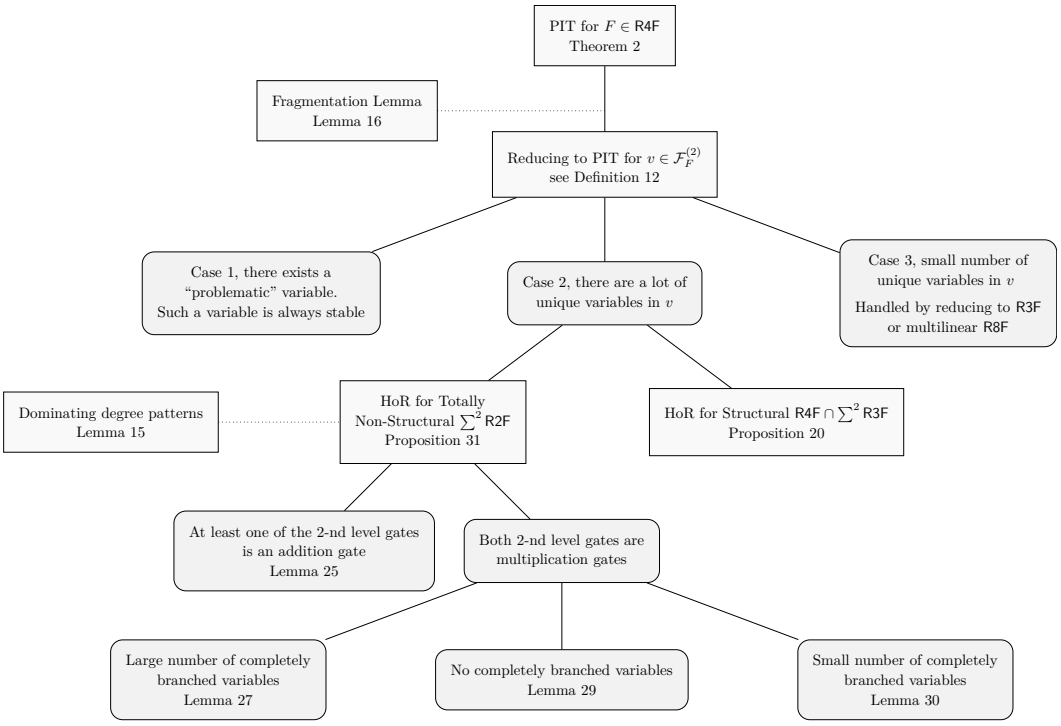


Figure 1 Structure of the proof of Theorem 2.

The fragmentation procedure discussed above allows us to focus our attention on a gate that is read-4 but that both its children are read-3.

If v is a multiplication gate then a hitting set for read-3 formulas hits p_v. When v is an addition gate we use hardness of representation together with restrictions by a generic assignment to obtain a hitting set. In the following part we give a short description of these techniques and how they are used together.

2.1.1 Hardness of Representation, Restrictions and Preserving Assignments

An idea that considerably simplifies the analysis is the use of *generic assignments* to a subset of the variables. A generic assignment is an assignment that does not alter any algebraic property[6] of the polynomials under consideration.

In the *hardness of representation* technique, we need to prove that a formula of a certain structure, cannot compute any polynomial divisible by the monomial $\mathcal{P}_n = \prod_{i=1}^{n} x_i$. Once such a result is proved, there is a way to use it to construct a hitting set. This technique has been successfully employed in several prior works on bounded read formulas [50, 6, 9, 39].

In general, one may observe that the monomial $\mathcal{P}_n$ can be computed even by a very restricted class of polynomials, such as the class of ROFs. Thus, hardness of representation for a class of polynomials $\mathcal{C}$ is obtained by identifying a collection of algebraic properties associated with polynomials in the class. More precisely, for every $p \in \mathcal{C}$, these properties are captured by a finite set of polynomials related to p which are simpler than p, for example, subformulas of F_p. The goal is to ensure that if $p \in \mathcal{C}$ is translated by a common nonzero of this finite set of polynomials, then the resulting polynomial is not divisible by $\mathcal{P}_n$.

[6] An algebraic property is one that can be expressed by a finite set of polynomial equalities.

For example, [48] showed that in order to prove hardness of representation for sums of ROFs, it suffices to translate by a common nonzero $\boldsymbol{\sigma}$ of all the partial derivatives of the individual ROFs. This condition is captured by the algebraic property of being **0**-justified, which ensures that no variable on which some term in the sum depends disappears after another variable is set to 0. The translation $\boldsymbol{\sigma}$ is then called a *justifying assignment*.

We call assignments that preserve a desired property of the polynomial under consideration *preserving assignments*. In this work, we use two additional types of preserving assignments. The first type, introduced in [9], is called an *irreducibility-preserving* assignment. A polynomial is **0**-irreducible if it is irreducible and remains irreducible after setting any strict subset of its variables to 0.

In this work, we introduce the notion of *degree-preserving* assignments. Such assignments preserve the degrees of the remaining variables in the polynomial for which they are constructed. Thus, they may be viewed as a stronger form of justifying assignments. A polynomial translated by such an assignment has its individual degrees preserved under substitutions of variables by 0, and we call such a polynomial **0**-preserved.

In the following subsections we define the sets of polynomials that are used to describe these preservation properties. In order to prove hardness of representation for $\mathsf{R4F} \cap \sum^2 \mathsf{R3F}$ we will first translate the polynomial by a common nonzero of certain polynomials from these sets.

In our analysis we combine this approach with generic assignments. Given a generic assignment we can *restrict* a formula to a subset of its variables, by substituting values to the other variables according to the assignment and simplifying the resulting formula. Observe that if the monomial $\mathcal{P}_n$ divides the original polynomial, then $\mathcal{P}_S = \prod_{i \in S} x_i$ must divide the restricted one. Hence, it is enough to prove hardness of representation results for the restricted polynomial.

The main advantage of restrictions is that it greatly simplifies the formula and enables us to reduce the analysis to arguing about polynomials in few variables.[7]

For example, to restrict our attention to p_v we may restrict the variables not appearing in its subtree.

We stress that the generic assignment is only used for the analysis, and not for the construction of the hitting set.

The difficult case, as explained, is when there are many non-structural variables, but let us first explain what we do in the mostly structural case.

2.1.2 The structural case

In the structural case, we use two sets of polynomials in order to find a translation that yields hardness of representation. The first set was introduced in [6] and is used to prove hardness of representation for structurally multilinear RkFs. Throughout, when we write $w \in F$ for a formula F, we view F as the set of its gates.

▶ **Definition 18** ($\mathcal{Q}_{F_1,F_2,\ldots,F_m}$)**.** *Let $F_1, F_2, \ldots, F_m$, be algebraic formulas such that F_i is an $\mathsf{Rk_iF}$ and let $k = \sum_{i\in[m]} k_i$. Define*

$$b_{m,k} := (k - m + 1) \cdot 4k \cdot (k + 1)^2 .$$

[7] For us, "few" can be 5^8 many variables, but in most cases we will eventually reduce to polynomials on at most 4 variables.

Then we define

$$\mathcal{Q}_{F_1,F_2,\ldots,F_m} \coloneqq \left\{ \partial_{Y^{\deg}}(p_w)\big|_{Z\leftarrow \mathbf{0}} \;\middle|\; w \in \bigcup_{i\in[m]} F_i,\ Z\subseteq[n],\ Y\subseteq[n]\setminus Z,\ |Z\cup Y|\le b_{m,k} \right\}.$$

In words, F_w is a subformula of some F_i, and $Z, Y \subseteq [n]$ are disjoint subsets satisfying $|Z \cup Y| \le b_{m,k}$. ⌟

The second set is defined as follows.

▶ **Definition 19.** *Let F, G be algebraic formulas. Define*

$$\mathcal{T} \coloneqq \left\{ \partial_{x_i^{\deg}}\left(\partial_{x_j^{\deg}} p\right) \;\middle|\; p \in \{p_F, p_G\},\ x_i, x_j \in var(p) \right\}.^{8}$$

If we translate $p \in \mathsf{R4F} \cap \sum^2 \mathsf{R3F}$ by a common nonzero of the two sets above, then hardness of representation is easy to prove in each of the following two cases:

1. "Many" variables appear with degree 1 in p.
2. The degrees in the two summands are unbalanced.

Thus, by removing the "few" variables with degree 1 in p using a generic assignment, we are left with the case where $p \in \sum^2 \mathsf{R2F}$ and every variable has individual degree 2 in both summands. To handle this case, we introduce the notion of *splitness* and show that it implies hardness of representation.

Splitting a polynomial $h = f + g$ with respect to variables x_i and x_j corresponds to transforming h into a polynomial of the form $\widetilde{h} = f_i f_j + g_i g_j$, where f_i and g_i do not depend on x_j, and f_j and g_j do not depend on x_i. We achieve this by taking a partial derivative with respect to a carefully chosen variable x_t.

More precisely, the three variables i, j, t depend on each other, and on the formula computing h, and we find them using combinatorial analysis of labeled trees, relying on the well-known Erdős–Szekeres theorem.

We then prove the main point which is that a split polynomial cannot be divisible by a high degree monomial. Going back to h we obtain the same result and hence conclude hardness of representation in the structural case.

▶ **Proposition 20.** *Let F, G be $\mathsf{R3F}$s such that the formula $F + G$ is a structural $\mathsf{R4F}$. Let U be a set containing all the read-4 variables in the formula $F + G$.*

Assume $U \neq \emptyset$ and that no nonzero polynomial in $\mathcal{Q}_{F,G} \cup \mathcal{T}$ vanishes at $\mathbf{0}$.

Then, if $n \ge r_{2,6} + 5^8$, there exists $t \in U$ such that $\mathcal{P}_{[n]\setminus\{t\}} \nmid \partial_{x_t^{\deg}}(F + G)$.

Let us now switch gears and consider the non-structural case.

2.1.3 Totally non-structural case

First, observe that a totally non-structural polynomial in $\mathsf{R4F} \cap \sum^2 \mathsf{R3F}$ must have the structure of a $\sum^2 \mathsf{R2F}$. Indeed, each of the two summands has degree 2 in every variable, yet all such degree-2 monomials cancel. In particular, the resulting polynomial is multilinear. In this part, we use this special structure to prove hardness of representation.

---

[8] Observe that $\partial_{x_i^{\deg}}\left(\partial_{x_j^{\deg}} p\right) \neq \partial_{x_i^{\deg} x_j^{\deg}} p$. On the left-hand side, we differentiate with respect to x_i as many times as its degree in $\partial_{x_j^{\deg}} p$, whereas on the right-hand side, we differentiate as many times as its degree in p.

We now describe the sets of polynomials that we use in the totally non-structural case to obtain hardness of representation. We denote the root of F by o, its left child by $(o)_{\mathrm{L}}$, and its right child by $(o)_{\mathrm{R}}$. We also use the following notation, recalling Definition 14:

$$\partial_{I^{\mathrm{dom}}} p_F = \left\{ \partial_{\mathfrak{c}} p_F \;\middle|\; \begin{array}{c} \mathfrak{c} \text{ is a dominating degree} \\ \text{pattern in } p_F \text{ with domain } I \end{array} \right\} .$$

▶ **Definition 21** ($\mathcal{A}_F$)**.** *Let F be an algebraic formula and $t \in \mathbb{N}$. Define,*

$$\mathcal{A}_F^t \coloneqq \bigcup_{I \in \binom{[n]}{\leq t}} \partial_{I^{dom}} p_F \ ,$$

In words, this is the set obtained by taking all possible derivatives according to all dominating degree patterns of size at most t. ⌟

▶ **Definition 22** ($\mathcal{E}_F^m$)**.** *Let F be an algebraic formula and let $m \in \mathbb{N}$. Define:*

$$\mathcal{E}_F^m \coloneqq \bigcup_{u \in F_{(o)_L} \cup F_{(o)_R}} \mathcal{A}_{F_u}^m .$$ ⌟

The utility of these definitions is captured in the next claim.

▷ Claim 23. Let F be an algebraic formula. If no polynomial in $\mathcal{E}_F^{m+1}$ vanishes at $\mathbf{0}$, then, for every subformula of $F_{(o)_{\mathrm{L}}}$ or $F_{(o)_{\mathrm{R}}}$, every derivative corresponding to a dominating degree pattern whose domain has size at most m is $\mathbf{0}$-preserved.

Additionally, the next set is used to make the ROF subformulas of the polynomial under consideration $\mathbf{0}$-irreducible.

▶ **Definition 24** ($\mathcal{C}_F$)**.** *Let F be an algebraic formula. For any pair of gates $u, v \in F$ such that u is an ancestor of v, let $\gamma_v \in \mathbb{F}$ denote the additive constant of the gate v. Define*

$$\mathcal{C}_u \coloneqq \left\{ p_{F_u|_{v=\gamma_v}} \;\middle|\; v \in F_u \right\},$$

In words, $\mathcal{C}_u$ is obtained by replacing, for each descendant v of u, the subtree rooted at v with a leaf labeled by the scalar γ_v and then reducing the formula.

We further define

$$\mathcal{C}_F \coloneqq \bigcup_{u \in F_{(o)_L} \cup F_{(o)_R}} \mathcal{C}_u \, .$$

When the formula F is clear from context, we simply write $\mathcal{C}$. ⌟

Let $F \in \sum^2 \mathsf{R2F}$ be totally non-structural. Since each child of the root of F computes a polynomial of degree 2 in every variable, the first common gate of the two leaves labeled by any variable x_i in $F_{(o)_{\mathrm{L}}}$ must be a multiplication gate, and the same holds in $F_{(o)_{\mathrm{R}}}$. Consequently, if, say, $(o)_{\mathrm{L}}$ is an addition gate, then there must exist variables $x_i, x_j \in \boldsymbol{x}$ whose first common gate is $(o)_{\mathrm{L}}$. Taking a partial derivative with respect to these two variables then reduces the problem to PIT for R2Fs.

▶ **Lemma 25.** *Let $F \in \sum^2 \mathsf{R2F}$ be a totally non-structural formula. Suppose that $(o)_L$ is an addition gate. Moreover, suppose that no nonzero polynomial in $\mathcal{A}^2_{F_{(o)_L}} \cup \mathcal{A}^2_{F_{(o)_R}}$ vanishes at $\mathbf{0}$. Then, if $n \geq 3$, we have $\mathcal{P}_n \nmid p_F$.*

Hence, the challenging case occurs when both $(o)_{\mathrm{L}}$ and $(o)_{\mathrm{R}}$ are multiplication gates. To handle this case, we introduce the notion of *branched variables*. A variable x_i *branches* in a product gate u if each child of u computes a linear polynomial in x_i. A variable that branches in the two subformulas computed by the children of the root is called *completely branched*, while one that branches in only one child is *partially branched*.

Note that if x_1 completely branches in F then F computes a polynomial of the form $(a_1x_1 + b_1)(a_2x_1 + b_2) + (a_3x_1 + b_3)(a_4x_1 + b_4)$, where the a_is and b_js are subformulas of F.

We further divide the analysis into three cases:

1. there exists a "large" set of variables that branch completely in F;
2. every variable that branches in F branches only partially;
3. the set of completely branched variables is "small".

To handle the first case, we first strengthen, in an appropriate sense, [9, Theorem 1]. This theorem provides a PIT algorithm for the class $\sum^2 \prod \mathsf{ROF}$. However, the proof does not proceed via a hardness of representation argument, which is what we require, since we already used a generic assignment in order to reduce to the $\sum^2 \mathsf{R2F}$ case. Accordingly, we first establish such a hardness of representation result.

▶ **Lemma 26.** *Let $F_1, F_2 \in \prod^k \mathsf{ROF}$ such that $F_i = \prod_{j=1}^{m_i} f_{i,j}$, where all $f_{i,j}$ (for $i \in [2]$, $j \in [m_i]$) are irreducible. Moreover, assume these irreducible factors are $\mathbf{0}$-irreducible and $\mathbf{0}$-preserved. Then, if $n \geq 2k + 1$, we have $\mathcal{P}_n \nmid p_{F_1} + p_{F_1}$.*

Now assume there exists a set $S \subseteq \boldsymbol{x}$ such that every variable in S branches completely in F. Observe that if we further restrict to the variables in S then the formula simplifies further and becomes a $\sum^2 \prod^2 \mathsf{ROF}$. Therefore, if S is large enough to apply the hardness of representation result from Lemma 26, the proof is complete. This is indeed the case whenever $|S| \geq 5$.

▶ **Lemma 27.** *Let $F \in \sum^2 \mathsf{R2F}$ be totally non-structural. Assume that both $(o)_L$ and $(o)_R$ are multiplication gates. Moreover, suppose there exists a set of variables S with $|S| \geq 5$ such that every $x_i \in S$ branches completely in F.*

If no nonzero polynomial $g \in \mathcal{C} \cup \mathcal{E}^2$ vanishes at $\mathbf{0}$, then $\mathcal{P}_n \nmid p_F$.

The second case is more delicate and relies on exploiting the structural constraints imposed by the definition of partial branching. The high level idea is that we first prove that if hardness of representation does not hold, i.e., that a high degree monomial divides p_F, then some variable must branch in one of the children of the root.

▶ **Lemma 28.** *Let $F \in \sum^2 \mathsf{R2F}$ be a totally non-structural formula, and suppose that both $(o)_L$ and $(o)_R$ are multiplication gates. If, for some $\boldsymbol{\alpha} \in \mathbb{F}^n$, we have $p_F(\boldsymbol{x} + \boldsymbol{\alpha}) \sim \mathcal{P}_n$, then there exists a variable $x_i \in \boldsymbol{x}$ that branches in F.*

Assume then that x_1 branches in $(o)_{\mathrm{L}}$. Since it does not completely branch, this means that in $(o)_{\mathrm{R}}$ it appears in exactly one child (i.e., a grandchild of the root). We exploit this structure to eventually prove that p_F cannot be divisible by high degree monomial.

▶ **Lemma 29.** *Let $F \in \sum^2 \mathsf{R2F}$ be a totally non-structural formula. Suppose that both $(o)_L$ and $(o)_R$ are multiplication gates, and that no variable branches completely in F. Moreover, Suppose that no polynomial in $\mathcal{A}^3_{F_{(o)_L}} \cup \mathcal{A}^3_{F_{(o)_R}}$ vanishes at $\mathbf{0}$. Then, if $n \geq 5$, we have $\mathcal{P}_n \nmid p_F$.*

In the third case, we present a procedure that attempts to reduce the setting to one of the two previous cases by applying generic assignments, to get rid of the few variables that completely branch. Note, however, that once we assign a value to a variable, we change the

structure of the formula, and new completely branched variables may appear. We show that if this is indeed the case, and it repeats itself through several iterations of fixing completely branched variables, then this outs a very rigid structure on the formula, which allows us to prove that the resulting polynomial cannot be divisible by $\mathcal{P}_n$.

▶ **Lemma 30.** *Let $F \in \sum^2 \mathsf{R2F}$ be a totally non-structural formula. Suppose that both $(o)_L$ and $(o)_R$ are multiplication gates. Let S_1 be the set of completely branched variables in F, and assume that $1 \leq |S_1| \leq 4$. Furthermore, assume that no nonzero polynomial in $\mathcal{C} \cup \mathcal{E}^3$ vanishes at $\mathbf{0}$.*

Then, if $n \geq 23$, we have $\mathcal{P}_n \nmid p_F$.

Combining all these results we obtain our hardness of representation result for the totally non-structural case.

▶ **Proposition 31.** *Let $F \in \sum^2 \mathsf{R2F}$ be totally non-structural. Suppose that no nonzero polynomial in $\mathcal{C} \cup \mathcal{E}^3$ vanishes at $\mathbf{0}$. Then, if $n \geq 23$, we have $\mathcal{P}_n \nmid p_F$.*

2.2 Proof overview of our whitebox algorithm (Theorem 3)

We follow the high-level approach of the whitebox algorithm introduced in [6]. Let S_4 denote the set of read-4 variables in a formula $F \in \mathsf{R4F}$. Our algorithm proceeds iteratively: in each iteration, it removes at least one variable from S_4 without introducing any new variables into it, while preserving nonzeroness of F. After at most n iterations, the resulting formula is an $\mathsf{R3F}$. We then verify its identity using the whitebox algorithm for $\mathsf{R3F}$s from [34].

To achieve this, we follow the cases handled in the proof of Theorem 2, and remove variables by following its arguments.

2.3 Proof overview of our orbit results

All of our orbit results follow from the simple observation that hardness of representation results for polynomial classes that are closed under translations and have low individual degree imply PIT not only for the backbone class[9], but also for the orbit class itself.

The following is an equivalent definition of hardness of representation, this is the notion that we primarily use in the full version.

▶ **Definition 32** (Hardness of Representation)**.** *A polynomial class $\mathcal{C} \subseteq \mathbb{F}[\boldsymbol{x}]$ is said to be m-*hard *if it is closed under $\mathbf{0}$-substitutions and no polynomial $p \in \mathcal{C}$ is divisible by a monomial with support size at least m.* ⌟

Assume we are concerned with an m-hard polynomial class $\mathcal{C}$. Then every polynomial $f(\boldsymbol{y}) \in \mathcal{C}$ contains a monomial whose support size is at most $m-1$. Since the action of every nonzero matrix $\boldsymbol{A} \in \mathrm{GL}_n(\mathbb{F})$ on $\mathbb{F}[\boldsymbol{y}]$ induces a degree-preserving isomorphism, if f has low degree, then $f(\boldsymbol{A}\boldsymbol{x})$ must contain a monomial of small support. We extend this observation to the action of $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ by first translating our polynomials by an appropriate vector of scalars.

This observation immediately lifts a large number of known results for backbone classes to their corresponding orbit classes.

[9] $\mathcal{C}$ denotes the backbone class corresponding to the orbit class $\mathcal{C}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$.

References

1. Manindra Agrawal. Proving lower bounds via pseudo-random generators. In Sundar Sarukkai and Sandeep Sen, editors, *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science*, pages 92–105, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/11590156_6.
2. Manindra Agrawal and Somenath Biswas. Primality and identity testing via chinese remaindering. *J. ACM*, 50(4):429–443, 2003. doi:10.1145/792538.792540.
3. Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Annals of Mathematics*, 160(2):781–793, September 2004. doi:10.4007/annals.2004.160.781.
4. Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, October 25-28, 2008*, pages 67–75. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.32.
5. Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. *ACM Trans. Comput. Theory*, 10(1):3:1–3:30, 2018. doi:10.1145/3170709.
6. Matthew Anderson, Dieter van Melkebeek, and Ilya Volkovich. Deterministic polynomial identity tests for multilinear bounded-read formulae. *Computational Complexity*, 24(4):695–776, December 2015. doi:10.1007/s00037-015-0097-4.
7. Robert Andrews. Algebraic hardness versus randomness in low characteristic. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 37:1–37:32. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.37.
8. Vishwas Bhargava and Sumanta Ghosh. Improved hitting set for orbit of roabps. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*, pages 30–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.30.
9. Pranav Bisht, Nikhil Gupta, and Ilya Volkovich. Towards Identity Testing for Sums of Products of Read-Once and Multilinear Bounded-Read Formulae. In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023)*, volume 284 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:23, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2023.9.
10. Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On algebraic branching programs of small width. *J. ACM*, 65(5), 2018. doi:10.1145/3209663.
11. Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure results for polynomial factorization. *Theory of Computing*, 15(1):1–34, 2019. doi:10.4086/TOC.2019.V015A013.
12. Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, June 1978. doi:10.1016/0020-0190(78)90067-4.
13. Pranjal Dutta and Sumanta Ghosh. Sigact news complexity theory column 121. *SIGACT News*, 55(2):53–88, 2024. doi:10.1145/3674159.3674165.
14. Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM Journal on Computing*, 39(4):1279–1293, 2010. doi:10.1137/080735850.
15. Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Communications of the ACM*, 62(3):109–115, 2019. doi:10.1145/3306208.
16. Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 163–172. ACM, 2012. doi:10.1145/2213977.2213995.

17 Michael A Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 527–542. Springer, 2013. doi:10.1007/978-3-642-40328-6_37.

18 Michael A. Forbes and Amir Shpilka. A pspace construction of a hitting set for the closure of small algebraic circuits. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 1180–1192, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3188745.3188792.

19 Michael Andrew Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs*. PhD thesis, Massachusetts Institute of Technology, 2014.

20 Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from Algebraic Hardness. *SIAM J. Comput.*, 51(2):315–335, 2022. doi:10.1137/20M1347395.

21 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic Circuits: A Chasm at Depth Three . In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 578–587, Los Alamitos, CA, USA, October 2013. IEEE Computer Society. doi:10.1109/FOCS.2013.68.

22 Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Comput. Complex.*, 26(4):835–880, 2017. doi:10.1007/S00037-016-0141-Z.

23 Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-nc. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 821–830, 2017. doi:10.1145/3055399.3055440.

24 J. Heintz and C. P. Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC '80, pages 262–272, New York, NY, USA, 1980. Association for Computing Machinery. doi:10.1145/800141.804674.

25 Ivan Hu, Dieter van Melkebeek, and Andrew Morgan. Polynomial identity testing via evaluation of rational functions. *Theory of Computing*, 20(1):1–70, 2024. doi:10.4086/toc.2024.v020a001.

26 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *computational complexity*, 13(1):1–46, December 2004. doi:10.1007/s00037-004-0182-6.

27 Nimrod Kaplan and Amir Shpilka. Polynomial identity testing for read-4 arithmetic formulas. *Electronic Colloquium on Computational Complexity (ECCC)*, page 95, 2026. URL: https://eccc.weizmann.ac.il/report/2026/076/.

28 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. doi:10.1016/J.TCS.2012.03.041.

29 Mrinal Kumar and Ramprasad Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bulletin of EATCS*, 3(129), 2019.

30 Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-Optimal Bootstrapping of Hitting Sets for Algebraic Models. *Theory Comput.*, 19:1–30, 2023. doi:10.4086/TOC.2023.V019A012.

31 László Lovász. On determinants, matchings and random algorithms. *Fundamentals of Computation Theory*, 79:565–574, January 1979.

32 Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. doi:10.1145/146585.146605.

33 Meena Mahajan, B. V. Raghavendra Rao, and Karteek Sreenivasaiah. Building above read-once polynomials: Identity testing and hardness of representation. *Algorithmica*, 76(4):890–909, December 2016. doi:10.1007/s00453-015-0101-z.

34 Meena Mahajan, B.V. Raghavendra Rao, and Karteek Sreenivasaiah. Monomials, multilinearity and identity testing in simple read-restricted circuits. *Theoretical Computer Science*, 524:90–102, 2014. doi:10.1016/j.tcs.2014.01.005.

35 Dori Medini and Amir Shpilka. Hitting sets and reconstruction for dense orbits in $\mathrm{VP_e}$ and $\Sigma\Pi\Sigma$ circuits. *CoRR*, abs/2102.05632, 2021. arXiv:2102.05632.

36 Daniel Minahan and Ilya Volkovich. Complete derandomization of identity testing and reconstruction of read-once formulas. *ACM Trans. Comput. Theory*, 10(3), May 2018. doi:10.1145/3196836.

37 Ketan Mulmuley. Geometric complexity theory v: Efficient algorithms for noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.

38 Øystein Ore. Über höhere kongruenzen. *Norske Videnskaps-Akademi i Oslo. Forhandlinger (Proceedings of the Norwegian Academy of Science and Letters)*, 1922(12):1–8, 1922. Contains the original root bound for nonzero multivariate polynomials over finite fields, later used in polynomial identity testing (Schwartz–Zippel lemma).

39 Gautam Prakriya. Derandomizing isolation and polynomial identity testing. Doctoral thesis, 2019.

40 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.*, 14(1):1–19, 2005. doi:10.1007/S00037-005-0188-8.

41 Chandan Saha and Bhargav Thankey. Hitting sets for orbits of circuit classes and polynomial families. *ACM Trans. Comput. Theory*, 16(3), 2024. doi:10.1145/3665800.

42 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity, 2021. URL: https://github.com/dasarpmar/lowerbounds-survey.

43 Nitin Saxena. Progress on Polynomial Identity Testing. *Bull. EATCS*, 99:49–79, 2009.

44 Nitin Saxena. Progress on polynomial identity testing-II. *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, pages 131–146, 2014.

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

46 Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992. doi:10.1145/146585.146609.

47 Nadav Shamir. Polynomial identity testing for read-2 formulas and linear separation between read-3 and read-2 multilinear formulas. Master thesis, 2022.

48 Amir Shpilka and Ilya Volkovich. Improved polynomial identity testing for read-once formulas. In Irit Dinur, Klaus Jansen, Joseph Naor, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 700–713, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. doi:10.1007/978-3-642-03685-9_52.

49 Amir Shpilka and Ilya Volkovich. On reconstruction and testing of read-once formulas. *Theory of Computing*, 10(18):465–514, 2014. doi:10.4086/toc.2014.v010a018.

50 Amir Shpilka and Ilya Volkovich. Read-once polynomial identity testing. *Comput. Complex.*, 24(3):477–532, September 2015. doi:10.1007/S00037-015-0105-8.

51 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3–4):207–388, 2010. doi:10.1561/0400000039.

52 Ola Svensson and Jakub Tarnawski. The Matching Problem in General Graphs Is in Quasi-NC. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.70.

53 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. doi:10.1016/J.IC.2014.09.004.

54 Richard Zippel. *Probabilistic algorithms for sparse polynomials*, pages 216–226. Springer Berlin Heidelberg, 1979. doi:10.1007/3-540-09519-5_73.

A Simple Sub-Polynomial Degree Coboundary Expander

Max Hopkins
Institute for Advanced Study, Princeton, NJ, USA

Arka Ray
Indian Institute of Science, Bengaluru, India

Abstract

High dimensional expanders simultaneously satisfying spectral and combinatorial (coboundary) expansion have recently played a major role in breakthroughs in PCP and coding theory, but the only known construction of such complexes is extremely involved, requiring deep algebraic number theory. In this work, we give an extremely simple combinatorial construction of a sub-polynomial degree complex based on projections of the flags complex (subspace chains) that is (i) a local spectral expander, (ii) a coboundary expander, and (iii) a swap coboundary expander. As a corollary, we also give the first near-linear size combinatorial hypergraphs with good agreement tests in the '1%' regime, and a simple PCP construction with near-linear size.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases high dimensional expansders, agreement testing, probabilistically checkable proofs

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.26

Related Version *Full Version*: https://arxiv.org/abs/2605.22173

Funding *Max Hopkins*: Supported by NSF Award DMS-2424441.
Arka Ray: Supported in part by the Walmart Center for Tech Excellence at IISc (CSR Grant WMGT-23-0001).

Acknowledgements We thank Yotam Dikstein and Roy Meshulam for helpful discussions. We are grateful to the International Centre for Theoretical Sciences (ICTS) for organizing the "ICTS Workshop on HDXs and Codes" (code: ICTS/HDXandCodes2025/04) which greatly facilitated our discussions with Roy Meshulam. We thank the anonymous reviewers of CCC 2026 for their comments and suggestions.

1 Introduction

High dimensional expanders, sparse but robustly connected hypergraphs, have seen a surge of interest and application in theoretical computer science, driving breakthrough progress in coding theory [40, 25, 2, 57, 73, 42, 31, 62, 51], approximate sampling [8, 9, 6, 7, 64], property testing [35, 21, 59, 47, 22, 13, 26], hardness of approximation [32, 53, 14], and pseudorandomness [54, 28]. Unlike their widely-studied graph analogs, there are very few known constructions of high dimensional expanders, and all known methods are algebraic. In this work, we give a simple construction of sub-polynomial degree hypergraphs based on subspaces that are both spectral and combinatorial high dimensional expanders, and a corresponding simple near-linear size PCP construction.

We consider three core notions of expansion: local spectral expansion (see Definition 16), a generalization of spectral expansion in graphs, coboundary expansion (see Definition 20), a generalization of edge expansion, and swap-coboundary expansion, a key notion related to the construction of PCPs measuring the coboundary expansion of higher level disjoint faces

© Max Hopkins and Arka Ray;
licensed under Creative Commons License CC-BY 4.0

(see Definition 25). Unlike the graph setting, these generalizations behave quite differently in high dimensions, e.g., there are random two-dimensional simplicial complexes that are good local spectral expanders but not good coboundary expanders with high probability [49]. Constructing sparse hypergraphs satisfying both spectral and coboundary expansion has been a major challenge since the notions were introduced [65, 48, 58], and has received renewed attention after it was shown such hypergraphs admit low soundness "agreement tests" [47, 22, 13] and efficient PCPs [15].

The first sparse hypergraphs satisfying all three notions of expansion were constructed recently in [17, 26, 12], based on deep algebraic number theory. This was followed very recently (and independently to our work) by [71], who gave a simpler algebraic construction based on variants of the elementary coset complexes of [60]. In this work, we show that if one is willing to sacrifice somewhat on sparsity, it is possible to give completely elementary combinatorial constructions of local-spectral and (swap)-coboundary HDX of sub-polynomial degree. Following [14], we show these give rise to a simpler PCP construction roughly matching the parameters of the breakthrough algebraic PCP of [69] in the small constant soundness regime.

Our complexes are based upon projections of the *flags complex*, a well-known dense hypergraph more typically studied as the local structure of [67]'s quotients of the affine building, and shown to be a swap-coboundary expander in [24]. The vertices of flags complex are the subspaces of $\mathbb{F}_q^d$ (excepting 0 and $\mathbb{F}_q^d$). Its hyperedges are given by inclusion chains ("flags") of these subspaces.

▶ **Definition 1** (Flags Complex). *Given $d \in \mathbb{N}$ and prime power q, the flags complex is the $(d-1)$-partite complex with vertices given by all subspaces of $\mathbb{F}_q^d$, and top-level faces given by inclusion chains (flags) of subspaces:*

$$\mathfrak{G}_q^d(d-2) = \{(W_1 \subseteq W_2 \subseteq \ldots \subseteq W_{d-1}) : \dim(W_i) = i\}.$$

While the flags complex is known to be a high-dimensional expander under all three definitions [24], it has two fatal flaws: it is *dense*, with roughly $q^{d^2/2}$ top faces over only $q^{d^2/4}$ vertices (a quadratic blow-up), and has *extremely high max-degree*. In fact, both flaws stem from the same source: the number of i-dimensional subspaces of $\mathbb{F}_q^d$ varies wildly with i. In particular, since there are roughly $q^{i(d-i)}$ subspaces of dimension i, the levels are sharply imbalanced: each of the q^{d-1} lines, for instance, show up in far more hyperedges than any of the $q^{d^2/4}$ subspaces of dimension $d/2$. Measured by size and max-degree (the key parameters used in applications), the flags complex is thus, asymptotically, essentially no better than simply taking an imbalanced product $[q^{d/4}]^{d-1} \times [q^{d^2/4}]$. As such, despite its fantastic expansion, the flags complex has seen little use outside its role as the local structure of [67]'s Ramanujan complexes in the literature.

The above, however, suggests a simple fix. The imbalance in the flags complex comes from *extremal* dimensions, those near 1 and d, whereas *central* dimensions, i.e., those near $d/2$, are much closer to balanced, each containing $q^{d^2/4 \pm \Theta(d)}$ subspaces. As such, we should simply discard the problematic levels and only consider inclusion chains through the well-behaved ones.

▶ **Definition 2** (Projected Flags Complex). *Given $d \in \mathbb{N}$, a prime power q, and a subset of dimensions $S = \{s_1 < s_2 < \cdots < s_k\} \subseteq [d-1]$, the* projected flags complex $\mathfrak{G}_q^{d,S}$ *is the k-partite simplicial complex with vertices given by subspaces $W \subset \mathbb{F}_q^d$ such that $\dim(W) \in S$ (partitioned into parts by dimension), and top-level faces given by inclusion chains using one subspace of each dimension in S:*

$$\mathfrak{G}_q^{d,S}(k-1) = \{W_1 \subset W_2 \subset \cdots \subset W_k \ : \ \dim(W_i) = s_i\}.$$

Equivalently, $\mathfrak{G}_q^{d,S}$ is the complex whose top faces are obtained by projecting each flag of $\mathfrak{G}_q^d$ onto its components of dimension in S. We drop d and q from the notation when clear from context.

Taking S to be the $2k+1$ central dimensions of the flags complex over $\mathbb{F}_q^{2d}$,

$$S_k = \{d-k, \ldots, d-1, d, d+1, \ldots, d+k\},$$

every level has $q^{d^2/4 \pm \Theta(kd)}$ of subspaces, eliminating the imbalance up to lower order factors. Fixing k and letting $d \to \infty$, we get a strongly explicit infinite family of complexes with $q^{\Theta(d^2)}$ vertices and max-degree $q^{O(kd)} \approx 2^{O(k\sqrt{\log N})}$ – sub-polynomial in the number of vertices N. The main technical content of this work is to show this projection retains the spectral and (swap)-coboundary expansion of the full complex, yielding simple near-linear-size low-soundness PCPs.

1.1 Our Results

Local Spectral Expansion. A k-dimensional simplicial complex $X = \{\emptyset\} \sqcup X(0) \sqcup X(1) \sqcup \cdots \sqcup X(k)$, where the set of i-dimensional faces $X(i)$ contains faces with cardinality $i+1$, is a (two-sided) λ-local spectral expander if (a) the underlying graph of X (i.e. the graph with vertices $X(0)$ and edges $X(1)$) is a (two-sided) spectral expander, and (b) for any face $s \in X$ with dimension at most $k-2$, the underlying graph of the s-link $X_s = \{\tau \setminus s | \tau \in X,\ \tau \supseteq s\}$ is also a (two-sided) spectral expander.

Our first result states any projection of the flags complex is a local spectral expander.[1]

▶ **Theorem 3** (Spectral Expansion of the Projected Flags Complex). *Fix any $d \geqslant 3$, any prime power q, and a set $S \subseteq [d-1]$ of size at least 2. Let $s \in \mathfrak{G}_q^{d,S}$ be any face with co-dimension at least 2. Then, the second eigenvalue of the underlying graph of the s-link of $\mathfrak{G}_q^{d,S}$ is $\frac{2}{(|S|-|s|-1)(\sqrt{q}-1)}$. In particular, the complex $\mathfrak{G}_q^{d,S}$ is a $\frac{2}{\sqrt{q}-1}$-local-spectral expander for any prime power q. Moreover, the complex $\mathfrak{G}_q^{d,S}$ is also a $\frac{1}{\sqrt{q}}$-product for any prime power q.*

We remark that while the local-spectral expansion of the projected flags complex has not been studied explicitly to our knowledge, similar bounds (up to a worse constant in the denominator) can be derived from combining improved "Trickling-Down" Theorems [1, 64] with well known bounds on the spectrum of subspace inclusion (Grassmann) graphs [20]. We give a simple and direct proof that does not require the former machinery.

Coboundary Expansion. One way to view combinatorial (edge) expansion of a (connected) graph is that if a graph has expansion η, then for any function $f : V \to \mathbb{F}_2$ there exists a constant function $\delta g : V \to \mathbb{F}_2$ such that

$$\eta \cdot \mathsf{dist}(f, \delta g) \leqslant \mathsf{wt}(\delta f)$$

where $\delta f(u,v) = f(u) - f(v)$ is the "coboundary" of f (similarly δg is the "coboundary" of some constant $g : \{\emptyset\} \to \mathbb{F}_2$). In other words, δf is the size of the cut formed by f, and any f whose cut has few edges must be close to 0 (a very small set) or 1 (a very large set).

[1] The below also references the notion of a λ-product, a slight variant of local-spectral expansion in the partite case which promises every bi-partite component of every link has expansion $\lambda_2 \leqslant \lambda$. See Definition 19.

For general groups, this definition can be extended to anti-symmetric[2] functions taking values from any group Γ instead of $\mathbb{F}_2$ by taking $\delta f(u,v) = f(u)f(v)^{-1}$. For a simplicial complex, it is possible to generalize this notion to the 1-coboundary expansion with respect to a group Γ that compares weights of the coboundary of 1-dimensional faces to their distance from 0-dimensional coboundary, as opposed to comparing weights of coboundary functions on 0-dimensional faces to their distance from the corresponding coboundary. In particular, a simplicial complex has 1-coboundary expansion $h^1(X,\Gamma)$ if for any anti-symmetric function $f : \vec{X}(1) \to \Gamma$

$$h^1(X,\Gamma) \cdot \mathsf{dist}(f, \delta g) \leqslant \mathsf{wt}(\delta f)$$

for some anti-symmetric function $g : X(0) \to \Gamma$ on the directed edges $\vec{X}(1)$ and vertices $X(0)$, where $\delta f(u,v,w) = f(u,v)f(v,w)f(w,u)$ is the "coboundary" of f. Our second result states that the projection of the flags complex onto its central dimensions is a coboundary expander.

▶ **Theorem 4** (Coboundary Expansion of Projected Flags Complex). *There exists a constant q_0 such that the following holds. Let Γ be any group. Let $d > k \geqslant 2$ be two integers and $q \geqslant q_0$ be a prime power. Let*

$$S_k = \{d-k, \ldots, d-1, d, d+1, \ldots, d+k\}\,.$$

Then, for any face $s \in \mathfrak{G}_q^{2d,S_k}$ with codimension at least 3, $\mathfrak{G}_{q,s}^{2d,S_k}$ is 1-coboundary expander with

$$h^1(\mathfrak{G}_{q,s}^{2d,S_k}, \Gamma) = \Omega\left(1\right).$$

As a corollary, we get a completely elementary construction of local-spectral coboundary expanders of sub-polynomial degree:

▶ **Corollary 5** (Elementary Sub-polynomial Degree HDX). *There exists a constant q_0 such that for any $d \geqslant k+1 \geqslant 3$, prime power $q \geqslant q_0$, and group Γ, $X = \mathfrak{G}_q^{2d,S_k}$ has $O(q^{d^2})$ vertices and*

- **(i)** ***Local-spectral Expansion:*** *$\frac{2}{(j-1)(\sqrt{q}-1)}$-spectral expansion at co-dimension j,*
- **(ii)** ***Coboundary Expansion:*** *$h^1(X,\Gamma) \geqslant \Omega(1)$,*
- **(iii)** ***Maximum Degree:*** *$\Delta_{max}(X) \leqslant q^{O(kd)}$.*

Moreover, the complex is strongly explicit.

In particular, for any $\lambda > 0$ we get a family of λ-local-spectral and coboundary expanders with degree $\exp(k\sqrt{\log(N)\log(1/\lambda)})$. To the best of our knowledge, these are the first sub-polynomial degree high dimensional expanders for which the cohomology is known to vanish for *every group*, rather than for all sufficiently small copies of Sym_m. As we discuss below, this is an important point toward the construction of better soundness agreement tests and PCPs.

We briefly comment on the proof before moving on, which is divided into two main steps. First, we need to argue that the projected flags complex (and all its links) have *vanishing cohomology* (Lemma 32), meaning that every $f : E \to \Gamma$ with $\delta f = 0$ is a coboundary δg for some $g : V \to \Gamma$. This is a well-known fact for the full flags-complex, and can be derived for projections of the flags complex via fairly standard tools in algebraic topology, e.g., from [16, 74]. We give a simplified, self-contained proof via [16]'s theory of shellability.[3]

[2] We discuss this standard mild restriction on the family of functions in Section 2.3.
[3] We thank Roy Meshulam for pointing us to this approach.

Second, we show that the central projections of the flags complex also satisfy "cocycle" expansion, which states that if $|\delta f|$ is small, then f must be close to a function in $\mathrm{Ker}(\delta)$ (which due to vanishing cohomology, is exactly the set of coboundaries). The proof relies on several techniques from [39, 23] to reduce the analysis to simple local components of the complex. In particular, we first use a local-to-global argument (Lemma 44) from [23] to move to links of the complex, and a "color restriction lemma" (Lemma 45) to reduce to showing most projections of the links to three colors are coboundary expanders. Finally, we use the cones method [61] (see Section 2.5) to argue directly that most projections to three colors indeed have constant coboundary expansion.

Swap Coboundary Expansion. Towards construction of agreement testers, [24, 13] define a third notion of expansion called swap coboundary expansion. Swap coboundary expansion essentially says the faces complexes (defined below) of the simplicial complex under consideration is a 1-coboundary expander.

▶ **Definition 6.** *Let X be a d-dimensional simplicial complex. Let $r \leqslant d$. We use $\mathrm{F}^r X$ to denote the simplicial complex whose vertices are $\mathrm{F}^r X(0) = X(r)$ and whose faces are all $\{s_0, s_1, ..., s_j\}$ such that $s_0 \sqcup s_1 \sqcup \cdots \sqcup s_j \in X((j+1)(r+1)-1)$. We call this the* r-faces complex *of X.*

Now, we can define swap coboundary expansion.

▶ **Definition 7.** *A simplicial complex X is said to have* (β, r)-swap coboundary expansion *if $\mathrm{F}^r X$ has 1-coboundary expansion at least β with respect to every group.*

Finally, we show swap coboundary expansion of the projections of the flags complex.

▶ **Theorem 8** (Swap coboundary expansion of the projected flags complex)**.** *Let d, k, r be integers such that $d > k \geqslant r^5$. There is some $q_0 = q_0(k, r)$ such that the following holds. Let $q \geqslant q_0$ be any prime power, $\mathfrak{G}$ the $\mathfrak{G}_q^{2d}$-flags complex, and*

$$S_k = \{d-k, \ldots, d-1, d, d+1, \ldots, d+k\}\,.$$

Then $\mathfrak{G}^{S_k}$ is a $(\exp(-O(\sqrt{r}\log^2 r)), r)$-swap coboundary expander.

Theorem 8 is an adaptation of the strategy of [24] for the full un-projected flags complex. We break the faces complex $\mathrm{F}^r(\mathfrak{G}^{S_k})$ into progressively simpler complexes (projections and deep links) until we can bound the coboundary expansion directly via the cones method. At a high level, the critical observation is simply that once one has moved to the link of a vertex, the projected flags complex behaves very similarly to a product (join) of full flags complexes, allowing us to apply [24]'s machinery without substantial modification.

Applications. Following [22, 13, 14], we give two main applications of our swap coboundary expansion result: (i) an agreement test in the low-soundness regime, and (ii) a corresponding simple, near-linear size PCP construction.

Agreement testing. An agreement test is a randomized property tester that, given an ensemble of local functions $\{f_s : s \to \Sigma | s \in \mathcal{F}\}$ on $\mathcal{F} \subseteq 2^{[n]}$, tries to determine whether there is a global function $G : [n] \to \Sigma$ such that $f_s = G|_s$ for all $s \in \mathcal{F}$. Many recent works have examined the case where the local functions are defined on some level of a high-dimensional expander. In this work, we consider the standard two-query agreement test (the "V test")

where we pick two functions $f_s, f_{s'}$ on sets of size ℓ where $|s \cap s'| = \sqrt{\ell}$, and the test accepts if $f_s|_{s\cap s'} = f_{s\cap s'}$, i.e., the two functions agree on all the common points in their domain. [22, 13, 14] recently showed that if a complex has good local-spectral and swap coboundary expansion, then it supports this two-query test. Thus, as a corollary of our results above, we immediately get that the projected flags complex gives rise to a simple near-linear size low soundness agreement tester.

▶ **Theorem 9** (Agreement testing over the projected flags complex). *Fix a $\delta > 0$. There is some q_0 such that the following holds.*

Let $q \geqslant q_0$ be any prime power. Let d, k, ℓ be such that $d > k \geqslant \mathsf{poly}(\ell)$ and $\ell \geqslant \exp(\mathsf{poly}(1/\delta))$. Let $\mathfrak{G}$ denote the $\mathfrak{G}_q^{2d}$-flags complex and

$$S_k = \{d-k, \ldots, d-1, d, d+1, \ldots, d+k\}.$$

Then, the $(\ell, \sqrt{\ell})$-agreement test over $\mathfrak{G}^{S_k}(\ell)$ with respect to every alphabet Σ has soundness δ, i.e., if $F : \mathfrak{G}^{S_k}(\ell-1) \to \Sigma^\ell$ passes the $(\ell, \sqrt{\ell})$-agreement test with respect to $\mathfrak{G}^{S_k}$ with probability at least δ, then there is $f : \mathfrak{G}^{S_k}(0) \to \Sigma$ such that

$$\mathop{\mathbb{P}}_{A\sim\mu_\ell}[\mathsf{dist}(F[A], f|A) \leqslant \delta] \geqslant \mathsf{poly}(\delta).$$

We remark that on the complete complex, it is known that the V test has soundness $\delta = 1/\mathsf{poly}(\ell)$, rather then $1/\mathsf{poly}\log(\ell)$ as given above. In prior HDX constructions, there is a barrier toward improving the soundness that comes from the cohomology of the complex becoming non-trivial for larger $\mathrm{Sym}_m = \Gamma$. Since our complexes are coboundary expanders over *every group*, this obstruction vanishes and raises the interesting possibility of showing agreement tests with optimal soundness (an important factor in the eventual goal of constructing subconstant soundness PCPs from HDX). Indeed, it is plausible that 3-query tests like the Z-test [56, 36] may have soundness $2^{-\Omega(\ell)}$.

Probabilistically Checkable Proofs (PCPs). The PCP theorem [41, 11, 10, 30] is one of the most celebrated results in theoretical computer science. Here, we will take the classic view of PCPs through hardness of approximation for the Label Cover problem.

▶ **Definition 10.** *An instance of* Label Cover $\Psi = (G = (L \cup R, E), \Sigma_\ell, \Sigma_r, \Phi = \{\phi_e\}_{e\in E})$ *consists of a bipartite graph G, alphabets Σ_ℓ, Σ_r and constraints $\phi_e : \Sigma_\ell \to \Sigma_r$, for each edge $e \in E$.*

Given a label cover instance Φ, the goal is to find assignments $A_\ell : L \to \Sigma_\ell$ and $A_r : R \to \Sigma_r$ that satisfy as many of the constraints as possible, namely that maximize the quantity

$$\mathsf{val}_\Phi(A_\ell, A_r) = \frac{1}{|E|}\big|\{e = (u,v) \in E \mid \phi_e(A_\ell(u)) = A_r(v)\}\big|.$$

This maximum value is referred to as the value of the instance Φ and denoted by

$$\mathsf{val}(\Phi) = \max_{A_\ell, A_r} \mathsf{val}_\Phi(A_\ell, A_r).$$

When viewed through the label cover problem, the PCP theorem takes the form of NP-hardness of distinguishing between satisfiable label cover instances and instances with value less than δ provided the alphabet size is large enough. Proofs of the PCP theorem proceed by

reducing the 3SAT problem to a label cover instance. A longstanding goal in such reductions is to achieve small δ (soundness) without dramatically blowing up the size of the resulting label-cover instance.

[14] give the first construction of quasi-linear sized PCP with low constant soundness. They do this by showing how to use a local-spectral expander with few additional structural properties and that supports the aforementioned agreement test to construct such instances. Finally, they show that variants of the complexes from [17] have the required properties. Unfortunately, these complexes also rely on extremely involved algebra. In contrast, our construction is elementary and combinatorial, and as a corollary to our agreement testing result (Theorem 9) we obtain the following elementary low-soundness PCP construction.

▶ **Theorem 11.** *For all $\delta \in (0,1)$, there exist a constant $C > 0$ such that for all sufficiently large integers d and n there is a polynomial time reduction mapping a 3SAT instance ϕ of size n to a label cover instance Ψ over a bipartite graph G of size n' for some $n' \leqslant n \exp\left(C\sqrt{\log n \log\log n}\right)$, such that*

1. *If ϕ is satisfiable, then Ψ is satisfiable.*
2. *If ϕ is unsatisfiable, then $\mathsf{val}(\Psi) \leqslant \delta$.*
3. *The left alphabet of Ψ is $\Sigma^{\exp(C/\delta)}$ and right alphabet is $\Sigma^{\sqrt{\exp(C/\delta)}}$ for some alphabet Σ with $\log|\Sigma| \leqslant \exp\left(C\sqrt{\log n \log\log n}\right)$.*

We remark that instantiating the base PCP with [68], one can make the entire construction combinatorial with the exception of the use of a multiplication code, of which only algebraic constructions are known. Finally, we note the alphabet size in the above statement is considerably larger compared to [14]. There are standard methods to reduce this [34], but they also rely on algebraic techniques. It would be interesting to give a purely combinatorial method, and we leave this as a direction for future work.

1.2 Further Related Works

Simple High Dimensional Expanders. Beyond the celebrated construction of Kaufman and Oppenheim [60] and its variants [70, 19], a number of recent works have considered simplified constructions of local-spectral expanders. [66, 45] construct bounded degree local-spectral HDX via a product of expanders and the complete complex, but cannot achieve strong enough expansion for most applications [27]. [46] constructs local-spectral HDX of any expansion from low-rank matrices, achieving roughly the same degree as our construction. This was improved in [29] via a recursive procedure to degree $2^{\log(N)^{\varepsilon}}$ for any constant $\varepsilon > 0$. Neither construction has vanishing cohomology, and therefore cannot be used in the construction of PCPs.

Comparison with [71]. In recent independent work, O'Donnell and Singer [71], building on [63], showed that [14]'s PCPs may be instantiated on a variant of the coset-complex HDX construction of Kaufman and Oppenheim [60]. These complexes are remarkably simpler than those originally used in [14], relying only elementary matrix groups, and achieve parameters matching [14]'s construction while our flags-complex based PCP is closer to the original construction of Moshkovitz-Raz [69]. Nevertheless, [71] still requires a fairly substantial amount of algebraic manipulation, and while the complexes are undeniably simpler than [17], they are certainly more involved than the completely elementary subspace-based construction suggested in this work. Finally, as discussed our complexes have vanishing cohomology over *all* groups, rather than just Sym_m for small m, removing a major barrier toward better agreement tests and PCPs with strong sub-constant soundness.

2 Preliminaries

General Notations. We will use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We will use Sym_A to denote the group of permutations on a set A. We will simply use Sym_n for the group $\mathrm{Sym}_{[n]}$. For a tuple t, we will write $\mathsf{set}(t)$ to denote the set of elements that occur in the tuple. For a set of sets J, we write $\cup J = \bigcup_{s \in J} s$. We abuse the symbol 0 to mean the subspace $\{0\}$ as well; such usage should be clear from context. We denote by $\lambda(\mathsf{A})$ the second largest eigenvalue of an operator A. We denote by $|\lambda|(\mathsf{A})$ the second largest eigenvalue of an operator A in absolute value. We will use $\mathbb{F}_q$ to denote the field of order q.

Let A, B be two set and $\mathcal{D}$ and distribution on A. Then, for any two functions $f : A \to B$ and $g : A \to B$, we define

$$\mathsf{dist}_{\mathcal{D}}(f, g) = \mathop{\mathbb{P}}_{x \sim \mathcal{D}} [f(x) \neq g(x)] .$$

When the distribution is not specified the uniform distribution is assumed.

2.1 Graphs and Expansion

We will use $V(G)$ and $E(G)$ to denote the vertices and the edges of a graph G, respectively. We will drop the parenthetical and simply write V and E when G is clear from the context. We will use $\Delta(G)$ to denote the max degree of the graph G. We will use $\mathsf{diam}(G)$ denote the diameter of the graph G, i.e., the maximum (over all pairs of vertices) length of shortest paths.

Probability density over Graphs. We can define an arbitrary probability density π_1 over the edges of a graph. This induces a distribution π_0 over the vertices in which we first sample an edge $\{u, v\} \sim \pi_1$ and then uniformly pick u or v. Note that this is the stationary distribution for the random walk on the graph.

Partite Graphs. A *k-partite* graph is a graph such that we can decompose $V = V_1 \sqcup V_2 \sqcup \cdots \sqcup V_k$ such that for every edge $\{u, v\} \in E$ we have $u \in V_i$ and $v \in V_j$ for $i \neq j$. If $v \in V_i$, we define the color $\mathtt{col}(v) = i$. Let $c \subseteq [k]$ be a set of colors. We define the *projection* of a graph G onto c as the graph G^c induced by the vertex set $V(G^c) = \{v \in V | \mathtt{col}(v) \in c\}$. We will often shorten $V(G^c)$ as V^c and $E(G^c)$ as E^c.

Expansion. Let $G = (V, E)$ be a graph and let $\pi_1 : E \to (0, 1]$ be a probability distribution. Let A_G be the *normalized adjacency operator*. This operator takes as input $f : V \to \mathbb{R}$ and outputs $\mathsf{A}_G f : V \to \mathbb{R}$, with

$$\mathsf{A}_G f(v) = \frac{1}{2\pi_0(v)} \sum_{u \sim v} \pi_1(uv) f(u).$$

The (normalized) adjacency operator is self adjoint with respect to the inner product on $\ell_2(V) = \{f : V \to \mathbb{R}\}$ given by

$$\langle f, g \rangle = \sum_{v \in V} \pi_0(v) f(v) g(v).$$

We write $\lambda(G) := \lambda(\mathsf{A}_G)$ and $|\lambda|(G) := |\lambda|(\mathsf{A}_G)$ for the the second largest eigenvalue and the second largest absolute value of the eigenvalue of the (normalized) adjacency operator, respectively. We say that G is a *λ-one sided spectral expander* if for every $\lambda(G) \leqslant \lambda$ and say

that G is a λ-*two sided spectral expander* if $|\lambda|(G) \leqslant \lambda$. We say that G is an η-edge expander if for every subset $S \subseteq V$, $S \neq \emptyset$, it holds that

$$\mathop{\mathbb{P}}_{uv\sim\mu_1}[u \in S, v \in V \setminus S] \geqslant \eta \mathop{\mathbb{P}}_{\mu_0}[S] \mathop{\mathbb{P}}_{\mu_0}[V \setminus S].$$

Spectral and edge expansion are classically related to each other via Cheeger's Inequality [4, 5]. We will only use the following "easy direction" of Cheeger.

▶ **Lemma 12** (Folklore). *Let G be a λ-one sided spectral expander, then G is a $(1-\lambda)$-edge expander.*

We will also need the standard expander-mixing lemma, which states that the weight of edges between any two sets A, B is roughly the expected weight based on the sizes of A and B.

▶ **Lemma 13** (Expander Mixing Lemma). *Let $G = (U, V, E)$ be a bipartite graph in which the $\lambda(G) \leqslant \lambda$, and let μ be the stationary distribution over G. Then for all $A \subseteq U$ and $B \subseteq V$ we have that*

$$\left| \mathop{\mathbb{P}}_{(u,v)\in E}[u \in A, v \in B] - \mu(A)\mu(B) \right| \leqslant \lambda\sqrt{\mu(A)(1-\mu(A))\mu(B)(1-\mu(B))}.$$

We also note a well-known sampling property of the bipartite expanders.

▶ **Lemma 14.** *Let $G = (U, V, E)$ be a weighted bipartite graph with $\lambda(G) \leqslant \lambda$. Let $B \subseteq U$ be a subset with $\mu(B) = \delta$ and set*

$$T = \left\{ v \in V \;\middle|\; \mathop{\mathbb{P}}_{u \text{ neighbour of } v}[u \in B] - \delta > \varepsilon \right\}.$$

Then $\mathbb{P}[T] \leqslant \lambda^2\delta/\varepsilon^2$.

Majority and expansion. A second very useful fact about expander graphs is that local agreement on the graph implies global agreement with majority. Let $G = (V, E)$ be a graph and $g : V \to \{1, 2, ..., n\}$ be some function. Let $S_i = \{v \in V | g(v) = i\}$. The majority assignment $\mathsf{maj}(g) \in \{1, 2, ..., n\}$ is the i such that $\mathbb{P}[S_i]$ is largest (ties broken arbitrarily). Observe that if $\mathbb{P}_v[g(v) = \mathsf{maj}(g)] \approx 1$ then for most edges $uv \in E$, it holds that $g(v) = g(u)$ (since with high probability they are both equal to $\mathsf{maj}(g)$). In expander graphs a converse to this statement also holds. That is, if for most edges $g(v) = g(u)$, then $\mathbb{P}_v[g(v) = \mathsf{maj}(g)] \approx 1$.

▶ **Lemma 15.** *Let G be an η-edge expander. Let $S_1, S_2, \ldots, S_m$ be a partition of the vertices of V. Assume that*

$$\mathop{\mathbb{P}}_{uv\sim\mu_1}[\exists i, u \in S_i \text{ and } v \notin S_i] \leqslant \varepsilon.$$

Then, there exists i such that $\mathbb{P}[S_i] \geqslant 1 - \frac{\varepsilon}{\eta}$. Stated differently, for every $g : V \to \{1, 2, ..., n\}$, we have

$$\mathop{\mathbb{P}}_{v}[g(v) \neq \mathsf{maj}(g)] \leqslant \mathop{\mathbb{P}}_{uv\sim\mu_1}[g(u) \neq g(v)].$$

2.2 Basics of Simplicial Complexes

We will now describe simplicial complexes and related notions. Our exposition and notation in this section closely follow [23, 24].

Simplicial Complexes. A *simplicial complex* X is a downward closed set system, i.e., if $s \in X$ and $t \subseteq s$, then $t \in X$.

- The sets in a simplicial complex are called *faces*. The *dimension* of a face s is $|s| - 1$.
- The dimension of a complex is the dimension of the largest face. In a *pure* simplicial complex, all maximal faces have the same dimension.
- $X(i)$ is the set of all the faces of dimension i of X. In particular, we have $X(-1) = \{\emptyset\}$. We will sometimes refer to $X(i)$ as the level i of the complex.
- The set of *oriented* k-faces is $\vec{X}(k) = \{(v_0, v_1, \ldots, v_k) | \{v_0, v_1, \ldots, v_k\} \in X(k)\}$.
- Let $s = (v_0, v_1, \ldots, v_k)$ be an oriented face and $i \in \{0, 1, \ldots, k\}$ be an index. Then, we get the face s_i by removing v_i from s.
- For convenience, typically we drop the set brackets and the tuple brackets for faces.

We will use Δ_n to denote the *complete complex* on n vertices, i.e., the $n-1$-dimensional complex with $[n]$ as the top face.

Partite Complexes. A $(d+1)$-partite simplicial complex is a d-dimensional simplicial complex such that one can decompose $X(0) = A_0 \sqcup A_1 \sqcup \cdots \sqcup A_d$ such that for every $s \in X(d)$ and $i \in \{0, 1, \ldots, d\}$ it holds that $|s \cap A_i| = 1$. The color of a vertex $\mathtt{col}(v) = i$ such that $v \in A_i$. More generally, the color of a face s is $c = \mathtt{col}(s) = \{\mathtt{col}(v) | v \in s\}$. We denote by X^c, the *projection* of X to c, the set of faces $s \in X$ with the color $\mathtt{col}(s) \subseteq c$, and for a singleton $\{i\}$ we sometimes write X^i instead of $X^{\{i\}}$. Note X^c is a $|c|$-partite complex.

Links, Underlying Graphs and Skeletons. Let X be a d-dimensional simplicial complex. For $k \leqslant d$, the *k-skeleton* of X is the complex $X^{\leqslant k} = \bigcup_{j=-1}^{k} X(j)$. When $k = 1$, we call this complex the *underlying graph* of X, as it consists of the vertices and edges in X (as well as the empty face). A d-dimensional complex is called a *clique complex* if every $(d+1)$-clique in its underlying graph is a face in X. The *diameter* $\mathsf{diam}(X)$, and the *max degree* $\Delta(X)$ are the diameter $\mathsf{diam}(G = (X(0), X(1)))$, and the max degree $\Delta(G)$ of the underlying graph of X. The *link* of a face $s \in X$ is the $d' = (d - |s|)$-dimensional complex $X_s = \{t \setminus s | t \in X, t \supseteq s\}$.

Joins. Given two simplicial complexes X_1 and X_2, we define the *join* $X_1 * X_2$ to be the simplicial complex defined on the vertices $X_1(0) \sqcup X_2(0)$ with

$$X_1 * X_2 = \{\sigma \sqcup \tau | \sigma \in X_1, \tau \in X_2\}.$$

Extend this definition inductively to define joins of multiple complexes.

Probability density over a simplicial complex. Let X be a d-dimensional (pure) simplicial complex. The top faces can be assigned an arbitrary probability density π_d. The probability density π_d induces a probability density π_k on the k-dimensional faces in which we first sample $t \sim \pi_d$ and then uniformly sample a subface $s \subseteq t$. We observe that

$$\pi_k(s) = \frac{1}{\binom{d+1}{k+1}} \sum_{t \supseteq s} \pi_d(t).$$

This also defines a probability density on oriented faces where all permutations of a face have the same probability. So, we have $\pi_k(s) = \frac{1}{(k+1)!}\pi_k(\mathsf{set}(s))$ for any face $s \in \vec{X}(k)$. For a simplicial complex X with a measure $\pi_d : X(d) \to (0, 1]$, the induced measure $\pi_{d', X_s} : X_s(d - |s|) \to (0, 1]$ on X_s, for $t \in X_s(d - |s|)$, i.e., the probability $t \sim \pi_{d', X_s}$ is the probability of sampling $t \sqcup s$ from π_d conditioned on containing s. Observe that we have

$$\pi_{d', X_s}(t \setminus s) = \frac{\pi_d(t)}{\sum_{t' \supseteq s} \pi_d(t')}.$$

Whenever the complex X is clear we shall write $\pi_{d', s}$ instead of π_{d', X_s}.

2.3 Expansion on a Simplicial Complex

Expansion on a simplicial complex can be defined in numerous ways. We discuss *local spectral expansion* and *coboundary expansion* here.

Local Spectral Expansion. The notion of local spectral expansion requires that underlying graph of each of its links is a spectral expander. Before formally defining local spectral expansion, let us setup some notation. We use $\lambda(X_s)$ to denote the second largest eigenvalue of the (normalized) adjacency operator of the underlying graph of X_s and we use $|\lambda|(X_s)$ to denote $|\lambda|(G_{X_s})$, where G_{X_s} is the underlying graph of X_s.

▶ **Definition 16** (Local Spectral Expansion [58, 72])**.** *Let X be a d-dimensional simplicial complex and let $\lambda \in (0,1)$. X is a λ-one sided local spectral expander if for every $s \in X^{\leqslant d-2}$ it holds that $\lambda(X_s) \leqslant \lambda$. Similarly, X is a λ-two sided local spectral expander if for every $s \in X^{\leqslant d-2}$ it holds that $|\lambda|(X_s) \leqslant \lambda$.*

▶ Remark 17. In the literature, the term *high-dimensional expander* is also sometimes used to mean local spectral expanders.

The following useful result relates the expansion of the links of faces with codimension 2 with local spectral expansion of the entire complex.

▶ **Theorem 18** (Trickle Down Theorem [72])**.** *Let X be a d-dimensional simplicial complex such that the 1-skeleton of every link is connected and for every $s \in X(d-2)$, the link X_s is a one-sided λ-expander. Then X is a μ-local spectral expander, where $\mu = \frac{\lambda}{1-(d-1)\lambda}$.*

We will also use the following variant of local-spectral expansion for partite complexes which asks that the projection of any link onto two colors is a good expander.

▶ **Definition 19** (λ-product [50])**.** *A d-partite complex X is called a λ-product if for every link X_τ of co-dimension at least 2 and colors $i, j \notin \mathtt{col}(\tau)$, we have:*

$$\lambda(X_\tau^{ij}) \leqslant \lambda.$$

Coboundary Expansion. Another useful notion of expansion in a simplicial complex is coboundary expansion. Let X be a d-dimensional simplicial complex for $d \geqslant 1$ and let Γ be any group. First, we define the notion of cochains. For $i = -1, 0$, the set of *i-cochains* is $C^i(X,\Gamma) = \{f : X(i) \to \Gamma\}$. We sometimes identify $C^{-1}(X,\Gamma) \cong \Gamma$. For $i = 1, 2$, the i-cochains are defined as follows. The 1-cochains

$$C^1(X,\Gamma) = \left\{ f : \overrightarrow{X}(1) \to \Gamma \;\middle|\; f(u,v) = f(v,u)^{-1} \right\}$$

and the 2-cochains

$$C^2(X,\Gamma) = \left\{ f : \overrightarrow{X}(i) \to \Gamma \;\middle|\; \begin{array}{l} \forall \pi \in \mathrm{Sym}_3, (v_0,v_1,v_2) \in X(2) \\ f(v_{\pi(0)}, v_{\pi(1)}, v_{\pi(2)}) = f(v_0,v_1,v_2)^{\mathsf{sign}(\pi)} \end{array} \right\}$$

are the spaces of the so-called anti-symmetric functions on edges and triangles, respectively. For $i = -1, 0, 1$, we define the "coboundary operators" as $\delta_i : C^i(X,\Gamma) \to C^{i+1}(X,\Gamma)$ by

(i) $\delta_{-1} : C^{-1}(X,\Gamma) \to C^0(X,\Gamma)$ is $\delta_{-1}h(v) = h(\emptyset)$,
(ii) $\delta_0 : C^0(X,\Gamma) \to C^1(X,\Gamma)$ is $\delta_0 h(v,u) = h(v)h(u)^{-1}$,
(iii) $\delta_1 : C^1(X,\Gamma) \to C^2(X,\Gamma)$ is $\delta_1 h(v,u,w) = h(v,u)h(u,w)h(w,v)$.

We write just δf for $\delta_i f$ when the dimension i is clear from context.

Although we want results on a general group Γ, it is instructive to consider the case where $\Gamma = \mathbb{F}_2$. Specifically, in this case, each i-cochain f is an indicator for some subset $S_f \subseteq X(i)$ of i-faces. Furthermore, the coboundary of an 0-cochain f corresponds to the indicator edges crossing the set S_f.

Let $Id = Id_i \in C^i(X, \Gamma)$ be the function that always outputs the identity element. It is easy to check that $\delta_{i+1} \circ \delta_i h = Id_{i+2}$ for all $i = -1, 0$ and $h \in C^i(X, \Gamma)$. Thus, if we write

$$Z^i(X, \Gamma) = \ker(\delta_i) \subseteq C^i,$$

and

$$B^i(X, \Gamma) = \mathrm{im}(\delta_{i-1}) \subseteq C^i,$$

for the sets of the so-called *cocycles* and *coboundaries*, respectively, we see that $B^i \subseteq Z^i$. We say i-th cohomology is *trivial* whenever $B^i = Z^i$.

▶ **Definition 20** (Cocycle and coboundary expansion [65, 48]). *Let X be a d-dimensional simplicial complex for $d \geqslant 2$. Let $-1 \leqslant k \leqslant d-1$. The k-cocycle expansion $h^k(X, \Gamma)$ of X with respect to a group Γ is given by*

$$h^k(X, \Gamma) = \min_{f \in C^k \setminus Z^k} \frac{\mathsf{wt}(\delta f)}{\mathsf{dist}(f, Z^k)}.$$

Additionally, if the k-th cohomology is trivial, i.e., we have $B^k(X, \Gamma) = Z^k(X, \Gamma)$, then we say $h^k(X, \Gamma)$ is the k-coboundary expansion of X with respect to a group Γ.

We remark the above is often called "cosystolic" rather than "cocycle" expansion in the literature. However the former originally referred to complexes with other additional properties [58], so there has been a recent shift in the it the literature to using "cocycle expansion" for this notion instead.

2.4 Walks on Simplicial Complexes

Simplicial complexes come equipped with several natural families of random walks generalizing the standard random walk on a graph.

Up-down and Down-up walks. Let $\ell \leqslant k \leqslant d$. The (k, ℓ)-*containment graph* $G_{k,\ell} = G_{k,\ell}(X)$ is the bipartite graph whose vertices are $L = X(k)$, $R = X(\ell)$ and whose edges are all (t, s) such that $t \supseteq s$. The distribution we associate to the edges is the natural distribution induced by the complex X, i.e., sampling $t \sim \pi_k$ and then uniformly sampling $s \subseteq t$ of size $|s| = \ell + 1$.

The *down operator* $\mathsf{D}_{k,\ell} : \ell_2(X(k)) \to \ell_2(X(\ell))$ is the bipartite operator of the containment graph $G_{k,l}$, i.e.,

$$\mathsf{D}_{k,\ell} f(s) = \mathop{\mathbb{E}}_{t \supseteq s} [f(t)].$$

The adjoint of $\mathsf{D}_{k,\ell}$ is the *up operator*, $\mathsf{U}_{\ell,k} : \ell_2(X(\ell)) \to \ell_2(X(k))$, given by:

$$\mathsf{U}_{\ell,k} g(t) = \mathop{\mathbb{E}}_{s \subseteq t} [g(s)].$$

Then the composition of the up and down operators $\mathsf{U}_{\ell,k}\mathsf{D}_{k,\ell}$ is called the "down-up" walk, and the "down-up" walk is similarly defined.

Swap Walks. Let X be a d-dimensional simplicial complex. Let i, j be so that $i+j+1 \leqslant d$. The swap walk $\mathsf{S}_{i,j} = \mathsf{S}_{i,j}(X)$ is the bipartite adjacency operator of the graph $(X(i), X(j), E)$, where an edge $\{s_i, s_j\}$ is chosen in this graph by first selecting a face $t \sim \pi_{i+j+1}$, and then partitioning it to $t = s_i \sqcup s_j$ where $s_i \in X(i)$ and $s_j \in X(j)$. This walk was defined and studied independently by [3] and [21].

▶ **Lemma 21** ([50]). *Let X be a λ-two-sided local spectral expander. Then the second largest eigenvalue of $\mathsf{S}_{k,\ell}(X)$ is upper bounded by $\sqrt{(k+1)(\ell+1)}\lambda$.*

Colored Swap Walks. The standard swap walks are not well behaved on partite complexes, but there is a useful analog for this setting defined in [21]. Let X be a d-partite simplicial complex, and $F_1, F_2 \subseteq [d]$ be two disjoint subsets. The colored swap walk $\mathsf{S}_{F_1,F_2} = \mathsf{S}_{F_1,F_2}(X)$ is the bipartite adjacency operator of the graph (X^{F_1}, X^{F_2}, E), where an edge $\{s_1, s_2\}$ is chosen in this graph by first selecting a face $t \in X^{F_1 \sqcup F_2}$, and then partitioning it to $t = s_1 \sqcup s_2$ according to its colors.

▶ **Lemma 22** ([50]). *Let X be a d-partite λ-product, and $F_1, F_2 \subseteq [d]$ disjoint subsets. Then*

$$\lambda(\mathsf{S}_{F_1,F_2}) \leqslant \sqrt{|F_1||F_2|}\lambda.$$

2.5 Cones

Cones are the main machinery used in most proofs of coboundary expansion. We define (non-abelian) cones. First, we define the composition of two paths $P_0 = (v_0, v_1, \ldots, v_m)$ and $P_1 = (v_m, v_{m+1}, \ldots, v_n)$ by $P_0 \circ P_1 = (v_0, v_1, \ldots, v_m, v_{m+1}, \ldots, v_n)$. Note that composition is only defined when the end point of the first path is equal to the starting point of the second.

Now, fix a simplicial complex X and some $v_0 \in X(0)$. We define two symmetric relations on loops around v_0:

(BT) We say that $P_0 \overset{BT}{\sim} P_1$ if $P_i = Q_0 \circ (u, v, u) \circ Q_1$ and $P_{1-i} = Q_0 \circ (u) \circ Q_1$ for $i = 0, 1$ (i.e. going from u to v and then backtracking is trivial).

(TR) We say that $P_0 \overset{TR}{\sim} P_1$ if $P_i = Q_0 \circ (u, v) \circ Q_1$ and $P_{1-i} = Q_0 \circ (u, w, v) \circ Q_1$ for some triangle $uvw \in X(2)$ and $i = 0, 1$.

We denote by $P \sim_1 P'$ if there is a sequence of loops $(P_0 = P, P_1, \ldots, P_m = P')$ and $j \in [m-1]$ such that:

1. $P_j \overset{TR}{\sim} P_{j+1}$ and
2. For every $j' \neq j$, $P_{j'} \overset{BT}{\sim} P_{j'+1}$.

In other words, We can get from P to P' by a sequence of "equivalences", where exactly one equivalence is by (TR).

Finally, for a path $P = (u_0, u_1, \ldots, u_m)$ on the graph underlying X, we denote by P^{-1} the inverse path $(u_m, \ldots, u_1, u_0)$.

▶ **Definition 23** (Cone). *A cone is a triple $C = (v_0, \{P_u\}_{u \in X(0)}, \{T_{uw}\}_{uw \in X(1)})$ such that*

1. *$v_0 \in X(0)$,*
2. *for every $v_0 u \in X(0)$, we have a walk P_u from v_0 to u,*
3. *for $u = v_0$, we take P_{v_0} to be the loop with no edges from v_0,*
4. *and for every $uw \in X(1)$, T_{uw} is a sequence of walks $(P_0, P_1, \ldots, P_m)$ such that*
   a. *$P_0 = P_u \circ (u, w) \circ P_w^{-1}$,*
   b. *for every $i = 0, 1, \ldots, m-1$, $P_i \sim_1 P_{i+1}$ and*
   c. *P_m is equivalent to the trivial loop by a sequence of (BT) relations.*

We call T_{uw} a contraction, and we write $|T_{uw}| = m$. The diameter of a cone is defined as follows.

$$\mathrm{diam}(C) = \max_{uw \in X(1)} |T_{uw}|.$$

▶ **Lemma 24** ([61, 24]). *Let X be a simplicial complex such that $\mathrm{Aut}(X)$ is transitive on k-faces. Suppose that there exists a cone C with diameter R. Then the 1-coboundary expansion of X at least $\frac{1}{\binom{k+1}{3} \cdot R}$.*

2.6 Faces Complexes

▶ **Definition 25.** *Let X be a d-dimensional simplicial complex. Let $r \leqslant d$. We use $\mathrm{F}^r X$ to denote the simplicial complex whose vertices are $\mathrm{F}^r X(0) = X(r)$ and whose faces are all $\{s_0, s_1, ..., s_j\}$ such that $s_0 \sqcup s_1 \sqcup \cdots \sqcup s_j \in X((j+1)(r+1)-1)$. We call this the r-*faces complex *of X.*

Observe that this is a $\left(\lfloor \frac{d+1}{r+1} \rfloor - 1\right)$-dimensional complex.

Probability density over a faces complex. Let X be a d-dimensional simplicial complex, and let $r < d$. The distribution on the top-level faces of $\mathrm{F}^r X$ is given by the following. Let $m = \lfloor \frac{d+1}{r+1} \rfloor - 1$.

1. Sample a d-face $t = \{v_0, v_1, \ldots, v_d\} \in X(d)$.
2. Sample $s_0, s_1, \ldots, s_m \subseteq t$ such that $|s_i| = r+1$, $s_i \cap s_j = \emptyset$ and output $\{s_0, s_1, \ldots, s_m\}$.

A faces complex can be viewed as a subcomplex of the following (non-pure) simplicial complex.

▶ **Definition 26.** *Let X be a d-dimensional simplicial complex. We denote by $\mathrm{F}X$ the simplicial complex whose vertices are $\mathrm{F}X(0) = X \setminus \{\emptyset\}$ and whose faces are all $\{s_0, s_1, ..., s_j\}$ such that $s_0 \sqcup s_1 \sqcup \cdots \sqcup s_j \in X$.*

This complex is not pure in general, hence we do not define a distribution on it.

Links of a faces complex. The links of a faces complex correspond to a faces complexes of the links of the original complex. More precisely, we have the following statement.

▶ **Observation 27.** *Let $s \in \mathrm{F}X$. Then $\mathrm{F}X_s = \mathrm{F}(X_{\cup s})$, where $\cup s = \bigcup_{t \in s} t$. Furthermore, if $s \in \mathrm{F}^r X$, then we have $\mathrm{F}^r X_s = \mathrm{F}^r(X_{\cup s})$.*

We introduce the notion of *generalized links* to simplify notation for links of faces complex.

▶ **Definition 28** (Generalized links). *Let $w \in X$. We write $\mathrm{F}X_w$ for $\mathrm{F}(X_w)$. We also write $\mathrm{F}^r X_w$ for $\mathrm{F}^r X \cap \mathrm{F}X_w$.*

Note that $\mathrm{F}^r X_w$ is not necessarily a proper link of $\mathrm{F}^r X$. However, using this notation we can write the conclusions of Observation 27 more compactly as $\mathrm{F}X_s = \mathrm{F}X_{\cup s}$ and $\mathrm{F}^r X_s = \mathrm{F}^r X_{\cup s}$ for a face $s \in \mathrm{F}^r X$.

Colors on a faces complex. Let X be a k-partite complex. We denote the set of colors of $\mathrm{F}^r X$ by $\mathcal{C} = \binom{[k]}{r+1}$. Fix a set $J \subset \Delta_k$, denoted $J = \{c_1, \dots, c_m\}$, such that $c_j \subset [k]$ are pairwise disjoint. Let $\mathrm{F}^J X = \{s \in \mathrm{F}X | \mathtt{col}(s) \subseteq J\}$ be the sub-complex of $\mathrm{F}X$ whose vertex colors are in J, so $\mathrm{F}^J X(0) = \bigcup_{j=1}^m X^{c_j}$. We call such a complex a J-faces complex and also a *colored* faces complex. We will be particularly interested in the case where $J \in \mathrm{F}^r\Delta$, namely, J consists of pairwise disjoint subsets of cardinality $r+1$. In this case, $\mathrm{F}^J X$ is $|J|$-partite. We abuse notation in this section allowing multiple c_j's to be empty sets. In this case X^{c_j} are copies of $\emptyset$, and every empty set is in all top level faces of $\mathrm{F}^J X$.

The measure induced on the top level faces of $\mathrm{F}^J X$ is obtained by sampling $t \in X^{\cup J}$ and partitioning it into $t = s_1 \sqcup s_2 \sqcup \cdots \sqcup s_m$ such that $s_i \in X^{c_i}$. Finally, throughout the paper, we use the following notation. Let $J', J \subseteq \mathrm{F}\Delta$. We write $J' \leqslant J$, if $J = \{c_1, c_2, \dots, c_m\}$ and $J' = \{c'_1, \dots, c'_m\}$ where $c'_j \subseteq c_j$.

Swap Coboundary Expansion. We end this section by defining swap coboundary expansion.

▶ **Definition 29.** *A simplicial complex X is said to have* (β, r)-swap coboundary (cocycle) expansion *if $\mathrm{F}^r X$ has 1-coboundary (cocycle) expansion (with respect to every group) at least β.*

In general, our goal is to prove swap co-boundary expansion for $\beta \geqslant 2^{-o(r)}$, which is the pre-condition to a 1% agreement test and resulting PCP construction [13, 22, 14].

3 Expansion of the Projected Flags Complex

3.1 Local Spectral Expansion Bound

In this section, we show Theorem 3, that any projection of the flags complex is a good local-spectral-expander.

▶ **Theorem 3** (Spectral Expansion of the Projected Flags Complex). *Fix any $d \geqslant 3$, any prime power q, and a set $S \subseteq [d-1]$ of size at least 2. Let $s \in \mathfrak{G}_q^{d,S}$ be any face with co-dimension at least 2. Then, the second eigenvalue of the underlying graph of the s-link of $\mathfrak{G}_q^{d,S}$ is $\frac{2}{(|S|-|s|-1)(\sqrt{q}-1)}$. In particular, the complex $\mathfrak{G}_q^{d,S}$ is a $\frac{2}{\sqrt{q}-1}$-local-spectral expander for any prime power q. Moreover, the complex $\mathfrak{G}_q^{d,S}$ is also a $\frac{1}{\sqrt{q}}$-product for any prime power q.*

Towards proving Theorem 3, we first show Proposition 30 which relates the spectral expansion of a k-partite graph to its bipartite components.

▶ **Proposition 30.** *Let G be a k-partite graph with the vertex-set $V = \bigsqcup_{i\in[k]} P_i$ with (normalized) edge weights given by $\pi_2 : E \to [0,1]$ and stationary distribution given by $\pi_1 : V \to [0,1]$. Suppose $\lambda_2(G^{\{i,j\}}) \leqslant \lambda_{ij}$ for all $i \neq j \in [k]$. Let K be the complete graph on k vertices with edge weight $\kappa_2(i,j) = \pi_2(E^{\{i,j\}})$. Then, the second eigenvalue of G is at most*

$$\lambda_2(G) \leqslant \lambda_2(K) + \max_{i\in[k]} \mathop{\mathbb{E}}_{j\sim\kappa_2[i]} [\lambda_{ij}] .$$

where $\kappa_2[i]$ denotes the distribution on the neighbors of i induced by κ_2.

The proof is fairly elementary linear algebra, and defer it to the full version. We will also need the following bound on the expansion of the Grassmann graphs.

▶ **Fact 31** ([20, 43]). *Fix any $d \geqslant 3$. For any $i < j \leqslant d-1$, the second eigenvalue of the underlying graph of $\mathfrak{G}_d^q[i,j]$ is at most $q^{-(j-i)/2}$.*

Proof of Theorem 3. The underlying graph of the complex is a k-partite graph where $k = |S|$ with vertex-set $V = \bigsqcup_{i \in S} P_i$ for P_i subspaces of dimension i with edges given by subspace inclusion and weight given by the number of chains containing end points of the edge with appropriate normalization. Moreover, the weights $\pi_2(E^{\{i,j\}})$ are all equal. Thus, using Proposition 30 we can bound the second eigenvalue of this graph by

$$\begin{aligned} 0 + \max_{i \in S} \mathop{\mathbb{E}}_{j \in S \setminus \{i\}} [\lambda_{ij}] &\leqslant \max_{i \in [d-1]} \mathop{\mathbb{E}}_{j \in S \setminus \{i\}} [\lambda_{ij}] \\ &\leqslant \frac{1}{(|S|-1)} \max_{i \in [d-1]} \sum_{j \in S \setminus \{i\}} \lambda_{ij} \\ &\leqslant \frac{1}{(|S|-1)} \max_{i \in [d-1]} \sum_{j \in [d-1] \setminus \{i\}} \lambda_{ij} \end{aligned}$$

Using the bounds on random walks on the projections of flags complex stated in Fact 31 we get the final bound of

$$\begin{aligned} \frac{1}{(|S|-1)} \max_{i \in [d-1]} \sum_{j \in [d-1] \setminus \{i\}} \lambda_{ij} &\leqslant \frac{1}{(|S|-1)} \max_{i \in [d-1]} \left(\sum_{j=1}^{i-1} q^{-(i-j)/2} + \sum_{j=i+1}^{d-1} q^{-(j-i)/2} \right) \\ &\leqslant \frac{1}{(|S|-1)} \max_{i \in [d-1]} \frac{(1 - q^{-(i-1)/2}) + (1 - q^{-(d-i-1)/2})}{\sqrt{q} - 1} \\ &\leqslant \frac{2}{(|S|-1)(\sqrt{q}-1)}. \end{aligned}$$

Now, consider a face $(W_{i_1} \subseteq W_{i_2} \subseteq \ldots \subseteq W_{i_\ell})$ of $\mathfrak{G}^S(\ell - 1)$. Notice that the link of this face is isomorphic to the join of projected complexes of form $\mathfrak{G}^{i_j - i_{j-1}, S_j}$ where $S_j \subseteq [i_j - i_{j-1}]$ with $i_0 = 0$. As before, the underlying graph is some $(k-\ell)$-partite graph G with each $G^{\{p,q\}}$ being either the underlying graph of $\mathfrak{G}^{i_j - i_{j-1}, S_j}$ or a complete bipartite graph. Again, similarly applying Proposition 30, Fact 31, and the fact that the second eigenvalue of a complete bipartite graph is 0 gives us the required bound of $\frac{2}{(|S|-\ell-1)(\sqrt{q}-1)}$. This observation about the structure of the links, Fact 31, and the fact that the second eigenvalue of a complete bipartite graph is 0 also lets us conclude that the complex is a $\frac{1}{\sqrt{q}}$-product. ◀

3.2 Coboundary Expansion

In this section we prove Theorem 4, that central projections of the Flags complex and their links have good 1-coboundary expansion over any group.

▶ **Theorem 4** (Coboundary Expansion of Projected Flags Complex)**.** *There exists a constant q_0 such that the following holds. Let Γ be any group. Let $d > k \geqslant 2$ be two integers and $q \geqslant q_0$ be a prime power. Let*

$$S_k = \{d-k, \ldots, d-1, d, d+1, \ldots, d+k\}.$$

Then, for any face $s \in \mathfrak{G}_q^{2d, S_k}$ with codimension at least 3, $\mathfrak{G}_{q,s}^{2d,S_k}$ is 1-coboundary expander with

$$h^1(\mathfrak{G}_{q,s}^{2d,S_k}, \Gamma) = \Omega(1).$$

Broadly speaking, the proof of Theorem 4 has two main steps, vanishing cohomology (Lemma 32) and co-cycle expansion (Lemma 44 + Lemma 47). We start with the former, which we prove via the theory of shellability.

3.2.1 Shellability of the Flags Complex

In this section, we prove projections of the flags complex have trivial cohomology over any group Γ.

▶ **Lemma 32.** *Fix any $d \geqslant 3$, any prime power q, any group Γ, and a set $S \subseteq [d-1]$ of size at least 2. Let $s \in \mathfrak{G}_q^{d,S}$ be any face of codimension at least 3. Then, the 1-cohomology of $\mathfrak{G}_{q,s}^{d,S}$ with respect to Γ is trivial.*

Shellability and Vanishing Cohomology. To prove Lemma 32, it is enough to show projections of the Flags complex are *shellable*, a condition roughly stating that the complex can be built up one (top-level) face at a time in such a way that each new face is glued to the prior faces along co-dimension 1 facets. It is a standard fact that shellable complexes have vanishing 1-cohomology, since they are homotopy equivalent to a wedge of spheres [16]. Here, we give a simple, self-contained proof of vanishing cohomology that does not rely on the latter machinery.

First, we formalize the standard notion of shellability.

▶ **Definition 33** (Topological Shellability)**.** *Suppose that X is a simplicial complex. A total ordering $\sigma_1, \sigma_2, \ldots, \sigma_k$ of maximal faces of X is a* topological shelling *of X if*

$$\dim(\sigma_1) \geqslant \dim(\sigma_2) \geqslant \ldots \geqslant \dim(\sigma_k),$$

and for any $j > 1$,

$$\left(\bigcup_{i<j} \sigma_i\right) \cap \sigma_j$$

is a pure $(\dim(\sigma_j) - 1)$-dimensional complex.[4] *A complex X is called* shellable *whenever a topological shelling exists.*

In other words, a complex is shellable if it may be constructed by sequentially gluing top level faces along a (union of) co-dimension 1 facets. As discussed, it is a standard fact that shellable complexes have vanishing 1-cohomology over every group (see [18, Proposition B.44], [52, Corollary 2.11], and [18, Proposition B.42]). We give an elementary proof of this fact by inductively constructing a contraction for any loop along the shelling order, where the inductive step is essentially a simple special case of Van Kampen's Theorem we handle directly.

▶ **Theorem 34.** *If X is a shellable complex of dimension at least 2, its 1-cohomology vanishes with respect to every group Γ.*

Proof. It is enough to prove any finite loop in the 1-skeleton of X can be contracted via a series of triangle and backtracking relations. We show this inductively. Given a shelling order $\sigma_1 \ldots \sigma_k$, define

$$X_j \coloneqq \bigcup_{i \leqslant j} \sigma_j$$

[4] Note the union denotes the simplicial complex given by taking all the σ_i and their downward closures.

Observe $X_1 = \sigma_1$, and $X_k = X$. We will prove by induction that every loop in X_i can be contracted via a series of triangle and backtracking relations. The base case is immediate from the fact that X_1 is a simplex (complete complex), for which it is a standard exercise to construct such a contraction.

Assume the inductive hypothesis holds for X_{j-1}, and let γ be a loop in X_j. Assume further that the loop γ does not lie entirely inside σ_j, else a contraction exists within the full simplex σ_j itself. We will prove in this case it is always possible to contract the loop γ to a loop entirely contained within X_{j-1}, at which point we may apply the inductive hypothesis.

Assume γ is not already entirely inside X_{j-1} (else we are done). Let $v \in \gamma$ be an offending vertex in $\sigma_j \setminus X_{j-1}$, and let $(v_i, \ldots, v_j)$ be the largest arc of γ containing v such that the entire arc lies inside σ_j. We will rely on the following two useful facts:

1. v_i and v_j are in $X_{j-1} \cap \sigma_j$, and
2. The 1-skeleton of $X_{j-1} \cap \sigma_j$ is connected.

Let's complete the proof assuming these facts. Since $X_{j-1} \cap \sigma_j$ is connected, we may find a path between v_i and v_j entirely inside $X_{j-1} \cap \sigma_j$. Concatenating this path with the arc (v_i, v_j) gives a loop entirely inside σ_j, so in particular, the arc can be contracted to this path by triangle and backtracking relations. Repeating this procedure until there are no further loop elements outside X_{j-1} completes the proof.[5]

It is left to prove the two facts. The latter is immediate from shellability, which promises $X_{j-1} \cap \sigma_j$ is a pure $(d-1)$-dimensional complex on at most $(d+1)$ vertices. Such complexes are automatically connected (indeed have diameter at most 2), since for any vertices w, w' in the complex, either (w, w') are already in a shared face, or the top-level faces σ_w and $\sigma_{w'}$ containing them respectively share a joint vertex v, so the path (w, v, w') exists on the 1-skeleton as desired.

Toward the former, observe that since we may assume $(v_i, \ldots, v_j)$ is not the entire loop (else γ is fully contained in σ_j), the arc has *boundary point(s)* v_{i-1} and v_{i+1} which are in $X_{j-1} \setminus \sigma_j$. Further, since (v_{i-1}, v_i) and (v_j, v_{j+1}) are edges in X_j, observe by definition they must come from some face σ_i for $i \leqslant j$ in the shelling order. Since v_{i-1} and v_{j+1} are not in σ_j, the edges cannot come from σ_j, so in particular must come from a prior σ_i included in X_{j-1}, so v_i and v_j are also in X_{j-1} as desired. ◀

Posets and Shellability of Projected Flags. It is left to prove that projections of the flags complex are shellable. To do so, we will rely on the fact that the flags complex arises from taking maximal chains of a graded poset, i.e., it is the so-called "order complex" of the Grassmann poset. Such complexes have a simple combinatorial characterization of shellability. Moreover, if the poset is ranked, the notion is closed under projecting to a subset of ranks, exactly our projecting operation.

We need a few background definitions concerning posets. A poset P is a set P endowed with a relation $\preceq_P$ that is (i) reflexive, (ii) anti-symmetric, and (iii) transitive. Whenever it is clear we drop the subscript P and use $\preceq$. Also, if $a \neq b$ and $a \preceq b$, we write $a \prec b$.

Graded Poset. A graded poset additionally has a rank function $\rho : P \to \mathbb{N}$ that has the following properties.

(i) If $x \prec y$, then $\rho(x) < \rho(y)$.
(ii) If $x \prec y$ and there is no element z such that $x \prec z \prec y$, then $\rho(y) = \rho(x) + 1$.

[5] Note the process indeed terminates by an elementary potential argument measuring the number of loop vertices in $\sigma_j \setminus X_{j-1}$ since 1) the loop is finite and 2) in every round we strictly decrease the number of elements in $\sigma_j \setminus X_{j-1}$.

We stress that such a rank function may not exist for every poset. We assume all posets we work with this in this section are graded.

Joins. Given two posets P_1 and P_2, the *join* $P_1 * P_2$ is the poset on the set $P_1 \sqcup P_2$ with $\leqslant_{P_1 * P_2}$ being such that if $a \leqslant_{P_1 * P_2} b$, then either (i) $a \preceq_{P_1} b$, or (ii) $a \preceq_{P_2} b$, or (iii)$a \in P_1$ and $b \in P_2$. Extend this definition inductively to define joins of multiple posets.

▶ **Definition 35** (Order Complex). *Let P be a poset. The* order complex X_P *of the poset P is a simplicial complex consisting of chains of the poset. In other words, whenever $a_1 \preceq_P a_2 \preceq_P \cdots \preceq_P a_k$ we have $\{a_1, a_2, \ldots, a_k\} \in X_P$.*

▶ **Definition 36** (Grassmann Poset). *Given $d \in \mathbb{N}$ and prime power q, the* Grassmann poset GP_q^d *is the poset whose elements are the subspaces of $\mathbb{F}_q^d$ and $V_1 \preceq_{\mathrm{GP}_q^d} V_2$ whenever $V_1 \subseteq V_2$.*

Thus the flags complex is exactly the order complex of the Grassmann poset. We now move to defining shellability of posets.

▶ **Definition 37** (Shellability of Posets). *Suppose $(P, \preceq)$ is a poset. Moreover, assume that P is* pure*, i.e., all its maximal chains are of the same length. If its maximal chains can be ordered as $m_1, m_2, \ldots, m_t$ so that for every $1 \leqslant i < j \leqslant t$, there exists a $1 \leqslant k < j$ and $x \in m_j$ such that*

$$m_i \cap m_j \subseteq m_k \cap m_j = m_j \setminus \{x\},$$

then we say P is a (pure) shellable poset*. Moreover, such an ordering is called a* shelling order*.*

The following two lemmas, which state respectively that 1) projections of shellable posets are shellable, and 2) the order complex of a shellable poset is topologically shellable, are simple exercises and can be found in [18].

▶ **Lemma 38** ([18, Theorem C.9]). *Suppose $(P, \preceq, \rho)$ is a pure ranked poset and r is the rank of the maximal elements. If P is shellable, then for any $S \subseteq [r]$, $P_S = \{x \in P | \rho(x) \in S\}$ is a shellable poset.*

▶ **Lemma 39** ([18, Proposition C.6]). *Let P be a shellable poset. Then, the order complex X_P is pure and shellable.*

Thus it suffices for us to prove the Grassmann poset is shellable. This fact is fairly standard, and can be considered a special case of [16, Theorem 3.7]. We give a self-contained direct proof here adapted from the proof of shellability of the Boolean poset in [18, Proposition C.8].

▶ **Lemma 40.** *For d any positive integer and q a prime power, the Grassmann poset GP_q^d is shellable.*

Proof. The idea is to label each maximal chain by an ordered basis $(v_1, \ldots, v_d)$ of $\mathbb{F}_q^d$ with $W_i = \mathrm{span}\{v_1, \ldots, v_i\}$ for all i, taking each v_i to be the smallest vector in $W_i \setminus W_{i-1}$ under a fixed total order on $\mathbb{F}_q^d$. We show lexicographically ordering chains by these labels gives a shelling. We formalize the argument below.

Labelling maximal chains. Fix an arbitrary total order on $\mathbb{F}_q^d$. To each maximal chain

$$m\colon \quad 0 = W_0 \subseteq W_1 \subseteq \cdots \subseteq W_d = \mathbb{F}_q^d,$$

associate the label-vector $\lambda(m) = (v_1, \ldots, v_d)$, where v_i is the smallest vector in $W_i \setminus W_{i-1}$ under our ordering. Note by definition we have $W_i = \operatorname{span}\{v_1, \ldots, v_i\}$.

We also extend the labeling to maximal chains lying *between* two subspaces $M \subseteq N$.[6] In particular, given $M \subseteq N$ and a maximal chain $c\colon M = W_0 \subseteq \cdots \subseteq W_k = N$, let $\lambda(c) = (v_1, \ldots, v_k)$ where v_i is the smallest vector in $W_i \setminus W_{i-1}$, so that $W_i = W_0 \oplus \operatorname{span}\{v_1, \ldots, v_i\}$.

The proposed shelling order. Order all maximal chains of GP_q^d as $m_1, m_2, \ldots$ in lexicographic order of their labels:

$$\lambda(m_1) < \lambda(m_2) < \lambda(m_3) < \cdots .$$

We claim this is a shelling. Fix $i < j$; we must produce some $k < j$ together with a vertex $V_t \in m_j$ such that

$$m_j \cap m_k \;=\; m_j \setminus \{V_t\} \quad \text{and} \quad m_j \cap m_i \;\subseteq\; m_j \cap m_k.$$

Setup. Write the two chains as

$$m_i\colon \quad 0 = U_0 \subseteq U_1 \subseteq \cdots \subseteq U_d = \mathbb{F}_q^d,$$
$$m_j\colon \quad 0 = V_0 \subseteq V_1 \subseteq \cdots \subseteq V_d = \mathbb{F}_q^d,$$

and let $r \geqslant 0$ be the largest index such that $V_i = U_i$ for all $i \leqslant r$, and $s > r$ be the smallest index such that we again have $V_s = U_s$. Abusing notation, we write $\lambda(V_a)$ for the a-th entry of $\lambda(m_j)$.

Reduction to finding a descent. We claim it suffices to produce an index $r < t < s$ at which the labels of m_j *descend*, i.e., $\lambda(V_t) > \lambda(V_{t+1})$. Suppose we have such a t, and write $\beta = \lambda(V_{t+1})$. Set

$$V \;=\; V_{t-1} + \operatorname{span}\{\beta\}.$$

Since $\beta \in V_{t+1} \setminus V_{t-1}$, V is a subspace strictly between V_{t-1} and V_{t+1} and distinct from V_t, so replacing V_t with V in m_j produces a maximal chain

$$m_k\colon \quad V_0 \subseteq \cdots \subseteq V_{t-1} \subseteq V \subseteq V_{t+1} \subseteq \cdots \subseteq V_d.$$

We now argue that

1. m_k comes before m_j in the order, and
2. $m_j \cap m_i \subseteq m_j \setminus \{V_t\} = m_j \cap m_k$.

We start with the former. Because $\beta \in V \setminus V_{t-1}$, the t-th entry of $\lambda(m_k)$ is at most β. On the other hand, $\beta = \lambda(V_{t+1}) < \lambda(V_t)$ since t is our descent. Since $\lambda(m_k)$ and $\lambda(m_j)$ agree in positions $1, \ldots, t-1$, we therefore have $\lambda(m_k) < \lambda(m_j)$ lexicographically as desired.

Towards the latter, first observe that by construction $m_j \cap m_k = m_j \setminus \{V_t\}$. As $r < t < s$, the chains m_i and m_j disagree on V_t. Thus, we have $V_t \notin m_i$ and $m_j \cap m_i \subseteq m_j \setminus \{V_t\} = m_j \cap m_k$ as desired.

---

[6] A chain from A to B is *maximal between them* if no element $A \preceq C \preceq B$ can be added.

Existence of a descent. It remains to produce such a t. This is a consequence of the following fact.

▷ Claim 41. Let $M \subseteq N$ be two subspaces in this poset. Then, there is a unique maximal chain c starting at M and ending at N with increasing label-vector $\lambda(c)$. Moreover, $\lambda(c)$ is lexicogaphically minimal among labelings of maximal chains between M and N.

Namely, consider the sub-chains of m_i and m_j between the matching endpoints $V_r = U_r$ and $V_s = U_s$. Because $\lambda(m_i) < \lambda(m_j)$ and the two chains first disagree at step $r+1$, the labels of the m_i-sub-chain are lexicographically smaller than those of the m_j-sub-chain. In particular, the m_j-sub-chain is *not* the lexicographical minimum, and so its label-vector is not strictly increasing and there must exist a descent. It is left to prove the claim.

Proof of Claim 41. We start with uniqueness. First note that a maximal chain $M = W_0 \subseteq W_1 \subseteq \cdots \subseteq W_k = N$ starting at M and ending at N with increasing label-vector can be constructed by taking $W_0 = M$ and then at each step i adding the smallest vector $w_i \in N \setminus W_{i-1}$ to W_{i-1} to get $W_i = W_{i-1} \oplus \mathsf{span}\{w_i\}$. This implies $(w_1 < w_2 < \cdots < w_k)$ is the label-vector for the constructed chain.

Towards a contradiction, assume that there are two (distinct) maximal chains c, c' starting at M and ending at N both with increasing label vector. Suppose the chains c, c' are given by

$$M = W_0 \subseteq W_1 \subseteq \cdots \subseteq W_k = N,$$
$$M = W'_0 \subseteq W'_1 \subseteq \cdots \subseteq W'_k = N,$$

respectively, and the label-vectors are given by $(w_1, w_2, \ldots, w_k)$ and $(w'_1, w'_2, \ldots, w'_k)$, respectively. Suppose i is the first index such that $w_i \neq w'_i$ and without loss of generality assume that $w_i < w'_i$. We must have $w_i \notin W'_i$; otherwise, $W_i = W'_i$ which implies w_i is the correct vector to associate with $W'_i \setminus W'_{i-1}$. This also means that the w_i must be contained in some W'_j with $k \geqslant j > i$. So, the corresponding label $w'_j \leqslant w_i$. On the other hand, since $(w'_1, w'_2, \ldots, w'_k)$ is increasing we must have $w'_j > w'_i > w_i$, which gives us a contradiction.

To see $\lambda(c)$ is minimal, consider another chain $\tilde{c}$ given by

$$M = S_0 \subseteq S_1 \subseteq \cdots \subseteq S_k = N$$

with label-vector $(s_1, \ldots, s_k)$. If $s_1 > w_1$, then we are done. Otherwise, let i be the smallest index such that $s_i \neq w_i$ but $s_{i-1} = w_{i-1}$. Then, as w_i is the smallest vector in $N \setminus W_{i-1} = S_{i-1}$, we must have $s_i > w_i$. ◁

◀

We already have everything we need to conclude that the cohomology of the projected flags complex is trivial. Furthermore, we can use the following observations to conclude that every link of the the projected flags complex also has trivial cohomology.

The following observation trivially follows from the definitions of join of simplicial complexes and join of posets.

▶ **Observation 42.** *Let P_1, P_2 be two posets. Then, we have $X_{P_1 * P_2} = X_{P_1} * X_{P_2}$.*

▶ **Observation 43.** *Let P_1 and P_2 be shellable posets. Then, the join $P_1 * P_2$ is also shellable.*

Proof. Let $m_1, m_2, \ldots, m_t$ be the maximal chains of P_1 in a shelling order. Let $m'_1, m'_2, \ldots, m'_{t'}$ be the maximal chains of P_2 in a shelling order. Then, it is easy to verify that the following is the sequence of maximal chains of $P_1 * P_2$ in a shelling order:

$$m_1 \circ m'_1, m_2 \circ m'_1, \ldots, m_t \circ m'_1, m_1 \circ m'_2, m_2 \circ m'_2, \ldots, m_t \circ m'_2, \ldots\ldots, m_1 \circ m'_{t'}, m_2 \circ m'_{t'}, \ldots, m_t \circ m'_{t'},$$

where $\circ$ is used to denote concatenation. ◀

Since links of the flags complex are joins of flags complexes in lower dimension, Lemma 32 immediately follows.

3.2.2 Cocycle Expansion of the Projected Flags Complex

With vanishing co-homology out of the way, it is enough to prove the complexes are also good 1-cocycle expanders. Toward this end, we'll rely on some useful techniques from [23] for coboundary and cocycle expansion of a complex with respect to a general (possibly non-abelian) group. The first allows us to derive cocycle expansion from spectral and coboundary expansion of the links of X.

▶ **Lemma 44** ([23]). *Let $\beta, \lambda > 0$ and let $k > 0$ be an integer. Let X be a d-dimensional simplicial complex for $d \geqslant k+2$ and assume that X is a λ-one-sided local spectral expander. Let Γ be any group. Assume that for every non-empty $r \in X$, X_r is a coboundary expander and that $h^{k+1-|r|}(X_r, \Gamma) \geqslant \beta$. Then*

$$h^k(X, \Gamma) \geqslant \frac{\beta^{k+1}}{(k+1)! \cdot 4} - e\lambda.$$

Here, $e \approx 2.71$ is the Euler's number.

The second tool is the "color restriction lemma" which states that if most projections of a complex are coboundary expanders, then the complex itself is a coboundary expander.

▶ **Lemma 45** ([23]). *Let ℓ, d be integers so that $3 \leqslant \ell \leqslant d$ and let $\beta, p \in (0, 1]$. Let Γ be some group. Let X be a $(d+1)$-partite simplicial complex so that*

$$\mathbb{P}_{I \in \binom{[d+1]}{\ell}} \left[X^I \text{ has 1-coboundary expansion } \beta \text{ and } \forall s \in X(0), X^I_s \text{ is a } \frac{1}{2}\text{-local spectral expander} \right] \geqslant p.$$

Then X is a coboundary expander with $h^1(X) \geqslant \frac{p\beta}{36}$.

To take advantage of Lemma 45, we start by showing any projection of the flags complex onto a "well-separated" set of three colors has constant coboundary expansion by directly constructing a family of cones.

▶ **Lemma 46.** *Fix any integer $d \geqslant 3$, any prime power q, and Γ any group. Let $I = \{i_0 < i_1 < i_2\} \subseteq [d-1]$. Let $s \in \mathfrak{G}_q^{d,[d-1]\setminus I}$. Assume that*

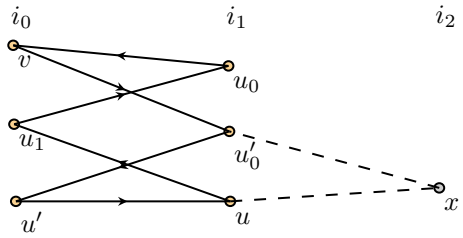
1. *$2i_0 \leqslant i_1$ or there is a subspace $V \in s$ with $i_0 < \mathtt{col}(V) < i_1$, and*
2. *either $2i_1 - i_0 \leqslant i_2$, or $2i_1 - \dim(V) \leqslant i_2$ where $V \in s$ is a subspace with $i_0 < \mathtt{col}(V) < i_1$, or there is a subspace $W \in s$ with $i_1 < \mathtt{col}(W) < i_2$.*

Then, the complex $\mathfrak{G}^{d,I}_{q,s}$ has 1-coboundary expansion at least

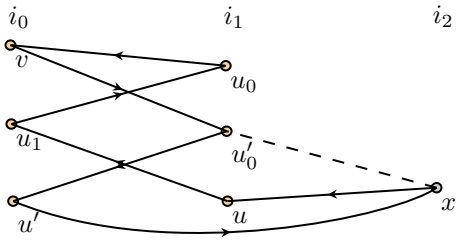
$$h^1\left(\mathfrak{G}^{d,I}_{q,s}, \Gamma\right) \geqslant \frac{1}{13}. \tag{1}$$

Proof. Let $\mathfrak{G} = \mathfrak{G}^d_q$. It is elementary to see the group of automorphisms $\mathrm{Aut}(\mathfrak{G}^I_s)$ acts transitively on the triangles of $\mathfrak{G}^I_s$ so by Lemma 24 it is sufficient to construct a cone of diameter 13.[7]

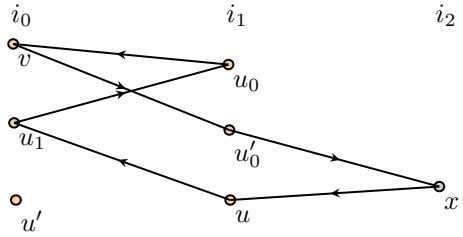
[7] Namely taking an appropriate basis for the vector spaces in any two such flags, we can transform one to the other with the corresponding basis exchange matrix.



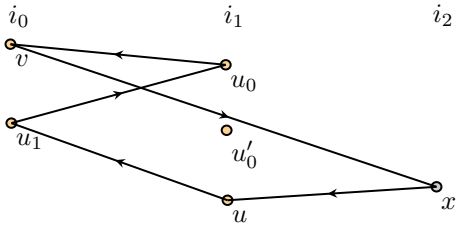
(a) $P_0 = P_u \circ (u', u) \circ P_{u'}^{-1} = vu_0'u'uu_1u_0v$.



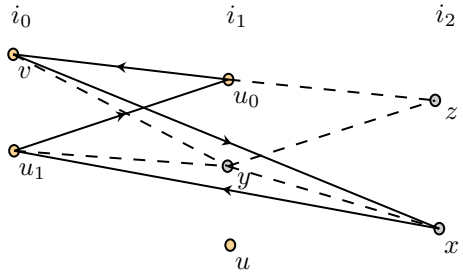
(b) $P_1 = vu_0'u'xuu_1u_0v$.



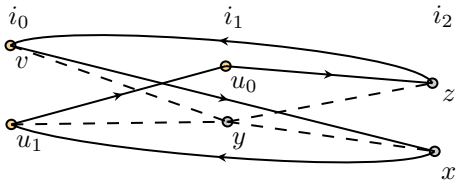
(c) $P_2 = vu_0'xuu_1u_0v$.



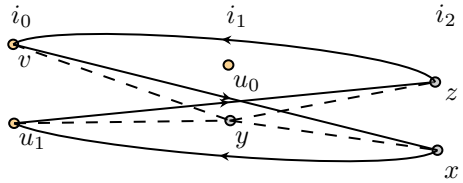
(d) $P_3 = vxuu_1u_0v$.



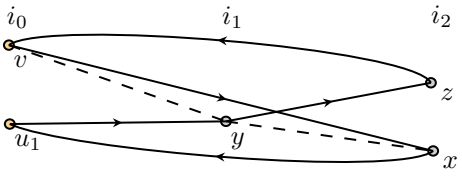
(e) $P_4 = vxu_1u_0v$.



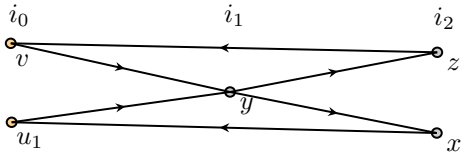
(f) $P_5 = vxu_1u_0zv$.



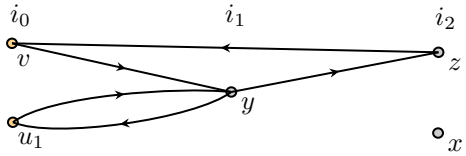
(g) $P_6 = vxu_1zv$.



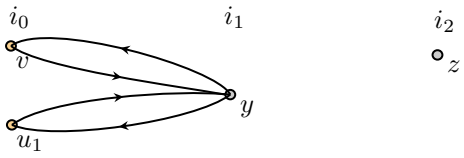
(h) $P_7 = vxu_1yzv$.



(i) $P_8 = vyxu_1yzv$.

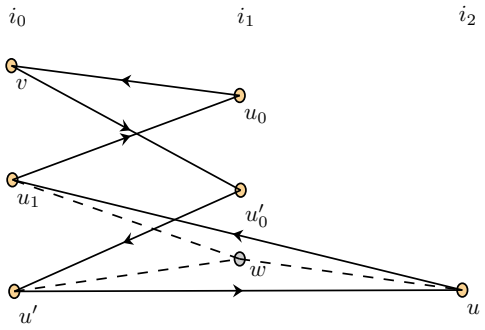


(j) $P_9 = vyu_1yzv$.

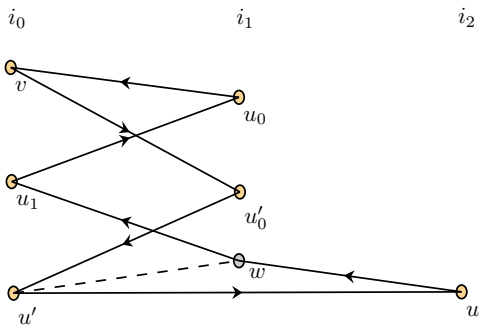


(k) $P_{10} = vyu_1yv$.

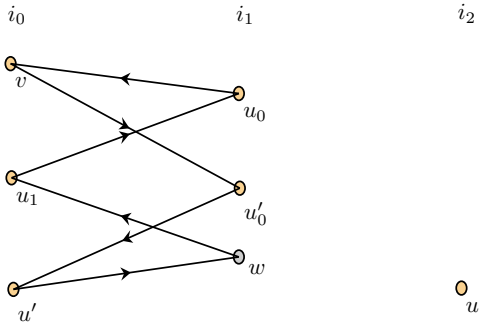
Figure 1 contraction $T_{u'u}$ when $u \in \mathfrak{G}^{\{i_1\}}$ and $u' \in \mathfrak{G}_s^{\{i_0\}}$.



(a) $P_0 = P_u \circ (u', u) \circ P_{u'}^{-1} = vu'_0u'uu_1u_0v$.

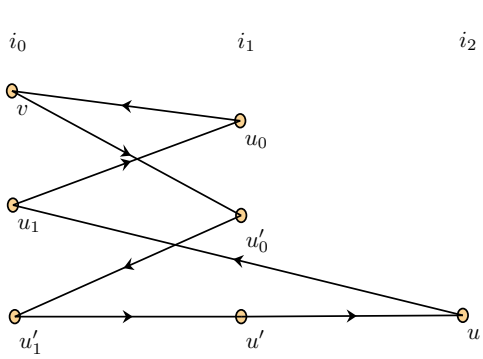


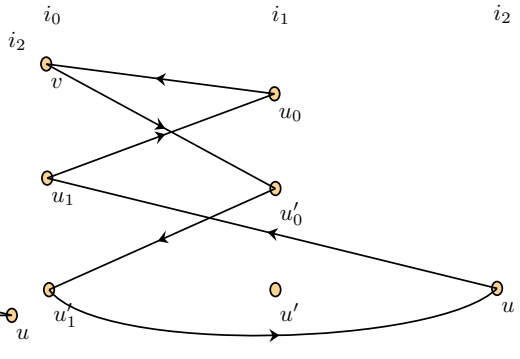
(b) $P_1 = vu'_0u'wu_1u_0v$.



(c) $P_2 = vu'_0u'wu_1u_0v$.

Figure 2 contraction $T_{u'u}$ when $u \in \mathfrak{G}^{\{i_2\}}$ and $u' \in \mathfrak{G}_s^{\{i_0\}}$ (reduction to $u \in \mathfrak{G}^{\{i_1\}}$ and $u' \in \mathfrak{G}_s^{\{i_0\}}$).





(a) $P_0 = P_u \circ (u', u) \circ P_{u'}^{-1} = vu'_0u'_1u'uu_1u_0v$. **(b)** $P_1 = vu'_0u'_1uwu_1u_0v$.

Figure 3 contraction $T_{u'u}$ when $u \in \mathfrak{G}^{\{i_2\}}$ and $u' \in \mathfrak{G}_s^{\{i_1\}}$ (reduction to $u \in \mathfrak{G}^{\{i_2\}}$ and $u' \in \mathfrak{G}_s^{\{i_0\}}$).

Fix an arbitrary $v \in \mathfrak{G}_s^{\{i_0\}}$. For any vertex $u \in \mathfrak{G}_s^{\{i_0\}} \setminus \{v\}$, we take $P_u = vu_0u$, where u_0 is an arbitrary vertex in $\mathfrak{G}_s^{\{i_1\}}$ such that $u_0 \supseteq v + u$. For any vertex $u \in \mathfrak{G}_s^{\{i_1,i_2\}}$, take $P_u = P_{u_1} \circ (u_1, u)$, where u_1 is an arbitrary vertex in $\mathfrak{G}_s^{\{i_0\}} \setminus \{v\}$ such that $u \supseteq u_1$.

We now describe the contraction $T_{u'u}$ for each edge $u'u$. First, suppose $u \in \mathfrak{G}^{\{i_1\}}$ and $u' \in \mathfrak{G}_s^{\{i_0\}}$. Then the first walk in the contraction is given by

$$P_0 = P_u \circ (u', u) \circ P_{u'}^{-1} = vu_0'u'uu_1u_0v.$$

As either $u_0' \cap u \supseteq u'$ and $i_2 \geqslant 2i_1 - i_0$, or $u_0' \cap u \supseteq V$ and $i_2 \geqslant 2i_1 - \dim(V)$, or $W \supseteq u_0', u$ there exists a vertex $x \in \mathfrak{G}_s^{\{i_2\}}$ such that $x \supseteq u + u_0'$. Now we get the following set of $\sim_1$ relations.

1. $P_0 \sim_1 P_1 = vu_0'u'xuu_1u_0v$ using the loop uxu and then the triangle $u'xu$.
2. $P_1 \sim_1 P_2 = vu_0'xuu_1u_0v$ using the triangle $u_0'u'x$.
3. $P_2 \sim_1 P_3 = vxuu_1u_0v$ using the triangle $vu_0'x$.
4. $P_3 \sim_1 P_4 = vxu_1u_0v$ using the triangle xuu_1.

Observe that there exists $y \in \mathfrak{G}_s^{\{i_1\}}$ such that $v + u_1 \subseteq y \subseteq x$ as $2i_0 \leqslant i_1$ or $v + u_1 \subseteq V$. Also there is a $z \in \mathfrak{G}_s^{\{i_2\}}$ such that $z \supseteq y + u_0$ since $y \cap u_0 \supseteq v$ or $y + u_0 \subseteq W$. This gives us another sequence of $\sim_1$ relations.

5. $P_4 \sim_1 P_5 = vxu_1u_0zv$ using the triangle u_0zv.
6. $P_5 \sim_1 P_6 = vxu_1zv$ using the triangle u_1u_0z.
7. $P_6 \sim_1 P_7 = vxu_1yzv$ using the triangle u_1yz.
8. $P_7 \sim_1 P_8 = vyxu_1yzv$ using the triangle vyx.
9. $P_8 \sim_1 P_9 = vyu_1yzv$ using the triangle yxu_1.
10. $P_9 \sim_1 P_{10} = vyu_1yv$ using the triangle yzv.

This completes the contraction $T_{u'u}$ as P_{10} is just two loops. Now, we reduce the task of finding contraction in the other two cases to the case above. Suppose $u \in \mathfrak{G}_s^{\{i_2\}}$ and $u' \in \mathfrak{G}_s^{\{i_0\}}$. Then, we start with $P_0 = vu_0'u'uu_1u_0v$. Note that there exists a $w \in \mathfrak{G}_s^{\{i_1\}}$ such that $u \supseteq w \supseteq u' + u_1$. Observe that $P_0 \sim_1 P_1 = vu_0'u'uwu_1u_0v$ using the triangle uwu_1, and $P_1 \sim_1 P_2 = vu_0'u'wu_1u_0v$ using the triangle $u'wu$ and then the loop wuw. This is precisely the walk at the start of the first case (except for the labeling of vertices). Finally, assume $u \in \mathfrak{G}^{\{i_2\}}$ and $u' \in \mathfrak{G}_s^{\{i_1\}}$. Then, we start with $P_0 = vu_0'u_1'u'uu_1u_0v$. We reduce it to the second case by observing $P_0 \sim_1 P_1 = vu_0'u_1'uu_1u_0v$ using the triangle $u_1'u'u$. ◀

While we cannot use Lemma 46 directly to show the projected flags complex is a coboundary expander (since *no* 3-color projection there-in is well-separated), we *can* use it to show the complex's *links* are good coboundary expanders. We may then combine this with the local-to-global lemma Lemma 44 to complete the proof.

▶ **Lemma 47.** *Fix integers $d > k \geqslant 2$, and a prime power $q \geqslant 25$. Let*

$$S_k = \{d - k, \dots, d - 1, d, d + 1, \dots, d + k\}$$

and let $s \in \mathfrak{G}_q^{2d,S_k}$ be a non-empty face in the projected building $\mathfrak{G}_q^{2d,S_k}$ with co-dimension at least 3. Then, the link $\mathfrak{G}_{q,s}^{2d,S_k}$ has 1-coboundary expansion

$$h^1(\mathfrak{G}_{q,s}^{2d,S_k}, \Gamma) = \Omega(1).$$

Proof. Let $c^* = \texttt{col}(s) \subseteq S_k$ and write $c^* = \{j_1, \dots, j_b\}$. Let $S' = S_k \setminus \texttt{col}(s)$. Now, we want to show that for a constant fraction of $I = \{i_0 < i_1 < i_2\} \in \binom{S'}{3}$, $\mathfrak{G}_{q,s}^{2d,I}$ has 1-coboundary expansion at least β for some constant $\beta > 0$. The proof is then complete using Lemma 45, as by Theorem 3 we already know that $\mathfrak{G}_{q,s}^{2d,I'}$ is $\frac{1}{2}$-local spectral expander for any I' for sufficiently large q. Pick a uniformly random set $I = \{i_0 < i_1 < i_2\} \in \binom{S'}{3}$.

Define $b+1$ *bins*

$$B_i = \{j \in S_k | j_i < j < j_{i+1}\}$$

for $0 < i < b$ and also $B_0 = \{j \in S_k | j < j_1\}$ and $B_b = \{j \in S_k | j > j_b\}$. We divide the proof into two cases based on the size of the s

Case 1: Small s. Assume $|s| \leqslant 2k+1-80$. In this case, we may assume there is a bin B_i with $|B_i| > \frac{2k+1-|s|}{4} \geqslant 20$, as otherwise a union bound implies the probability our three random indices lie in separate bins is at least

$$1 - \frac{(2k+1-|s|)/4}{2k+1-|s|} - \frac{2(2k+1-|s|)/4}{2k+1-|s|} \geqslant \frac{1}{4},$$

so $h^1(\mathfrak{G}_{q,s}^{2d,I}, \Gamma) \geqslant \frac{1}{13}$ by Lemma 46 and $h^1(\mathfrak{G}_{q,s}^{2d,S_k}, \Gamma) \geqslant \frac{1}{1872}$. We may also assume $j_i \neq 0$; otherwise, observe that any projected building is isomorphic to the projected building formed by replacing each subspace with its orthogonal complement, where $j_i = b$ instead. Since $|B_i| > \frac{2k+1-|s|}{4}$, with probability at least

$$\frac{|B_i|/20}{2k+1-|s|} \cdot \frac{|B_i|/5}{2k+1-|s|} \cdot \frac{|B_i|/4}{2k+1-|s|} \geqslant \frac{1}{25600}$$

i_0 is in the second twentieth of the indices B_i, i_1 is in the second fifth of the indices B_i, and i_2 is in the final quarter of the indices B_i. In this case, the link is isomorphic to $\mathfrak{G}_q^{j_{i+1}-j_i, \{i_0-j_i, i_1-j_i, i_2-j_i\}}$ where $\{i_0-j_i, i_1-j_i, i_2-j_i\}$ satisfy the spreadness conditions of Lemma 46. Thus $h^1(\mathfrak{G}_{q,s}^{2d,I}, \Gamma) \geqslant \frac{1}{13}$ with probability at least $\frac{1}{25600}$ and $h^1(\mathfrak{G}_{q,s}^{2d,S_k}, \Gamma) \geqslant 10^{-8}$ as desired.

Case 2: Large s. When $|s| > 2k+1-80$, it suffices to show there exists a single choice of I with constant coboundary expansion, since each valid I appears with probability at least $\frac{1}{\binom{2k+1-|s|}{3}} = \Omega(1)$. We now show such a choice of I always exists. We break into cases based on maximum bin size

1. Suppose we have a bin B_i with at least three colors. As before, by taking orthogonal complements if necessary, we may assume $i \neq 0$ (similarly for the cases below). We may then take $I = \{j_i+1, j_i+2, j_i+3\}$. As $\mathfrak{G}_{q,s}^{2d,I}$ isomorphic to $\mathfrak{G}_q^{j_{i+1}-j_i, \{1,2,3\}}$, Lemma 46 implies $h^1(\mathfrak{G}_{q,s}^{2d,I}, \Gamma) \geqslant \frac{1}{13}$.
2. Suppose we have a bin B_i with two colors and there is a non-empty bin $B_{i'}$ where $i' > i > 0$. Then, taking $I = \{j_i+1, j_i+2, j_{i'}+1\}$ we get that $\mathfrak{G}_{q,s}^{2d,I}$ is isomorphic to $\mathfrak{G}_{q,s'}^{j_{i'+1}-j_i, \{1,2,j_{i'}-j_i+1\}}$ for the appropriate $s' \subset s$ between these indices. In particular, s' contains a subspace of dimension $j_{i'} - j_i$, so $\mathfrak{G}_{q,s}^{2d,I}$ has constant coboundary expansion by Lemma 46. If we instead have B_i with one color and a bin $B_{i'}$ with two colors where $i' > i$, the proof is analogous.
3. Finally, we may have three non-empty bins $B_i, B_{i'}, B_{i''}$, where $i'' > i' > i$. Taking $I = \{j_i+1, j_{i'}+1, j_{i''}+1\}$ we get that $\mathfrak{G}_{q,s}^{2d,I}$ is isomorphic to $\mathfrak{G}_{q,s'}^{j_{i''+1}-j_i, \{1, j_{i'}-j_i+1, j_{i''}-j_i+1\}}$ where $s' \subset s$ contains subspaces of dimensions $j_{i'} - j_i$ and $j_{i''} - j_i$, and constant coboundary expansion again follows from Lemma 46. ◀

The proof of Theorem 4 now follows directly from Lemma 44: if all links of a local spectral expander with vanishing cohomology are coboundary expanders, then so is the complex itself.

Proof of Theorem 4. By Lemma 47, it is enough to handle just the case of the full complex $h^1(\mathfrak{G}_q^{d,S_k},\Gamma)$. Toward this end, let q_0 be large enough so that we have $\frac{2e}{\sqrt{q_0}-1} \leqslant \frac{\beta^2}{16}$. Let $V,W \in \mathfrak{G}_q^{2d,S_k}(1)$ be any edge. By Theorem 3, we get that $\mathfrak{G}_{q,\{W,V\}}^{d,S_k}$ is a 1/2-local spectral expander and by Lemma 47 we have $h^1(\mathfrak{G}_{q,V}^{d,S_k},\Gamma) = \Omega(1)$. So, there is a constant $\beta > 0$ such that $h^0(\mathfrak{G}_{q,\{W,V\}}^{d,S_k},\Gamma) \geqslant \beta$ and $h^1(\mathfrak{G}_{q,V}^{d,S_k},\Gamma) \geqslant \beta$. As $\frac{2e}{\sqrt{q_0}-1} \leqslant \frac{\beta^2}{16}$ by Lemma 44 we get $h^1(\mathfrak{G}_q^{d,S_k},\Gamma) \geqslant \beta^2/16$. ◀

4 The Faces Complexes

In this section, we analyze the expansion properties of the faces complexes of the projected flags complex.

4.1 Local Spectral Expansion

We first bound the local spectral expansion of the faces complexes. The following lemma, which relates the local spectral expansion of the faces complex on disjoint colors to the local spectral expansion of the partite base complex, is standard but we include it for completeness.

▶ **Lemma 48.** *Let X be a k-partite λ-product. Let $2r+2 \leqslant k$. Let $J \subseteq \binom{[k]}{r+1}$ be a subset of disjoint colors. Then $\mathrm{F}^J X$ is a $\frac{(r+1)\lambda}{1-(k-r)\lambda}$-one-sided local spectral expander.*

Proof. Let $s \in \mathrm{F}^J X$ with co-dimension exactly 2. Observe that the random walk matrix of $(F^J X)_s$ is

$$\mathsf{W}_s = \begin{bmatrix} 0 & \mathsf{S}_{c_1,c_2} \\ \mathsf{S}_{c_2,c_1} & 0 \end{bmatrix},$$

where $c_1, c_2 \sqcup \mathtt{col}(s) = J$. By Lemma 22, we get that $\lambda(W_s) \leqslant (r+1)\cdot\lambda$. Similarly, links of co-dimension i are i-partite graphs where each bi-partite component is a corresponding swap walk in the original complex as above, and are therefore connected. The Trickle-Down Theorem (Theorem 18) then gives the desired statement. ◀

Now, the following lemma is an easy consequence of Theorem 3 and Lemma 48.

▶ **Lemma 49.** *Fix any $d \geqslant 3$, any prime power q, and a set $S \subseteq [d-1]$ of size at least 2. Let $2r+2 \leqslant |S|$ and $J \subseteq \binom{S}{r+1}$ be disjoint color sets. Then $\mathrm{F}^J(\mathfrak{G}_q^{d,S})$ is a $\frac{2(r+1)}{\sqrt{q}-1-2(|S|-r)}$-local-spectral expander for any prime power q.*

4.2 Coboundary Expansion

We now prove the faces complexes associated with the projected flags complex is good 1-coboundary expander.

▶ **Theorem 50** (Swap coboundary expansion of the projected flags complex)**.** *Let d,k,r be integers such that $d > k \geqslant r^5$. There is some $q_0 = q_0(k,r)$ such that the following holds. Let $q \geqslant q_0$ be any prime power, $\mathfrak{G}$ be the $\mathfrak{G}_q^{2d}$-flags complex, and*

$$S_k = \{d-k,\dots,d-1,d,d+1,\dots,d+k\}.$$

Let $\mathcal{Z} = \mathrm{F}^r(\mathfrak{G}^{S_k})$ be its faces complex. Then $\mathcal{Z}$ is a coboundary expander and $h^1(\mathcal{Z}) \geqslant \exp(-O(\sqrt{r}\log^2 r))$.

As in [24, Lemma 8.2], we start by showing it is enough to argue that a constant fraction of projections of the faces complex are themselves co-boundary expanders, a variant of Lemma 45 from [23] for the faces complex. In particular, we start with the r-faces complex of a k-partite complex and consider the projections onto m disjoint r-sets of colors from $[k]$, where each such r-set acts as a color for the faces complex. If (i) all the 1-skeletons of vertex-links of all such projections are spectral expanders, and (ii) a large fraction of such projections are 1-coboundary expanders, then (abstracting [24, Lemma 8.2]) this implies coboundary expansion of the faces complex.

▶ **Lemma 51.** *Let k, r be integers such that $k \geqslant r^5$. Let X be a k-partite simplicial complex and $\mathcal{Y} = \mathrm{F}^r X$. Let $m = \sqrt{r+1}$. Assume that for each $I \in \mathrm{F}\Delta_k(m-1)$ and for each $v \in \mathcal{Y}^I$, the underlying graph of $\mathcal{Y}^I_v$ is a $\Theta(1)$-edge expander. If*

$$\mathop{\mathbb{P}}_{I \sim \mathrm{F}^r \Delta_k(m-1)} \left[\mathcal{Y}^I \text{ is a 1-coboundary expander with } h^1(\mathcal{Y}^I) \geqslant \beta \right] = p$$

where I is a uniformly chosen set of $m = \sqrt{r+1}$ pairwise disjoint colors from $\mathcal{C}$, then $\mathcal{Y}$ is a 1-coboundary expander with $h^1(\mathcal{Y}) = \Omega(\beta p^2)$.

We defer the proof of Lemma 51 to the full version of this paper as it is essentially identical to the proof of [24, Lemma 8.2] up to changes in nomenclature. We will also rely on [24, Proposition 7.1.1] to relate the coboundary expansion of the faces complex to the base complex.

▶ **Proposition 52** ([24, Proposition 7.1.1])**.** *Let X be an k-partite complex that is a λ-local spectral expander for $\lambda \leqslant \frac{1}{2r^2}$. Let $\ell \geqslant 5$ and let $J = \{c_1, c_2, \ldots, c_\ell\}$ be a set of mutually disjoint colors $c_j \subseteq [k]$, with $|c_j| \leqslant r$. Denote by $R = \sum_{j=1}^{\ell} |c_j|$. Let $\beta > 0$ and assume that for every $I = \{i_1, i_2, \ldots, i_\ell\}$ such that $i_j \in c_j$ and every $w \in X^{\cup J \setminus I}$, we have $h^1(X^I_w) \geqslant \beta$. Then $h^1(\mathrm{F}^J X) \geqslant \beta_1^R$ for $\beta_1 = \Omega_\ell(\beta)$.*

Before we begin proving Theorem 50, it will be helpful to establish some notation and definitions.

Notations.

- Let $\mathfrak{G} = \mathfrak{G}_q^{2d}$. Note that $\mathfrak{G}$ is $(2d-1)$-partite.
- Let $\mathcal{Z} = \mathrm{F}^r(\mathfrak{G}^{S_k})$ and $\tilde{\mathcal{Z}} = \mathrm{F}(\mathfrak{G}^{S_k})$.
- Let $\mathcal{C} = \binom{S_k}{r+1}$.
- Let $m = \sqrt{r+1}$

Given $w \in \mathfrak{G}^{S_k}$, let $c^* = \mathtt{col}(w) \subseteq S_k$ and write $c^* = \{j_1, \ldots, j_b\}$. Define $b+1$ *c^*-bins*

$$B_i = \{j \in S_k | j_i < j < j_{i+1}\}$$

for $0 < i < b$ and also $B_0 = \{j \in S_k | j < j_1\}$ and $B_b = \{j \in S_k | j > j_b\}$. Note that the link $\mathfrak{G}^{S_k}_w$ is isomorphic to the join of $\left\{ (\mathfrak{G}_q^{j_{i+1}-j_i})^{\tilde{B}_i} \right\}_{i=0}^{b}$ where we have taken $j_0 = 0$, $j_{b+1} = 2d$ and $\tilde{B}_i = \left\{ \tilde{j} | \tilde{j} = j - j_i \text{ where } j \in B_i \right\}$ for $i = 1, 2, \ldots, b$. We will refer to a c^*-bin as just a bin when the c^* is clear from context.

Now, recall that for $J', J \subseteq \binom{S_k}{\leqslant r+1}$ we write $J' \leqslant J$, if $J' = \{c'_1, \ldots, c'_m\}$ where $c'_j \subseteq c_j$. Let $c^i_j = c_j \cap B_i$.

▶ **Definition 53** (Classification of bins). *Let $J = \{c_1, c_2, \dots, c_\ell\} \subseteq \mathcal{C}$ be a set of colors of $\mathcal{Z}$. A bin B_i is* J-crowded *if $J_i = (c_1^i, \dots, c_\ell^i)$ has more than one coordinate for which $c_j^i \neq \emptyset$. It is* J-lonely *if there is exactly one such coordinate. Otherwise, it is* J-empty*. When the set J is clear from the context we simply say crowded, lonely, and empty for J-crowded, J-lonely, and J-empty, respectively.*

▶ **Definition 54** (Well-spread colors). *Let J be a set of m colors in $\mathcal{C}$. We say that J is* well-spread *if the following properties hold.*

1. *J consists of pairwise disjoint colors.*
2. *For every $\ell_1, \ell_2 \in (\cup J) \cup \{d-k, d+k\}$ it holds that $|\ell_1 - \ell_2| \geqslant \frac{2k}{(m(r+1))^3}$.*
3. *For every $J' \subseteq J$ of size $|J'| = 5$ and $\overline{J'} = J \setminus J'$ the following hold.*
   a. *For every $\ell \in \cup\overline{J'} \cup \{d-k\}$, there exists some $\ell' \in \cup\overline{J'} \cup \{d+k\}$ such that $1 < \ell' - \ell \leqslant \frac{200k\log(r+1)}{(r+1)m}$.*
   b. *For every $c \in J$, the number of colors $i \in c$ that are in J'-crowded $\cup\overline{J'}$-bins is at most $\frac{100(r+1)\log(r+1)}{m\log m} = \frac{200(r+1)}{m}$.*

▶ Remark 55. Our definition of well-spread colors does not include an analogue of [24, Item 3(c)]. The bin size bounds in Item 3(c) lets them obtain a stronger bound for the flags complex. Here, we have adapted the proof used for weaker swap-expansion parameter where we don't need the bin size bounds.

Proof of Theorem 50. Our proof strategy follows the overall framework of [24] for the full flags complex. The idea is to break $\mathcal{Z}$ into progressively simpler complexes (projections, then deep links) until we have a complex whose coboundary expansion we can bound directly via our bounds for $\mathfrak{G}_w^{S_k}$.

Moving to projections of $\mathcal{Z}$ to well-spread colors. First, we move from bounding the coboundary expansion of $\mathcal{Z}$ to coboundary expansion of the projections of $\mathcal{Z}$ on well-spread color-sets J (see Definition 54). We do this in two (sub)steps: (i) first, we show most color-sets are well-spread (see Claim 56), (ii) then, we show that if most color-sets are well-spread and all well-spread color sets are good co-boundary expanders, then $\mathcal{Z}$ itself is a good co-boundary expander (see Claim 57).

More concretely, denote by $\mathcal{J}$ the set of well-spread J's as per Definition 54. We claim the measure of $\mathcal{J}$ tends to 1 as r grows large, i.e., most projections are well-spread as follows.

▷ Claim 56. Let k, r be such that $k \geqslant r^5$. Let $6 \leqslant m \leqslant (r+1)$. The probability that m colors chosen uniformly from S_k are well-spread tends to 1 as $r \to \infty$.

Claim 56 is an elementary combinatorial calculation and follows directly from an analogous computation in [24, Proposition 8.4.1] for the correct setting of parameters,[8] so we omit the proof.

By Claim 56, for large enough r at least half of the sets J are in $\mathcal{J}$. Thus, applying, Lemma 51 to $\mathfrak{G}$, we immediately get the coboundary expansion of $\mathcal{Z}$ is lower bounded by the worst-case expansion over spread projections as stated in Claim 57.

▷ Claim 57. $h^1(\mathcal{Z}, \Gamma) \geqslant \Omega(1) \cdot \min_{J \in \mathcal{J}} h^1(\mathcal{Z}^J, \Gamma)$.

[8] To be precise, the dimension parameter in [24, Proposition 8.4.1] should be set to $2k+1$ and the arity parameter to r. After this, observe that the bijection from $[2k+1]$ to S_k that maps $x \mapsto d-(k+1)+x$ has the property that (i) a well-spread set of color according the definition of [24] maps to a well-spread set of colors according to our definition, and (ii) a uniform distribution on $[2k+1]$ induces a uniform distribution on S_k.

▶ Remark 58. We note that

$$\mathcal{Z}^J = \mathrm{F}^r(\mathfrak{G}^{S_k})^J = \mathrm{F}^r(\mathfrak{G})^J.$$

Comparing this to the proof of flags coboundary expansion in [24] we find that they also end up with the projection of the r-faces complex of the flags complex to well-spread colors. As a result, we will be able to proceed with the proof in exactly the same manner from hereon with the changes due to changes in parameters of the well-spreadness definition being addressed when they come up.

Trickling down the links. In the previous step, we reduced the task of showing the coboundary expansion of $\mathcal{Z}$ to coboundary expansion of $\mathcal{Z}^J$ for spread color sets. Our next step will be to reduce the analysis of $h^1(\mathcal{Z}^J)$ to the coboundary expansion of its links. In particular, in Claim 59 below, we fix $J \in \mathcal{J}$ and use a "local to global" argument to move to the links of $\mathcal{Z}^J$. We note that we need the local spectral expansion to be constant for Claim 59 to hold and indeed there is a q_0 such that we have local spectral expansion $\mathcal{Z}^J$ to be at least $1/2$ for any $q \geqslant q_0$ due to Claim 59.

▷ Claim 59 ([24, Claim 8.5.1]). Let J be a set of m pairwise disjoint colors for $\mathcal{Z}$, and let $i = -1, 0, \ldots, m-6$, then

$$h^1(\mathcal{Z}^J, \Gamma) \geqslant \exp(-O(i)) \cdot \min_{s \in \mathcal{Z}^J(i)} h^1(\mathcal{Z}^J_s, \Gamma),$$

where we understand $\mathcal{Z}^J_s$ to mean $\mathcal{Z}^{J'}_s$ for $J' = J \setminus \mathtt{col}(s)$.

We note that, as stated in [24, Claim 8.5.1] the set J should be well-spread, but it is easy to check the proof only relies on disjointness. Thus, by Claim 59,

$$h^1(\mathcal{Z}^J, \Gamma) \geqslant \exp(-O(m)) \cdot \min_{s \in \mathcal{Z}^J(m-6)} h^1(\mathcal{Z}^J_s, \Gamma).$$

Reducing to Crowded Colors. Fix any $s \in \mathcal{Z}^J(m-6)$. Our goal is now to reduce from analyzing coboundary expansion of $\mathcal{Z}^J_s$ to $\mathcal{Z}^{\tilde{J}}_s$ where $\tilde{J}$ are the crowded colors in the link. In particular, since by definition of spreadness there are few crowded colors, this will allow us to apply Proposition 52 with much smaller $R = \sum_{j=1}^{5} |\tilde{c}_i|$ to achieve the final desired coboundary expansion. Toward this end, we'd like to use the following result of [24].

▶ **Corollary 60** ([24, Corollary 8.7]). *Let $w \in \mathfrak{G}$. There is some constant $\beta > 0$ such that*

$$h^1(\tilde{\mathcal{Z}}^J_w, \Gamma) \geqslant \beta \cdot h^1(\tilde{\mathcal{Z}}^{\tilde{J}}_w, \Gamma),$$

where $\tilde{J} = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_\ell\}$ and $\tilde{c}_j = \{i \in c_j | i$ is not in a lonely or empty bin$\}$.

As stated, the only problem is the above bounds $\tilde{\mathcal{Z}}^J_w$ for a vertex $w \in \tilde{\mathcal{Z}}$ (equivalently a face in $\mathfrak{G}$), while we need to bound $\mathcal{Z}^J_s$. However, by Observation 27 we have $\mathcal{Z}^J_s \cong \mathrm{F}^J\mathfrak{G}_{\cup s}$, and since J consists entirely of $(r+1)$-sets we indeed have $\mathrm{F}^J\mathfrak{G}_{\cup s} \cong \tilde{\mathcal{Z}}^J_{\cup s}$. Thus, applying Corollary 60, we have

$$h^1(\mathcal{Z}^J_s, \Gamma) = h^1(\mathrm{F}^J\mathfrak{G}_{\cup s}, \Gamma) = h^1(\tilde{\mathcal{Z}}^J_{\cup s}, \Gamma) \geqslant \Omega(h^1(\tilde{\mathcal{Z}}^{\tilde{J}}_{\cup s}, \Gamma)).$$

Reduction to the Base Complex. Next, denote by $\beta = \min_{w,I} h^1(\mathfrak{G}^{\cup I}_{\cup s \sqcup w}, \Gamma)$ where the minimum is taken over sets I consisting of five singletons, i.e., $I = \{\{i_0\}, \dots, \{i_4\}\}$ such that $I \cup \mathtt{col}(s) \leqslant \tilde{J}$, and $w \in \mathfrak{G}_{\cup s}$ such that $\mathtt{col}(w) \subseteq \bigcup \tilde{J}$ and $\mathtt{col}(w) \cap I = \emptyset$. Since $\tilde{Z}^{\tilde{J}}_s \cong \mathrm{F}^{\tilde{J}} \mathfrak{G}_{\cup s}$, by Proposition 52,

$$h^1(\tilde{Z}^{\tilde{J}}_s, \Gamma) \geqslant \mathrm{const} \cdot \beta_1^R,$$

where $\beta_1 = \Omega(\beta)$ and $R = \sum_j |\tilde{c}_j|$. By Item 3b of Definition 54, for every c_j,

$$|\tilde{c}_j| = \text{the number of indices from } c_j \text{ in crowded bins} = O\left(\frac{r \log r}{m \log m}\right),$$

so, in total $R = O\left(\frac{r \log r}{m \log m}\right) = O\left(\sqrt{r}\right)$

$$h^1(\mathcal{Z}, \Gamma) \geqslant \mathrm{const} \cdot \exp(-O(\sqrt{r})) \cdot \beta^{\sqrt{r}}. \tag{2}$$

Using bounds for the Flags complex. It is left to bound β, the coboundary expansion of deep spread out links in the projected complex.

▷ Claim 61. Let $I = \{\{i_0\}, \dots, \{i_4\}\}$, and let $s \in \mathcal{Z}$ be such that $|s| = m - 5$ and there is a well-spread set of colors $J \in \mathcal{J}$ such that $I \cup \mathtt{col}(s) \leqslant J$. Let $w' \in \mathfrak{G}^{\cup J}_{\cup s}$ be such that $\mathtt{col}(w') \cap I = \emptyset$. Then

$$h^1(\mathfrak{G}^{\cup I}_{\cup s \sqcup w'}, \Gamma) \geqslant \exp(-O(\log^2 r)).$$

Claim 61 is a variant of [24, Lemma 8.8], which makes the same statement for the full flags complex under a modified notion of spreadness dependent on the ambient dimension d (n in their definition), while our definition of spreadness critically depends only on k, as d grows with the number of vertices and none of the projections we consider would be spread under [24]'s definition. Nevertheless, because we are inside the link of s, "morally" we may treat the ambient dimension as k, in which case our definition reduces to theirs.

In somewhat more detail, the proof of Claim 61 essentially proceeds by (1) first, using Lemma 45 to move to projections to 4 indices, (2) relating the coboundary expansion of the complexes to the diameter, and (3) finally, obtaining a bound on the diameter as result of well-spreadness of J. Because we are inside a link, the diameter bounds depend on the ratio of the parameters in Item 2 and parameters in Item 3 which remains unchanged from the definition of [24]. We reproduce relevant parts of the proof of Claim 61 in the full version of the paper.

We can now complete the proof. Namely, by Claim 61, we have

$$\beta = \min_{w,I} h^1(\mathfrak{G}^{\cup I}_{\cup s \sqcup w}, \Gamma) \geqslant \exp(-O(\log^2 r)).$$

Plugging this back into Equation (2) we get

$$h^1(\mathcal{Z}, \Gamma) \geqslant \mathrm{const} \cdot \exp(-O(\sqrt{r} \log^2 r))$$

as needed. ◀

5 Applications

5.1 Background

5.1.1 Agreement testing

Agreement Testing. Let $\mathcal{F} \subseteq \binom{[n]}{k}$ be a family of k element subsets of $[n]$ and let $\{f_s : s \to \Sigma | s \in \mathcal{F}\}$ be an ensemble of local functions, each defined over a subset $s \in \mathcal{F}$. An agreement test is a randomized property tester for the question:

$$\textit{Is there a global function } G : [n] \to \Sigma \textit{ such that } f_s = G|_s \textit{ for all } s \in \mathcal{F}?$$

Towards answering this question, we will work with the following basic restricted type of 2-query test.

▶ **Definition 62** (2-query agreement test)**.** *Let Σ be an arbitrary finite set which we will call the* alphabet*. Given a family $\mathcal{F}$ of k element subsets of $[n]$ and an ensemble of functions $\mathcal{E} = \{f_s : s \to \Sigma | s \in \mathcal{F}\}$ from $[n]$ to Σ which we will call* local functions*, a 2-query agreement test is a distribution $\mathcal{D}$ over a pair of subsets $s_1, s_2 \in \mathcal{F}$, with the agreement of the ensemble being*

$$\mathsf{agr}_{\mathcal{D}}(\mathcal{E}) = \mathop{\mathbb{P}}_{s_1,s_2\sim\mathcal{D}}[f_{s_1}|_{s_1\cap s_2} = f_{s_2}|_{s_1\cap s_2}]\,.$$

▶ Remark 63. In the literature, sometimes more general agreement tests are considered where the distribution $\mathcal{D}$ is over ℓ subsets in $\mathcal{D}$.

We note that for any property testing question we would like the test to be complete, i.e., we would want an object having the property in consideration to pass the test. By design, agreement tests (as described above) have completeness as one can clearly see restriction of a global function to sets in $\mathcal{F}$ would pass any 2-query test of the above form.

We also want agreement tests to be sound, i.e., whenever the tests passes there is a global function whose restrictions agree with most functions in the ensemble. There are two regimes of soundness of agreement tests that are typically considered:

The 99% regime or the high acceptance regime. In this regime, we want to claim that there is a global function $G : [n] \to \Sigma$ such that $G|_s = f|_s$ for most $s \in \mathcal{F}$ whenever $f_s|_{s\cap s'} = f_{s'}|_{s\cap s'}$ for most $s, s' \in \mathcal{F}$. More formally, we want to show

$$\mathsf{agr}_{\mathcal{D}}(\{f_s\}_{s\in\mathcal{F}}) \geqslant 1-\varepsilon \quad\Longrightarrow\quad \exists G : [n] \to \Sigma,\ \mathbb{P}\left[G|_s = f_s\right] \geqslant 1 - O(\varepsilon).$$

The 1% regime or the low acceptance regime. In this regime, if there is at least some, say ε, fraction of $s, s' \in \mathcal{F}$ so that $f_s|_{s\cap s'} = f_{s'}|_{s\cap s'}$, we want to claim that there is a global function $G : [n] \to \Sigma$ such that $G|_s \approx f|_s$ for some fraction of $s \in \mathcal{F}$, say $\mathsf{poly}(\varepsilon)$. More formally, we want to show there there is some polynomial poly such that for any $\varepsilon \geqslant \varepsilon_0$,

$$\mathsf{agr}_{\mathcal{D}}(\{f_s\}_{s\in\mathcal{F}}) \geqslant \varepsilon \quad\Longrightarrow\quad \exists G : [n] \to \Sigma,\ \mathbb{P}\left[\mathsf{dist}(G|_s, f_s) \leqslant \eta\right] \geqslant \mathsf{poly}(\varepsilon),$$

where η is some "closeness" parameter. Assuming (for simplicity) that the polynomial is of form $\mathsf{poly}(x) = x^c$, we call (ε_0, η, c) the *soundness parameters* and call a test $\mathcal{D}$ with these parameters a (ε_0, η, c)*-sound test*. More consisely, we may leave ε_0, c unspecified say a test is η-sound, i.e., a η-sound test is (ε_0, η, c)-sound for some $\varepsilon_0, c > 0$.

In the low soundness regime, agreement testing is quite non-trivial even over $\mathcal{F} = \binom{[n]}{k}$, and there is a long line of works which settled this [44, 33, 37, 38, 55, 36].

Here, we will study agreement testing on sparser families of sets, in particular, we will take $\mathcal{F} = X(k-1)$ for some simplicial complex X of dimension $d > k-1$ and define an agreement test based on its weights. In particular, we will consider the following agreement test.

$(k, \sqrt{k})$-test. The $(k, \sqrt{k})$-test $\mathcal{D}$ is a 2-query test given by the following process.

1. Sample $t \sim \Pi_d$.
2. Sample $s \sim \Pi_{\sqrt{k}-1}$, conditioned on $s \subseteq t$.
3. Sample $s_1, s_2 \sim \Pi_{k-1}$, conditioned on $s \subseteq s_1, s_2 \subseteq t$.

Finally, we remark that for any function $G : [n] \to \Sigma$ the ensemble $\{G|_s\}_{s \in \mathcal{F}}$ can be thought of as an encoding of G that is locally testable using the agreement test. This code is often referred to as a derandomized direct product code and agreement testing is referred to as direct product testing in this context.

5.1.2 Probabilistically Checkable Proofs (PCPs)

For the most part, we will forego a discussion of the "proofs" view of probabilistically checkable proofs and take a more combinatorial view of PCPs using the language of the Label Cover problem.

▶ **Definition 64.** *An instance of* Label Cover $\Psi = (G = (L \cup R, E), \Sigma_\ell, \Sigma_r, \Phi = \{\phi_e\}_{e \in E})$ *consists of a bipartite graph G, alphabets Σ_ℓ, Σ_r and constraints $\phi_e : \Sigma_\ell \to \Sigma_r$, for each edge $e \in E$.*

Given a label cover instance Φ, the goal is to find assignments $A_\ell : L \to \Sigma_\ell$ and $A_r : R \to \Sigma_r$ that satisfy as many of the constraints as possible, namely that maximize the quantity

$$\mathsf{val}_\Phi(A_\ell, A_r) = \frac{1}{|E|} |\{e = (u, v) \in E | \phi_e(A_\ell(u)) = A_r(v)\}|.$$

This maximum value is referred to as the value of the instance Φ and denoted by

$$\mathsf{val}(\Phi) = \max_{A_\ell, A_r} \mathsf{val}_\Phi(A_\ell, A_r).$$

The label cover problem is a special case of the 2-ary constraint satisfaction problem where each constraint is a relation on the alphabet instead of being a function.

▶ **Definition 65.** *An instance $\Psi = (G = (V, E), \Sigma, \{\Phi_e\}_{e \in E})$ of a 2-CSP consists of a weighted digraph G, alphabet Σ, and constraints $\Phi_e \subseteq \Sigma \times \Sigma$, one for each edge.*

Let $\mathsf{gapLabelCover}[\alpha, \beta]$ be the promise problem wherein the input is an instance Φ of label cover promised to either have $\mathsf{val}(\Phi) \geqslant \alpha$ or else $\mathsf{val}(\Phi) \leqslant \beta$, and the goal is to distinguish between these two cases.

▶ **Definition 66.** *An instance of* Generalized Label Cover $\Psi = (G = (L \cup R, E), \Sigma_\ell, \Sigma_r, \Phi = \{\phi_e\}_{e \in E})$ *consists of a bipartite graph G, alphabets Σ_ℓ, Σ_r, $\{\Sigma_\ell(u)\}_{u \in L}$ with $\Sigma_\ell(u) \subseteq \Sigma_\ell$, for each $u \in L$, and constraints $\phi_e : \Sigma_\ell(u) \to \Sigma_r$, for each edge $e = (u, v) \in E$.*

5.2 New Agreement Testers

[22, 13, 14] showed the following agreement testing result for complexes with strong expansion properties. We refer the reader to [13, Remark 4.2] for the specific parameters.

▶ **Theorem 67** ([22]). *There is a $c > 0$ such that for all $\delta > 0$, there is $r \in \mathbb{N}$ such that for any $\ell \geqslant \exp(\mathsf{poly}(1/\delta))$, any $k \geqslant \mathsf{poly}(\ell)$ and there is a $\gamma_0(k)$ such that for any $\gamma \leqslant \gamma_0$ the following holds.*

If a k-dimensional clique complex X is a γ-local spectral expander and $(c2^{-o(r)}, r)$-swap coboundary expander, then the $(\ell, \sqrt{\ell})$-agreement test over $X(\ell)$ with respect to every alphabet Σ has soundness δ. Namely, if $F : X(\ell-1) \to \Sigma^\ell$ passes the $(\ell, \sqrt{\ell})$-agreement test with respect to X with probability at least δ, then there is $f : X(0) \to \Sigma$ such that

$$\mathop{\mathbb{P}}_{A \sim \mu_\ell} [\mathsf{dist}(F[A], f|A) \leqslant \delta] \geqslant \mathsf{poly}(\delta).$$ ⌟

As an immediate corollary we get that the $(\ell, \sqrt{\ell})$-test on the projected flags complex has a low soundness test.

▶ **Theorem 9** (Agreement testing over the projected flags complex). *Fix a $\delta > 0$. There is some q_0 such that the following holds.*

Let $q \geqslant q_0$ be any prime power. Let d, k, ℓ be such that $d > k \geqslant \mathsf{poly}(\ell)$ and $\ell \geqslant \exp(\mathsf{poly}(1/\delta))$. Let $\mathfrak{G}$ denote the $\mathfrak{G}_q^{2d}$-flags complex and

$$S_k = \{d-k, \ldots, d-1, d, d+1, \ldots, d+k\}.$$

Then, the $(\ell, \sqrt{\ell})$-agreement test over $\mathfrak{G}^{S_k}(\ell)$ with respect to every alphabet Σ has soundness δ, i.e., if $F : \mathfrak{G}^{S_k}(\ell-1) \to \Sigma^\ell$ passes the $(\ell, \sqrt{\ell})$-agreement test with respect to $\mathfrak{G}^{S_k}$ with probability at least δ, then there is $f : \mathfrak{G}^{S_k}(0) \to \Sigma$ such that

$$\mathop{\mathbb{P}}_{A \sim \mu_\ell} [\mathsf{dist}(F[A], f|A) \leqslant \delta] \geqslant \mathsf{poly}(\delta).$$

Proof. Let $c > 0, r, \gamma_0(k)$ be as in Theorem 67. First note that by taking q_0 the flags complex $\mathfrak{G}_q^{2d,S_k}$ has $\gamma \leqslant \gamma_0$ by Theorem 3. Now, by Theorem 50 we have the the swap coboundary expansion required by Theorem 67. Hence, by Theorem 67, the $(\ell, \sqrt{\ell})$-agreement test over $\mathfrak{G}^{S_k}(\ell)$ with respect to every alphabet Σ has soundness δ for any $\ell \geqslant \exp(\mathsf{poly}(1/\delta))$. ◀

5.3 A Near-linear Size PCP

Finally, we show the projected flags complex can be used to construct PCPs using techniques in [14].

▶ **Theorem 11.** *For all $\delta \in (0,1)$, there exist a constant $C > 0$ such that for all sufficiently large integers d and n there is a polynomial time reduction mapping a 3SAT instance ϕ of size n to a label cover instance Ψ over a bipartite graph G of size n' for some $n' \leqslant n \exp\left(C\sqrt{\log n \log\log n}\right)$, such that*

1. *If ϕ is satisfiable, then Ψ is satisfiable.*
2. *If ϕ is unsatisfiable, then $\mathsf{val}(\Psi) \leqslant \delta$.*
3. *The left alphabet of Ψ is $\Sigma^{\exp(C/\delta)}$ and right alphabet is $\Sigma^{\sqrt{\exp(C/\delta)}}$ for some alphabet Σ with $\log|\Sigma| \leqslant \exp\left(C\sqrt{\log n \log\log n}\right)$.*

We first record the set of properties of the projected flags complex in Theorem 69 that are required for the PCP construction. The statement of Theorem 69 is an analog of Theorem 2.13 in [14], albeit with slightly different parameters, as we replace the Chapman-Lubotzky complexes [17] with the projected buildings.

▶ **Definition 68** (Symmetrically-Connected Complex)**.** *Let L be a k-partite complex. We call L* symmetrically-connected *if*

1. *For any $i \neq j \in [k]$, the diameter of L^{ij} is bounded by $O(\frac{k}{|i-j|})$.*
2. *Sym_L acts transitively on its top faces.*

▶ **Theorem 69.** *For all $\delta \in (0,1)$, and for all $C > 0$ the following holds. For large enough $\ell, k \in \mathbb{N}$, and for large enough $n \in \mathbb{N}$, there is a prime power $q = \Theta(\log^C n)$ and $d = \Theta\left(\sqrt{\frac{\log(n)}{\log\log(n)}}\right)$, one can construct in time $\mathsf{poly}(n)$ a $2k$-dimensional complex X for which $n \leqslant |X(0)| \leqslant nq^{O(d)}$ such that the following holds.*

1. *The complex X is $(2k+1)$-partite.*
2. *The complex X is a clique complex.*
3. *$\Delta_{max}(X) \leqslant q^{O(kd)}$.*
4. *Each vertex-link L of X is symmetrically-connected. In other words, we have the following:*
   a. *For each vertex link L of X there is a group $\mathrm{Sym}(L)$ that acts transitively on the $(k-2)$-faces $L(k-2)$.*
   b. *For each vertex link L of X and any colors $i \neq j$, the bipartite graph $(L_i(0), L_j(0))$ is uniformly weighted (over edges) and has diameter $O(\frac{\max\{i,j\}}{|i-j|})$.*
5. *X is a $\frac{1}{\sqrt{q}}$-product.*
6. *The 1-skeleton $G = (X(0), X(1))$ satisfies $\lambda(G) \leqslant \frac{1}{k(\sqrt{q}-1)}$ and $|\lambda|(G) \leqslant \frac{1}{2k}$.*
7. *The $(\ell, \sqrt{\ell})$-direct product test on X has soundness δ.*

Proof. Given $\delta > 0$, $C > 0$, and n, take $q = 2^{\lceil \log_2 \log_2^C n \rceil}$ and $d = \lceil \sqrt{\log_q n} \rceil$. If q and k are large enough, then the complex $X = \mathfrak{G}_q^{2d,S_k}$ has $(\ell, \sqrt{\ell})$-agreement test with soundness δ for large enough ℓ by Theorem 9 (this shows Item 7). We get Items 5 and 6 from Theorem 3, Fact 31, and the Trickle-Down theorem (Theorem 18). It is elementary to see the group of automorphisms $\mathrm{Aut}(L)$ acts transitively on the top faces of L where L is a vertex link (this gives us Item 4a).[9] Item 4b follows directly from Claim 8.7.2 in [24] and hence we get Item 4. Item 3 is a straight-forward counting which we omit for the sake of brevity. Items 1 and 2 also follow directly from the definition. ◀

Overview of PCP construction in [14]. The first step of [14]'s PCP involves reducing an arbitrary 2-query PCP, viewed as a 2-CSP, to a 2-CSP whose constraint graph underlies an HDX. In particular, they show how to embed an arbitrary 2-CSP on a graph G while maintaining the soundness provided that G has a fault-tolerant routing protocol, and show any sufficiently local-spectral HDX (see Items 5 and 6) with dense symmetric links (see Items 4a and 4b) indeed satisfies such a protocol. The second step uses a size-efficient direct product tester from HDX. This allows derandomized parallel repetition for 2-CSPs on an HDX, and reduce their soundness to a small constant close while incurring a poly-logarithmic blow-up in size. Combining the first and second steps they get a quasi-linear PCP with small soundness, albeit with a large alphabet.

Since the projected flags complex satisfies essentially the same properties as the Chapman-Lubotzky complex albeit with different parameters (compare Theorem 69 with [14, Theorem 2.13]), the same construction also works even if we use projected flags complex. Theorem 70 states a general version of this theorem. In the full version of this paper, we provide a sketch of the proof to highlight the changes in parameters.

[9] As in the proof of Lemma 46, taking an appropriate basis for the vector spaces in any two such flags, we can transform one to the other with the corresponding basis exchange matrix.

▶ **Theorem 70** ([14]). *There exist $\alpha, \varepsilon_0, d_0 > 0$, and $r, C \in \mathbb{N}$ such that for large enough $n \in \mathbb{N}$ the following holds. Let X be d-partite λ-product such that (i) its underlying graph G has $|\lambda|(G) \leqslant \alpha/4$; (ii) its vertex links are symmetrically-bounded; (iii) its max degree $\Delta(X)$ is bounded by Δ; (iii) the dimension $d \geqslant d_0$ and its size $|X(0)| = n$. Also, assume that the $(\ell, \sqrt{\ell})$-direct product tester on X has soundness $\delta^2 \leqslant \varepsilon_0^2/16$. Then there is a $\mathsf{poly}(n)$-time procedure mapping any 2-CSP Ψ' with n vertices on a r-regular graph H and alphabet Σ, to an instance of Label Cover Φ over the weighted inclusion graph $(X(\sqrt{\ell}), X(\ell))$, over left alphabet $\Sigma^{\mathsf{poly}(\Delta \log(n) \log|\Sigma|)\ell}$ and right alphabet $\Sigma^{\mathsf{poly}(\Delta \log(n) \log|\Sigma|)\sqrt{\ell}}$ satisfying the following properties.*

- **(i)** ***Completeness:*** *If $\mathsf{val}(\Psi') = 1$ then $\mathsf{val}(\Phi) = 1$.*
- **(ii)** ***Soundness:*** *If $\mathsf{val}(\Psi') \leqslant 1 - \mathsf{poly}(d) r^2 \lambda^2 \log^C n - \frac{1}{\log^C n}$, then $\mathsf{val}(\Phi) \leqslant \delta$.*

We now start with a size efficient construction of PCP due to [68] stated in a more convenient formulation in terms of hardness of 2-CSPs.[10]

▶ **Theorem 71** ([68]). *There exist constants $c > 0, \delta \in (0,1)$ such that there is a polynomial-time reduction mapping a 3-SAT instance ϕ of size n to a 2-CSP instance Ψ over the regular graph $G = (V, E)$ and alphabet Σ where*

- *$\mathsf{size}(G) \leqslant n(\log n)^{c \log\log n}$ and $|\Sigma| = O(1)$.*
- *If ϕ is satisfiable, then $\mathsf{val}(\Psi) = 1$.*
- *If ϕ is not satisfiable, then $\mathsf{val}(\Psi) \leqslant 1 - \delta$.*

▶ Remark 72. The construction of PCP in [68] is almost combinatorial. The only algebraic component is the multiplicative codes used in the paper.

Applying this to our complex, i.e., combining Theorems 69–71, we obtain Theorem 73.

▶ **Theorem 73.** *For all $\delta \in (0,1)$, there exist constants $C, C' > 0$ such that for all sufficiently large integers d and n, there exist k, ℓ with $k \geqslant \mathsf{poly}(\ell)$ and $\ell \geqslant \exp(\mathsf{poly}(1/\delta))$ so that the following holds. Let $\{X_{n'}\}_{n' \in \mathbb{N}}$ be the infinite sequence of complexes from Theorem 69, where every $X_{n'}$ is a $(2k+1)$-dimensional complex on n' vertices with parameters $q = \Theta(\log^{C'} n')$ and δ that is constructible in time $\mathsf{poly}(n')$. There is a polynomial time reduction mapping a 3SAT instance ϕ of size n to a label cover instance Ψ over the weighted inclusion graph $(X_{n'}(\ell), X_{n'}(\sqrt{\ell}))$ for some $n' \leqslant n \log^{C \log\log n} n$, such that*

1. *If ϕ is satisfiable, then Ψ is satisfiable.*
2. *If ϕ is unsatisfiable, then $\mathsf{val}(\Psi) \leqslant \delta$.*
3. *The left alphabet of Ψ is Σ^ℓ and right alphabet is $\Sigma^{\sqrt{\ell}}$ for some alphabet Σ with $\log|\Sigma| \leqslant q^{Ckd}$.*

As an immediate corollary, we get the version of the PCP theorem as stated in Theorem 11.

References

1 Dorna Abdolazimi and Shayan Oveis Gharan. An improved trickle down theorem for partite complexes. In *Proceedings of the conference on Proceedings of the 38th Computational Complexity Conference*, pages 1–16, 2023. doi:10.4230/LIPIcs.CCC.2023.10.

2 Vedat Levi Alev, Fernando Granha Jeronimo, Dylan Quintana, Shashank Srivastava, and Madhur Tulsiani. List decoding of direct sum codes. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1412–1425. SIAM, 2020. doi:10.1137/1.9781611975994.85.

[10] We need a regularization step [30, Lemma 4.2] on top of the original statement to obtain this form.

3 Vedat Levi Alev, Fernando Granha Jeronimo, and Madhur Tulsiani. Approximating constraint satisfaction problems on high-dimensional expanders. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science*, pages 180–201. IEEE Comput. Soc. Press, Los Alamitos, CA, [2019] ©2019. doi:10.1109/FOCS.2019.00021.

4 Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. doi:10.1007/BF02579166.

5 Noga Alon and Vitali D Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985.

6 Nima Anari, Vishesh Jain, Frederic Koehler, Huy Tuan Pham, and Thuy-Duong Vuong. Entropic independence: optimal mixing of down-up random walks. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1418–1430. ACM, 2022. doi:10.1145/3519935.3520048.

7 Nima Anari, Kuikui Liu, and Shayan Oveis Gharan. Spectral independence in high-dimensional expanders and applications to the hardcore model. *SIAM J. Comput.*, 53(6):S20–1, 2024. doi:10.1137/20M1367696.

8 Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: High-dimensional walks and an FPRAS for counting bases of a matroid. In *STOC'19—Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1–12. ACM, New York, 2019. doi:10.1145/3313276.3316385.

9 Nima Anari, Kuikui Liu, Shayan Oveis Gharan, Cynthia Vinzant, and Thuy-Duong Vuong. Log-concave polynomials IV: approximate exchange, tight mixing times, and near-optimal sampling of forests. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 408–420. ACM, 2021. doi:10.1145/3406325.3451091.

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

11 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

12 Mitali Bafna, Noam Lifshitz, and Dor Minzer. Constant degree direct product testers with small soundness. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 862–869. IEEE, 2024. doi:10.1109/FOCS61266.2024.00059.

13 Mitali Bafna and Dor Minzer. Characterizing direct product testing via coboundary expansion. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1978–1989. ACM, 2024. doi:10.1145/3618260.3649714.

14 Mitali Bafna, Dor Minzer, Nikhil Vyas, and Zhiwei Yun. Quasi-linear size pcps with small soundness from HDX. *CoRR*, abs/2407.12762, 2024. doi:10.48550/arXiv.2407.12762.

15 Mitali Bafna, Dor Minzer, Nikhil Vyas, and Zhiwei Yun. Quasi-linear size PCPs with small soundness from HDX. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 45–53. ACM, 2025. doi:10.1145/3717823.3718197.

16 Anders Björner. Shellable and cohen-macaulay partially ordered sets. *Transactions of the american mathematical society*, 260(1):159–183, 1980.

17 Michael Chapman and Alexander Lubotzky. Stability of homomorphisms, coverings and cocycles ii: Examples, applications and open problems. *Advances in Mathematics*, 463:110117, 2025.

18 Mark De Longueville. *A course in topological combinatorics*. Springer Science & Business Media, 2013.

19 Laura Grave de Peralta and Inga Valentiner-Branth. High-dimensional expanders from kac–moody–steinberg groups. *European Journal of Combinatorics*, 126:104131, 2025. doi:10.1016/J.EJC.2025.104131.

20 Philippe Delsarte. Association schemes and t-designs in regular semilattices. *Journal of Combinatorial Theory, Series A*, 20(2):230–243, 1976. doi:10.1016/0097-3165(76)90017-0.

21 Yotam Dikstein and Irit Dinur. Agreement testing theorems on layered set systems. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science*, pages 1495–1524. IEEE Comput. Soc. Press, Los Alamitos, CA, [2019] ©2019. doi:10.1109/FOCS.2019.00088.

22 Yotam Dikstein and Irit Dinur. Agreement theorems for high dimensional expanders in the low acceptance regime: The role of covers. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1967–1977. ACM, 2024. doi:10.1145/3618260.3649685.

23 Yotam Dikstein and Irit Dinur. Coboundary and cosystolic expansion without dependence on dimension or degree. In Amit Kumar and Noga Ron-Zewi, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, August 28-30, 2024, London School of Economics, London, UK*, volume 317 of *LIPIcs*, pages 62:1–62:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.APPROX/RANDOM.2024.62.

24 Yotam Dikstein and Irit Dinur. Swap cosystolic expansion. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1956–1966. ACM, 2024. doi:10.1145/3618260.3649780.

25 Yotam Dikstein, Irit Dinur, Prahladh Harsha, and Noga Ron-Zewi. Locally testable codes via high-dimensional expanders. *Electron. Colloquium Comput. Complex.*, TR20-072, 2020. URL: https://eccc.weizmann.ac.il/report/2020/072.

26 Yotam Dikstein, Irit Dinur, and Alexander Lubotzky. Low acceptance agreement tests via bounded-degree symplectic hdxs. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 826–861. IEEE, 2024. doi:10.1109/FOCS61266.2024.00058.

27 Yotam Dikstein and Max Hopkins. Chernoff bounds and reverse hypercontractivity on hdx. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 870–919. IEEE, 2024. doi:10.1109/FOCS61266.2024.00060.

28 Yotam Dikstein, Max Hopkins, Russell Impagliazzo, and Toniann Pitassi. High rate efficient local list decoding from hdx. *arXiv preprint arXiv:2601.22535*, 2026. doi:10.48550/arXiv.2601.22535.

29 Yotam Dikstein, Siqi Liu, and Avi Wigderson. Sparser abelian high dimensional expanders. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference, CCC 2025, Toronto, Canada, August 5-8, 2025*, volume 339 of *LIPIcs*, pages 7:1–7:98. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CCC.2025.7.

30 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.

31 Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, and Shahar Mozes. Locally testable codes with constant rate, distance, and locality. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 357–374. ACM, 2022. doi:10.1145/3519935.3520024.

32 Irit Dinur, Yuval Filmus, Prahladh Harsha, and Madhur Tulsiani. Explicit sos lower bounds from high-dimensional expanders. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, Virtual Conference, January 6-8, 2021*, volume 185 of *LIPIcs*, pages 38:1–38:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.38.

33 Irit Dinur and Elazar Goldenberg. Locally testing direct product in the low error range. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, October 25-28, 2008*, pages 613–622. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.26.

34 Irit Dinur and Prahladh Harsha. Composition of low-error 2-query pcps using decodable pcps. *SIAM Journal on Computing*, 42(6):2452–2486, 2013. doi:10.1137/100788161.

35 Irit Dinur and Tali Kaufman. High dimensional expanders imply agreement expanders. In *58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*, pages 974–985. IEEE Computer Soc., Los Alamitos, CA, 2017. doi:10.1109/FOCS.2017.94.

36 Irit Dinur and Inbal Livni Navon. Exponentially small soundness for the direct product z-test. *Theory Comput.*, 19:1–56, 2023. doi:10.4086/TOC.2023.V019A003.

37 Irit Dinur and Omer Reingold. Assignment testers: Towards a combinatorial proof of the PCP theorem. *SIAM J. Comput.*, 36(4):975–1024, 2006. doi:10.1137/S0097539705446962.

38 Irit Dinur and David Steurer. Direct product testing. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 188–196. IEEE Computer Society, 2014. doi:10.1109/CCC.2014.27.

39 Shai Evra and Tali Kaufman. Bounded degree cosystolic expanders of every dimension. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 36–48, 2016. doi:10.1145/2897518.2897543.

40 Shai Evra, Tali Kaufman, and Gilles Zémor. Decodable quantum LDPC codes beyond the square root distance barrier using high dimensional expanders. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 218–227. IEEE, 2020. doi:10.1109/FOCS46700.2020.00029.

41 Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996. doi:10.1145/226643.226652.

42 Uriya A. First and Tali Kaufman. On good 2-query locally testable codes from sheaves on high dimensional expanders. *CoRR*, abs/2208.01778, 2022. doi:10.48550/arXiv.2208.01778.

43 Jason Gaitonde, Max Hopkins, Tali Kaufman, Shachar Lovett, and Ruizhe Zhang. Eigenstripping, spectral decay, and edge-expansion on posets. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPIcs*, pages 16:1–16:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.16.

44 Oded Goldreich and Shmuel Safra. A combinatorial consistency lemma with application to proving the PCP theorem. *SIAM J. Comput.*, 29(4):1132–1154, 2000. doi:10.1137/S0097539797315744.

45 Louis Golowich. Improved product-based high-dimensional expanders. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, University of Washington, Seattle, Washington, USA (Virtual Conference), August 16-18, 2021*, volume 207 of *LIPIcs*, pages 38:1–38:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.38.

46 Louis Golowich. From grassmannian to simplicial high-dimensional expanders. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1639–1648. IEEE, 2023. doi:10.1109/FOCS57990.2023.00099.

47 Roy Gotlib and Tali Kaufman. List agreement expansion from coboundary expansion. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 61:1–61:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.61.

48 M. Gromov. Singularities, expanders and topology of maps. part 2: from combinatorics to topology via algebraic isoperimetry. *Geom. Funct. Anal.*, 20:416–526, 2010. doi:10.1007/s00039-010-0073-8.

49 Anna Gundert and Uli Wagner. On eigenvalues of random complexes. *Israel Journal of Mathematics*, 216:545–582, 2016.

50 Tom Gur, Noam Lifshitz, and Siqi Liu. Hypercontractivity on high dimensional expanders. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 176–184. ACM, 2022. doi:10.1145/3519935.3520004.

51 Tom Gur, Dor Minzer, Guy Weissenberg, and Kai Zhe Zheng. 3-query rldcs are strictly stronger than 3-query ldcs. *arXiv preprint arXiv:2512.12960*, 2025. doi:10.48550/arXiv.2512.12960.

52 Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2002.

53 Max Hopkins and Ting-Chun Lin. Explicit lower bounds against Ω(n)-rounds of sum-of-squares. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 662–673. IEEE, 2022. doi:10.1109/FOCS54457.2022.00069.

54 Jun-Ting Hsieh, Alexander Lubotzky, Sidhanth Mohanty, Assaf Reiner, and Rachel Yun Zhang. Explicit lossless vertex expanders. *arXiv preprint arXiv:2504.15087*, 2025. doi:10.48550/arXiv.2504.15087.

55 Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: Simplified, optimized, and derandomized. *SIAM J. Comput.*, 39(4):1637–1665, 2010. doi:10.1137/080734030.

56 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query pcps. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 131–140, 2009. doi:10.1145/1536414.1536435.

57 Fernando Granha Jeronimo, Shashank Srivastava, and Madhur Tulsiani. Near-linear time decoding of ta-shma's codes via splittable regularity. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1527–1536. ACM, 2021. doi:10.1145/3406325.3451126.

58 Tali Kaufman, David Kazhdan, and Alexander Lubotzky. Isoperimetric inequalities for ramanujan complexes and topological expanders. *Geometric and Functional Analysis*, 26(1):250–287, 2016.

59 Tali Kaufman and David Mass. Local-to-global agreement expansion via the variance method. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 74:1–74:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.74.

60 Tali Kaufman and Izhar Oppenheim. Construction of new local spectral high dimensional expanders. In *STOC'18—Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 773–786. ACM, New York, 2018. doi:10.1145/3188745.3188782.

61 Tali Kaufman and Izhar Oppenheim. Coboundary and cosystolic expansion from strong symmetry. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, Glasgow, Scotland (Virtual Conference), July 12-16, 2021*, volume 198 of *LIPIcs*, pages 84:1–84:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.84.

62 Tali Kaufman and Izhar Oppenheim. High dimensional expansion implies amplified local testability. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPIcs*, pages 5:1–5:10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.5.

63 Tali Kaufman, Izhar Oppenheim, and Shmuel Weinberger. Coboundary expansion of coset complexes. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1722–1731, 2025. doi:10.1145/3717823.3718136.

64 Jonathan Leake, Kasper Lindberg, and Shayan Oveis Gharan. Optimal trickle-down theorems for path complexes via c-lorentzian polynomials with applications to sampling and log-concave sequences. *CoRR*, abs/2503.01005, 2025. doi:10.48550/arXiv.2503.01005.

65 Nathan Linial and Roy Meshulam. Homological connectivity of random 2-complexes. *Comb.*, 26(4):475–487, 2006. doi:10.1007/s00493-006-0027-9.

66 Siqi Liu, Sidhanth Mohanty, and Elizabeth Yang. High-dimensional expanders from expanders. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, Seattle, Washington, USA, January 12-14, 2020*, volume 151 of *LIPIcs*, pages 12:1–12:32. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.12.

67 Alexander Lubotzky, Beth Samuels, and Uzi Vishne. Explicit constructions of Ramanujan complexes of type $\tilde{A}_d$. *European J. Combin.*, 26(6):965–993, 2005. doi:10.1016/j.ejc.2004.06.007.

68 Or Meir. Combinatorial pcps with short proofs. *Comput. Complex.*, 25(1):1–102, 2016. doi:10.1007/S00037-015-0111-X.

69 Dana Moshkovitz and Ran Raz. Two-query pcp with subconstant error. *Journal of the ACM (JACM)*, 57(5):1–29, 2008.

70 Ryan O'Donnell and Kevin Pratt. High-dimensional expanders from chevalley groups. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, Philadelphia, PA, USA, July 20-23, 2022*, volume 234 of *LIPIcs*, pages 18:1–18:26. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.18.

71 Ryan O'Donnell and Noah G. Singer. Low-soundness direct-product testers and pcps from kaufman-oppenheim complexes. *CoRR*, abs/2511.10514, 2025. doi:10.48550/arXiv.2511.10514.

72 Izhar Oppenheim. Local spectral expansion approach to high dimensional expanders Part II: Mixing and geometrical overlapping. *Discrete Comput. Geom.*, 64(3):1023–1066, 2020. doi:10.1007/s00454-019-00117-7.

73 Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical LDPC codes. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 375–388. ACM, 2022. doi:10.1145/3519935.3520017.

74 David B Surowski. Covers of simplicial complexes and applications to geometry. *Geometriae Dedicata*, 16(1):35–62, 1984.

Optimal Depth-Three Circuits for Inner Product

Mohit Gurumukhani
Cornell University, Ithaca, NY, USA

Daniel Kleber
Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA, USA

Ramamohan Paturi
Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA, USA

Christopher Rosin
Constructive Codes, USA

Navid Talebanfard
University of Sheffield, UK

Abstract

We show that Inner Product in $2n$ variables, $\mathsf{IP}_n(x,y) = x_1y_1 \oplus \ldots \oplus x_ny_n$, can be computed by depth-3 bottom fan-in 2 circuits of size $\mathsf{poly}(n) \cdot (9/5)^n$, matching the lower bound of Göös, Guan, and Mosnoi (Inform. Comput.'24). Our construction is obtained via the following steps.

1. We provide a general template for constructing optimal depth-3 circuits with bottom fan-in k for an arbitrary function f. We do this in two steps. First, we partition $f^{-1}(1)$ into *orbits* of its automorphism group. Second, for each orbit, we construct *one* k-CNF that (a) accepts the largest number of inputs from that orbit and (b) rejects all inputs rejected by f.
2. We instantiate the template for IP_n and $k = 2$. Guided by the intuition (which we call *modularity principle*) that optimal 2-CNFs can be constructed by taking the conjunction of variable-disjoint copies of smaller 2-CNFs, we use computer search to identify a small set of *building block* 2-CNFs over at most 4 variables.
3. We again use computer search to discover appropriate combinations (disjoint conjunctions) of building blocks to arrive at optimal 2-CNFs and analyze them using techniques from analytic combinatorics.

We believe that the approach outlined in this paper can be applied to a wide range of functions to determine their depth-3 complexity.

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases Depth 3 circuits, Circuit lower bounds, Inner product, Analytic combinatorics

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.27

Related Version *Full Version*: `https://arxiv.org/abs/2601.04446`

Supplementary Material *Software*: `https://github.com/mjguru/Inner-Product` archived at `swh:1:dir:22913678e86e5ba7afef126b288dbeeeb431a715`

Funding *Mohit Gurumukhani*: Supported by a Sloan Research Fellowship, NSF CAREER Award 2045576, and NSF Award CCF-2514586.
Daniel Kleber: Partially supported by NSF grant 2212136.
Ramamohan Paturi: Partially supported by NSF grant 2212136.

1 Introduction

A Σ_3 circuit is a depth-3 OR-AND-OR circuit with unbounded fan-in. Despite their simplicity, these circuits are surprisingly powerful. Any n-variate Boolean function can be computed by a Σ_3 circuit of size $O(2^{n/2})$ [4], a quadratic speed-up over the trivial construction. More

© Mohit Gurumukhani, Daniel Kleber, Ramamohan Paturi, Christopher Rosin, and Navid Talebanfard;
licensed under Creative Commons License CC-BY 4.0

strikingly, Valiant [19] showed that any linear-size fan-in 2 circuit of logarithmic depth can be computed by a Σ_3 circuit of size $2^{O(n/\log\log n)}$ size. Therefore, truly exponential lower bounds for depth-3 circuits imply non-linear lower bounds for log-depth circuits. However, the best known depth-3 lower bound for an explicit function is only $2^{\Omega(\sqrt{n})}$ and beating this barrier remains a major open problem in the field [9].

Even a further restriction of depth-3 circuits shows surprising power. Let Σ_3^k be the class of Σ_3 circuits in which every bottom OR gate is connected to at most k input bits. Equivalently, these circuits can be viewed as disjunctions of k-CNFs and the number of these formulas serves as a natural measure of size for the circuit. We denote the smallest number of k-CNFs needed to express a function f by $\mathsf{Size}_3^k(f)$. [1] Valiant's depth reduction shows that linear-size log-depth circuits can in fact be computed by $\Sigma_3^{n^\epsilon}$ circuits of subexponential size where $\epsilon > 0$ can be chosen arbitrarily. More recently, Golovnev, Kulikov, and Williams [7] showed that *unrestricted* circuits of size cn for every $c < 3.9$ can be computed by Σ_3^{16} circuits of size $2^{\epsilon n}$ for some $\epsilon < 1$. Therefore, *near-maximal* lower bounds for Σ_3^{16}, i.e., lower bounds of the form $2^{n-o(n)}$, imply unrestricted circuit lower bounds beyond the state-of-the-art which is only $3.1n - o(n)$ [5, 12].

Near-maximal lower bounds are only known for Σ_3^2 and are due to Paturi, Saks, and Zane [17]. For $k \geq 3$, the state-of-the-art is due to Paturi, Pudlák, Saks, and Zane [15] who showed a Σ_3^k lower bound of $2^{cn/k}$ where $c > 1$ is a constant, building on a previous work by Paturi, Pudlák, and Zane [16] who proved a tight lower bound of $\Omega(2^{n/k})$ for Parity.

Near-maximal lower bound arguments

Paturi, Saks, and Zane [17] proved a crucial property of 2-CNFs. They showed that any 2-CNF with $2^{\Omega(n)}$ satisfying assignments, must accept a large *projection*, i.e., an affine space with dimension $\Omega(n)$ defined by equations of the form $x = 0, 1$, $x = y$, and $x = y + 1$. Therefore, any good *affine disperser*, i.e., any function that is not constant under large affine spaces, requires Σ_3^2 circuits of size $2^{n-o(n)}$. Several explicit constructions of such functions are known including with asymptotically optimal dependence on dimension - $O(\log(n))$ by Li [13]. Affine dispersers serve as a natural candidate for near-maximal lower bounds; the known explicit constructions are complicated poly-time algorithms, and the only known circuit construction has super-linear size [10]. Furthermore, state-of-the-art unrestricted circuit lower bounds hold for affine dispersers [5, 12].

In order to prove strong Σ_3^k lower bounds for affine dispersers, we cannot hope to apply the projection argument; there are 3-CNF formulas accepting exponentially many assignments which only accept projections of constant dimension [6]. However, a more general statement can be true. In particular, we do not know the answer to the following question: *is it true that every k-CNF with exponentially many satisfying assignments must accept an affine space of linear dimension?* An affirmative answer yields a near-maximal Σ_3^k lower bound for every good affine disperser. However, if the answer is negative, then there may not be a unified proof showing the hardness of *every* affine disperser. Instead, the hardness of each affine disperser may have to be demonstrated in a unique way. This suggests that by determining the complexity of *concrete* affine dispersers, we may be able to develop techniques that will ultimately allow us to prove near-maximal lower bounds.

[1] Note that this quantity is off by a factor of n^k from the smallest Σ_3^k sized circuit computing f. However, since k is small enough in this paper and the circuit size is exponential, this difference is immaterial.

Depth-3 complexity of Inner Product

The Inner Product modulo 2 function IP_n on n coordinates defined as $\mathsf{IP}_n(x,y) = x_1y_1 \oplus \cdots \oplus x_ny_n$ serves as an ideal starting point since it has a very simple and concrete definition and it is known to be a moderately good affine disperser (for dimension $n+1$). Frankl, Gryaznov, and Talebanfard [6] pointed out that even for Σ_3^2 circuits, the exact complexity of IP_n is unclear as the projection technique fails to yield a strong bound. Golovnev, Kulikov, and Williams [7] initiated the study of depth-3 complexity of IP_n for independent reasons.

IP_n can be trivially computed by Σ_3^2 circuits of size $O(2^n)$: $\mathsf{IP}_n = \bigvee_{I\subseteq[n],|I|=\text{odd}}(\wedge_{i\in I}(x_i \wedge y_i) \wedge_{i\notin I}(\neg x_i \vee \neg y_i))$. A simple lower bound of $\Omega(2^{n/2})$ follows by a reduction from Parity. Göös, Guan, and Mosnoi [8] showed, somewhat unexpectedly, that both of these bounds can be improved: $(9/5)^n \simeq 2^{0.847n} \leq \mathsf{Size}_3^2(\mathsf{IP}_n) \leq 2^{0.965n}$. They proved the lower bound by identifying a hard probability distribution on inputs to IP_n and characterizing the formulas that maximize the probability of accepting an input under this restriction. For the upper bound, they used the fomulas in the lower bound to cover solutions with small weight, i.e., the number of coordinates on which both x and y variables are 1 is small. For the remaining solutions, they used a small number of the *matching* formulas. We will discuss these formulas later as we will also use them in our construction.

Amano [1] improved the upper bound even further to $2^{0.952n}$; this was done by observing that a depth-3 circuit for IP_n can be constructed by partitioning the coordinates into blocks of size t for a small value of t, and then combining depth-3 circuits that compute IP_t and $\neg\mathsf{IP}_t$ on these blocks. To find such depth-3 circuits for a fixed t, the paper used computer search.

Our results

In this paper, we determine $\mathsf{Size}_3^2(\mathsf{IP}_n)$ up to polynomial factors by showing that the lower bound of [8] is, in fact, tight.

▶ **Theorem 1** (Main result). $\mathsf{Size}_3^2(\mathsf{IP}_n) \leq \mathsf{poly}(n) \cdot (9/5)^n$

The paper makes two main contributions. Firstly, we give a general framework for the construction of optimal Σ_3^k circuits for a large class of *highly symmetric* functions. We show that it is sufficient to find consistent *extremal* k-CNFs to construct Σ_3^k optimal circuits for a function f. Secondly, our approach to circuit design is somewhat novel: Guided by modularity principle we use computer search to find the building blocks and search again for optimal disjoint conjunctions of building blocks to arrive at the desired constructions. Our methodology is general and flexible and we are optimistic that it can be applied to other functions.

2 A recipe for constructing depth-3 circuits

Let $f : \{0,1\}^n \to \{0,1\}$ be an arbitrary Boolean function and consider the task of proving a tight upper bound on $\mathsf{Size}_3^k(f)$. In this section, we will provide a general framework for tackling such a task and will instantiate it to construct optimal circuits for IP. Our framework reduces the task of constructing Σ_3^k circuits for f to the task of constructing many different k-CNFs, each *consistent* with f, and each one maximizing the number of satisfying assignments with a certain property. We say a k-CNF F is consistent with the function f if for all $x \in \{0,1\}^n$, it holds that $F(x) = 1 \implies f(x) = 1$, i.e., $\mathbf{sol}(F) \subseteq f^{-1}(1)$, where $\mathbf{sol}(F)$ is the set of satisfying assignments of F. We use the term *consistent* k-CNF to mean k-CNF consistent with a function when the function is clear from the context.

To describe our framework, we will require some basic concepts from group theory; see Section 6 for a quick refresher. We define the notion of an automorphism group and orbits of a function as follows:

▶ **Definition 2** (Automorphism group and orbits). *For any $f : \{0,1\}^n \to \{0,1\}$, define the* automorphism group of f, Aut_f, *to be the group of permutations π such that for all $x \in \{0,1\}^n$, $f(x_{\pi(1)}, \ldots, x_{\pi(n)}) = f(x_1, \ldots, x_n)$. The action $\mathsf{Aut}_f \times f^{-1}(1) \mapsto f^{-1}(1)$ of Aut_f on $f^{-1}(1)$ is defined as follows. $\pi \in \mathsf{Aut}_f$ and $x_1 \ldots x_n \in f^{-1}(1)$ maps to $x_{\pi(1)} \ldots x_{\pi(n)}$. We denote the set of orbits of this action by Orb_f.*

Note that for each $S \in \mathsf{Orb}_f$ we have $S \subseteq f^{-1}(1)$. We now define the following quantities associated with optimal consistent k-CNFs for each orbit.

▶ **Definition 3.** *For any $f : \{0,1\}^n \to \{0,1\}$ and $S \in \mathsf{Orb}_f$, define*

$$\mu_{f,S,k} := \max_{F:F \text{ is an } n\text{-variate consistent } k\text{-CNF}} |\mathbf{sol}(F) \cap S|,$$
$$\rho_{f,S,k} := |S|/\mu_{f,S,k}, \text{ and } \rho^*_{f,k} = \max_{S \in \mathsf{Orb}_f} \rho_{f,S,k}.$$

In words, $\mu_{f,s,k}$ is the largest number of assignments S that are accepted by a consistent k-CNF, $\rho_{f,s,k}$ is a lower bound on $\mathsf{Size}^k_3(f)$ due to S and $\rho^*_{f,k}$ is the maximum of such lower bounds over all orbits. We now present the main result of our framework which is proved in Section 2.

▶ **Lemma 4.** *For all $k, n \in \mathbb{N}$ and all functions $f : \{0,1\}^n \to \{0,1\}$, we have $\rho^*_{f,k} \leq \mathsf{Size}^k_3(f) \leq O\left(n \cdot |\mathsf{Orb}_f| \cdot \rho^*_{f,k}\right)$.*

We note that if $|\mathsf{Orb}_f|$ of a function f is bounded by a polynomial, then $\mathsf{Size}^k_3(f)$ and $\rho^*_{f,k}$ are within a polynomial factor of each other. Therefore, to determine the Σ^k_3 complexity of f, it is enough to find a consistent k-CNF with the maximum number of solutions in S for every orbit S. This is indeed the case for IP_n and is exactly the strategy we will use to construct optimal Σ^2_3 circuits for it.

Depth-3 upper bound – Proof of Lemma 4

We will first show that we can cover an entire orbit S by using isomorphic copies of a consistent k-CNF F where the number of copies is bounded by $O(n|S|/|\mathbf{sol}(F) \cap S|)$.

▷ Claim 5. Let $f : \{0,1\}^n \to \{0,1\}$ be an arbitrary function and let F be a k-CNF consistent with f. Then, for any $S \in \mathsf{Orb}_f$, there exist consistent k-CNFs $F_1, \ldots, F_t$ such that $S \subseteq \mathbf{sol}(F_1) \cup \ldots \cup \mathbf{sol}(F_t)$ where $t = O(n|S|/|\mathbf{sol}(F) \cap S|)$.

We first show how the proof of Lemma 4 follows from the claim and present the proof of the claim following it.

Proof of Lemma 4. We prove the lower bound and the upper bound part of the inequality separately. For the lower bound, fix any Σ^k_3 circuit C for f . We express $C = \bigvee_{i=1}^m F_i$ where $m \in \mathbb{N}$ and each F_i is a k-CNF. We will show that for all $S \in \mathsf{Orb}_f$, it holds that $m \geq \rho_{f,S,k}$, which implies $m \geq \rho^*_{f,k}$. Fix any $S \in \mathsf{Orb}_f$. We observe that each F_i must be consistent with f and so for all $i \in [m]$, we must have that $|\mathbf{sol}(F_i) \cap S| \leq \mu_{f,S,k}$. Moreover, each assignment in S must be covered by some F_i. Indeed, for each $\alpha \in S$, since $C(\alpha) = f(\alpha) = 1$ and $C = \bigvee_{i=1}^m F_i$, there must exist some $i \in [m]$ such that $F_i(\alpha) = 1$. Hence, we must have $m \geq |S|/\mu_{f,S,k} = \rho_{f,S,k}$ as desired.

For the upper bound, we construct a circuit C for f as follows. let $S \in \mathsf{Orb}_f$ and let F_S be the k-CNF consistent with f that maximizes $|\mathbf{sol}(F_S) \cap S|$. By definition, $|\mathbf{sol}(F_S) \cap S| = \mu_{f,S,k}$. From Claim 5 we get consistent k-CNFs $F_{S,1}, \ldots, F_{S,t}$ which cover S where $t = t(S) = O(n|S|/|\mathbf{sol}(F) \cap S|) = O(n\rho_{f,S,k})$. We define $C = \bigvee_{S \in \mathsf{Orb}_f, i \in [t(S)]} F_{S,i}$. C accepts all assignments from each of the orbits S. Since each $F_{S,i}$ is consistent with f, C correctly rejects all assignments from $f^{-1}(0)$ and hence C indeed computes the function f. The total number of k-CNFs in C $O(n \sum_{S \in \mathsf{Orb}_f} \rho_{f,S,k}) = O(n \cdot |\mathsf{Orb}_f| \cdot \rho^*_{f,k})$ as claimed. ◀

Proof of Claim 5. Let $S \in \mathsf{Orb}_f$ and F be an n-variate k-CNF consistent with f. Let $t = 2|S| \ln |S|/|\mathbf{sol}(F) \cap S| = O(n|S|/|\mathbf{sol}(F) \cap S|)$. For each $i \in [t]$, define $F_i = F^{\pi_i}$ to be the permutation of F under a random permutation $\pi_i \in \mathsf{Aut}_f$. Fix any $\alpha \in S$. It follows (see Lemma 21 for the general group theoretic claim and its proof) that for a random $\pi \in \mathsf{Aut}_f$, $\Pr[\alpha \in \mathbf{sol}(F^\pi)] = |\mathbf{sol}(F) \cap S|/|S|$, which implies

$$\begin{aligned}\Pr[\alpha \notin \mathbf{sol}(F_1) \cup \ldots \cup \mathbf{sol}(F_t)] &\leq (1 - |\mathbf{sol}(F) \cap S|/|S|)^t \\ &\leq \exp\left(-t \cdot |\mathbf{sol}(F) \cap S|/|S|\right) \\ &< 1/|S|\end{aligned}$$

where the last inequality follows by the choice of t and the fact that $|S| \leq 2^n$. By a union bound over all $\alpha \in S$, we conclude that there exists a choice of F_is covering all assignments in S. ◁

3 Construction for IP

In this section, we instantiate our framework from Section 2 to construct optimal Σ^2_3 circuits for IP. We will prove the following main lemma which together with Lemma 4 proves Theorem 1. In the following, we also use IP^0_n for IP_n and $\mathsf{IP}^1_n = 1 - \mathsf{IP}^0_n$ for the complement of IP^0_n.

▶ **Lemma 6.** *There exists a constant c such that for all integer $n \geq 2$, $\rho^*_{\mathsf{IP}_n,2} \leq n^c \cdot (9/5)^n$.*

To prove the lemma, we will identify the orbits of IP_n in Section 3.1 and the building block 2-CNFs in Section 3.2. In Section 3.3, we will reduce Lemma 6 to a version of itself (Lemma 14) that holds only for infinitely many even n and reduce the constructions for IP^1_n to those of $\mathsf{IP}^0_{n'}$. Our plan is to construct 2-CNFs for each of the orbits by taking a *disjoint* conjunction of copies of a small number of the building-block 2-CNFs. For this purpose, we divide the orbits into six regions in Section 3.4, state Lemma 16 which is a version of Lemma 14 specialized to regions. We prove Lemma 16 in Section 7.

3.1 Orbits of IP and spectra of 2-CNFs

We now describe the orbits of IP^1_n and IP^0_n and introduce the notion of the *spectrum* of a 2-CNF.

We observe that $\mathsf{Aut}_{\mathsf{IP}^1_n} = \mathsf{Aut}_{\mathsf{IP}^0_n}$ and they contain all permutations that permute the n coordinates, and permute x_i and y_i within each coordinate. For input $(x, y) \in (\{0,1\}^n)^2$, define the following quantities:

1. $d_2(x, y) = |\{i \in [n] : x_i = y_i = 1\}|$.
2. $d_1(x, y) = |\{i \in [n] : x_i \neq y_i\}|$.
3. $d_0(x, y) = |\{i \in [n] : x_i = y_i = 0\}|$.

We have $d_2(x,y) + d_1(x,y) + d_0(x,y) = n$. More importantly, for $j \in \{0,1,2\}$, $d_i(x,y)$ are invariant under a permutation $\pi \in \mathsf{Aut}_{\mathsf{IP}^1_n}$, that is, $d_j(x',y') = d_j(x,y)$ where $(x',y') = \pi(x,y)$. We also observe that if $d_j(x,y) = d_j(x',y')$ for all $j \in \{0,1,2\}$, then there exists a permutation $\pi \in \mathsf{Aut}_{\mathsf{IP}^1_n}$ such that $\pi(x,y) = (x',y')$. Hence, d_j precisely determine the orbits of IP^1_n and IP^0_n In particular we have $|\,\mathsf{Orb}_{\mathsf{IP}^1_n}\,| = |\,\mathsf{Orb}_{\mathsf{IP}^0_n}\,| = O(n^2)$. We parameterize the orbits of IP_n as follows:

▶ **Definition 7** (Orbits of IP^1_n and IP^0_n)**.** *For $d_2, d_1, d_0 \in [n]$ such that $d_2 + d_1 + d_0 = n$, let $S(d_2,d_1,d_0) \subseteq (\{0,1\}^n)^2$ denote the set of assignments $(x,y) \in (\{0,1\}^n)^2$ such that $d_j(x,y) = d_j$ for all $j \in \{0,1,2\}$. Note that $S(d_2,d_1,d_0)$ is an orbit of the function $\mathsf{IP}^{d_2 \mod 2}_n$ and for any $(x,y) \in S(d_2,d_1,d_0)$, $\mathsf{IP}^0_n(x,y) = d_2 \mod 2$.*

We record the following fact regarding size of each of the orbits:

▶ **Fact 8.** $|S(d_2,d_1,d_0)| = \binom{n}{d_2}\binom{n-d_2}{d_0}2^{n-d_2-d_0} = \binom{n}{d_2}\binom{d_1+d_0}{d_1}2^{d_1}$.

We now define the spectrum of a 2-CNF F as the generating function that encodes the number of its satisfying assignments from each of the orbits.

▶ **Definition 9** (Spectrum of 2-CNF)**.** *For a 2-CNF F over $2n$ variables, define the spectrum of F to be the generating function $G(x,y,z) = \sum_{p,q,r \in \mathbb{N}} C_{p,q,r} x^p y^q z^r$ where $C_{p,q,r}$ is the number of assignments $\alpha \in \{0,1\}^{2n}$ such that $F(\alpha) = 1$, $d_2(\alpha) = p, d_1(\alpha) = q$, and $d_0(\alpha) = r$.*

The spectrum of 2-CNF of F is always a finite homogeneous polynomial. If F is consistent with IP^0_n, then $C_{p,q,r} = 0$ if (p,q,r) is not an orbit of IP^0_n.

3.2 Building blocks for our constructions

In this section, we identify a small number of building block 2-CNFs which will be used to construct 2-CNFs; we will also define the notion of a disjoint conjunction of the building blocks. Since any function over 2 variables can be expressed as a 2-CNF, our building block descriptions will be expressed as AND of functions over 2 variables.

▶ **Definition 10** (Building block 2-CNFs and their spectra)**.** *We define the following building blocks:*

- $\mathbf{Id}_2$ *denotes the 2-CNF over 1 coordinate (2 variables) that accepts iff both variables equal 1; its spectrum is x.*
- $\mathbf{Id}_1$ *denotes the 2-CNF over 1 coordinate that accepts iff both variables do not equal each other; its spectrum is $2y$.*
- $\mathbf{Id}_0$ *denotes the 2-CNF over 1 coordinate that accepts iff both variables equal 0; its spectrum is z.*
- **NAND** *denotes the 2-CNF over 1 coordinate that accepts iff both variables do not equal 1; its spectrum is $2y + z$.*
- **Matching** *denotes the 2-CNF over over 2 coordinates defined as follows. On input $(x,y) \in (\{0,1\}^2)^2$, the 2-CNF accepts iff $x_1 = x_2$, and $y_1 = y_2$; its spectrum is $x^2 + 2y^2 + z^2$.*
- **2Imp** *denotes the 2-CNF over 2 coordinates defined as follows. On input $(x,y) \in (\{0,1\}^2)^2$, the 2-CNF accepts iff the following holds: $x_1 = x_2$, $x_1 \implies y_1$, and $x_2 \implies y_2$; its spectrum is $x^2 + y^2 + 2yz + z^2 = x^2 + (y+z)^2$.*

▶ **Definition 11** (Disjoint conjunction)**.** *Let F_1 and F_2 be arbitrary 2-CNFs over n_1 and n_2 variables respectively. Then, $F_1 \wedge F_2$ is the 2-CNF over $n_1 + n_2$ variables obtained by adding all clauses from both F_1 and F_2 where we associate the first n_1 variables with variables of F_1 and the last n_2 variables with variables of F_2.*

We next record a couple of crucial observations regarding the spectrum of the disjoint conjunction of 2-CNFs and 2-CNFs consistent with either IP^0_n or IP^1_n.

▶ **Fact 12** (Spectra of disjoint conjunctions 2-CNFs). *Let F_1 and F_2 be arbitrary 2-CNFs over n_1 and n_2 variables with spectra G_1 and G_2 respectively. Then, the spectrum of $F_1 \wedge F_2$ equals $G_1 \cdot G_2$. Furthermore, if F_1 is consistent with $\mathsf{IP}^{b_1}_{n_1}$ and that F_2 is consistent with $\mathsf{IP}^{b_2}_{n_2}$ for some $b_1, b_2 \in \{0,1\}$, then $F_1 \wedge F_2$ is consistent with $\mathsf{IP}^{b_1 \oplus b_2}_{n_1+n_2}$.*

We note that each building block from Section 3.2 is consistent with IP^b_n for some $b \in \{0,1\}$ for an appropriate n. This implies:

▶ **Corollary 13.** *Let F be a 2-CNF over n variables obtained by (repeatedly) applying disjoint conjunction to building blocks from Definition 10. Then, F is consistent with IP^b_n for some $b \in \{0,1\}$.*

3.3 Reducing to infinitely often case

We now state a version of Lemma 6 that only holds for infinitely many n, each a multiple of some even constant.

▶ **Lemma 14.** *There exist constants c, even $m \in \mathbb{N}$ such that the following holds: For all $p, q, r, n \in \mathbb{N}$ such that $n = p+q+r$ and m divides each of p, q, r and n, there exists a 2-CNF F over n variables that is consistent with IP^0_n and satisfies:*

$$\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} \leq n^c \cdot (9/5)^n.$$

We will now show that Lemma 6 follows from Lemma 14. Lemma 14 will be proven by reducing it to Lemma 16 in Section 3.4.

Proof of Lemma 6. Fix any integer $n \geq 2$ and let c_0 and m be as given by Lemma 14. Now consider any orbit S of IP_n. It suffices to show that there exists a 2-CNF F consistent with IP_n such that $\frac{|S|}{|\mathbf{sol}(F) \cap S|} \leq n^c \cdot (9/5)^n$ where c is a constant. As discussed in Section 3.1, S is parameterized by integers $p, q, r \in \mathbb{N}$ such that $p+q+r = n$ where p is odd and $S = S(p,q,r)$.

Let $p' \leq p, q' \leq q, r' \leq r$ be the largest multiples of m less than or equal to p, q, r respectively. Let $n' = p'+q'+r'$. n' must also be a multiple of m and $n - n' \leq 3m$. We apply Lemma 14 for p', q', r', n' to obtain 2-CNF F' over n' variables such that $\frac{|S(p',q',r')|}{|\mathbf{sol}(F') \cap S(p',q',r')|} \leq (n')^{c_0} \cdot (9/5)^{n'}$. Let F be the 2-CNF over n variables obtained by the disjoint conjunction of $p - p'$ copies of $\mathbf{Id}_2$, $(q - q')$ copies of $\mathbf{Id}_1$, $(r - r')$ copies of $\mathbf{Id}_0$, and one copy of F'.

We know that F' is consistent with $\mathsf{IP}^{p' \pmod 2}_{n'}$, $\mathbf{Id}_1$ and $\mathbf{Id}_0$ are consistent with IP^0_1 and $\mathbf{Id}_2$ is consistent with IP^1_1. Applying Fact 12, we get that F is consistent with $\mathsf{IP}^{p \pmod 2}_n$ as desired. Next, using Fact 8 and the fact that $n - n' \leq 3m, p - p' \leq m, q - q' \leq m, r - r' \leq m$, we infer that there exists a constant c_1 such that $\frac{|S(p,q,r)|}{|S(p',q',r')|} \leq n^{c_1}$. We also have $|\mathbf{sol}(F) \cap S(p,q,r)| = 2^{q-q'} |\mathbf{sol}(F') \cap S(p',q',r')|$. We thus have

$$\begin{aligned}
\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} &= \frac{|S(p,q,r)|}{|S(p',q',r')|} \cdot \frac{|S(p',q',r')|}{|\mathbf{sol}(F) \cap S(p,q,r)|} \\
&= 2^{q'-q} \cdot \frac{|S(p,q,r)|}{|S(p',q',r')|} \cdot \frac{|S(p',q',r')|}{|\mathbf{sol}(F') \cap S(p',q',r')|} \\
&\leq 2^m \cdot n^{c_1} \cdot n^{c_0} \cdot (9/5)^n \\
&\leq n^c \cdot (9/5)^n
\end{aligned}$$

for a suitable constant c. ◀

3.4 Splitting orbits into regions

In this subsection, we divide the orbits of IP_n^1 and IP_n^0 into six regions and prove Lemma 14 for each of the six regions.

▶ **Definition 15.** *For $i \in [6]$ and $n \in \mathbb{N}$, let $\mathcal{R}_i(n) \subseteq \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ is a set of triples $(p, q, r) \in \mathbb{N}$ such that $p + q + r = n$ and*

1. $\mathcal{R}_1(n)$*:* $\frac{1}{2}n - \frac{25}{32}p \geq 0$*,* $\frac{5}{4}n - \frac{25}{32}p - \frac{25}{4}r \geq 0$*, and* $-\frac{5}{4}n + \frac{25}{16}p + \frac{25}{4}r \geq 0$*.*
2. $\mathcal{R}_2(n)$ *:* $\frac{1}{2}n - \frac{25}{32}p \leq 0$*.*
3. $\mathcal{R}_3(n)$ *:* $\frac{1}{2}n - \frac{25}{32}p \geq 0$*,* $\frac{5}{4}n - \frac{25}{32}p - \frac{25}{4}r \leq 0$*, and* $\frac{1}{4}n - p \leq 0$*.*
4. $\mathcal{R}_4(n)$ *:* $\frac{1}{2}n - \frac{25}{32}p \geq 0$*,* $-\frac{5}{4}n + \frac{25}{16}p + \frac{25}{4}r \leq 0$*,* $-\frac{1}{20}n + r \geq 0$*, and* $\frac{1}{4}n - p \leq 0$*.*
5. $\mathcal{R}_5(n)$ *:* $\frac{1}{2}n - \frac{25}{32}p \geq 0$*, and* $-\frac{1}{20}n + r \leq 0$*.*
6. $\mathcal{R}_6(n)$ *:* $\frac{1}{4}n - p \geq 0$*.*

From here on, we refer to (p, q, r) as the orbit corresponding to it.

For each region $\mathcal{R}_i$ and each orbit in the region, we will construct a 2-CNF such that its spectrum satisfies the following property.

▶ **Lemma 16.** *For all $i \in [6]$, there exist constants K_i and even $m_i \in \mathbb{N}$ such that the following holds: For all $p, q, r, n \in \mathbb{N}$ such that $(p, q, r) \in \mathcal{R}_i$, $n = p + q + r$, and m_i divides each of p, q, r, there exists a 2-CNF F over $2n$ variables that is consistent with IP_n^0 and satisfies:*

$$\frac{|S(p, q, r)|}{|\mathbf{sol}(F) \cap S(p, q, r)|} \leq n^{K_i} \cdot (9/5)^n.$$

We first show that Lemma 14 follows from Lemma 16. We will prove Lemma 16 in Section 7.

Proof of Lemma 14. We apply Lemma 16 and obtain even m_i and K_i. Let m be the least common multiple of m_1, m_2, m_3, m_4, m_5, and m_6 and let c to be the maximum of $K_1, K_2, K_3, K_4, K_5, K_6$. We observe that the six regions cover the set of all triples such that $p + q + r = n$, completing our proof. ◀

4 An overview of constructions

In this section we provide an overview of our constructions that will lead to the proof of Lemma 16.

For each $i \in [6]$ and $(p, q, r) \in \mathcal{R}_i$, we construct a 2-CNF by taking a disjoint conjunction of an appropriate number of building block 2-CNFs while ensuring that F is consistent with IP_n^0. We then argue that F captures many assignments from $S(p, q, r)$ by lower bounding the coefficient of the monomial $x^p y^q z^r$ in its spectrum. This is the common framework of construction and analysis. To illustrate how the framework works, we provide a sketch of the construction and its analysis for region 1 in the following. Our analysis of the constructions will vary in their techniques across regions, ranging from direct calculations to reducing to an optimization problem that we solve using a numerical optimization software (see Section 7.3 for an example). For a lot of regions, we will use tools from multivariate analytic combinatorics to lower bound the coefficient of the desired monomial.

A more complete description of the constructions for each region and their analysis is presented in Section 7 and Section 8.

Sketch of the construction and analysis for $\mathcal{R}_1$

Let $A = \frac{5}{4}n - \frac{25}{32}p - \frac{25}{4}r, B = \frac{-5}{4}n + \frac{25}{16}p + \frac{25}{4}r, C = \frac{1}{2}n - \frac{25}{32}p$. We select $m_1 = 8000$ (which is not necessarily optimized) and require that m_1 divides p, q, and r. A, B, C are non-negative integers since $(p, q, r) \in \mathcal{R}_1$. We take a disjoint conjunction of A copies of **Matching**, B copies of **2Imp** and $2C$ copies of **NAND** building blocks to construct our 2-CNF F. Since m_1 is even, we have that F is consistent with IP^0_n.

The spectrum of F equals $P(x, y, z) = (x^2 + 2y^2 + z^2)^A (x^2 + (y + z)^2)^B ((2y + z)^2)^C$ and the coefficient of $x^p y^q z^r$ in $P(x, y, z)$ equals $|\mathbf{sol}(F) \cap S(p, q, r)|$. To bound the coefficient, we define $f(u, v) = P(u^{1/2}, v, 1)$. Since $p + q + r = n$ and P is a homogeneous polynomial, the coefficient of $u^{p/2} v^q$ in $f(u, v)$ equals the coefficient of $x^p y^q z^r$ in $P(x, y, z)$. We set the gradient of $f(u, v)$ to 0 and compute the non-negative real solutions of this equation to get the unique *critical point* at $(16, 2)$. We then apply a standard result in analytic combinatorics – Lemma 29 – that states that for *well-behaved* f, the value of the coefficient of $u^{p/2} v^q$ (up to a polynomial factor) equals $\frac{f(16,2)}{16^{p/2} 2^q} = \frac{25^{A+B+C}}{16^{p/2} 2^q} = \frac{5^n}{2^{2p+q}}$ where we used the fact that $A + B + C = n/2$. We now have $|\mathbf{sol}(F) \cap S(p, q, r)| = \frac{5^n}{2^{2p+q}}$ up to a polynomial factor. We upper bound the number of 2-CNFs required to cover the orbit $S(p, q, r)$.

$$\begin{aligned}
\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} &\leq \binom{n}{p}\binom{n-p}{r} \cdot 2^{n-p-r} \cdot \frac{2^{2p+q}}{5^n} \\
&= \binom{n}{p,q,r} \cdot 4^{-r} \cdot (4/5)^n \\
&\leq \sum_{i,j,k \in \mathbb{N}: i+j+k=n} \binom{n}{i,j,k} \cdot 1^i \cdot 1^j \cdot 4^{-k} \cdot (4/5)^n \\
&= (9/4)^n \cdot (4/5)^n = (9/5)^n
\end{aligned}$$

where we used Fact 8 to establish the first inequality.

▶ Remark 17. From above calculations, we can identify that for the orbit of $(4n/9, 4n/9, n/9)$, we require $(9/5)^n$ many 2-CNFs to cover it.

5 Conclusions

The immediate question left open is to determine $\mathsf{Size}^k_3(\mathsf{IP}_n)$ for every $k \geq 3$. We believe that a similar methodology should work for upper bounds. However, the challenge is to find an argument that scales with k. It also remains open to find a non-trivial lower bound argument for $k \geq 3$.

Another related open question is to construct an explicit degree 2 polynomial that requires Σ^k_3 circuits of size $2^{n-o(n)}$ for all constant k; this is open even for $k = 2$. Existentially, Impagliazzo, Paturi, and Zane showed that a random degree 2 polynomial requires maximal Σ^k_3 circuits [11].

How did we discover these circuits?

We here document our journey for constructing optimal circuits for IP_n. We hope this inspires future work to find optimal or close to optimal Σ^k_3 circuits for various explicit functions.

Modular constructions

As a starting point, we believed that optimal circuits for IP_n must be modular, i.e., obtained by combining disjoint copies of smaller 2-CNFs. Our reasons for such a belief are twofold:

All previous best known constructions of IP_n had this property and optimal (and conjectured to be optimal) Σ_3^k circuits for other simple functions such as Parity and Majority have this property.

Searching for better circuits for small n

Our first improvement came from examining the construction of Σ_3^2 circuits for IP_n by Amano [1]. To construct such circuits, the paper used an integer programming solver on a set of constraints to find a disjunction of 7 2-CNFs that compute IP_4^1 and another disjunction of 7 2-CNFs that compute IP_4^0. For the general construction, a combination of these formulas is used.

We searched for better circuits using Kissat SAT solver under a similar set of constraints and identified a disjunction of 14 2-CNF formulas that computes IP_5^1 and another disjunction of 13 2-CNF formulas that computes IP_5^0. Repeated use of these formulas as in Amano's construction gives a slightly improved construction for IP_n with size 2^{cn} where $c \approx 0.9509$. This inspired us to seek further improvements.

Search for small Pareto-optimal building blocks

Suppose A and B are 2-CNFs consistent with IP_n^b for some $b \in \{0,1\}$ and $n \in \mathbb{N}$. Then, we say A dominates B if each coefficient in A's spectrum is at least as large as the corresponding coefficient in B's spectrum. We performed an exhaustive computer search for Pareto-optimal (non-dominated) 2-CNF building blocks for IP_2^0 that identified three distinct Pareto-optimal building blocks: **Matching**, **2Imp** and disjoint conjunction of 2 copies of **NAND**. In particular, **2Imp** was a new building block first identified by this exhaustive search and is *crucial* in our optimal constructions. We also identified other Pareto-optimal building blocks for small values of n that we did not end up using.

Search for compositions of building blocks using disjoint conjunctions

After we identified a suitable set of building blocks, we are left with the task of finding disjoint conjunctions of the building blocks to cover the assignments in each orbit. This task turned out to be highly nontrivial and we resorted to computer search again to gain intuition to guide us towards optimal general constructions. We searched for efficient disjoint conjunctions of building blocks for each orbit of IP_n for some large fixed values of n. For this, we enumerated all distinct compositions of the building blocks. This exercise pointed to the best composition for each orbit and helped identify the hardest orbits.

For the search, we used $n = 50$, $n = 100$, and $n = 200$. We have also enumerated restricted compositions for $n = 400$ and $n = 800$ near the conjectured hardest orbits. The search identified best compositions involving the building blocks **Matching**, **2Imp**, and **NAND** which led to the general constructions. From our best compositions for each value of n, we obtained depth-3 circuits of size $\mathsf{poly}(n) \cdot 2^{cn}$ for IP_n where the dependence of c on n is presented below.

$n = 50$	$c = 0.8344320$
$n = 100$	$c = 0.8414042$
$n = 200$	$c = 0.8447647$
$n = 400$	$c = 0.8463819$
$n = 800$	$c = 0.8471913$

A least-squares fit to the data yields $0.8479999 - 0.6470509/n$, suggesting an eventual convergence to 0.8479999 for large n. This is extremely close to $\log(9/5) = 0.8479969$, which would match the lower bound from [8], motivating us to pursue an upper bound with $c = \log(9/5)$.

Restricted compositions to cover regions

Finally, we analytically solved for optimal constructions using saddle point methods with **Matching**, **2Imp**, and **NAND** as building blocks. Our analysis identified regions 2, 3, and 4 and that optimal constructions for these regions use at most 1 or 2 building blocks. We then performed a search for optimal compositions (using at most 2 types of building blocks) for each of the regions for $n = 200$. The goal of the search was to find a small number of distinct compositions that would cover a region so that the ratio of the size of an orbit to number of solutions in the orbit is at most $(9/5)^n$. This search successfully identified the desired compositions that are sufficient for our purpose.

6 Preliminaries

We use $\log(x)$ to denote logarithm of x with base 2 and $\ln(x)$ to denote the natural logarithm. The binary entropy function $H : [0,1] \to [0,1]$ is defined as $H(x) = x\log(1/x) + (1-x)\log(1/(1-x))$.

We will utilize the following well known approximation for binomial coefficients (see, for instance, equation 7.14 from [3]):

▶ **Theorem 18.** *For all $m, k \in \mathbb{N}$ such that $m \geq k$, we have that*

$$\frac{1}{m+1}2^{mH(k/m)} \leq \binom{m}{k} \leq 2^{mH(k/m)}.$$

Group theory

▶ **Definition 19** (Group action and orbits). *An action of a finite group G on a finite set S is a mapping $(\cdot)$ from $G \times S$ to S such that*

- *$e \cdot x = x$ for all $x \in S$ where e is the identity element of G,*
- *$g \cdot (h \cdot x) = gh \cdot x$ for all $g, h \in G$ and $x \in S$.*

The orbit of an element $x \in S$ is $G \cdot x = \{g \cdot x : g \in G\}$. The properties of the group show that orbits partition S into natural equivalence classes.

For $g \in G$ and $T \subseteq S$ define $g \cdot T = \{g \cdot y : y \in T\}$. For $x \in S$ let the stabilizer subgroup of x be defined as $G_x = \{g \in G : g \cdot x = x\}$.

▶ **Theorem 20** (Orbit-Stabilizer Theorem). *For every $x \in S$, $|G \cdot x| = |G|/|G_x|$.*

We need the following lemma which is an easily corollary of Orbit-Stabilizer Theorem.

▶ **Lemma 21.** *Let $x \in S$ and let $T \subseteq S$. Then $\Pr_g[x \in g \cdot T] = \frac{|G \cdot x \cap T|}{|G \cdot x|}$, where g is sampled uniformly from G.*

Proof. We have $\Pr_g[x \in g \cdot T] = \Pr_g[\exists y \in T : x = g \cdot y]$. The events $x = g \cdot y$ are disjoint and have non-zero probability only when y is in the same orbit as x. Therefore we can write the probability as $\sum_{y \in G \cdot x \cap T} \Pr_g[x = g \cdot x]$. It remains to show that $\Pr_g[x = g \cdot y] = 1/|G \cdot x|$ for every $y \in G \cdot x \cap T$.

Observe that there is a bijection between $A := \{g : g \cdot y = x\}$ and G_y. Pick any $\sigma \in A$. Then the mapping π to $\sigma^{-1}\pi$ is a bijection. Therefore $\Pr_g[x = g \cdot y] = |A|/|G| = |G_y|/|G| = 1/|G \cdot x|$, where the last equality follows from Orbit-Stabilizer Theorem and the fact that y and x are in the same orbit. ◀

7 Constructing 2-CNFs for each region

In this section, we will prove Lemma 16 for each of the regions by constructing 2-CNFs for orbits in that region. Proofs provided in this section depend on results regarding asymptotes of coefficients of multivariate generating functions which are proved in Section 8.

7.1 2-CNFs for orbits in region 1

In this subsection, we will prove Lemma 16 for orbits in $\mathcal{R}_1$. We will rely on the following result regarding asymptotes of a coefficient of a monomial from a generating function, which we will prove in Section 8.1.

▶ **Lemma 22.** *There exists a constant C_0 such that the following holds. Let A, B, C, p, q, r, n be nonnegative integers such that $n \geq C_0$, $n = p + q + r, A = \frac{5}{4}n - \frac{25}{32}p - \frac{25}{4}r, B = \frac{-5}{4}n + \frac{25}{16}p + \frac{25}{4}r, C = \frac{1}{2}n - \frac{25}{32}p$. Then, the coefficient of the monomial $x^p y^q z^r$ in $P(x, y, z) = (x^2 + 2y^2 + z^2)^A (x^2 + (y + z)^2)^B ((2y + z)^2)^C$ is at least $\frac{1}{n^{C_0}} \frac{5^n}{2^{2p+q}}$.*

Proof of Lemma 16 for region 1. Let $m_1 = 32$. Now consider any $(p, q, r) \in \mathcal{R}_1$ such that m_1 divides each of p, q, and r and hence n.

Let $A = \frac{5}{4}n - \frac{25}{32}p - \frac{25}{4}r, B = \frac{-5}{4}n + \frac{25}{16}p + \frac{25}{4}r, C = \frac{1}{2}n - \frac{25}{32}p$. A, B, C are non-negative integers since m_1 divides each of n, p, q, r and by the definition of $\mathcal{R}_1$ we have that $A, B, C \geq 0$.

Our 2-CNF F is obtained by a disjoint conjunction of A copies of **Matching**, B copies of **2Imp** and $2C$ copies of **NAND**. It is easy to see that F is consistent with $\mathsf{IP}_n^{p \bmod 2} = \mathsf{IP}_n^0$.

The spectrum of F is $P(x, y, z) = (x^2 + 2y^2 + z^2)^A (x^2 + (y + z)^2)^B ((2y + z)^2)^C$. We now apply Lemma 22 to get

$$|\mathbf{sol}(F) \cap S(p, q, r)| \geq n^{-K_1} \cdot \frac{5^n}{2^{2p+q}}$$

where K_1 is a constant. Since $|S(p, q, r)| = \binom{n}{p}\binom{n-p}{r} \cdot 2^{n-p-r}$. Using these, we get

$$\begin{aligned}\frac{|S(p, q, r)|}{|\mathbf{sol}(F) \cap S(p, q, r)|} &\leq \binom{n}{p}\binom{n-p}{r} \cdot 2^{n-p-r} \cdot n^{-K_1} \cdot \frac{2^{2p+q}}{5^n} \\ &= n^{-K_1} \cdot \binom{n}{p, q, r} \cdot 4^{-r} \cdot (4/5)^n\end{aligned}$$

where the last equality follows from the fact that $p + q + r = n$. To prove $\frac{S(p,q,r)}{|\mathbf{sol}(F) \cap S(p,q,r)|} \leq n^{-K_0} \cdot (9/5)^n$ for some constant K_0, it suffices to show that $\binom{n}{p,q,r} \cdot 4^{-r} \leq (9/4)^n$. We have

$$\begin{aligned}\binom{n}{p, q, r} \cdot 4^{-r} &= \binom{n}{p, q, r} 1^p \cdot 1^q \cdot 4^{-r} \\ &\leq \sum_{i,j,k \in \mathbb{N} : i+j+k=n} \binom{n}{i, j, k} \cdot 1^i \cdot 1^j \cdot 4^{-k} \\ &= (9/4)^n\end{aligned}$$

where the last equality follows from the binomial theorem. Hence, our construction F indeed has the desired properties. ◀

7.2 2-CNFs for orbits in region 2

In this subsection, we will prove Lemma 16 for orbits in $\mathcal{R}_2$. In the proof, we use the following general claim regarding binomial coefficients which will be proved at the end of the subsection.

▷ Claim 23. For all $m, k \in \mathbb{N}$ such that $m \geq k$ and m, k are even, we have that

$$\binom{m}{k} \leq (m/2+1)^2 \binom{m/2}{k/2}^2$$

Proof of Lemma 16 for region 2. Let $m_2 = 4$ and consider any $(p, q, r) \in \mathcal{R}_2$ such that m_2 divides each of p, q, and r and hence n.

Let F be the 2-CNF obtained by taking the disjoint conjunction of $n/2$ copies of **Matching**. Since **Matching** is consistent with IP^0_n, F is also consistent with IP^0_n as desired. The spectrum of F is given by $P(x, y, z) = (x^2 + 2y^2 + z^2)^{n/2}$ and that $|\mathbf{sol}(F) \cap S(p, q, r)|$ equals the coefficient of the monomial $x^p y^q z^r$ in $P(x, y, z)$. Examining P, we easily deduce that this coefficient equals $\binom{n/2}{p/2}\binom{n/2-p/2}{q/2}2^{q/2}$. Since $S(p, q, r) = \binom{n}{p}\binom{n-p}{r}2^q$, we get

$$\frac{S(p,q,r)}{|\mathbf{sol}(F) \cap S(p,q,r)|} = \frac{\binom{n}{p}\binom{n-p}{r}2^q}{\binom{n/2}{p/2}\binom{(n-p)/2}{q/2}2^{q/2}} = \frac{\binom{n}{p}}{\binom{n/2}{p/2}} \cdot \frac{\binom{n-p}{q}}{\binom{(n-p)/2}{q/2}} \cdot 2^{q/2}$$

where the last equality follows using the fact that $p + q + r = n$. We apply Claim 23 to the last expression and obtain:

$$\begin{aligned}
\frac{S(p,q,r)}{|\mathbf{sol}(F) \cap S(p,q,r)|} &\leq n^4 \cdot \binom{n/2}{p/2} \cdot \binom{(n-p)/2}{q/2} \cdot 2^{q/2} \\
&\leq n^4 \cdot \binom{n/2}{p/2} \cdot \sum_{i=0}^{(n-p)/2} \binom{(n-p)/2}{i} \cdot 2^i \\
&= n^4 \cdot \binom{n/2}{p/2} \cdot 3^{(n-p)/2} && \text{(by binomial theorem)}
\end{aligned}$$

Since we are in $\mathcal{R}_2$, we have $p \geq p_0 = 16n/50 \geq (n/2)/2$, which implies that the term $\binom{n/2}{p/2}$ decreases as p increases. Also the term $3^{(n-p)/2}$ decreases as p increases. Hence, $n^4 \cdot \binom{n/2}{p/2} \cdot 3^{(n-p)/2} \leq n^4 \cdot \binom{n/2}{p_0/2} \cdot 3^{(n-p_0)/2}$. Therefore, we obtain

$$\frac{S(p,q,r)}{|\mathbf{sol}(F) \cap S(p,q,r)|} \leq n^4 \cdot \binom{n/2}{p_0/2} \cdot 3^{(n-p_0)/2}$$

We apply Theorem 18 to the last expression to get the desired upper bound on $\frac{S(p,q,r)}{|\mathbf{sol}(F) \cap S(p,q,r)|}$:

$$\begin{aligned}
\frac{S(p,q,r)}{|\mathbf{sol}(F) \cap S(p,q,r)|} &\leq n^4 \cdot 2^{(n/2)H(p_0/n)} \cdot 3^{(n-p_0)/2} = n^4 \cdot \left(2^{H(16/25)/2} \cdot 3^{9/50}\right)^n \\
&< n^4 \cdot (1.7)^n < n^K \cdot (9/5)^n
\end{aligned}$$

where H is the binary entropy function and $K = 4$. ◀

We now prove our remaining claim, an inequality involving binomial coefficients.

Proof of Claim 23. This will follow by couple of applications of Theorem 18. We compute:

$$\begin{aligned}\binom{m}{k} &\leq 2^{mH(k/m)} && \text{(applying Theorem 18)}\\ &= (m/2+1)^2\left(\frac{1}{(m/2+1)^2}\cdot 2^{(m/2)H((k/2)/(m/2))}\right)^2 \\ &\leq (m/2+1)^2\binom{m/2}{k/2}^2 && \text{(applying Theorem 18)}\end{aligned}$$

as desired. ◁

7.3 Constructing 2-CNFs for orbits in region 3

We here prove Lemma 16 for region 3. We need the following lemma regarding the coefficients of certain generating functions which will be proved in Section 8.2.

▶ **Lemma 24.** *There exists a constant K such that the following holds. Let $B_1, C_1, B_2, C_2, p, q, r, n$ be nonnegative integers such that $n \geq K$, $p \leq 2\cdot\min(B_1, B_2)$, $n = p+q+r, r \leq n-8, B_1 = 0.34n, C_1 = 0.16n, B_2 = 0.465n, C_2 = 0.035n$. Then, for $i \in [2]$, the coefficient of the monomial $x^p y^q z^r$ in $P_i(x,y,z) = (x^2+(y+z)^2)^{B_i}((2y+z)^2)^{C_i}$ is at least $\frac{1}{n^K}\frac{(2v_i+1)^{2C_i}(u_i+(v_i+1)^2)^{B_i}}{u_i^{p/2}v_i^q}$ where $v_i = \frac{-\beta_i+\sqrt{\beta_i^2+8rq}}{4r}, u_i = \frac{p(v_i+1)^2}{2B_i-p}$, and $\beta_i = 4C_i+(2B_i-p)-3q$.*

We also need the following lemma regarding the solution of a bounded optimization problem which is solved using an optimization solver:

▶ **Lemma 25.** *Let $\hat{p}, \hat{q}, \hat{r}$ be arbitrary non-negative reals satisfying the following constraints:*

- $\hat{p}+\hat{q}+\hat{r} = 1$.
- $\frac{1}{2} - \frac{25}{32}\hat{p} \geq 0$.
- $\frac{5}{4} - \frac{25}{32}\hat{p} - \frac{25}{4}\hat{r} \leq 0$.
- $\frac{1}{4} - \hat{p} \leq 0$.

Let $\hat{B}_1 = 0.34, \hat{C}_1 = 0.16, \hat{B}_2 = 0.465, \hat{C}_2 = 0.035$. For $i \in [2]$, let $\hat{\beta}_i = 4\hat{C}_i + (2\hat{B}_i - \hat{p}) - 3\hat{q}, v_i = \frac{-\hat{\beta}_i+\sqrt{\hat{\beta}_i^2+8\hat{r}\hat{q}}}{4\hat{r}}$. Finally, let

$$\begin{aligned}T_i = &-\hat{p}\log(\hat{p}) - \hat{q}\log(\hat{q}) - \hat{r}\log(\hat{r}) + \hat{q}+\\ &\hat{q}\log(v_i) - 2\hat{C}_i\log(2v_i+1) - (2\hat{B}_i-\hat{p})\log(v_i+1)+\\ &0.5(2\hat{B}_i-\hat{p})\log(2\hat{B}_i-\hat{p}) + 0.5\hat{p}\log(\hat{p}) - \hat{B}_i\log(2\hat{B}_i)\end{aligned}$$

Then, $\min(T_1, T_2) \leq 0.841$.

Proof. This as an optimization problem where we maximize $\min(T_1, T_2)$ over all $\hat{p}, \hat{q}, \hat{r}$ subject to the constraints. In particular, we substitute $\hat{q} = 1-\hat{p}-\hat{r}$ and then maximize over all values of $\hat{p}, \hat{r}$ subject to the constraints. To solve this optimization problem, we use IbexOpt, a well known optimization tool that uses numeric interval arithmetic based library IBEX in C++ to solve global optimization problems [14]. We provide our code for this at `https://github.com/mjguru/Inner-Product/blob/main/region3.bch`. We note that the objective function we use in the code uses upper and lower bounds when evaluating logarithms to avoid numerical stability issues. These bounds only increase our objective function and so our upper bound still holds. ◀

We are now ready to prove Lemma 16 for region 3.

Proof of Lemma 16 for region 3. Let $m_3 = 2000$. Consider any $(p, q, r) \in \mathcal{R}_3$ such that m divides each of p, q, r and hence also $n = p + q + r$. Since p is even, we will construct 2-CNF F that is consistent with IP^0_n such that $\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|} \leq n^{K_0} \cdot (9/5)^n$ for some constant K_0.

When $r = n$, we have $|S(p, q, r) = 1|$ and in that case, we trivially construct a consistent 2-CNF covering the solution. Otherwise, since m_3 divides r and $r \leq n$, it must be that $r \leq n - m \leq n - 8$ and we use this fact when we apply Lemma 24.

Let $B_1 = 0.34n, C_1 = 0.16n, B_2 = 0.465n, C_2 = 0.035n$. We easily see that each of them is a positive even integer. For $i \in [2]$, let F_i be the 2-CNF obtained by a disjoint conjunction of B_i copies of **2Imp** and $2C_i$ copies of **NAND**. Since B_i and C_i are even, we infer that F_i is consistent with IP^0_n. We will let F equal one of F_1, F_2, whichever one attains larger value of $|\mathbf{sol}(F_i) \cap S(p, q, r)|$.

The spectrum of F_i is $P_i = (x^2 + (y + z)^2)^{B_i}((2y + z)^2)^{C_i}$. The coefficient of $x^p y^q z^r$ in P_i equals $|\mathbf{sol}(F_i) \cap S(p, q, r)|$.

Recall that $|S(p, q, r)| = \binom{n}{p}\binom{n-p}{r} \cdot 2^q$. We apply Theorem 18 to get

$$|S(p, q, r)| \leq 2^{-p\log(p/n)-q\log(q/n)-r\log(r/n)+q}.$$

Using Lemma 24 for each P_i, we infer that

$$|\mathbf{sol}(F_i) \cap S(p, q, r)| \geq n^{-K_1} \cdot \frac{(2v_i + 1)^{2C_i}(u_i + (v_i + 1)^2)^{B_i}}{(u_i)^{p/2} v_i^q}$$

where $v_i = \frac{-\beta_i+\sqrt{\beta_i^2+8rq}}{4r}, u_i = \frac{p(v_i+1)^2}{2B_i-p}$, and $\beta_i = 4C_i + (2B_i - p) - 3q$. Combing the last two inequalities, we get

$$\frac{|S(p, q, r)|}{|\mathbf{sol}(F_i) \cap S(p, q, r)|} \leq \frac{2^{-p\log(p/n)-q\log(q/n)-r\log(r/n)+q}(u_i)^{p/2}(v_i)^q}{n^{-K_1}(2v_i + 1)^{2C_i}(u_i + (v_i + 1)^2)^{B_i}}.$$

Let $\sigma_i = \frac{|S(p,q,r)|}{|\mathbf{sol}(F_i)\cap S(p,q,r)|}$ and let $\hat{p} = \frac{p}{n}, \hat{q} = \frac{q}{n}, \hat{r} = \frac{r}{n}, \hat{B}_i = \frac{B_i}{n}, \hat{C}_i = \frac{C_i}{n}$. Then we rewrite:

$$\begin{aligned}\frac{\log(\sigma_i)}{n} - K_1\log(n) \leq &-\hat{p}\log(\hat{p}) - \hat{q}\log(\hat{q}) - \hat{r}\log(\hat{r}) + \hat{q} + \\ &0.5\hat{p}\log(u_i) + \hat{q}\log(v_i) - 2\hat{C}_i\log(2v_i + 1) - 2\hat{B}_i\log(u_i + (v_i + 1)^2).\end{aligned}$$

We substitute $u_i = \frac{p(v_i+1)^2}{2B_i-p}$ and simplify to obtain:

$$\begin{aligned}\frac{\log(\sigma_i)}{n} - K_1\log(n) \leq &\hat{p}\log(\hat{p}) + \hat{q}\log(\hat{q}) + \hat{r}\log(\hat{r}) + \hat{q} + \\ &\hat{q}\log(v_i) - 2\hat{C}_i\log(2v_i + 1) - (2\hat{B}_i - \hat{p})\log(v_i + 1) + \\ &0.5(2\hat{B}_i - \hat{p})\log(2\hat{B}_i - \hat{p}) + 0.5\hat{p}\log(\hat{p}) - \hat{B}_i\log(2\hat{B}_i)\end{aligned}$$

We now apply Lemma 25, verifying that we satisfy all the constraints, to infer that

$$\min\left(\frac{\log(\sigma_1)}{n} - K_1\log(n), \frac{\log(\sigma_2)}{n} - K_1\log(n)\right) \leq 0.8398 \tag{2}$$

By definition of F, we have that $\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|} = \min(\sigma_1, \sigma_2)$. Hence, we compute that

$$\begin{aligned}\frac{|S(p, q, r)|}{|\mathbf{sol}(F) \cap S(p, q, r)|} &= \min(\sigma_1, \sigma_2) \\ &= 2^{n\cdot\min\left(\frac{\log(\sigma_1)}{n}, \frac{\log(\sigma_1)}{n}\right)} \\ &\leq n^{K_1} \cdot 2^{n\cdot 0.8398} && \text{(using Equation (2))} \\ &\leq n^{K_1} \cdot (9/5)^n\end{aligned}$$

as desired. ◀

7.4 Constructing 2-CNFs for orbits in region 4

In this subsection, We prove Lemma 16 for region 4. We need the following lemma regarding the coefficients of certain generating functions which itself will be proved in Section 8.3

▶ **Lemma 26.** *There exists a constant K such that the following holds. Let $A_1, C_1, A_2, C_2, p, q, r, n$ be nonnegative integers such that $n \geq K_0$, $p \leq 2 \cdot \min(A_1, A_2)$, $n = p + q + r, 0 < r \leq n - 8, A_1 = 0.34n, C_1 = 0.16n, A_2 = 0.355n, C_2 = 0.145n$. Also, for $i \in [2]$, define the degree 3 real polynomial $Q_i(x) = (4r)x^3 + (4A_i - 2p - 2q)x^2 + (4C_i - 2q)x - q$. Then, for $i \in [2]$, the following holds: (1) $Q_i(x)$ has a unique non-negative real root. (2) The coefficient of the monomial $x^p y^q z^r$ in $P_i(x, y, z) = (x^2 + 2y^2 + z^2)^{A_i}((2y + z)^2)^{C_i}$ is at least $\frac{1}{n^K} \frac{(2v_i+1)^{2C_i}(u_i+2v_i^2+1)^{A_i}}{u_i^{p/2} v_i^q}$ where v_i is the unique non-negative root of the cubic polynomial $Q_i(x)$, and $u_i = \frac{p(2v_i^2+1)}{2A_i - p}$.*

We also need the solution of a bounded optimization problem which is obtained using an optimization solver.

▶ **Lemma 27.** *Let $\hat{p}, \hat{q}, \hat{r}$ be arbitrary non-negative reals satisfying the following constraints:*

- $\hat{p} + \hat{q} + \hat{r} = 1$.
- $\frac{1}{2} - \frac{25}{32}\hat{p} \geq 0$.
- $-\frac{5}{4} + \frac{25}{16}\hat{p} + \frac{25}{4}\hat{r} \leq 0$.
- $-\frac{1}{20} + \hat{r} \geq 0$.
- $\frac{1}{4} - \hat{p} \leq 0$.

Let $\hat{A}_1 = 0.34, \hat{C}_1 = 0.16, \hat{A}_2 = 0.355, \hat{C}_2 = 0.145$. For $i \in [2]$, let v_i be the unique non-negative root of the cubic polynomial $P_i(x) = (4\hat{r})x^3 + (4\hat{A}_i - 2\hat{p} - 2\hat{q})x^2 + (4\hat{C}_i - 2\hat{q})x - \hat{q}$, and let $u_i = \frac{p(2v_i^2+1)}{2\hat{A}_i - \hat{p}}$ Finally, let

$$\begin{aligned} T_i = &- \hat{p}\log(\hat{p}) - \hat{q}\log(\hat{q}) - \hat{r}\log(\hat{r}) + \hat{q} + \\ &- 2\hat{C}_i \log(2\hat{v}_i + 1) - \hat{A}_i \log(\hat{u}_i + 2\hat{v}_i^2 + 1) + \\ &(\hat{p}/2)\log(\hat{u}_i) + \hat{q}\log(\hat{v}_i) \end{aligned}$$

Then, $\min(T_1, T_2) \leq 0.845$.

Proof. We will first assert the existence and uniqueness of the nonnegative real root of the polynomial $P_i(x)$ which follows from Lemma 26.

We now describe how we obtain the upper bound on $\min(T_1, T_2)$. This as an optimization problem where we maximize $\min(T_1, T_2)$ over all $\hat{p}, \hat{q}, \hat{r}$ subject to the constraints. In particular, we substitute $\hat{q} = 1 - \hat{p} - \hat{r}$ and then maximize over all values of $\hat{p}, \hat{r}$ subject to the constraints. To solve this optimization problem, we use IbexOpt, a well known optimization tool that uses numeric interval arithmetic based library IBEX in C++ to solve global optimization problems [14]. We provide our code for this at `https://github.com/mjguru/Inner-Product/blob/main/region4.bch`. We note that the objective function we use in the code uses upper and lower bounds when evaluating logarithms to avoid numerical stability issues. These bounds only increase our objective function and so our upper bound still holds. ◀

We are now ready to prove Lemma 16 for region 4.

Proof of Lemma 16 for region 4. We let $m_4 = 2000$. Consider any $(p, q, r) \in \mathcal{R}_4$ such that m divides each of p, q, r and hence also $n = p + q + r$. Since p is even, we will construct a 2-CNF F that is consistent with IP^0_n such that $\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|} \leq n^{K_0} \cdot (9/5)^n$ where K_0 is a constant.

Let $A_1 = 0.34n, C_1 = 0.16n, A_2 = 0.355n, C_2 = 0.145n$. Since 2000 divides n, each of these is a positive even integer. For $i \in [2]$, let F_i be the 2-CNF obtained by a disjoint conjunction of A_i copies of **Matching** and $2C_i$ copies of **NAND**. Using Corollary 13 and the fact that each of A_i and C_i are even integers, we infer that F_i is indeed consistent with IP^0_n. Let F equal one of F_1, F_2, whichever one attains larger value of $|\mathbf{sol}(F_i) \cap S(p, q, r)|$.

The spectrum of F_i is $P_i = (x^2 + 2y^2 + z^2)^{A_i}((2y + z)^2)^{C_i}$. The coefficient of $x^p y^q z^r$ in P_i equals $|\mathbf{sol}(F_i) \cap S(p, q, r)|$.

We know that $|S(p, q, r)| = \binom{n}{p}\binom{n-p}{r} \cdot 2^q$. We apply Theorem 18 and simplify using the fact $p + q + r = n$ to infer that $|S(p, q, r)| \leq 2^{-p\log(p/n)-q\log(q/n)-r\log(r/n)+q}$.

Using Lemma 26 for each P_i, we infer that

$$|\mathbf{sol}(F_i) \cap S(p, q, r)| \geq n^{-K_1} \cdot \frac{(2v_i + 1)^{2C_i}(u_i + 2v_i^2 + 1)^{A_i}}{(u_i)^{p/2} v_i^q}$$

where v_i is the unique non-negative root of the polynomial $(4r)x^3 + (4A_i - 2p - 2q)x^2 + (4C_i - 2q)x - q$, $u_i = \frac{p(2v_i^2+1)}{2A_i - p}$ and K_1 a constant. We now have

$$\frac{|S(p, q, r)|}{|\mathbf{sol}(F_i) \cap S(p, q, r)|} \leq \frac{2^{-p\log(p/n)-q\log(q/n)-r\log(r/n)+q}(u_i)^{p/2}(v_i)^q}{n^{-K_1}(2v_i + 1)^{2C_i}(u_i + 2v_i^2 + 1)^{A_i}}.$$

Let $\sigma_i = \frac{|S(p,q,r)|}{|\mathbf{sol}(F_i)\cap S(p,q,r)|}$ and let $\hat{p} = \frac{p}{n}, \hat{q} = \frac{q}{n}, \hat{r} = \frac{r}{n}, \hat{B}_i = \frac{B_i}{n}, \hat{C}_i = \frac{C_i}{n}$. We write the inequality as

$$\begin{aligned}\frac{\log(\sigma_i)}{n} - K_1 \log(n) \leq &- \hat{p}\log(\hat{p}) - \hat{q}\log(\hat{q}) - \hat{r}\log(\hat{r}) + \hat{q} + \\ &2\hat{C}_i \log(2v_i + 1) - \hat{A}_i \log(u_i + 2v_i^2 + 1) + \\ &(\hat{p}/2)\log(u_i) + \hat{q}\log(v_i)\end{aligned}$$

After verifying that we satisfy all the constraints of the lemma (since $(p, q, r) \in \mathcal{R}_5$, we must have that $r \leq 0.9n \leq n - 8$), we now apply Lemma 27, to infer that

$$\min\left(\frac{\log(\sigma_1)}{n} - K_1\log(n), \frac{\log(\sigma_2)}{n} - K_1\log(n)\right) \leq 0.844 \tag{2}$$

By definition of F, we have that $\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|} = \min(\sigma_1, \sigma_2)$. Hence, we conclude that

$$\begin{aligned}\frac{|S(p, q, r)|}{|\mathbf{sol}(F) \cap S(p, q, r)|} &= \min(\sigma_1, \sigma_2) \\ &= 2^{n\cdot\min\left(\frac{\log(\sigma_1)}{n}, \frac{\log(\sigma_1)}{n}\right)} \\ &\leq n^{K_1} \cdot 2^{n\cdot 0.844} && \text{(using Equation (2))} \\ &\leq n^{K_1} \cdot (9/5)^n\end{aligned}$$

as desired. ◀

7.5 Constructing 2-CNFs for orbits in region 5

In this subsection, we prove Lemma 16 for orbits in $\mathcal{R}_5$. We need the following result which we prove towards the end of this subsection:

▷ Claim 28. Let $g : (0, 0.64] \to \mathbb{R}$ be defined as $g(y) = H(y) + \varepsilon_R H(\varepsilon_R) + a - \frac{y}{2} - aH(y/2a)$ where $\varepsilon_R = \frac{1}{20}, a = 0.355$ and H is the binary entropy function. Then, $\max_{y\in[0,0.64]} g(y) \leq 0.828$.

We first show how Lemma 16 follows.

Proof of Lemma 16 for region 5. Let $m_5 = 2000$ and let $(p, q, r) \in \mathcal{R}_5$ be an arbitrary orbit such that m divides each of p, q, and r and hence n. Since p is even, it suffices to construct a 2-CNF F that is consistent with IP^0_n and $\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|} \leq n^{K_0}(9/5)^n$ where K_0 is a constant.

Let $A = 0.355n, C = 0.145n$. Let F be the 2-CNF obtained by disjoint conjunctions of A copies of **Matching** and $2C$ copies of **NAND** (as defined in Definition 10). Since both these building blocks are consistent with IP^0_n, we have that F is consistent with IP^0_n as desired.

We will now show that $\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|} \leq n^{K_0}(9/5)^n$. The spectrum of F is given by $P(x, y, z) = (x^2 + 2y^2 + z^2)^A(2y + z)^{2C}$ and $|\mathbf{sol}(F) \cap S(p, q, r)|$ equals the coefficient of $x^p y^q z^r$ in $P(x, y, z)$. We lower bound this coefficient as follows: Choose r out of $2C$ terms of $(2y + z)$ to equal z and remaining to equal $2y$; then choose $p/2$ terms out of A_i terms to be x^2 and remaining terms to equal $2y^2$ to obtain the monomial $2^{A-p/2+2C-r} \cdot x^p y^q z^r$. Since there are $\binom{2C}{r} \cdot \binom{A}{p/2} \geq \binom{A}{p/2}$ of obtaining such a monomial, we lower bound its coefficient by $\binom{A}{p/2} \cdot 2^{A+2C-p/2-r}$. Since $p \leq 0.64n$ and $r \leq 10^{-5}n$, we have $\binom{n-p}{r} \leq \binom{n}{r}$. Since $|S(p, q, r)| = \binom{n}{p}\binom{n-p}{r} \cdot 2^q$, we get

$$\begin{aligned}\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|} &\leq \frac{\binom{n}{p}\binom{n}{r}2^q}{\binom{A}{p/2}2^{A+2C-p/2-r}}\\ &= \frac{\binom{n}{p}\binom{n}{r}2^{A-p/2}}{\binom{A}{p/2}}\end{aligned}$$

where in the last line we used the fact $n = 2A + 2C = p + q + r$.

We now repeatedly apply Theorem 18 to the last expression to get the following bound:

$$\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|} \leq (A+1)\cdot 2^{n\cdot H(p/n)+r\cdot H(r/n)+A-p/2-A\cdot H(p/(2A))}$$

where H is the binary entropy function. Using the fact that $A + 1 \leq n$ and rearranging and taking logarithm of both sides, we have

$$\frac{\log\left(\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|}\right)}{n} - \log(n) \leq H(\hat{p}) + \hat{r}\cdot H(\hat{r}) + \hat{A} - \hat{p}/2 - \hat{A}\cdot H(\hat{p}/(2\hat{A}))$$

where $\hat{A} = \frac{A}{n}, \hat{p} = \frac{p}{n}, \hat{r} = \frac{r}{n}$. Since $0 \leq \hat{r} \leq \frac{1}{20}$ and $\hat{r} \cdot H(\hat{r})$ is an increasing function in that range, we infer

$$\frac{\log\left(\frac{|S(p,q,r)|}{|\mathbf{sol}(F)\cap S(p,q,r)|}\right)}{n} - \log(n) \leq H(\hat{p}) + \frac{1}{20}\cdot H\left(\frac{1}{20}\right) + \hat{A} - \hat{p}/2 - \hat{A}\cdot H(\hat{p}/(2\hat{A}))$$

Let $g(y) = H(y) + \frac{1}{20}H\left(\frac{1}{20}\right) + \hat{A} - y/2 - \hat{A} \cdot H(y/2\hat{A})$ where $\hat{A} = 0.355$. Then, using Claim 28, we get that $g(\hat{p}) \leq 0.828$ and

$$\frac{\log\left(\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|}\right)}{n} - \log(n) \leq 0.828$$

Rearranging and considering exponents, we conclude

$$\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} \leq n \cdot 2^{0.828n} \leq n^1 \cdot (9/5)^n$$

as desired. ◀

We now prove the claim regarding the real function g:

Proof of Claim 28. We will show that for all $y \in (0, 6.4]$, $g'(y) > 0$ which lets us conclude that $\max_{y \in [0,0.64]} g(y) = g(0.64) \leq 0.828$ as desired.

Recall that the derivative of the binary entropy function H is $H'(x) = \log((1-x)/x)$. Using this, we compute that

$$\begin{aligned} g'(y) &= \log((1-y)/y) - \frac{1}{2} - \frac{1}{2}\log((2a-y)/y) \\ &= \log\left(\frac{1-y}{\sqrt{2y(2a-y)}}\right) \end{aligned}$$

Therefore, to show that $g'(y) > 0$ for all $y \in (0, 0.64]$, it suffices to show that for y in this domain, $1 - y > \sqrt{2y(2a-y)}$. Since both sides of this inequality are positive, this is equivalent to showing that $(1-y)^2 > 2y(2a-y)$. This is equivalent to showing $3y^2 - y(4a+2) + 1 > 0$. Let $h(y) = 3y^2 - y(4a+2) + 1$. We will show that $h(y) > 0$ for all y. The discriminant of this quadratic $h(y)$ is $(4a+2)^2 - 12 = (3.42)^2 - 12 < 0$. This means $h(y)$ has no real roots. Since $h(0) = 1 > 0$, we conclude that $h(y) > 0$ for all y as desired. ◁

7.6 Constructing 2-CNFs for orbits in region 6

Proof of Lemma 16 for region 6. Let $m_6 = 2000$ and let $(p,q,r) \in \mathcal{R}_6$ be an arbitrary orbit such that m divides each of p, q and r and hence n. Since p is even, it suffices to show that we can construct a 2-CNF F that is consistent with IP^0_n such that $\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} \leq n^{K_0}(9/5)^n$ for some constant K_0.

Let $A = p/2, C = (n-p)/2$. Let F be the 2-CNF obtained by a disjoint conjunction of A copies of **Matching** and $2C$ copies of **NAND**. Since both these building blocks are consistent with IP^0_n, we have that F is consistent with IP^0_n as desired.

We will now show that $\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} \leq n^{K_0}(9/5)^n$. The spectrum of F is given by $P(x,y,z) = (x^2 + 2y^2 + z^2)^A(2y+z)^{2C}$ and $|\mathbf{sol}(F) \cap S(p,q,r)|$ equals the coefficient of $x^p y^q z^r$ in $P(x,y,z)$. Since $A = p/2$, this coefficient exactly equals $\binom{2C}{q} \cdot 2^q = \binom{n-p}{q} \cdot 2^q$ Also we have that $|S(p,q,r)| = \binom{n}{p}\binom{n-p}{q} \cdot 2^q$. Therefore,

$$\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} = \binom{n}{p}.$$

We apply Theorem 18 to get that

$$\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} \leq 2^{n \cdot H(p/n)}$$

Since $0 \leq p \leq \frac{n}{4}$ and $H(\cdot)$, the binary entropy function, is increasing in this range, we obtain

$$\frac{|S(p,q,r)|}{|\mathbf{sol}(F) \cap S(p,q,r)|} \leq 2^{n \cdot H(1/4)} \leq 2^{n \cdot (0.82)} \leq (9/5)^n$$

as desired. ◀

8 Coefficient extraction for each region

In this section, we obtain asymptotic bounds for coefficients of monomials of particular generating functions. To prove these results, we require the following lemma which bounds the coefficients of a power series:

▶ **Lemma 29.** *Let $p, q \in \mathbb{N}$ be such that $p^2 + q^2 \geq 4$. Let $f : \mathbb{C}^2 \to \mathbb{C}$ be analytic on a compact neighborhood $\mathcal{N}$ around $(0,0)$. Let $f(u,v) = \sum_{r=(r_1,r_2)\in\mathbb{N}^2} C_r u^{r_1} v^{r_2}$ where for C_r are non-negative. Let $h : \mathbb{C}^2 \to \mathbb{C}$ be defined as $h(u,v) = \ln(f(u,v)) - p\ln(u) - q\ln(v)$. Assume there exist non-negative real u_0 and v_0 such that $\nabla_h(u_0,v_0) = 0$ and that $\mathcal{H}_h(u_0,v_0)$ is positive definite. Assume that $|f(u,v)|$ attains its unique global maximum on the torus $|u| = u_0, |v| = v_0$ at (u_0,v_0). Then,*

$$C_{p,q} = \frac{f(u_0,v_0)}{2\pi u_0^{p+1} v_0^{q+1} \sqrt{\det(\mathcal{H}_h(u_0,v_0))}} \left(1 + O\left(\frac{1}{\sqrt{p^2+q^2}}\right)\right).$$

where $\mathcal{H}_h(u_0,v_0)$ is the Hessian of h evaluated at u_0, v_0.

This result seems to be a standard result in analytic combinatorics but since we could not find a ready reference to the formulation in Lemma 29, we will provide a somewhat self-contained proof in Appendix A. For now, we use Lemma 29 to obtain bounds on the coefficients of the generating functions that arise from the constructions of IP.

8.1 Bound for Region 1

Proof of Lemma 22. Let $f(u,v) = P(u^{1/2}, v, 1)$. Since the degree of x is even in $P(x,y,z)$, $2A + 2B + 2C = p + q + r = n$, and each of the three factor polynomials is homogeneous, we infer that the coefficient of $u^{p/2}v^q$ in $f(u,v)$ equals the coefficient of $x^p y^q z^r$ in $P(x,y,z)$.

Let $h : \mathbb{C}^2 \to \mathbb{C}$ as $h(u,v) = \ln(f(u,v)) - (p/2)\ln(u) - q\ln(v)$. We observe that $\nabla_h(16,2) = 0$. We argue that the Hessian $\mathcal{H}_h(16,2)$ of h at $u = 16$ and $v = 2$ is positive definite. To show this, let $\alpha(x,y) = h(\exp(x), \exp(y))$. We know the function α is convex since that h has nonnegative coefficients in its power series (see section 4.5 from [2]). We also observe that α is strictly convex and hence $\mathcal{H}_\alpha(16,2)$ is positive definite. Since $\nabla_h(16,2) = 0$, we apply the chain rule to get that $\mathcal{H}_\alpha(16,2) = D \cdot \mathcal{H}_h(16,2) \cdot D$ where $D = \begin{bmatrix} 16 & 0 \\ 0 & 2 \end{bmatrix}$ is a diagonal matrix. Since $\mathcal{H}_\alpha(16,2)$ is positive definite and D is positive definite, we get that $\mathcal{H}_h(16,2)$ is positive definite as well. We next apply triangle inequality to infer that in the (complex) torus defined by $|u| = 16, |v| = 2$, the unique global maximum occurs at $u = 16$ and $v = 2$. Lastly, we claim that at least one of p or q must have value at least $0.01n$. Indeed, if they both are smaller, then it must be that $r \geq 0.98n$, implying that $A < 0$. However, that is a contradiction since we assume A is a nonnegative integer. Let C_0 be large enough so we have $(p/2)^2 + q^2 \geq 4$ for $n \geq C_0$.

We satisfy the conditions to apply Lemma 29 with $u_0 = 16, v_0 = 2$ to extract the coefficient $u^{p/2}v^q$. We make sure that C_0 is large enough to conclude that the coefficient $u^{p/2}v^q$ equals

$$\frac{25^{A+B+C}}{C_0 4^p 2^q \mathsf{poly}(p,q)}\left(1+\frac{1}{\sqrt{p^2+q^2}}\right) \geq \frac{5^n}{C_0 4^p 2^q \mathsf{poly}(n)}\left(1+\frac{1}{\mathsf{poly}(n)}\right)$$

where in the last equality we used the fact that either p or q will be at least $0.01n$. ◀

8.2 Bound for region 3

In this subsection, we obtain asymptotic bounds on the coefficients of generating functions related to $\mathcal{R}_3$.

Proof of Lemma 24. For $i \in [2]$, let $f_i(u,v) = P_i(u^{1/2}, v, 1)$. Since x appears in squared form, f_i is still product of three polynomials. Furthermore, since $2B_i + 2C_i = p+q+r = n$, and both the polynomials are homogeneous, we infer that the coefficient of $u^{p/2}v^q$ in $f_i(x,y,z)$ equals the coefficient of $x^p y^q z^r$ in $P_i(x,y,z)$.

Define $h_i : \mathbb{C}^2 \to \mathbb{C}$ by $h(u,v) = \ln(f_i(u,v)) - (p/2)\ln(u) - q\ln(v)$. We will find non-negative $(u_i, v_i) \in \mathbb{R}^2$ such that $\nabla_{h_i}(u_i, v_i) = 0$. We compute that $\frac{\partial h_i}{\partial u} = -\frac{p}{2}\cdot\frac{1}{u} + \frac{B_i}{u+(v+1)^2}$. Setting this to 0 and rearranging, we infer that

$$u_i = \frac{p(v_i+1)^2}{2B_i - p} \tag{1}$$

We next compute that $\frac{\partial h_i}{\partial v} = \frac{4C_i}{2v+1} + \frac{2B_i(v+1)}{u+(v+1)^2} - \frac{q}{v}$. Setting this to 0, rearranging, substituting in Equation (1), and using the fact that $4C_i + 2(2B_i - p) - 2q = 2r$ (since $2B_i + 2C_i = p+q+r = n$), we infer that v_i satisfies the following quadratic equation:

$$2r\cdot(v_i)^2 + \beta_i \cdot v_i - q = 0$$

where $\beta_i = 4C_i + (2B_i - p) - 3q$. This means v_i equals $\frac{\beta_i \pm \sqrt{\beta_i^2 + 8rq}}{4r}$. By the inequalities satisfied by p, q, r, it must be the case that rq is at least a constant. This makes the value inside the square root comes out to be strictly larger than β_i^2 and so one root is strictly negative and other is strictly positive. We reject the negative root since Lemma 29 requires the special point (u_i, v_i) to have non-negative real values.

With this value of u_i, v_i, we show that $\mathcal{H}_{h_i}(u_i, v_i)$ is positive definite where $\mathcal{H}_{h_i}$ is the the Hessian of h_i. For this, let $\alpha_i(x,y) = h_i(\exp(x), \exp(y))$. Using the fact that h has nonnegative coefficients in its power series, we use the well known result (see section 4.5 from [2]) that the function α_i as we defined is convex. Also by picking any three non-zero terms of the power series and checking that they don't lie in single line, we see that α_i is strictly convex and so $\mathcal{H}_{\alpha_i}(u_i, v_i)$ is positive definite. Using the fact that $\nabla_{h_i}(u_i, v_i) = 0$ and chain rule, we obtain that $\mathcal{H}_{\alpha_i}(u_i, v_i) = D \cdot \mathcal{H}_{h_i}(u_i, v_i) \cdot D$ where D is the diagonal matrix with diagonal entries u_i and v_i. Since D is positive definite, we infer that $\mathcal{H}_{h_i}(u_i, v_i)$ is indeed positive definite.

We next apply the triangle inequality to infer that in the complex torus with $|u| = u_i, |v| = v_i$, the unique global maximum for f_i occurs at (u_i, v_i) (carefully checking that the values at $(\pm u_i, \pm v_i)$ are smaller). Also since $r \leq n-8$, $p+q+r = n$ and $n \geq K_0$ where we will set K_0 to be large enough constant, it must be the case that $(p/2)^2 + q^2 \geq 4$. Therefore we satisfy all conditions to apply Lemma 29 with $(u_0, v_0) = (u_i, v_i)$, we infer that the coefficient of $u^{p/2}v^q$ in $f_i(u,v)$ equals

$$\frac{f(u_i, v_i)}{2\pi u_i^{p+1} v_i^{q+1}\sqrt{\mathcal{H}_{h_i}(u_i, v_i)}}\left(1 + O\left(\frac{1}{\sqrt{p^2+q^2}}\right)\right).$$

Since $p^2+q^2 \geq 4$, and $u_i, v_i \leq n$, and that the entries of $\mathcal{H}_{h_i}$ are bounded by $\mathsf{poly}(u_i, v_i)$, we infer that there exists a constant K_1 such that the coefficient is at least

$$n^{-K_1} \cdot \frac{f(u_i, v_i)}{u_i^{p/2} v_i^q} = n^{-K_1} \cdot \frac{(u_i + (v_i+1)^2)^{B_i} (2v_i+1)^{2C_i}}{u_i^{p/2} v_i^q}$$ ◀

8.3 Bound for region 4

In this subsection, we obtain asymptotic bounds on the coefficients of generating functions related to $\mathcal{R}_4$.

Proof of Lemma 26. We first prove that Q_i has a unique real nonnegative root. First, since Q_i has degree 3, it has at least one real root. Let x_0 be any such real root. Since $Q_i(x_0) = 0$, we rearrange to infer that x_0 must satisfy:

$$\frac{4C_i x_0}{2x_0+1} + \frac{2x_0^2(2A_i - p)}{2x_0^2+1} = q$$

We see that the left side at $x = 0$ equals 0 and as $x \to +\infty$, it approaches $2C_i + 2A_i - p = q + r$ (since $n = 2A_i + 2C_i = p + q + r$). Since q and r are nonnegative and right side equals q, by intermediate value theorem, there must exist a unique value $y \geq 0$ such that left side equals q, proving our result.

We now prove the asymptotic bound. For $i \in [2]$, let $f_i(u, v) = P_i(u^{1/2}, v, 1)$. Since x appears in squared form, f_i is still product of three polynomials. Furthermore, since $2A_i + 2C_i = p + q + r = n$, and both the polynomials are homogeneous, we infer that the coefficient of $u^{p/2} v^q$ in $f_i(x, y, z)$ equals the coefficient of $x^p y^q z^r$ in $P_i(x, y, z)$.

Define $h_i : \mathbb{C}^2 \to \mathbb{C}$ by $h(u, v) = \ln(f_i(u, v)) - (p/2)\ln(u) - q\ln(v)$. We will find nonnegative $(u_i, v_i) \in \mathbb{R}^2$ such that $\nabla_{h_i}(u_i, v_i) = 0$. We compute that $\frac{\partial h_i}{\partial u} = \frac{A_i}{u+2v^2+1} - \frac{p}{2u}$. Setting this to 0 and rearranging, we infer that

$$u_i = \frac{p(2v_i^2+1)}{2A_i - p} \tag{1}$$

We next compute that $\frac{\partial h_i}{\partial v} = \frac{4C_i}{2v+1} + \frac{4vA_i}{u+2v^2+1} - \frac{q}{v}$. Setting this to 0, rearranging, substituting in Equation (1), and using the fact that $8C_i + 4(2A_i - p) - 4q = 4r$ (since $2A_i + 2C_i = p+q+r = n$), we infer that v_i satisfies the following cubic equation:

$$(4r) \cdot v_i^3 + (4A_i - 2p - 2q) \cdot v_i^2 + (4C_i - 2q) \cdot v_i - q = 0$$

In other words, $Q_i(v_i) = 0$. From above, we know that there is a unique nonnegative root of Q_i and we let v_i be the unique nonnegative root.

With this value of u_i, v_i, we show that $\mathcal{H}_{h_i}(u_i, v_i)$ is positive definite where $\mathcal{H}_{h_i}$ is the the Hessian of h_i. For this, let $\alpha_i(x, y) = h_i(\exp(x), \exp(y))$. Using the fact that h has nonnegative coefficients in its power series, we use the well known result (see section 4.5 from [2]) that the function α_i as we defined is convex. Also by picking any three non-zero terms of the power series and checking that they don't lie in single line, we see that α_i will be strictly convex and so $\mathcal{H}_{\alpha_i}(u_i, v_i)$ is positive definite. Using the fact that $\nabla_{h_i}(u_i, v_i) = 0$ and chain rule, we obtain that $\mathcal{H}_{\alpha_i}(u_i, v_i) = D \cdot \mathcal{H}_{h_i}(u_i, v_i) \cdot D$ where D is the diagonal matrix with with diagonal entries u_i and v_i. Since D is positive definite, we infer that $\mathcal{H}_{h_i}(u_i, v_i)$ is indeed positive definite.

We next apply the triangle inequality to infer that in the complex torus with $|u| = u_i, |v| = v_i$, the unique global maximum for f_i occurs at (u_i, v_i) (carefully checking that the values at $(\pm u_i, \pm v_i)$ are smaller). Also since $r \leq n - 8$, $p + q + r = n$ and $n \geq K_0$

where we will set K_0 to be large enough constant, it must be the case that $(p/2)^2 + q^2 \geq 4$. Therefore we satisfy all conditions to apply Lemma 29 with $(u_0, v_0) = (u_i, v_i)$, we infer that the coefficient of $u^{p/2}v^q$ in $f_i(u, v)$ equals

$$\frac{f(u_i, v_i)}{2\pi u_i^{p+1} v_i^{q+1} \sqrt{\mathcal{H}_{h_i}(u_i, v_i)}} \left(1 + O\left(\frac{1}{\sqrt{p^2+q^2}}\right)\right).$$

Since $p^2 + q^2 \geq 4$, and $u_i, v_i \leq n$, and that the entries of $\mathcal{H}_{h_i}$ are bounded by $\mathsf{poly}(u_i, v_i)$, we infer that there exists a constant K_1 such that the coefficient is at least

$$n^{-K_1} \cdot \frac{f(u_i, v_i)}{u_i^{p/2} v_i^q} = n^{-K_1} \cdot \frac{(u_i + v_i^2 + 1)^{A_i} (2v_i + 1)^{2C_i}}{u_i^{p/2} v_i^q}$$ ◀

References

1. Kazuyuki Amano. Depth-three circuits for inner product and majority functions. In Satoru Iwata and Naonori Kakimura, editors, *34th International Symposium on Algorithms and Computation, ISAAC 2023, December 3-6, 2023, Kyoto, Japan*, volume 283 of *LIPIcs*, pages 7:1–7:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ISAAC.2023.7.
2. Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
3. Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.
4. Vlado Dancík. Complexity of boolean functions over bases with unbounded fan-in gates. *Inf. Process. Lett.*, 57(1):31–34, 1996. doi:10.1016/0020-0190(95)00182-4.
5. Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. Improving 3n circuit complexity lower bounds. *Comput. Complex.*, 32(2):13, 2023. doi:10.1007/S00037-023-00246-9.
6. Peter Frankl, Svyatoslav Gryaznov, and Navid Talebanfard. A variant of the VC-dimension with applications to depth-3 circuits. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 72:1–72:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.72.
7. Alexander Golovnev, Alexander S. Kulikov, and R. Ryan Williams. Circuit depth reductions. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 24:1–24:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.24.
8. Mika Göös, Ziyi Guan, and Tiberiu Mosnoi. Depth-3 circuits for inner product. *Inf. Comput.*, 300:105192, 2024. doi:10.1016/J.IC.2024.105192.
9. Johan Håstad, Stasys Jukna, and Pavel Pudlák. Top-down lower bounds for depth-three circuits. *Comput. Complex.*, 5(2):99–112, 1995. doi:10.1007/BF01268140.
10. Xuangui Huang, Peter Ivanov, and Emanuele Viola. Affine extractors and AC0-parity. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, volume 245 of *LIPIcs*, pages 9:1–9:14, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.9.
11. Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/JCSS.2001.1774.
12. Jiatu Li and Tianqi Yang. $3.1n - o(n)$ circuit lower bounds for explicit functions. In *54th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 1180–1193. ACM, 2022. doi:10.1145/3519935.3519976.
13. Xin Li. Two source extractors for asymptotically optimal entropy, and (many) more. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1271–1281. IEEE, 2023. doi:10.1109/FOCS57990.2023.00075.

14 Jordan Ninin. Global optimization based on contractor programming: An overview of the ibex library. In *International Conference on Mathematical Aspects of Computer and Information Sciences*, pages 555–559. Springer, 2015. doi:10.1007/978-3-319-32859-1_47.

15 Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364, 2005. doi:10.1145/1066100.1066101.

16 Ramamohan Paturi, Pavel Pudlák, and Francis Zane. Satisfiability coding lemma. *Chic. J. Theor. Comput. Sci.*, 1999, 1999. URL: http://cjtcs.cs.uchicago.edu/articles/1999/11/contents.html.

17 Ramamohan Paturi, Michael E. Saks, and Francis Zane. Exponential lower bounds for depth three boolean circuits. *Comput. Complex.*, 9(1):1–15, 2000. doi:10.1007/PL00001598.

18 Robin Pemantle, Mark C Wilson, and Stephen Melczer. *Analytic combinatorics in several variables*, volume 212. Cambridge University Press, 2024.

19 Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Mathematical foundations of computer science (Proc. Sixth Sympos., Tatranská Lomnica, 1977)*, pages 162–176. Lecture Notes in Comput. Sci., Vol. 53, 1977.

A Proving coefficient asymptotics

In this section, we will prove our required lemma regarding bounding coefficients of power series of an analytic function - Lemma 29.

To help prove this, we will require the following well known result that coefficient expansion of a generating function can be expressed as a Cauchy integral (see equation 1.5 from [18]):

▶ **Theorem 30** (Bivariate Cauchy integral formula). *Let $f : \mathbb{C}^2 \to \mathbb{C}$ be analytic on a compact neighborhood $\mathcal{N}$ around $(0,0)$. Let $f(u,v) = \sum_{r=(r_1,r_2)\in\mathbb{N}^2} C_r u^{r_1} v^{r_2}$ where for all $i \in \mathbb{N}^2$, $C_i \in \mathbb{R}$ and $C_i \geq 0$. Then, for all $r \in \mathbb{N}^2$, it holds that*

$$C_r = \oint_{x\in\mathcal{N}} \left(\frac{1}{2\pi i}\right)^2 f(x) \cdot x_1^{-r_1-1} \cdot x_2^{-r_2-1}\, dx.$$

We will utilize the following result regarding bounding Cauchy integral:

▶ **Theorem 31** (Theorem 5.2 from [18], specialized). *Let $A, \phi : \mathbb{C}^2 \to \mathbb{C}$ be analytic on a compact neighborhood $\mathcal{N}$ around $(0,0)$. Furthermore assume that the real part of ϕ is nonnegative on $\mathcal{N}$ and vanishes only at $(0,0)$, and that the Hessian matrix $\mathcal{H}_\phi$ of ϕ is such that $\mathcal{H}_\phi(0,0)$ is non-singular. Let, $I(\lambda) = \int_{\mathcal{N}} A(z)e^{-\lambda\phi(z)}$. Then for $\lambda \leq \frac{1}{2}$, we have that:*

$$I(\lambda) = A(0,0)\frac{2\pi}{\sqrt{\det(\mathcal{H}_\phi(0,0))}} \cdot \frac{1}{\lambda}\left(1 + O\left(\frac{1}{\lambda}\right)\right).$$

Using these, we obtain the following estimate regarding estimating coefficient of an analytic function:

Proof of Lemma 29. We apply Theorem 30 with the contour $\mathcal{C}$ being product of circles with $|u| = u_0, |v| = v_0$ so that

$$C_{p,q} = \frac{-1}{4\pi^2}\int_{(u,v)\in\mathcal{C}} f(u,v)u^{-p-1}v^{-q-1}dudv.$$

For $\theta = (\theta_1, \theta_2) \in [-\pi, \pi]^2$, we perform change of variables to express the points in the contour as $u = u(\theta) = u_0 e^{i\theta_1}, v = v(\theta) = v_0 e^{i\theta_2}$. We see that $du = u_0 i e^{i\theta_1} d\theta_1$ and $dv = v_0 i e^{i\theta_2} d\theta_2$. We also use the fact that $\exp(h(u,v)) = \frac{f(u,v)}{u^p v^q}$ to obtain that

$$C_{p,q} = \frac{1}{4\pi^2}\int_{\theta\in[\pi,\pi]^2} \exp(h(u(\theta), v(\theta)))d\theta.$$

Let $\lambda = \sqrt{p^2+q^2}$. We perform one last change of variables where we let $h'(u,v) = h(u,v)/\lambda$ to express the integral as:

$$C_{p,q} = \frac{\exp(h(u_0,v_0))}{4\pi^2} \int_{\theta\in[\pi,\pi]^2} \exp\left(\lambda \cdot (h'(u(\theta),v(\theta)) - h'(u_0,v_0))\right) d\theta.$$

We will apply Theorem 31 with $A(\theta) = 1$, $\phi(\theta) = h'(u_0,v_0) - h'(u(\theta),v(\theta))$. We claim the following:

▷ Claim 32.

$$\det(\mathcal{H}_\phi(0,0)) = \left(\frac{u_0v_0}{\lambda}\right)^2 \det(\mathcal{H}_{h'}(u_0,v_0)).$$

We will prove this at the end and for now will just assume it.

We first show that we satisfy all of the preconditions laid out by Theorem 31. First, we have that $\phi(0,0) = 0$ and by unique maximality of u_0, v_0 that for all $\theta \neq (0,0)$, it follows that $\phi(\theta) > 0$. Second, we show that $\mathcal{H}_\phi(0,0)$ is non-singular. By assumption, we know that $\mathcal{H}_h(u_0,v_0)$ is positive definite and so must be true for $\mathcal{H}_{h'}(u_0,v_0)$, making it singular. Therefore by Claim 32, it follows that $\mathcal{H}_\phi(0,0)$ is non-singular as well. Lastly, by assumption we have that $\lambda \leq \frac{1}{2}$. Hence, we can indeed apply Theorem 31 with the same λ to infer that

$$C_{p,q} = \frac{\exp(h(u_0,v_0))}{4\pi^2} \cdot \frac{2\pi}{\sqrt{\det(\mathcal{H}_\phi(0,0))}} \cdot \frac{1}{\lambda} \cdot \left(1 + O\left(\frac{1}{\lambda}\right)\right).$$

Using Claim 32 and the fact that $\exp(h(u_0,v_0)) = \frac{f(u_0,v_0)}{u_0^q v_0^q}$, our desired bound on $C_{p,q}$ follows.

We lastly prove our helper claim that relates $\det(\mathcal{H}_\phi(0,0))$ and $\det(\mathcal{H}_{h'}(u_0,v_0))$.

Proof of Claim 32. Let $g(\theta) = -h'(u(\theta),v(\theta))$ so that $\phi(\theta) = h'(u_0,v_0)+g(\theta)$. Since $h'(u_0,v_0)$ is a constant, we have that $\mathcal{H}_g(0,0) = \mathcal{H}_\phi(0,0)$. Hence, it suffices to show that $\det(\mathcal{H}_g(0,0)) = \left(\frac{u_0v_0}{\lambda}\right)^2 \det(\mathcal{H}_{h'}(u_0,v_0))$. Using chain rule, we see that for $i \in [2]$:

$$\frac{\partial g}{\partial \theta_i} = \frac{\partial h'}{\partial u}\frac{\partial u}{\partial \theta_i} + \frac{\partial h'}{\partial v}\frac{\partial v}{\partial \theta_i}$$

When we evaluate this at $(0,0)$, we will evaluate $\frac{\partial h'}{\partial u}$ and $\frac{\partial h'}{\partial v}$ at (u_0,v_0). By assumption $\nabla_h(u_0,v_0) = 0$ and so $\nabla_{h'}(u_0,v_0) = 0$. Using above, this implies that $\nabla_g(0,0) = 0$.

Using chain rule, the fact that $\nabla_g(0,0) = 0$, and recalling the definition of $u(\theta)$ and $v(\theta)$, we obtain that

$$\mathcal{H}_g(0,0) = D\mathcal{H}_{h'}(u_0,v_0)D$$

where D is the 2×2 diagonal matrix with entries iu_0, iv_0.

With this we finally compute that

$$\begin{aligned}
\det(\mathcal{H}_g(0,0)) &= \det(D\mathcal{H}_{h'}(u_0,v_0)D) \\
&= \det(D)\cdot\det(\mathcal{H}_{h'}(u_0,v_0))\cdot\det(D) \\
&= (u_0v_0)^2\det(\mathcal{H}_{h'}(u_0,v_0)) \\
&= \frac{(u_0v_0)^2}{\lambda^2}\det(\mathcal{H}_h(u_0,v_0))
\end{aligned}$$

where for the last equality we used the fact that $h' = h/\lambda$, which implies $\mathcal{H}_{h'}(u_0,v_0) = \frac{1}{\lambda}\mathcal{H}_h(u_0,v_0)$. ◁

◀

Plethysm is in #BQP

Matthias Christandl
University of Copenhagen, Denmark

Aram W. Harrow
Massachusetts Institute of Technology, Cambridge, MA, USA

Greta Panova
University of Southern California, Los Angeles, USA

Pietro M. Posta
University of Copenhagen, Denmark

Michael Walter
Ludwig-Maximilians-Universität München, Germany
Munich Center for Quantum Science and Technology (MCQST), Germany
Korteweg-de Vries Institute for Mathematics, Amsterdam, The Netherlands
QuSoft, University of Amsterdam, The Netherlands

Abstract

Some representation-theoretic multiplicities, such as the Kostka and the Littlewood-Richardson coefficients, admit a combinatorial interpretation that places their computation in the complexity class #P. Whether this holds more generally is considered an important open problem in mathematics and computer science, with relevance for geometric complexity theory and quantum information. Recent work has investigated the quantum complexity of particular multiplicities, such as the Kronecker coefficients and certain special cases of the plethysm coefficients.

Here, we show that a broad class of representation-theoretic multiplicities is in #BQP. This includes the result that plethysm coefficients are in #BQP, which was only known in certain cases. It also implies all known results on the quantum complexity of previously studied coefficients as special cases, thus unifying, simplifying, and extending prior work. We obtain our result by multiple applications of the Schur transform; recent work has improved its dependence on the local dimension, which is crucial for our work. We further describe a general approach for showing that representation-theoretic multiplicities are in #BQP that captures the approaches of our and previous work. We complement the above by showing that the same multiplicities are also naturally in GapP and obtain polynomial-time classical algorithms when certain parameters are fixed.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Theory of computation → Quantum complexity theory

Keywords and phrases Quantum witness counting, #BQP, algebraic combinatorics, representation theory, representation-theoretic multiplicities

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.28

Related Version *Full Version*: https://arxiv.org/abs/2602.08441 [1]

Funding *Matthias Christandl*: MC acknowledges support from the European Research Council (ERC Grant Agreement No. 818761), VILLUM FONDEN via the QMATH Centre of Excellence (Grant No. 10059) and the Novo Nordisk Foundation (grant NNF20OC0059939 "Quantum for Life").
Aram W. Harrow: AWH acknowledges support from a grant from the Simons Foundation (MP-SIP-00001553, AWH) and from a Fulbright Scholar Grant.
Greta Panova: GP acknowledges partial support from NSF grant CCF:2302174 and from an AMS Birman fellowship, and a residence at ICERM (NSF grant DMS-1929284) in Fall 2025.
Pietro M. Posta: PMP acknowledges support from the European Research Council (ERC Grant Agreement No. 818761), VILLUM FONDEN via the QMATH Centre of Excellence (Grant No. 10059), the Novo Nordisk Foundation (grant NNF20OC0059939 "Quantum for Life") and from a PhD scholarship on Quantum Algorithms from the Danish e-infrastructure Consortium (DeiC), and partial support from Istituto Nazionale di Alta Matematica "Francesco Severi"(INdAM).



© Matthias Christandl, Aram W. Harrow, Greta Panova, Pietro M. Posta, and Michael Walter;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 28; pp. 28:1–28:11



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Michael Walter: MW acknowledges support from the European Research Council (ERC Grant SYMOPTIC, 101040907) by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, 556164098), from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy– EXC-2111– 390814868, and from the German Federal Ministry of Research, Technology and Space (QuSol, 13N17173). Part of this work was conducted while MW was visiting Q-FARM and the Leinweber Institute for Theoretical Physics at Stanford University and the Simons Institute for the Theory of Computing at UC Berkeley.

Acknowledgements We are grateful to Christian Ikenmeyer, Vojtech Havlicek and Martin Larocca for fruitful discussions.

1 Introduction

Computing multiplicities is a central problem in representation theory and algebraic combinatorics, and is directly relevant to quantum information theory and geometric complexity theory. In most cases, no closed-form formulas are known. A natural and fruitful question is to ask whether these multiplicities count combinatorial objects. If so, then one obtains a *positive combinatorial interpretation*. Stanley stated the problem of finding such interpretations for important representation-theoretic multiplicities in his influential paper [48]. Specific multiplicities of interest include: the *Littlewood-Richardson (LR) coefficients*, discussed in 1934 [36], giving the multiplicity of an irreducible representation of the general linear group $\mathrm{GL}(n)$ in the tensor product of two others; the *Kronecker coefficients* of the symmetric group, defined in 1938 [38], giving the multiplicity of an irreducible representation of the symmetric group S_n in the tensor product of two others; and the *plethysm coefficients*, first mentioned by Littlewood in 1936 [34] and defined formally in Ref. [35], which gives the multiplicity of an irreducible $\mathrm{GL}(n)$-representation in the composition of two irreducible GL-representations (see Section 2). By Schur-Weyl duality, the Kronecker coefficients can also be defined in terms of the general linear group, and the LR and plethysm coefficients can also be defined in terms of the symmetric group.

Multiplicities arise naturally across a broad range of areas and applications in mathematics, physics, and computer science, and accordingly they have a long history of study using different methods, from representation theory and invariant theory, to geometry and algebraic combinatorics. For the plethysm coefficients in particular, initial properties were understood via geometric means, notably their stability [6]. The combinatorial investigations started with the unfulfilled hope of finding an interpretation as counting integer points in polytopes (as is known for the LR coefficients, see below), see e.g. the works of Kahle and Michalek [29]. Approaches via tableaux combinatorics were developed by many groups, see e.g. Refs. [21, 20, 39]. The general goal is to give an interpretation as counts of some simple-to-describe but exponentially large set. Ultimately, however, combinatorial formulas are only known in very special situations, see Refs. [22, 40, 43] for some of the known cases.

1.1 The complexity of computing multiplicities

While the notion of a "positive combinatorial interpretation" is not a priori formally defined, the modern interpretation formalizes it as "proving the computational problem is in #P, and deciding positivity is in NP" [41, 45]. Indeed, the complexity class #P captures the problem of counting gadgets (witnesses) which can be recognized efficiently (in polynomial time). This perspective uncovered surprising connections between mathematics and computer science and led to new insights in both fields. The LR coefficients turn out to be in #P, as they admit an interpretation as certain tableaux and integer points in polytopes, whereas their

positivity is even decidable in P [37, 12] as a consequence of the Knutson-Tao proof of the saturation conjecture [32]. In contrast, the problems for Kronecker and plethysm coefficients are still wide open [46]. We know that both coefficients are #P-hard to compute and NP-hard to decide positivity of [26, 23]. However, it is not known whether they are in #P, that is, whether they admit a positive combinatorial interpretation, or whether their positivity is in NP. [1]

As representation-theoretic multiplicities are linear-algebraic quantities defined in high dimensions, it is natural to study their complexity from the perspective of quantum computation. In particular, their very definition as the dimension of the space of multiplicities of some irreducible representation, gives the hope that one could place them in the quantum complexity class #BQP, the quantum analogue of #P, by the intuition that these multiplicities might count the dimension of easy-to-recognize subspaces in some bigger vector space. Indeed, in Ref. [5] it was shown that deciding positivity of Kronecker coefficients is in QMA, the quantum analogue of NP, and computing a certain multiple of these coefficients is in #BQP. Shortly after, it was shown in Ref. [28] that computing the Kronecker coefficients, as well as certain *special* plethysm coefficients (see below), is in #BQP. See also Ref. [17]. In certain parameter regimes, efficient quantum algorithms for computing these multiplicities were proposed in Ref. [33], however many of those cases were later shown to be polynomial-time computable by classical algorithms as well [47], see also Ref. [15]. Thus the quantum and classical complexity of representation-theoretic multiplicities is subtle even in special cases. Here we consider this problem from a general perspective, motivated by the outstanding problem of characterizing the complexity of the plethysm coefficients.

1.2 Relevance in algebraic complexity theory and quantum information theory

In algebraic complexity theory, geometric complexity theory (GCT) is a program that seeks to establish lower bounds, such as separating VNP from VP, by exploiting the symmetries of the universal polynomials, for example the permanent and determinant of a symbolic matrix. One approach via algebraic geometry and representation theory goes via so-called *multiplicity obstructions*. Namely, decompose the coordinate rings of the GL-orbit closures of a VNP-complete polynomial (usually the permanent) into irreducible components, and compare the multiplicities of these components to the ones corresponding to a universal polynomial for VP (usually the determinant). The plethysm coefficients (as well as the Kronecker coefficients) play a natural and significant role in such computations, for example as upper bounds or even as part of explicit formulas for such multiplicities, see Refs. [8, 11, 13, 14, 22, 27]. The GCT program can also be adapted to other situations, such as finding lower bounds for the complexity of the matrix multiplication problem [11], and representation-theoretic multiplicities again turn out to be crucial.

In quantum physics, representation-theoretic multiplicities are ubiquitous since the state of quantum systems are described by unit vectors in high-dimensional complex vector spaces, which are acted upon unitarily by symmetry groups. In some situations, such as high energy physics, the groups and representations of interest are fixed, while in quantum information and computation one naturally considers sequences of groups and representations. One important setting is the one-body quantum marginal problem. Indeed, the compatibility of the one-body

[1] However, it is known that, for a broad class multiplicities, a natural *asymptotic* version of the positivity problem is in NP ∩ coNP [9].

marginals with an overall pure state is guided by the Kronecker coefficients [16, 18, 30], while the plethysm coefficients appear in the fermionic version of the problem [2], known as the one-body N-representability problem in quantum chemistry [19], which has been studied at least since the early 1970s [4].

2 Our contributions

We now present our main results, starting with the quantum complexity of plethysm coefficients. We first give a brief definition of plethysms (see the full version of the paper [1] for more details). Let $V \cong \mathbb{C}^n$ be a finite-dimensional complex vector space. Recall that the irreducible polynomial complex representations of the general linear group $\mathrm{GL}(V)$ are labeled by integer partitions with no more than n parts. Given such a partition ν, we denote by $\rho_{\mathrm{GL}(V),\nu}\colon \mathrm{GL}(V) \to \mathrm{GL}(\{\nu\}_{\mathrm{GL}(V)})$ the corresponding irreducible representation, where $\{\nu\}_{\mathrm{GL}(V)}$ denotes the representation space or *Weyl module*. If ν has more than n parts, we define $\rho_{\mathrm{GL}(V),\nu}$ as the zero-dimensional representation. We abbreviate $H \coloneqq \mathrm{GL}(V)$, and consider the group $G \coloneqq \mathrm{GL}(\{\nu\}_H)$ along with one of its polynomial representations $\rho_{G,\mu}$, where μ is again an integer partition. Then the composition $\rho_{G,\mu} \circ \rho_{H,\nu}\colon H \to \mathrm{GL}(\{\mu\}_G)$ is a representation of $H = \mathrm{GL}(V)$. We shall denote the corresponding representation space by $\{\mu\}{\downarrow}_H^{\mathrm{GL}(\{\nu\}_H)}$ to remember that we regard it as a module of H, by *restriction* along the irreducible representation $\rho_{H,\nu}\colon H \to G = \mathrm{GL}(\{\nu\}_H)$. In general, the representation $\rho_{G,\mu} \circ \rho_{H,\nu}$ is reducible and hence it decomposes into irreducible H-representations. Thus:

$$\{\mu\}{\downarrow}_H^{\mathrm{GL}(\{\nu\}_H)} \cong \bigoplus_\lambda \underbrace{\{\lambda\}_H \oplus \cdots \oplus \{\lambda\}_H}_{a^\lambda_{\mu,\nu}} \cong \bigoplus_\lambda \{\lambda\}_H \otimes \mathbb{C}^{a^\lambda_{\mu,\nu}}. \tag{2.1}$$

The *plethysm coefficient* $a^\lambda_{\mu,\nu}$ is defined as the multiplicity of the irreducible module $\{\lambda\}_H$ in this decomposition. Here we assume that λ has no more than $n = \dim(V)$ parts, in which case the coefficient does not depend on the choice of n but only on the three partitions λ, μ, ν. An equivalent definition of the plethysm coefficient $a^\lambda_{\mu,\nu}$ is as the coefficient of s_λ in the expansion of $s_\mu[s_\nu]$. Here s_μ, s_ν, s_λ denote Schur functions, see the full version of the paper [1] for a precise definition.

▶ **Problem 1** (Plethysm coefficients).
Input: Three partitions λ, μ, ν, encoded as lists of positive integers, in unary.
Task: Compute the plethysm coefficient $a^\lambda_{\mu,\nu}$.

Encoding the partitions in unary is a standard convention in the literature. Thus the input size is $\Theta(|\lambda| + |\mu| + |\nu|)$. We give two concrete examples of plethysm coefficients:

- If both $\mu = (m)$ and $\nu = (d)$ are partitions with a single part, then $\{\mu\}{\downarrow}_H^{\mathrm{GL}(\{\nu\}_H)} = \mathrm{Sym}^m(\mathrm{Sym}^d(V))$ can be interpreted as the space of homogeneous polynomials of degree m in variables that are coefficients on the space of homogeneous polynomials of degree d. This space is central in geometric complexity theory and is the main object in the Foulkes conjecture [24].
- If $\mu = (m)$ and $\nu = (1^d) = (1, \ldots, 1)$, then $\{\mu\}{\downarrow}_H^{\mathrm{GL}(\{\nu\}_H)} = \mathrm{Sym}^m(\mathrm{Alt}^d(V))$ corresponds to the space of polynomials in the coefficients of a wavefunction of d fermions, each having Hilbert space V. This is directly relevant to the N-representability problem in quantum chemistry [2, 31].

The general case is a major problem in algebraic combinatorics as stated in Ref. [48, Problem 9]. Prior work [28] showed that the former example is in #BQP, but already the latter example was out of reach for earlier methods. Here we exhibit an algorithm which proves that the general plethysm problem is in #BQP:

▶ **Theorem 2.** *Problem 1 is in* #BQP*. As a consequence, the problem of deciding if a plethysm coefficient is positive,* $a^{\lambda}_{\mu\nu} > 0$*, is in* QMA*.*

The corresponding quantum algorithm is sketched in Figure 1 and we describe its main ideas in Section 3. The above also implies an analogous result for the *restriction coefficients* r^{μ}_{λ}, defined as the multiplicy of an irreducible S_n-representation (Specht module) in the restriction of an irreducible $\mathrm{GL}(n)$-representation (Weyl module) to the subgroup $S_n \subset \mathrm{GL}(n)$ of permutation matrices. Indeed, the restriction coefficients can be expressed as a positive sum of plethysms: using symmetric functions (see the full version [1] for definitions), we have $r^{\mu}_{\lambda} = \langle s_\mu, s_\lambda[1 + h_1 + h_2 + \cdots]\rangle$. Thus, computing restriction coefficients is in #BQP and their positivity is in QMA, as an immediate corollary of Theorem 2.

We obtain Theorem 2 by specialization of a general theorem that applies to a broad class of multiplicities known as *branching multiplicities (or coefficients)*. They are defined as follows: We consider groups G and H, together with a group homomorphism $\varphi\colon H \to G$. Any representation $\pi\colon G \to \mathrm{GL}(W)$ of G then *restricts* to a representation of H by composition with φ, i.e., $\pi \circ \varphi\colon H \to \mathrm{GL}(W)$. The corresponding branching multiplicities are defined as the multiplicities of irreducible H-representation in the restriction $\pi \circ \varphi$. Here we consider branching multiplicites in the setting where G and H are *products of general linear groups*, and where the group homomorphism φ and the G-representation π are specified succinctly in terms of representation-theoretic data, see the full version of the paper [1] for a formal definition. This encompasses, among others, all coefficients studied in prior work: Kostka numbers, LR coefficients, and Kronecker coefficients, (of course, the first two are even in $\#\mathsf{P} \subset \#\mathsf{BQP}$, while for the latter this is an open problem). It is known that computing branching multiplicities can be done in classical polynomial time when G and H are *fixed*. This holds not just for products of general linear groups, but more generally in the setting of complex reductive (or compact connected) Lie groups [15]. In contrast, here we allow the groups (as well as the representations) to be part of the input. This is substantially more difficult since it involves representations that in general have dimensions that are *exponential* in the input size.

▶ **Theorem 3.** *Computing branching multiplicities for products of general linear groups is in* #BQP*. As a consequence, deciding their positivity is in* QMA*.*

We prove Theorem 3 in the full version of the paper [1] by giving a quantum algorithm that is perfectly sound and complete (if the gate set is such that the circuit can be implemented perfectly), similarly to the algorithms in Refs. [5, 28]. In particular, this shows that the positivity problem is in QMA_1 if there exists a universal gate set such that the high-dimensional version of the quantum Schur transform on $(\mathbb{C}^d)^{\otimes N}$ can be implemented exactly for any choice of N and d. We discuss the main technical ideas behind our quantum algorithms in Section 3. Moreover, in the full version of the paper [1], we describe a general approach for showing that representation-theoretic multiplicities are in #BQP that subsumes our quantum algorithms as well as those of the prior works [5, 28] and explains their relation.

As discussed, the plethysm coefficients and in fact most representation-theoretic coefficients are #P-hard, and it is a central open question to determine when they are in #P, see Refs. [41, 45]. By Theorem 2, a negative answer to this question would imply a breakthrough separation between the complexity classes #P and #BQP. To gain further insight into their computational complexity, we complement our quantum results by showing that the branching multiplicities are also in GapP extending earlier results in Refs. [10, 42, 44].

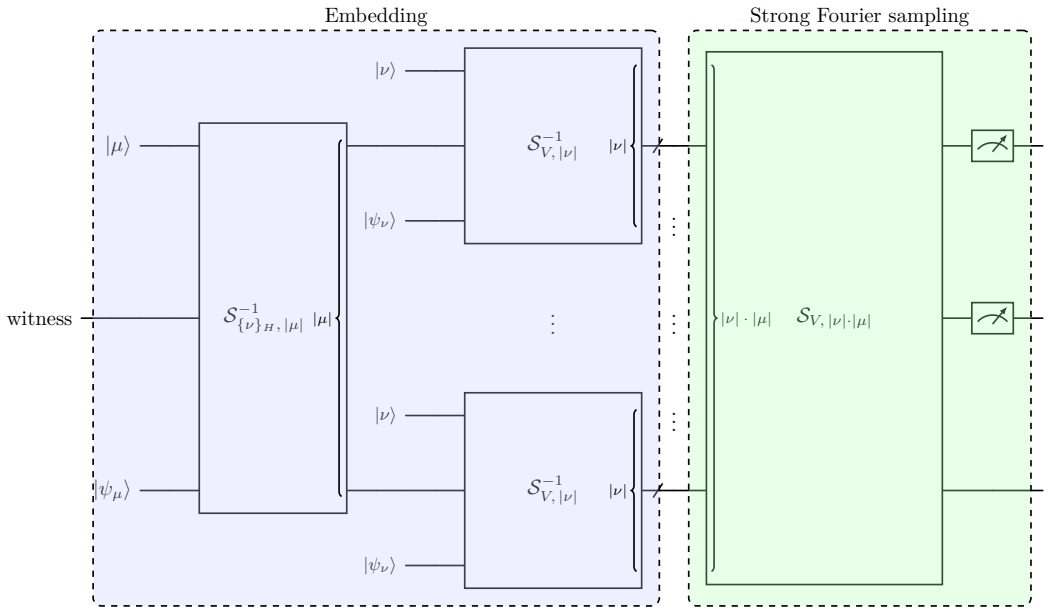


Figure 1 Quantum circuit for the plethysm coefficient $a^\lambda_{\mu,\nu}$ (Theorem 2). The witness is a state in the space $\{\mu\}_{\mathrm{GL}(\{\nu\}_H)}$. The first inverse Schur transform acts on the space $\{\nu\}_H^{\otimes|\mu|}$, taking an input in the Schur basis and transforming it in the computational basis, consisting of $|\mu|$ registers, each isomorphic to $\{\nu\}_H$. Each inverse Schur transform from the second layer acts on $V^{\otimes|\nu|}$, again transforming from the Schur basis to the computational basis, consisting of $|\nu|$ registers of dimension n. After the final Schur transform, taking an input in the computational basis of $V^{\otimes|\mu||\nu|}$ and transforming it into the Schur basis, we measure the registers corresponding to the partition and the Weyl module. We accept if and only if the former returns outcome λ and the latter p_0, where $|p_0\rangle \in \{\lambda\}_H$ is an arbitrary fixed basis state of the Weyl module.

▶ **Theorem 4.** *Computing branching multiplicities for products of general linear groups is in* GapP.

We prove Theorem 4 in the full version of the paper [1], and we also present a more specialized algorithm for computing the plethysm coefficients. In particular we obtain a polynomial time algorithm when the number of parts of λ and the size of μ are fixed, extending the results of Ref. [47].

3 Methods

We discuss the main ideas behind our results, focusing on the special case of plethysm coefficients for concreteness (Theorem 2). For further details see the full version of the paper [1].

Quantum algorithm for plethysm coefficients

Our algorithm is depicted in circuit form in Figure 1. Let λ, μ, ν be partitions and $V = \mathbb{C}^n$. We may assume that the number of parts of the partitions and n are such that the corresponding Weyl modules are nonzero, otherwise the plethysm coefficient $a^\lambda_{\mu,\nu} = 0$. Our approach uses Schur-Weyl duality to embed the H-module $\{\mu\}\downarrow^G_H$ inside $V^{\otimes|\nu|\cdot|\mu|}$, where $G = \mathrm{GL}(\{\nu\}_H)$ and $H = \mathrm{GL}(V)$ as before. This is achieved as follows: We first embed the G-module $\{\mu\}_G$

in a tensor power of $\{\nu\}_H$, which is the defining module of $G = \mathrm{GL}(\{\nu\}_H)$. We then consider each $\{\nu\}_H$ as a H-module and embed it into a tensor power of V, which is the defining module of $H = \mathrm{GL}(V)$. Since $\{\mu\}\downarrow_H^G$ is nothing but the restriction of the G-module $\{\mu\}_G$ to H, we obtain the following chain of H-equivariant inclusions

$$\{\mu\}\downarrow_H^G \hookrightarrow (\{\nu\}_H)^{\otimes|\mu|} \hookrightarrow V^{\otimes|\nu|\cdot|\mu|}. \tag{3.1}$$

In order to implement these embeddings, we employ *inverse* quantum Schur transforms, as these allow us to map from the *Schur basis* of the tensor power spaces to the standard basis (blue box in Figure 1). After the embeddings, we implement a final (forward) quantum Schur transform on all $|\nu| \cdot |\mu|$ registers and measure in the Schur basis – accepting if and only if we obtain the partition λ, meaning that the input state was in the λ-isotypic component of $\{\mu\}\downarrow_H^G$, and if we obtain an arbitrary but fixed basis state p_0 on the Weyl module register (green box in Figure 1). As inclusions are H-equivariant, they preserve the dimension of the isotypic component, and they act as the identity on the Weyl module. As such, the entire circuit implements a projective measurement, and the dimension of the space of witnesses that are accepted with probability one (assuming the Schur transform is implemented exactly) is precisely the multiplicity of $\{\lambda\}_H$ in $\{\mu\}\downarrow_H^G$. By Equation (2.1), this multiplicity is the plethysm coefficient $a^\lambda_{\mu,\nu}$. See the full version of the paper [1] for more details and for the analysis of the algorithm soundness, completeness and complexity.

Generalization to branching multiplicities

Our result for the branching problem for products of general linear groups (Theorem 3) is a natural generalization of this idea. The quantum algorithm is formally described and analyzed in detail in the full version of the paper [1]. One conceptual contribution, detailed in the full version, is to identify an appropriate definition of the computational problem. This is subtle, as it involves objects that will in general be exponentially large and hence need to be succinctly specified in order to obtain a nontrivial problem that captures the plethysm coefficients and other classical coefficients as special cases. On the algorithmic side, we employ a similar sequence of embeddings as before, this time on products of tensor powers of the defining modules, and implemented by a sequence of parallel quantum circuits. In our general setting, the defining modules for G need no longer be irreducible as H-modules, requiring us to implement intermediate measurements in order to realize the subsequent embeddings. See the full version of the paper [1] for further details.

Comparison with prior work

Previous results on the quantum computation of Kronecker and special plethysm coefficient employed the formulation of these coefficients in terms of the symmetric group S_n and rely on Beals' quantum Fourier transform for S_n [3]. In principle, any plethysm coefficient can be computed as branching multiplicities of representations of wreath products of symmetric groups. However, as pointed out in Ref. [33], a direct use of Beals' quantum Fourier transform leads to an exponential blowup in general. As such it is not immediately clear how to generalize the symmetric group approach to the general plethysm problem (Problem 1). We overcome this challenge by working with the representation theory of the general linear group. In this regard we may view this approach as the "Schur-Weyl dual" of prior approaches to treat representation-theoretic multiplicities. We emphasize that the local dimensions in our setting (such as the dimension of $\{\nu\}_H$) can be *exponentially large*. Accordingly, we require a version of the quantum Schur transform that only depends polylogarithmically

on the local dimensions. This has recently been achieved in Ref. [7], see also Ref. [25]. In contrast, for the Kronecker coefficients the relevant local dimension are polynomial and hence the original quantum Schur transform suffices. We discuss quantum algorithms for the Schur transform in more detail in the full version of the paper [1].

Unified approach towards multiplicities

Despite the difference between these approaches there is a unified picture that captures both and which applies even more broadly, to the general problem of the multiplicity of an irreducible H-module V_λ in an H-module V, for some group H. Consider the following two-step procedure:

1. Embed the H-module V we wish to decompose into an H-module V_{model}.
2. Apply strong Fourier sampling for V_{model} and accept if the outcome consists of is the label of the irreducible representation V_λ, together with a fixed basis state $|p\rangle \in V_\lambda$.

It is clear that if these two steps can be efficiently implemented, then the multiplicity problem is in #BQP, and the corresponding decision problem is in QMA. Remarkably, this captures both the approach of this work and the prior works:

- For the plethysm problem, $V = \{\mu\}{\downarrow}_H^G$ and $V_{\text{model}} = (\mathbb{C}^n)^{\otimes|\nu|\cdot|\mu|}$. The first (embedding) step is achieved by the two layers of inverse Schur transforms (blue box in Figure 1), while the second (strong Fourier sampling) step is achieved by the (forward) Schur transform and subsequent measurements (green box in Figure 1). Our more general algorithm for the branching problem of products of general linear groups likewise fits this picture.
- The key ingredient of Refs. [5, 28] is the *generalized phase estimation* circuit [25], which likewise admits a natural interpretation in this framework. Here, $V_{\text{model}} = \mathbb{C}[H] \otimes V$, where $\mathbb{C}[H]$ is the group algebra and H acts on $\mathbb{C}[H]$ but not on V. The first (embedding) step is realized using the controlled group action of V (that is, the unitary $|h\rangle\,|\psi\rangle \mapsto |h\rangle\,|h \cdot \psi\rangle$ on $\mathbb{C}[H] \otimes V$), and the second (strong Fourier sampling) step is implemented using the Fourier transform of the group. [2] This procedure applies whenever the controlled group action and the Fourier transform of the group can be efficiently implemented, for further details see the full version of the paper [1].

We may also combine these two ideas. For example, the restriction coefficients r_λ^μ can be captured by an embedding step that first uses an inverse Schur transform (as in the first bullet point) to realize $\{\lambda\}_{\mathrm{GL}(n)}$ in $V = (\mathbb{C}^n)^{\otimes|\lambda|}$, followed by GPE (as in the second bullet point), noting that the controlled group action of S_n on $\mathbb{C}^n$ can be efficiently implemented. We further expand on this general perspective in the full version of the paper [1].

References

1 Matthias Christandl, Aram W. Harrow, Greta Panova, Pietro M. Posta, and Michael Walter. Plethysm is in #BQP, full version, 2026. arXiv:2602.08441.

2 M. Altunbulak and A. Klyachko. The Pauli principle revisited. *Communications in Mathematical Physics*, 282(1):287–322, 2008. doi:10.1007/s00220-008-0552-z.

3 Robert Beals. Quantum computation of Fourier transforms over symmetric groups. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, pages 48–53. ACM, 1997. doi:10.1145/258533.258548.

[2] In Ref. [5] the authors use *weak* Fourier sampling and therefore prove membership in #BQP for a *scaled* version of the Kronecker coefficients (namely, the Kronecker coefficients multiplied by the dimension of corresponding irreducible representations).

4 R. E. Borland and K. Dennis. The conditions on the one-matrix for three-body fermion wavefunctions with one-rank equal to six. *Journal of Physics B: Atomic and Molecular Physics*, 5(1):7–15, 1972. doi:10.1088/0022-3700/5/1/009.

5 Sergey Bravyi, Anirban Chowdhury, David Gosset, Vojtěch Havlíček, and Guanyu Zhu. Quantum complexity of the Kronecker coefficients. *PRX Quantum*, 5(1):010329, 2024. doi:10.1103/PRXQuantum.5.010329.

6 Michel Brion. Stable properties of plethysm: on two conjectures of Foulkes. *Manuscripta Mathematica*, 80:347–371, 1993. doi:10.1007/BF03026558.

7 Adam Burchardt, Jiani Fei, Dmitry Grinko, Martin Larocca, Maris Ozols, Sydney Timmerman, and Vladyslav Visnevskyi. High-dimensional quantum Schur transforms, 2025. doi:10.48550/arXiv.2509.22640.

8 Peter Bürgisser, Matthias Christandl, and Christian Ikenmeyer. Even partitions in plethysms. *Journal of Algebra*, 328(1):322–329, 2011. doi:10.1016/j.jalgebra.2010.10.031.

9 Peter Bürgisser, Matthias Christandl, Ketan Mulmuley, and Michael Walter. Membership in moment polytopes is in NP and coNP. *SIAM Journal on Computing*, 46(3):972–991, 2017. doi:10.1137/15M1048859.

10 Peter Bürgisser and Christian Ikenmeyer. The complexity of computing Kronecker coefficients. In *FPSAC'08 - 20th International Conference on Formal Power Series and Algebraic Combinatorics*, 2008. doi:10.46298/dmtcs.3622.

11 Peter Bürgisser and Christian Ikenmeyer. Geometric complexity theory and tensor rank. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing (STOC '11)*, pages 509–518. ACM, 2011. doi:10.1145/1993636.1993704.

12 Peter Bürgisser and Christian Ikenmeyer. Deciding positivity of Littlewood–Richardson coefficients. *SIAM Journal on Discrete Mathematics*, 27(4):1639–1681, 2013. doi:10.1137/120892532.

13 Peter Bürgisser, Christian Ikenmeyer, and Jesko Hüttenhain. Permanent versus determinant: not via saturations. *Proceedings of the American Mathematical Society*, 145(3):1247–1258, 2017. doi:10.1090/proc/13310.

14 Peter Bürgisser, Christian Ikenmeyer, and Greta Panova. No occurrence obstructions in geometric complexity theory. *Journal of the American Mathematical Society*, 32(1):163–193, 2019. doi:10.1090/jams/908.

15 Matthias Christandl, Brent Doran, and Michael Walter. Computing multiplicities of Lie group representations. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 639–648. IEEE, 2012. doi:10.1109/FOCS.2012.43.

16 Matthias Christandl, Aram W. Harrow, and Graeme Mitchison. On nonzero Kronecker coefficients and what they tell us about spectra. *Communications in Mathematical Physics*, 270(3):575–585, 2007. doi:10.1007/s00220-006-0157-3.

17 Matthias Christandl, Aram W. Harrow, and Michael Walter. Presentation at the algorithms and complexity in algebraic geometry reunion workshop. Presented by M. Walter at the Simons Institute for the Theory of Computing, Berkeley, 2015.

18 Matthias Christandl and Graeme Mitchison. The spectra of quantum states and the Kronecker coefficients of the symmetric group. *Communications in Mathematical Physics*, 261(3):789–797, 2006. doi:10.1007/s00220-005-1435-1.

19 A. J. Coleman and V. I. Yukalov. *Reduced Density Matrices: Coulson's Challenge*. Lecture Notes in Chemistry, Vol. 72. Springer Berlin Heidelberg, 2000. doi:10.1063/1.1359715.

20 Laura Colmenarejo, Rosa Orellana, Franco Saliola, Anne Schilling, and Mike Zabrocki. The mystery of plethysm coefficients. *Open Problems in Algebraic Combinatorics (Proc. Sympos. Pure Math.)*, 110:275–292, 2024. doi:10.48550/arXiv.2208.07258.

21 Melanie de Boeck, Rowena Paget, and Mark Wildon. Plethysms of symmetric functions and highest weight representations. *Transactions of the American Mathematical Society*, 374(11):8013–8043, 2021. doi:10.1090/tran/8481.

22 Julian Dörfler, Christian Ikenmeyer, and Greta Panova. On geometric complexity theory: Multiplicity obstructions are stronger than occurrence obstructions. *SIAM Journal on Applied Algebra and Geometry*, 4(2):354–376, 2020. doi:10.1137/19M1287638.

23 Nick Fischer and Christian Ikenmeyer. The computational complexity of plethysm coefficients. *Computational Complexity*, 29:1–43, 2020. doi:10.1007/s00037-020-00198-4.

24 H. O. Foulkes. Concomitants of the quintic and sextic up to degree four in the coefficients of the ground form. *Journal of the London Mathematical Society*, 1(3):205–209, 1950. doi:10.1112/jlms/s1-25.3.205.

25 Aram W. Harrow. *Applications of coherent classical communication and the Schur transform to quantum information theory*. PhD thesis, Massachusetts Institute of Technology, 2005. arXiv:quant-ph/0512255.

26 Christian Ikenmeyer, Ketan Mulmuley, and Michael Walter. On vanishing of Kronecker coefficients. *Computational Complexity*, 26:949–992, 2017. doi:10.1007/s00037-017-0158-y.

27 Christian Ikenmeyer and Greta Panova. Rectangular Kronecker coefficients and plethysms in geometric complexity theory. *Advances in Mathematics*, 319:40–66, 2017. doi:10.1016/j.aim.2017.08.024.

28 Christian Ikenmeyer and Sathyawageeswar Subramanian. A remark on the quantum complexity of the Kronecker coefficients, 2023. arXiv:2307.02389.

29 Thomas Kahle and Mateusz Michałek. Plethysm and lattice point counting. *Foundations of Computational Mathematics*, 16(5):1241–1261, 2016. doi:10.1007/s10208-015-9275-7.

30 Alexander A. Klyachko. Quantum marginal problem and representations of the symmetric group, 2004. arXiv:quant-ph/0409113.

31 Alexander A. Klyachko. Quantum marginal problem and N-representability, 2005. arXiv:quant-ph/0511102.

32 Allen Knutson and Terence Tao. The honeycomb model of $\mathrm{GL}_n(\mathbb{C})$ tensor products I: Proof of the saturation conjecture. *Journal of the American Mathematical Society*, 12(4):1055–1090, 1999. doi:10.1090/s0894-0347-99-00299-4.

33 Martin Larocca and Vojtech Havlicek. Quantum algorithms for representation-theoretic multiplicities, 2024. arXiv:2407.17649.

34 D. E. Littlewood. Polynomial concomitants and invariant matrices. *Journal of the London Mathematical Society*, s1-11(1):49–55, 1936. doi:10.1112/jlms/s1-11.1.49.

35 D. E. Littlewood. Products and plethysms of characters with orthogonal, symplectic and symmetric groups. *Canadian Journal of Mathematics*, 10:17–32, 1958. doi:10.4153/CJM-1958-002-7.

36 Dudley Ernest Littlewood and Archibald Read Richardson. Group characters and algebra. *Philosophical Transactions of the Royal Society of London. Series A*, 233:99–141, 1934. doi:10.1098/rsta.1934.0015.

37 Ketan Mulmuley, Hariharan Narayanan, and Milind Sohoni. Geometric complexity theory III: on deciding nonvanishing of a Littlewood–Richardson coefficient. *Journal of Algebraic Combinatorics*, 36(1):103–110, 2012. doi:10.1007/s10801-011-0325-1.

38 Francis D. Murnaghan. On the analysis of the Kronecker product of irreducible representations of the symmetric group. *American Journal of Mathematics*, 60(3):761–784, 1938. doi:10.1073/pnas.41.7.515.

39 Rosa Orellana, Franco Saliola, Anne Schilling, and Mike Zabrocki. Plethysm and the algebra of uniform block permutations. *Algebraic Combinatorics*, 5(5):1165–1203, 2022. doi:10.5802/alco.243.

40 Rosa Orellana, Franco Saliola, Anne Schilling, and Mike Zabrocki. From quasi-symmetric to Schur expansions with applications to symmetric chain decompositions and plethysm. *The Electronic Journal of Combinatorics*, 31(4):P4.23, 2024. doi:10.37236/12950.

41 Igor Pak. What is a combinatorial interpretation? In *Open Problems in Algebraic Combinatorics*, volume 110 of *Proc. Sympos. Pure Math.*, pages 191–260. Amer. Math. Soc., 2024. doi:10.1090/pspum/110/02007.

42 Igor Pak and Greta Panova. On the complexity of computing Kronecker coefficients. *Computational Complexity*, 26:1–36, 2017. doi:10.1007/s00037-015-0109-4.

43 Igor Pak, Greta Panova, and Joshua P. Swanson. A combinatorial interpretation for certain plethysm and Kronecker coefficients, 2025. Unpublished manuscript.

44 Igor Pak and Colleen Robichaux. Signed combinatorial interpretations in algebraic combinatorics, 2024. doi:10.48550/arXiv.2406.13902.

45 Greta Panova. Computational complexity in algebraic combinatorics. In *Current Developments in Mathematics*. International Press Boston, 2023. doi:10.4310/CDM.250429173831.

46 Greta Panova. Complexity and asymptotics of structure constants. In *Open Problems in Algebraic Combinatorics*, volume 110 of *Proc. Sympos. Pure Math.*, pages 61–86. Amer. Math. Soc., 2024. doi:10.1090/pspum/110/02016.

47 Greta Panova. Polynomial time classical versus quantum algorithms for representation theoretic multiplicities, 2025. doi:10.48550/arXiv.2502.20253.

48 Richard P. Stanley. Positivity problems and conjectures in algebraic combinatorics. *Mathematics: Frontiers and Perspectives*, pages 295–319, 2000. doi:10.5281/zenodo.18768172.

Frontier Space-Time Algorithms Using Only Full Memory

Petr Chmel
Computer Science Institute of Charles University, Prague, Czech Republic

Aditi Dudeja
The School of Data Science, The Chinese University of Hong Kong, Shenzhen, China

Michal Koucký
Computer Science Institute of Charles University, Prague, Czech Republic

Ian Mertz
Computer Science Institute of Charles University, Prague, Czech Republic

Ninad Rajgopal
Computer Science Institute of Charles University, Prague, Czech Republic

Abstract

We develop catalytic algorithms for fundamental problems in algorithm design that run in polynomial time, use only $\mathcal{O}(\log(n))$ workspace, and use sublinear catalytic space matching the best-known space bounds of non-catalytic algorithms running in polynomial time.

First, we design a polynomial time algorithm for directed s-t connectivity using $n/2^{\Theta(\sqrt{\log n})}$ catalytic space, which matches the state-of-the-art time-space bounds in the non-catalytic setting [20], and improves the catalytic space usage of the best known algorithm [43]. Furthermore, using only $\mathcal{O}(\log(n))$ random bits we get a randomized algorithm whose running time nearly matches the fastest time bounds known for space-unrestricted algorithms.

Second, we design polynomial time algorithms for the problems of computing Edit Distance, Longest Common Subsequence, and the Discrete Fréchet Distance, again using $n/2^{\Theta(\sqrt{\log n})}$ catalytic space. This again matches non-catalytic time-space frontier for Edit Distance and Least Common Subsequence [58].

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Complexity classes

Keywords and phrases Catalytic computation, Connectivity, Edit distance, Discrete Fréchet distance

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.29

Related Version *Full Version*: https://arxiv.org/abs/2602.21089

Funding *Michal Koucký, Ian Mertz, Ninad Rajgopal:* Partially supported by the Grant Agency of the Czech Republic under the grant agreement no. 24-10306S and by Charles Univ. project UNCE 24/SCI/008.
Petr Chmel: Partially supported by the Grant Agency of the Czech Republic under the grant agreement no. 24-10306S, by GAUK project No. 70924 and by Charles Univ. project UNCE 24/SCI/008.
Aditi Dudeja: This work was initiated while the author was affiliated with the University of Salzburg. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 947702).

© Petr Chmel, Aditi Dudeja, Michal Koucký, Ian Mertz, and Ninad Rajgopal;
licensed under Creative Commons License CC-BY 4.0

1 Introduction

1.1 Catalytic Computation

Recently, a new paradigm for space-bounded computation called *catalytic computing* has received a great deal of attention. In short, a catalytic machine is one whose memory is almost completely full with arbitrary data at the start; while it may freely use this full memory, known as *catalytic space*, in a read-write manner as with ordinary space, it must restore the initial data to the catalytic memory at the end of the computation. Catalytic computing was defined by Buhrman, Cleve, Koucký, Loff, and Speelman, in [32], who showed that the addition of polynomial catalytic space to a log-space machine gives it significant power, more so even than non-determinism, randomness, or both.

Since then there have been a host of results regarding the catalytic model, such as structural results [34, 51, 44, 41, 39, 60], placing further problems into the model [63, 6, 8, 5], and alternative catalytic models [23, 52, 50, 33, 24, 38] (see surveys of Koucký [59] and Mertz [62] for more background). Furthermore, the techniques of using full memory were also used in the context of ordinary space-bounded computation, giving rise to a novel paradigm for derandomization [45, 61, 46, 53] as well as shedding new light on the relationship of space and time [40, 42, 70, 68].

While all problems in NL are computable in polynomial time CL (i.e., the class of problems computable by a log-space machine with polynomial catalytic space), the approach of [32] incurs a large overhead in both time and catalytic space. In response to this, there been a recent focus, responding to a question posed by Mertz [62], of truly efficient catalytic algorithms whose time and catalytic space usage are both comparable to the best known results in the non-catalytic setting. Cook and Pyne [43] showed that connectivity in graphs (STCONN) can be solved in quadratic time using linear catalytic space; they also show a similar result for estimating random walks. These algorithms are conceptually simple enough to be implemented in practice – to wit, they are only a $\mathsf{poly}(\log(n))$ factor slower than breadth-first search – and redevelop pre-existing techniques from the catalytic literature in new and streamlined ways.

In this work we push the algorithmic frontier of connectivity and other problems, by asking whether we can capture the best *space-time* algorithms, namely those using (reasonable) polynomial time and sublinear space, entirely within the catalytic paradigm.

1.2 Sublinear-Space Polynomial-Time Algorithms

Connectivity is an extremely fundamental primitive in graph algorithms, both in theory and in practice. In complexity theory the many variants of STCONN characterize logspace and their adjacent classes, while on the algorithmic side there has been an extensive history of studying its space, time, space-time trade-offs and heuristics.

While the straightforward DFS and BFS algorithms for STCONN use linear time in the number of edges, the space requirement for these algorithms is linear in the number of vertices. In contrast, Savitch's theorem shows that this problem can be solved in $\mathcal{O}(\log^2(n))$ space and time $n^{\mathcal{O}(\log(n))}$ [67]. To meet both conditions, i.e. run in polynomial time and use sublinear space, a result by Barnes, Buss, Ruzzo, and Schieber [20] gives space complexity $\frac{n}{2^{\Theta\left(\sqrt{\log(n)}\right)}}$.

Except for restricted graph classes [10, 55, 65, 64], no better results are known. Furthermore, the algorithm of [20] can be implemented in a restricted model of computation called Node Naming Jumping Automata on Graphs (NNJAG), for which we have a matching space lower bound in the polynomial time regime due to Edmonds, Poon, and Achlioptas [48], even for probabilistic algorithms.

There are also a number of fundamental algorithmic questions which reduce to connectivity, including Edit Distance (ED), Longest Common Subsequence (LCS), and Discrete Fréchet Distance (DFD). The first two, which quantify how close two strings are in various ways, are very well-studied problems with applications in various areas such as data analysis, computational biology, and text processing [25, 1, 16, 17, 18, 2, 3, 11, 12, 13, 14, 15, 19, 21, 26, 29, 27, 28]. Meanwhile, DFD has been studied extensively in computational geometry, motivated by applications such as dynamic time-warping [56], matching of time series in databases [57], and analysis of moving objects [30, 31].

For ED and LCS, the classic dynamic programming algorithms need linear space, and similarly for DFD, the original dynamic programming algorithm of Eiter and Mannila [49] takes $\mathcal{O}(mn)$ time (where m and n are numbers of vertices of the polygonal chains); this was subsequently improved to $\mathcal{O}(n)$ space and $\mathcal{O}\left(\frac{mn\log\log(n)}{\log(n)}\right)$ time by Agarwal et al. [4]. On the other hand, if we allow for $n^{\mathcal{O}(\log(n))}$ runtime, then we can solve these problems in $\mathcal{O}(\log^2 n)$ space via a reduction to STCONN. With regards to space-time results, their respective reductions to STCONN involve a quadratic blowup in the input size n, which would render the result of [20] useless; however, a recent work of Kiyomi, Horiyama, and Otachi [58] gives a polynomial time algorithm for LCS and ED using $\frac{n}{2^{\Theta\left(\sqrt{\log(n)}\right)}}$ space. While better algorithms are known in the approximation and/or streaming regimes [35, 16, 66, 37], any improvements for these problems beyond the STCONN threshold seem firmly out of reach [37].

1.3 Our Results

In this work, we develop polynomial-time algorithms using catalytic space that matches the $\frac{n}{2^{\Omega\left(\sqrt{\log(n)}\right)}}$ space-bound frontier for polynomial-time algorithms that solve the aforementioned problems, with all the space being borrowed from full memory except for $\mathcal{O}(\log n)$ bits.

We begin with STCONN, where additionally $\mathcal{O}(\log(n))$ random bits are sufficient to nearly match the time bounds of the best space-*unrestricted* algorithms.

▶ **Theorem 1** (Sublinear catalytic space algorithms for STCONN). *There exists a deterministic algorithm solving* STCONN *on n-vertex graphs in time* poly(n)*, space $\mathcal{O}(\log(n))$, and catalytic space $\frac{n}{2^{\Omega\left(\sqrt{\log(n)}\right)}}$.*

Furthermore, for every $\epsilon > 0$, there is a randomized algorithm that runs in time $\mathcal{O}(|E| \cdot n^{2+\epsilon})$ given $\mathcal{O}(\log(n))$ random bits which uses the same space (both ordinary and catalytic) as the deterministic algorithm.

The randomized algorithm only makes *one-sided errors*, where it always answers correctly when s is not connected to t in G. As such, the deterministic algorithm stated above is obtained by direct derandomization within the same space, i.e., by going over all possible random strings and accepting, if at least one of them makes the randomized algorithm accept. Our algorithms also improve on the $\tilde{O}(n)$ (or the $\tilde{O}(n^2)$) catalytic space bound observed by the randomized (or deterministic) algorithms from [43], achieving a sublinear bound in both cases.

Next, we extend this result to ED, LCS, and DFD, by designing catalytic algorithms that exploit the special graph structure underlying the problems. To the best of our knowledge, no sublinear space algorithm was known for DFD even without the catalytic restriction.

▶ **Theorem 2** (Sub-linear catalytic space algorithms for ED, LCS, and DFD)**.** *There exist deterministic algorithms for* ED*,* LCS*, and* DFD *which use time* $\mathsf{poly}(n)$*, space* $\mathcal{O}(\log(n))$*, and catalytic space* $\frac{n}{2^{\Omega\left(\sqrt{\log(n)}\right)}}$*.*

Furthermore, the running time of our algorithm for DFD *is* $\mathcal{O}(n^{4+\varepsilon})$*, for any* $\epsilon > 0$*, if all the distances of the points are integers from the range* $\{0, \ldots, \mathsf{poly}(n)\}$*.*

We do not know what is the exact running time of our algorithms for edit distance and longest common subsequence as it depends on the running time of a log-space procedure[1] for conversion between Chinese remainder representation and the usual binary representation of integers [54].

To the best of our knowledge, no sublinear space algorithm was known for DFD even without the catalytic restriction. However, we note that such a theorem can be derived via the reduction to STCONN and existing results: Chakraborty and Tewari [36] give a polynomial time algorithm for STCONN in directed layered planar graphs requiring $\mathcal{O}(n^{\varepsilon})$ space which we can directly plug in to obtain a deterministic algorithm for DFD that uses time $\mathsf{poly}(n)$ and space $\mathcal{O}(n^{\varepsilon})$.

This result is incomparable to our catalytic algorithm, as it uses less space overall but significantly more free memory. We also note that the results on grid-graph reachability of Allender et al. [9] are not sufficient for our purposes as in the process of reducing DFD to grid reachability, some instances can have both multiple sources and multiple sinks. Moreover, the grids we consider are NL-complete as opposed to the planar grids considered in [9] which are not known to be NL-complete.

Independently, Edenhofer [47] considered catalytic space-work space tradeoffs for STCONN using techniques similar to ours: for any $k \in [n]$, he gives an algorithm for solving STCONN using $\mathcal{O}(\log n \cdot \log k + \log n)$ space and $\mathcal{O}\left(\frac{n}{k} \cdot \log^2(n)\right)$ catalytic space.

1.4 Proof Overview

Our technique will primarily rely on the structure of two different STCONN algorithms: first, the sublinear-space polynomial-time algorithm of Barnes et al. [20] in the non-catalytic world; and second, the recent polynomial-time algorithm of Cook and Pyne [43] using linear catalytic space. We merge their approaches using analysis from other areas of catalytic computing, as well as incorporating new graph tools for overcoming one of the core barriers that arises in the approach of [20], to resolve the case of STCONN. Lastly, we use a common framework to attack ED, LCS, and DFD, working on an implicit grid graph whose structure will allow us to focus on small subgraphs and thus keep our space small.

1.4.1 Previous algorithms for STCONN: [20] & [43]

1.4.1.1 Long and Short Paths Framework ([20])

The starting point for Theorem 1 is the deterministic algorithm by [20]. For any parameter λ, [20] constructs a *representative set* of vertices $U \subset V$ of size roughly n/λ such that, if any pair u is connected to v in G by a path $\rho_{u,v} = (u, p_1, \ldots, p_L, v)$, then there exists a sequence of vertices $\tilde{\rho}_{u,v} = (u, w_1, \ldots, w_{L'}, v)$, such that each w_i is in U, and consecutive vertices in $\tilde{\rho}_{u,v}$ are connected by a path of length at most λ. This provides a sparsification $H = (U, E_U)$

[1] The running time for this procedure was not explicitly calculated and it boils down to the running time of a highly sophisticated log-space algorithm for multiplying n integers of $\mathcal{O}(\log n)$ bits each.

of G that still preserves connectivity within G: (s_i, s_j) is in E_U, if and only if there exists a path of length λ from s_i to s_j in G. Thus by setting λ to be $2^{\mathcal{O}\left(\sqrt{\log(n)}\right)}$, storing U in the workspace suffices to decide STCONN efficiently, if one can, in polynomial time and n/λ workspace,

- **construct** the representative set U;
- check if (s_i, s_j) forms an edge in the sparsified graph H, through a "**short path**" algorithm that checks for connectivity over paths of length λ in G; and
- using this short-paths algorithm as a sub-routine, find a "**long path**" in H, say $w_1, \ldots, w_{L'}$, such that the distances from s to w_1, and from $w_{L'}$ to t are at most λ in G.

To construct the set U, [20] employs a breadth-first search (BFS) from s to partition the vertices into λ candidate sets $U_1, \ldots, U_\lambda$, where intuitively each U_i contains the vertices at distance $i, i+\lambda, \ldots, i+\lfloor n/\lambda \rfloor \cdot \lambda$ from s in the BFS-tree. Each U_i is a representative set by design, and by a simple averaging argument one of these sets, which we choose to be our true representative set, has at most n/λ vertices. Thus if our "short paths" algorithm can test whether two nodes are at distance exactly λ, the vertices and edges of our sparsified graph H can be computed in polynomial time and $|U| = \mathcal{O}(n/\lambda)$ space using BFS; this gives us our "long paths" algorithm.

To check if vertices u and v are connected by a path of length λ in G, [20] interpolate between ordinary BFS and the recursive approach by Savitch [67]. As a first step to reduce the space, they arbitrarily partition set V into λ color classes, each of size n/λ, and look only for u–v paths which traverse over a fixed sequence of $\lambda + 1$ (possibly repeatable) color classes, where the first class contains u and the last class contains v. This is achieved by a basic BFS algorithm, iteratively computing connectivity between consecutive color classes in the sequence edge-by-edge and only storing the bit-vector of length n/λ of which nodes in the current color class are reachable from u.

By looping over all possible sequences of color classes this discovers every possible path of length λ; however, the runtime of this procedure is at least λ^λ, which is superpolynomial when λ is $2^{\sqrt{\log(n)}}$ as desired. Taking inspiration from Savitch's recursive algorithm, they instead choose a middle color class and test connectivity from the first color class, i.e. the one containing u, to this middle class and then from this middle class to the final class containing v. Such a test reduces connectivity of distance λ to 2λ tests of distance $\lambda/2$, and recursively this gives a runtime of $(2\lambda)^{\log \lambda}$ instead, giving us our final choice of $\lambda = 2^{\mathcal{O}(\sqrt{\log(n)})}$.

We aim to design a polynomial time algorithm for STCONN which uses only $\mathcal{O}(\log(n))$ workspace and $\mathcal{O}(n/\lambda)$ catalytic space. The first issue with adapting their approach is that storing U requires $\mathcal{O}(n/\lambda)$ clean workspace, which we cannot afford. Furthermore, even if one generates U in low-space, it is unclear how to construct catalytic algorithms for computing long and short paths that satisfy our time-space requirements. With regards to the second issue, we now turn to the catalytic algorithm of Cook and Pyne [43], using $\mathcal{O}(\log(n))$ workspace and $\tilde{\mathcal{O}}(n)$ catalytic space, which gives us an initial clue for our subroutines.

Connectivity via Propagating Sums ([43])

In a word, the Cook-Pyne algorithm implements a catalytic version of BFS. To illustrate their main ideas, consider a layered graph G with n layers of n vertices each, with edges only going between consecutive layers, and we want to test connectivity between s in the first layer and t in the last. We describe an algorithm using $\tilde{\mathcal{O}}(n^2)$ catalytic space, which is linear in the number of vertices of G (of course their algorithm works on arbitrary graphs, albeit with slightly less efficiency than in the layered case).

Let $R \coloneqq \mathbb{Z}/2^m\mathbb{Z}$ be the ring of integers modulo 2^m, for some m which we fix shortly. Their algorithm first allocates n^2 registers $r_{1,1} \dots r_{n,n}$ on the catalytic tape, with each $r_{i,j}$ being assigned to corresponding node $v_{i,j}$ in G, and let their initial contents be $\tau_{1,1} \dots \tau_{n,n} \in R$; this gives a total catalytic memory of $n^2 \cdot m$ bits. They start by "pushing" the value of $r_{1,j}$ through all its outgoing edges to the second layer, in particular by adding $\tau_{1,j}$ to each node $v_{2,j'}$ connected to $v_{1,j}$. They then push the values of the $r_{2,j}$ nodes forward to the third level, where now each $r_{2,j}$ has not just $\tau_{2,j}$ but also any value added to it in the previous round.

Repeating this procedure for all layers i in turn, we obtain some final value σ which arrives at the vertex (n, t) at the end of this process. They reset all catalytic memory by performing a "reverse edge push", i.e. repeating the whole procedure but with subtraction and in reverse, thus getting back $\tau_{i,j}$ in each register $r_{i,j}$. Finally, they repeat the entire algorithm but with the register $r_{1,s}$ for the source incremented by 1, collecting the value σ' at the vertex (n, t) and resetting as before.

For each pair of nodes $(v_{1,i}, v_{n,j})$ in the first and final layer respectively, let $\Gamma_{i,j}$ be the collection of paths between them. By linearity, the net result of our first forward push before resetting is

$$r_{n,j} \mathrel{+}= \sum_{i \in [n]} \sum_{P \in \Gamma_{i,j}} \sum_{v_{k,\ell} \in P} \tau_{k,\ell} \pmod{2^m}$$

because each value $\tau_{k,\ell}$ is added forward along the path P. In particular this holds for t, which gives us our σ value, while the same argument in the second forward push gives

$$\sigma' = \left(\sum_{i \neq s} \sum_{P \in \Gamma_{i,t}} \sum_{v_{k,\ell} \in P} \tau_{k,\ell} \right) + \left(\sum_{P \in \Gamma_{s,t}} \left(1 + \sum_{v_{k,\ell} \in P} \tau_{k,\ell} \right) \right) \pmod{2^m}$$

Hence $\sigma' - \sigma$ is exactly $|\Gamma_{s,t}| \pmod{2^m}$, i.e. the number of paths from s to t modulo 2^m.

When $2^m > \sigma' - \sigma$ this resolves the connectivity between s and t and even accurately counts the exact number of s–t paths. However, the number of paths can be as large as $\mathcal{O}(n^n)$, and setting each register size to be of size $m = \tilde{\mathcal{O}}(n)$ increases the catalytic space as well as the run-time by a factor of at least n. To deal with this, they finish the argument by using the fact that with large probability, a random prime p of value $\mathsf{poly}(n)$ satisfies $\sigma' - \sigma \not\equiv 0 \pmod{p}$ if the number of paths is non-zero. Hence, instead of computing over the ring R they compute over the field $\mathbb{F}_p$, thus reducing the catalytic space to $\mathcal{O}(n^2 \cdot \log n)$ as required.

Assuming one could stitch shorter segments together, we could use this algorithm for the long path procedure (putting aside the layered issue) on the graph H of representative nodes. However, using this for our short-paths sub-routine would still require the algorithm to push values across the edges associated with all the n vertices, necessitating catalytic space of $\tilde{\mathcal{O}}(n)$.

1.4.2 Our approach for STCONN

We now move on to our approach. The two conceptual contributions of our work are the following, which we discuss in turn: 1) we adapt the flow-pushing argument of [43] to the recursive framework of [20] to get the best of both worlds; and 2) we construct the representative set U for the long paths algorithm using tools from the theory of pseudo-randomness in a space and time efficient manner.

Merging Recursion and Flows

Assuming we have access to our representative set U, the long paths procedure can be handled by the [43] algorithm for unlayered graphs, and so we focus on finding paths of length λ in G. In combining the flow-based algorithm of [43] with the recursive approach of [20], we show both that the catalytic space of [43] can be reduced using techniques from [20] and that the space of [20] can be made catalytic using [43].

If we want to adopt the recursive structure of [20], then we cannot use the idea of adding 1 to the initial vertex and comparing the sums before and after; this would require too much free memory and would make it unclear how to "stitch together" the two sides of a recursive call. Instead, we take inspiration from previous catalytic algorithms, such as that of Ben-Or and Cleve [22], which handle arithmetic cancellations in recursion. Our goal will be to count the number of paths between any pair of vertices u and v.

For $\ell = 1 \dots \log(\lambda)$, let $\Gamma^{\ell}_{u,v}$ be the set of u–v paths of distance exactly 2^{ℓ}. [22] give the following recursive invariant:

$$r_v \mathrel{+}= \tau_u \cdot |\Gamma^{\ell}_{u,v}|$$

Following [20], we may consider all possible midpoints for a u–v path of length 2^{ℓ}. Where before we simply took $\vee_w (u \to_{2^{\ell-1}} w) \wedge (w \to_{2^{\ell-1}} v)$, the crucial insight is that moving to arithmetic $\vee$ and $\wedge$, i.e. $+$ and $\times$, faithfully counts the number of paths in the same way, as any u–w path can be paired with any w–v path to give a u–v path:

$$|\Gamma^{\ell}_{u,v}| = \sum_w |\Gamma^{\ell-1}_{u,w}| \cdot |\Gamma^{\ell-1}_{w,v}|$$

Thus utilizing the [22] procedure for recursive multiplication, fixing a midpoint w, allows us to evaluate this inner product using four recursive calls:

1. $r_v \mathrel{-}= r_w \cdot |\Gamma^{\ell-1}_{w,v}|$ // $r_v = \tau_v - \tau_w \cdot |\Gamma^{\ell-1}_{w,v}|$
2. $r_w \mathrel{+}= r_u \cdot |\Gamma^{\ell-1}_{u,w}|$ // $r_w = \tau_w + \tau_u \cdot |\Gamma^{\ell-1}_{u,w}|$
3. $r_v \mathrel{+}= r_w \cdot |\Gamma^{\ell-1}_{w,v}|$ // $r_v = \tau_v + \tau_u \cdot |\Gamma^{\ell-1}_{u,w}| \cdot |\Gamma^{\ell-1}_{w,v}|$
4. $r_w \mathrel{-}= r_u \cdot |\Gamma^{\ell-1}_{u,w}|$ // $r_w = \tau_w$ (and $r_u = \tau_u$)

Repeating this for all potential midpoints w gives us the outer sum, and hence the invariant above. Repeating for every u–v pair, by linearity we arrive at a final value of

$$r_v \mathrel{+}= \sum_u \tau_u \cdot |\Gamma^{\ell}_{u,v}| \quad \forall v \in V$$

which we take as the recursive invariant for our algorithm. This fits the base case of $\ell = 0$, since we add r_u to r_v for each u which has an edge to v, while the analysis of [22] shows that it holds for ℓ given access to $\ell - 1$. Crucially for the catalytic restriction, at the end of each subroutine and hence the entire algorithm, r_u and r_w are left in their original configuration.

We now turn to the time of the recursion. The above procedure, looping over all w independently, is the structure of Savitch's algorithm [67], and so following [20] we generalize our recursive calls to work over choosing middle color classes rather than individual vertices. Once again we choose to have λ color classes of size n/λ each, and in making four recursive calls per middle color class (rather than the two from [20]), our runtime becomes $(4\lambda)^{\log(\lambda)}$, which does not change our asymptotics for $\lambda = 2^{\sqrt{\log(n)}}$. Our vectors of length n/λ are entirely catalytic, with our free space being only the $\mathcal{O}(\log n)$ necessary to compute sums and keep track of our runtime.

As a final note, to make the analysis of our recursive condition simpler, we use the *same* recursive algorithm for the long paths as well, with some minor interfacing between the two to reflect the change in graphs. In the outer algorithm we run for paths of length up to $|U|$,

i.e. maximal length paths, but in exchange we remove the color class structure and operate on the entire graph at every step. This gives a runtime of $4^{\log |U|} = \mathcal{O}(n^2)$, which is linearly less efficient than [20] and [43].

Efficiently Constructible Representative Sets

We return to our final loose end for STCONN, which is the construction of the representative set for our long paths algorithm using low space. To do this, we use *pairwise-independent hash function families* $\mathcal{H}_{m,n}$ to construct a family of *multisets* $\{U_\sigma\}$, indexed by $\sigma \in \{0,1\}^{(6+\varepsilon)\log(n)}$, and prove that with high probability over the choice of σ, a multi-set from this family is a representative set for G. For convenience of exposition, we assume $\{U_\sigma\}$ as a family of sets, and deal with the technical challenges that arise in the STCONN algorithm from U_σ being a multi-set in Section 4.

Recall (from [69]) that for $m \leq n$, $\mathcal{H}_{m,n}$ is a pair-wise independent family of hash functions $h : [m] \to [n]$ indexed by $2\log(n)$ bits, such that an h chosen at random from $\mathcal{H}_{m,n}$ maps any pair $i \neq j \in [m]$ into $[n]$ independently. Moreover, each hash function can be evaluated on an input $i \in [m]$, in $\mathsf{poly}(\log(n))$ time and $\mathcal{O}(\log(n))$ space (see Lemma 10 for the formal statement).

For the construction, we set $m = \mathcal{O}(n/\lambda)$, and suppose that we sample $K = \mathcal{O}(\log(n))$ many hash functions $h_1, \ldots, h_K$ from $\mathcal{H}_{m,n}$, and consider U_σ as the union of their images, where σ is the random seed used to sample them. Using a probabilistic argument, we show that with high probability over the choice of the $h_1, \ldots, h_K$ (or σ) the following holds: for every pair of vertices u, v from G that are connected by a path of length $\lambda/2$, there exists a walk between them traversing $\lambda/2 - 1$ distinct vertices that visits U_σ.

With high probability over σ, U_σ forms a representative set for G, over paths of length λ. Indeed, for any u, v that is connected, suppose we break this path into L' sub-paths of size $\lambda/2$. By the properties of a "good" U_σ, we see that each sub-path can be replaced by a walk traversing $\lambda/2-1$ distinct vertices and visiting U_σ. The distance between consecutive vertices from U_σ along this new path from u to v is at most λ, so U_σ can be used to sparsify G.

However, the length of σ is $2K\log(n) = \mathcal{O}(\log^2(n))$, and we cannot store this within our $\mathcal{O}(\log(n))$-sized workspace. As such, we use the well-known technique of reducing randomness by performing a random walk on a *constant-degree explicit expander* D, with suitably high expansion [7]. In particular, sampling K hash functions using a random walk on D specified over vertex set $\{0,1\}^{2\log(n)}$ (that describes all of $\mathcal{H}_{m,n}$), gives similar guarantees of intersecting with walks, as that of picking K hash functions uniformly at random. The number of random bits is now at most $\mathcal{O}(\log(n))$, and by carefully accounting the values of K, the degree and expansion of D, a seed length of $(6+\varepsilon)\log(n)$ suffices to describe U_σ.

Finally, any vertex in U_σ can be generated, given σ, in $\mathsf{poly}(\log(n))$ time and $\mathcal{O}(\log(n))$ space. This holds from standard explicit constructions of constant-degree expanders, and efficient evaluation of hash functions in $\mathcal{H}_{m,n}$.

1.4.3 ED, LCS, & DFD Without Direct Reductions To STCONN

We now turn our attention to computing ED, LCS, & DFD. A natural direction would be to use the reduction to reachability of grid-like graphs and use the previous algorithm. However, on inputs of length n, the underlying grid-like graph has quadratic size, and thus the space required would be $\frac{n^2}{2^{\Omega(\sqrt{\log n})}}$ – far larger than the space we would like to use.

LCS and ED in Sublinear Free Space ([58])

While [20] have the outer algorithm with a single color class (corresponding to the representative set) and recursion depth $\mathcal{O}(\log n)$ and the inner algorithm with λ color classes and recursion depth $\mathcal{O}(\log(\lambda)) = \mathcal{O}(\sqrt{\log n})$, [58] approach the problem by effectively using the ideas similar to the inner algorithm of [20] as their outer algorithm. In the outer algorithm, given a color class in the first layer[2] and a color class in the last layer, [58] choose a middle layer of the grid-like graph and then iterate over the λ color classes of size n/λ as the possible choices of a middle vertex of the path, doing this recursively until a base case is reached, when the two layers are at distance n/λ. In this base case, the inner algorithm takes over. Because the grid-like graph only has edges to close neighbors, either the two color classes are far enough so that there is no possible path between the two, or the graph containing the two color classes and all paths between them contains $\mathcal{O}(n/\lambda)$ rows and $\mathcal{O}(n/\lambda)$ columns, and therefore we can compute the shortest path from the first color class to the second color class by the standard dynamic programming algorithm that only requires us to remember two rows at a time.

It is not immediate that one can emulate the algorithm of [58] using catalytic tools. In particular, their algorithm is designed to find the shortest path which is necessary for computing edit distance. Thus the algorithm always has to choose the shortest path that leads to a given middle point. However, the "push" algorithm of Cook and Pyne sums all the paths leading to the middle point. It is not clear how to select the minimum path among them. More to the point, the values we are summing in the algorithm of Cook and Pyne have nothing to do with the true values we want to take minimum of as they are offset by some unknown initial values from the catalytic memory. Taking their minimum would be completely useless.

The standard approach to overcome this would be to layer the graph and count the length of the paths implicitly. Checking whether the length of a path from a source vertex to the target vertex is of length at most d is done by checking whether the source vertex in the first layer is connected to the target vertex in the d-th layer. However, the layering increases the size of the graph by a factor of $\mathcal{O}(n)$ so any possible savings in space from clever algorithms will be wiped-out.

To overcome this, we stick to the original graph but introduce multiplicative weights to the edges. It turns out the the algebraic framework of the Cook-Pyne algorithm can be modified to count a *weighted sum* of the paths. The weight of a path will be the product of its edge weights. By choosing weights that are large enough, different path weights will be separated so that from their weighted sum we can read off the count of the paths of different weights. This assumes we will work with large integers that we will not have space to store; the path weights will have $\mathcal{O}(n^2)$ bits. We overcome this by computing the final value modulo different primes, that is we compute it in Chinese remainder representation. Using a space-efficient algorithm that converts Chinese remainder representation into the usual binary representation, we get bit-by-bit access to the actual weighted sum. From it we can read off the length of the shortest path as well as the length of the longest path.

So our goal will be to design an algorithm for weighted grid graphs that computes the weighted sum of paths between two vertices modulo a small prime.

[2] The grid-like graph is not layered in the usual sense, which is not an issue in this algorithm as it only uses free space.

Weighted Reachability on Grid Graphs

We use a similar high-level technique to finding the shortest path in grid-like graphs as the algorithm [58]. Instead of finding the shortest path we will compute the weighted sum of the paths.

Again, our algorithm for weighted reachability on grid graphs will be composed of an outer and an inner algorithm. For practical purposes, we assume that the edge weights are given to us by an oracle, and we use this oracle as a parameter for our three functions of interest which we want to compute. Moreover, for this algorithm we assume all values to be over a field $\mathbb{F} = \mathbb{F}_p$ for some prime $p = \mathsf{poly}(n)$.

The outer algorithm will be essentially identical to the inner "short paths" algorithm from our algorithm for general STCONN. It will use recursion of depth $\mathcal{O}(\sqrt{\log n})$ with color classes of size n/λ, for $\lambda = 2^{\Theta(\sqrt{\log n})}$, to reduce the weighted reachability on a grid of size $n \times n$ to the weighted reachability on grid-graphs of size roughly $n/\lambda \times n/\lambda$. This smaller reachability will be solved by the inner algorithm.

However, we do not have the luxury of using our previous algorithm as our inner algorithm, given that our goal is to use only $\mathcal{O}(\log n)$ free space. Therefore, we use a similar idea to the outer algorithm of [20] together with the ideas of Cook and Pyne. We are interested in computing weighted reachability for two color classes of size n/λ at horizontal distance n/λ. Two color classes at such horizontal distance in a grid-graph can affect each other only if they are adjacent vertically. Thus, we get a subgrid with n/λ layers and at most three color classes, yielding $3 \cdot n/\lambda$ vertices per layer. We now use the idea of propagating sums for the outer algorithm in our STCONN algorithm while recursing again on the middle layers. We will now have only a single color class, which is not an issue with $\mathcal{O}(n/\lambda)$ vertices per layer. Our final base case here is the case of two neighboring layers, where we can iterate over all $\mathcal{O}(n/\lambda)$ edges and multiply intermediate values by their weight.

Here, we only need a constant number of bits of free space per level of recursion for the instruction counter, and therefore the total recursion depth $\mathcal{O}(\log n)$ does not pose any large issues space-wise. Moreover, we can implement the recursion in a constant number of steps per each recursion level, thus we get time complexity $c^{\mathcal{O}(\log n)} \cdot \mathcal{O}(n/\lambda) = \mathsf{poly}(n)$. Regarding catalytic space usage, if we use a new catalytic register vector at every recursion level, we require $\mathcal{O}(\log n)$ vectors, each using n/λ field elements, which each requires $\mathcal{O}(\log p)$ bits, which in total yields the required catalytic space complexity $\frac{n}{2^{\Omega(\sqrt{\log n})}}$.

To finally compute the weight of all u, v-paths modulo p, we first run the algorithm forward, recording the result r_1, then rerun the computation backwards. We than add 1 to the register corresponding to u, and run the computation again, recording the result r_2, and rerun the computation backwards for the final time. We then subtract 1 from the register corresponding to u and return $r_2 - r_1$ as the result.

Using Grid Reachability to Compute Discrete Fréchet Distance

For discrete Fréchet distance of point sequences $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$, we use the previous algorithm as a blackbox, and we provide a suitable edge weight oracle. We also assume that the distances of the points are integers in the range $\{0, \ldots, \mathsf{poly}(n)\}$. If we are given a more complicated metric, we can build a range query oracle out of it – that is, we can compute how many different point pairs have distance lower than the queried point pair (p, q), by iterating over all the other pairs, and thus we would get an oracle whose values are in the range $\{1, \ldots, n^2\}$.

One possible way of computing discrete Fréchet distance is by taking a grid-like graph on vertex set $\{0,\dots,n\}\times\{0,\dots,n\}$, where each vertex (i,j) has three outgoing edges into vertices $(i+1,j),(i,j+1),(i+1,j+1)$ if these vertices exist. We then build an induced subgraph which we call $\mathsf{Frech}_{P,Q,m}$, where we set the weight of each edge $((i,j),(k,\ell))$ to one if the distance between p_k and q_ℓ is at most m and zero otherwise. As there are only $\mathsf{poly}(n)$ possible values for m, we can try all possible values of m, and the smallest value of m such that there is a path between $(0,0)$ and (n,n) is the Fréchet distance. However, there may be exponentially many paths and we only allow computation modulo primes of polynomial size. We remedy this using the Chinese remainder theorem and the fact that the total weight is zero iff the total weight module the first $4n$ primes is zero. Thus, we run the algorithm for $4n$ primes with the same weights (as they are all either zero or one) to determine whether the discrete Fréchet distance is at most m, and we run binary search over all choices of m to compute DFD in polynomial time, logarithmic free space and catalytic space $\frac{n}{2^{\Omega(\sqrt{\log n})}}$.

Using Grid Reachability to Compute ED and LCS

For ED and LCS for n-character strings, we want to find the shortest or, respectively, longest path in a weighted grid-like graph – essentially the graph for DFD. However, unlike DFD, we need to take care of different weights of edges. Furthermore, in our approach, we multiply weights instead of adding them, and hence we must move to addition in the exponent.

For the grid-like graph, we note that there at most 3^{2n} paths (the longest path has length $2n$, and each vertex has out-degree at most three). In all cases, the edges in the grid-like graph have two possible weights: 0 or 1, but some edges might also be missing. Therefore, we can simulate addition by having the weights as follows: if the edge is missing, its weight will be set to 0. (And therefore, any path using such edge will have zero total weight, which effectively removes the edge.) For the existing edges, if an edge had weight $\mu(e)$ originally, then our new weight will be $16^{n\cdot\mu(e)}$. Therefore, weight-zero edges will have their new multiplicative weight set to 1, and weight-one edges will have their multiplicative weight set to 16^n. This also ensures that we can efficiently learn the longest/shortest path in the graph: given the result, which is a sum of weights of all s,t-paths, the lowest bit set to 1 corresponds to the length of the shortest path: indexing bits such that the least significant bit has index 0, any of the bits $4n(w+1)-1,\dots,4nw$ being set to one corresponds to there existing a path of weight w in the original additive weighting scheme. Therefore, we can just check all the bits of the number to find the largest and the smallest non-zero bit index.

There is one slight issue: if we were computing this over integers, the representation of the value r would have $\Theta(n^2)$ bits, which we cannot afford to represent directly. However, we can use the Hesse-Allender-Barrington algorithm [54] for reconstructing integers from their Chinese remainder representation that uses polynomial time, polynomial number of queries for the value of the r mod p_i for some primes p_i and logarithmic space to output a particular bit of r in the binary representation. Since our algorithm returns all results modulo p, each query to r mod p_i corresponds to us running our reachability algorithm over the field $\mathbb{F}_{p_i}$. Thus, we get again polynomial time, logarithmic free space and catalytic space $\frac{n}{2^{\Omega(\sqrt{\log n})}}$.

We note that this approach also works for the weighted versions of edit distance if the weights are $\mathcal{O}(\log n)$-bit integers with appropriate changes to the weighting scheme.

1.5 Future Work

We consider these algorithms to be an initial step towards bypassing the current $\frac{n}{2^{\Theta\left(\sqrt{\log(n)}\right)}}$ space barrier for these problems. The NNJAG lower bounds which substantiate this barrier are based on pebbling techniques, which was also the case for Tree Evaluation for over

a decade before finally being broken by the catalytic algorithm of [42]. Their algorithm bypasses these pebbling lower bounds by combining multiple pebbles in the same location, a process which has been called "fuzzy pebbling" by later reviews on the topic. This is also the situation for our STCONN algorithm, where the connectivity of many nodes in the graph are all stored on top of one another; thus there is hope that further development of catalytic techniques or more clever instantiations will allow us to go beyond this longstanding space-time barrier.

2 Preliminaries

2.1 Catalytic Model

Our machine model in this paper is an extension of ordinary space-bounded computation:

▶ **Definition 3** (Catalytic Turing machine). *A* catalytic Turing machine *is a Turing machine with four tapes: 1) a read-only* input *tape; 2) a write-only* output *tape; 3) a read-write* work *tape; and 4) a read-write* catalytic *tape.*

We say that a function $f : \{0,1\}^n \to \{0,1\}^m$ can be solved in $\mathsf{CSPACE}[s, c]$ *if there exists a catalytic Turing machine M whose input, output, work, and catalytic tapes have length n, m, s, and c respectively, such that for any $x \in \{0,1\}^n$ and $\tau \in \{0,1\}^c$, if M is initialized with x on the input tape and τ on the catalytic tape, then after execution M halts with $f(x)$ on the output tape and τ on the catalytic tape.*

We further say that f is in $\mathsf{CTIMESPACE}[t, s, c]$ *if M runs in time t for every x and τ.*

Since time is a consideration in our algorithm, we specify that in this paper we will use *Random Access Turing machines*, which allow us to jump with their heads to a particular tape cell, specified on an auxiliary *address* tape consisting of $\lceil \log(c + s) \rceil$ bits, as an atomic operation. This does not provide the machines with extra computational power, but only allows for faster simulation of RAM machines.

While our ultimate goal in catalytic computation is to compute a function into ordinary memory while leaving catalytic memory untouched, along the way it is necessary to alter the catalytic tape in predictable (and reversible) ways:

▶ **Definition 4** (Catalytic subroutine). *A* catalytic subroutine *is a program P whose net result is to 1) transform a section of the catalytic tape τ via addition, i.e. $\tau \mathrel{+}= v$ for some value v; 2) manipulate the free memory in any way; and 3) leave all other catalytic memory in its original configuration. We also require that every catalytic subroutine P has an inverse program P^{-1} whose result is to subtract v from the same region instead.*

Our algorithms will need to operate with catalytic memory containing values from a particular field $\mathbb{F}_p$. The details of this are left to the full version due to space constraints.

2.2 Function Statements

We first define all functions of interest in this paper:

▶ **Definition 5** (Connectivity). *For a directed graph $G = (V, E)$ and two vertices $s, t \in V$, we say s and t are* connected *if there is a path from s to t in G.*

▶ **Definition 6** (Longest common subsequence). *For a string $x \in \Sigma^*$, a* subsequence *is a string z that can be obtained by removing any number of characters of x.*

For two strings $x, y \in \Sigma^*$, the longest common subsequence *is the longest $z \in \Sigma^*$ that is a subsequence of both x and y, and we define the corresponding function* $\mathsf{LCS}(x, y) := |z|$.

▶ **Definition 7** (Edit distance). *Given a string $x \in \Sigma^*$, an* edit operation *is either a deletion, insertion or a substitution of a character in x.*

For two strings $x, y \in \Sigma^*$, their edit distance $\mathsf{ED}(x, y)$ *is the minimum number of edit operations needed to transform x into y.*

▶ **Definition 8** (Discrete Fréchet distance). *For two sequences $P = (p_1, \ldots, p_m), Q = (q_1, \ldots, q_n)$ of points from a metric space (M, d), their* monotone alignment *is a sequence of pairs $(i_k, j_k) \in [m] \times [n]$ of some length ℓ such that $(i_1, j_1) = (1, 1)$, $(i_\ell, j_\ell) = (m, n)$ and for all $2 \leq k \leq \ell$, $i_k \in \{i_{k-1}, i_{k-1} + 1\}$ and $j_k \in \{j_{k-1}, j_{k-1} + 1\}$.*

The discrete Fréchet distance *of P and Q is*

$$\mathsf{DFD}(P, Q) = \min_{A} \max_{(i,j) \in A} d(p_i, q_j),$$

where A ranges over all possible monotone alignments.

Informally, the discrete Fréchet distance describes the situation, where there are two frogs connected by a rope at points p_1, q_1, and in each step, each frog may jump forward to the next point in the sequence. The distance itself is then the minimum length of the rope such that the frogs can jump in a way that the rope does not snap.

In our algorithm, we assume that the metric d is represented by integers from the range $\{0, \ldots, \mathsf{poly}(n)\}$. This assumption is without loss of generality: there are $m \cdot n$ pairs in $P \times Q$, and it follows from the definition that the distance of one of the pairs is the discrete Fréchet distance, and our algorithm will only use the queries of the form (p_i, q_j) for some $p_i \in P, q_j \in Q$. Thus, we can represent the order of distances in the metric by simply ranking the distances between the pairs of points. These results are integers from the set $\{1, \ldots, m \cdot n\}$, which is sufficient. Provided the distances from the original metric can be compared using $\mathcal{O}(\log n)$ bits of space, we can translate from the metric into the ranks in logarithmic space and roughly quadratic time per query.

2.3 Pairwise Independent Hash Functions

Our algorithms for STCONN will require efficiently constructible *hash functions* for sparsifying our graph G:

▶ **Definition 9** (Pairwise Independent Hash Functions). *A family of functions $\mathcal{H}_{m,n} = \{h : [m] \to [n]\}$ is pairwise independent if for every $x_1 \neq x_2 \in [m]$, and every $y_1, y_2 \in [n]$, when a function h is chosen uniformly at random from $\mathcal{H}_{m,n}$,*

$$\Pr_{h \sim \mathcal{H}_{m,n}} [h(x_1) = y_1 \wedge h(x_2) = y_2] = \frac{1}{n^2}.$$

In addition, for any $h \in \mathcal{H}_{m,n}$, let $\mathsf{Im}(h)$ be the image of h.

It follows from Definition 9 that for every $x, y \in [m]$, $\Pr_{h \sim \mathcal{H}_{m,n}} [h(x) = y] = 1/n$.

Important for our result is that there exist pairwise independent hash functions that have a short description and can be constructed efficiently:

▶ **Lemma 10** ([69]). *For every $m, n \in \mathbb{N}$, there exists a pairwise independent hash function family $\mathcal{H}_{m,n}$, where a random function can be sampled from the family $\mathcal{H}_{m,n}$ using $r = \max\{\log(m), \log(n)\} + \log(n)$ bits.*

Moreover, given any $h \in \mathcal{H}_{m,n}$ (described by a string in $\{0,1\}^r$) and $x \in [m]$, $h(x)$ can be computed in time $\mathsf{poly}(\log(m, n))$ and space $\mathcal{O}(r)$.

2.4 Modular Arithmetic

In addition to hash functions, we will also require working over the integers modulo a prime p of logarithmic bit-size in an efficient manner:

▶ **Proposition 11.** *Given a prime p, we can compute addition, subtraction, and multiplication modulo p, in time* $\mathsf{poly}(\log(p))$ *and space* $\mathcal{O}(\log(p))$.

Working with modulo primes will be a useful proxy for testing whether or not a number is non-zero:

▶ **Proposition 12.** *Let $0 < m \leq 2^t$, and let $t \log t \leq p \leq 2t \log t$ be a randomly chosen prime. Then $m \not\equiv 0 \bmod p$ with probability 0.99 over p.*

Beyond testing for zero, we will sometimes need to represent numbers exactly, for which we use the *Chinese remainder representation* of a number x over many primes. There is a log-space procedure that can calculate bits of x in binary representation using *oracle* access to the Chinese remainder representation of x [54]:

▶ **Proposition 13** ([54])**.** *Let p_i denote the i-th prime. There is a procedure that works in space $\mathcal{O}(\log n)$ with query access to $x \bmod p_i$, for $i = 1, \ldots, n$, where $0 \leq x < 2^n$ is an arbitrary integer, that on input (n, j) represented in binary, where $n > j \geq 0$ are integers, outputs j-th bit of the binary representation of x. The procedure runs in polynomial time and makes in total $\mathcal{O}(n^2)$ queries to $x \bmod p_i$.*

We remark that the procedure would work as long as $x < \prod_{i=1}^{n} p_i$ which would provide $\mathcal{O}(\log n)$-factor savings in time and the number of queries. This is however irrelevant for us. Moreover the procedure can output all the bits of the binary representation of x in the usual order using the same space, time and number of queries. This could provide time savings of $\mathsf{poly}(n)$-factor for our edit distance and LCS algorithms.

3 Pushing Flow Through Walks

In this section we will utilize the short paths (or in our case, walks) decomposition from [20] as a generic setup for the flow-based algorithm of [43]. The latter assigned initial values to every node in the graph from the catalytic tape, and then "pushed" these values forward through the graph along every path via addition, using cancellation to collect up only terms from relevant (i.e., s–t) paths. To improve their catalytic space usage, we need to follow the color class framework of [20] and use recursion restricted to subsections of the graph.

▶ **Definition 14** (Walk matrix)**.** *Let $H = (V, E)$ be a graph on $m \in \mathbb{N}$ vertices, and for convenience assume m is a power of 2. Fix a field $\mathbb{F} = \mathbb{F}_p$ for some prime $p \leq \mathsf{poly}(m)$, and let $\mu : E \to \mathbb{F}$ be a weighting function on the edges of H. Given a walk W in H, we say the weight of W is the product of all edge weights in W with multiplicity, i.e. $\mu(W) = \prod_{e \in W} \mu(e)^{d_e}$ where d_e is the number of times e occurs in W.*

For any $\ell \in [\log m]$, and any $v_{in}, v_{out} \in V$, define $\Gamma_\ell(v_{in}, v_{out})$ to be the collection of all walks of length exactly 2^ℓ from v_{in} to v_{out} in H. We define the walk matrix *$\mathcal{W}_\ell \in \mathbb{F}^{V \times V}$ by*

$$\mathcal{W}_\ell[v_{in}, v_{out}] = \sum_{W \in \Gamma_\ell(v_{in}, v_{out})} \mu(W)$$

While we cannot store the walk matrix, we simplify our task in two ways: 1) we focus on a collection of input and output nodes, V_{in} and V_{out} respectively, of sufficiently small size to be stored; and 2) while we still cannot store the matrix $\mathcal{W}_\ell$ restricted to $V_{in} \times V_{out}$, we instead provide an input vector $\overrightarrow{\tau_{in}} \in \mathbb{F}^{V_{in}}$ and compute the output of $\overrightarrow{\tau_{in}} \cdot \mathcal{W}_\ell$ in the entries V_{out}.

▶ Remark 15. For convenience, given sets $V_{in}, V_{out} \subseteq V$ and a vector $\overrightarrow{\tau_{in}} \in \mathbb{F}^{V_{in}}$, we write $\overrightarrow{\tau_{in}} \cdot \mathcal{W}_\ell$ to mean the vector resulting from the product of matrix $\mathcal{W}_\ell$ with the (row) vector indexed by V which is $\overrightarrow{\tau_{in}}$ for all $v_{in} \in V_{in}$ and 0 everywhere else. If we add this to a vector $\overrightarrow{\tau_{out}} \in \mathbb{F}^{V_{out}}$, we mean that we restrict this vector to the entries indexed by $v_{out} \in V_{out}$ and do addition entry-wise by corresponding v_{out}.[3] Thus we have

$$(\overrightarrow{\tau_{in}} \cdot \mathcal{W}_\ell)[v_{out}] = \sum_{v_{in} \in V_{in}} \overrightarrow{\tau_{in}}[v_{in}] \cdot \sum_{W \in \Gamma_\ell(v_{in}, v_{out})} \mu(W)$$

Our V_{in} and V_{out} sets will be chosen via breaking V into easily computable blocks of equal size, often referred to as color classes.

▶ **Definition 16.** *Let H be a graph on $m \in \mathbb{N}$ vertices, and let $C \in [m]$ such that C and m are powers of 2. We partition V into sets $\{V_c\}_{c \in [C]}$, which we call* color classes*, such that $|V_c| = m/C$ for all c, by letting the first $\log C$ bits of any vertex v's label in $[m]$ indicate the color class of v.*

While the key property for our space and time bounds is the even distribution of nodes into classes, note that computing the color class and label within the color class of v can be done in time $\log C$ and $\log m/C$ respectively.

We now reach our main propagation subroutine which will drive all algorithms in this paper. Unlike [20] itself, we will require calculating each side of the recursion multiple times in order to get the cancellations necessary for [43]; however, by setting the parameters correctly we maintain an equivalent asymptotic runtime.

▶ **Lemma 17.** *Let $m \in \mathbb{N}$, let $\mathbb{F} = \mathbb{F}_p$ for some prime $p = \mathsf{poly}(m)$, and let $H = (V, E)$ be a graph on m vertices with edge weight function μ. Let $\mathcal{W}$ be defined with respect to H as per Definition 14 and let C and $\{V_i\}_{i \in [C]}$ be as per Definition 16.*

Assume that we have a catalytic subroutine P_0 which takes in $c_{in}, c_{out} \in [C]$ as well as sections $\overrightarrow{r_{in}}$, $\overrightarrow{r_{out}}$, $\overrightarrow{r_{mid}} \in \mathbb{F}^{m/C}$ from the catalytic tape, and adds $\overrightarrow{r_{in}} \cdot \mathcal{W}_0$ to $\overrightarrow{r_{out}}$.

Then there exists a catalytic subroutine P which takes in $\ell \in [\log m]$, $c_{in}, c_{out} \in [C]$, and sections $\overrightarrow{r_{in}}, \overrightarrow{r_{out}}, \overrightarrow{r_{mid}} \in \mathbb{F}^{m/C}$ from the catalytic tape, and whose effect is to add $\overrightarrow{r_{in}} \cdot \mathcal{W}_\ell$ to $\overrightarrow{r_{out}}$.

If P_0 runs in time T and uses free space S, then P runs in time $(4C)^\ell \cdot T$ and uses free space $\ell \log 4C + S$; it uses only the catalytic memory $\overrightarrow{r_{in}}, \overrightarrow{r_{out}}, \overrightarrow{r_{mid}}$ and the memory required to run P_0.

4 STCONN via Graph Decomposition

In this section we prove Theorem 1, i.e. deciding connectivity in an n-vertex graph. To do this, we start by discussing important parameterizations of Lemma 17 and what they say about the types of graphs and walks we can capture with them.

[3] Note that this definition is *not* the same as taking the product with submatrix $(\mathcal{W}_\ell)_{V_{in}, V_{out}}$; we will allow our walks to go via any vertices as long as they start in V_{in} and end in V_{out}.

In order to run in polynomial time we need to choose C and ℓ such that $(4C)^\ell \leq \mathsf{poly}(n)$, while for catalytic space $\frac{n}{2^{\Omega(\sqrt{\log n})}}$ we need to store $\overrightarrow{r_{in}}, \overrightarrow{r_{out}}, \overrightarrow{r_{mid}}$ using $(m/C) \cdot \mathcal{O}(\log n)$ bits, where m is the number of vertices of the graph in question. There are two extreme cases of interest:

- when $C = 1$, we can tolerate any 2^ℓ-length walk for $\ell \leq \log m$, which in particular can cover every vertex in the graph in question; with respect to this graph, however, we only get small catalytic space if $m \leq \frac{n}{2^{\Omega(\sqrt{\log n})}}$, i.e. working on graphs with many fewer vertices than n.
- when $\ell \leq \sqrt{\log n}$ and $C \leq 2^{\mathcal{O}(\sqrt{\log n})}$, we can only handle short walks, but in exchange we use sufficiently little catalytic storage even when working on large graphs, such as G or even larger graphs with $\mathcal{O}(n)$ vertices.

Thus we will have two phases of the algorithm as in [20]: 1) a *long walks* algorithm which works on a vertex-sparsified version of G whose edges represent walks of length λ in G; and 2) a *short walks* algorithm which runs on all of G but only for walks of length λ. For this we will use both of the extreme settings of Lemma 17 as discussed above. Due to space limitations, we postpone the proofs to the full version.

4.1 Long Walks via Graph Sparsification

In order to handle long walks in a graph, we need to reduce the number of vertices significantly while still not losing connectivity information about G. We do this through a randomized construction of a sufficiently large subset of vertices U, such that if any two vertices are connected in G, they are connected by taking short walks between the nodes in U. The following theorem may be of independent interest:

▶ **Theorem 18.** *There exists an algorithm $\mathcal{U}$ such that for every directed graph $G = (V, E)$ on n vertices, value $\lambda = \omega(\log(n))$, seed $\sigma \in \{0,1\}^{(6+\epsilon)\log(n)}$, and constant $\varepsilon > 0$, it satisfies the following properties:*

- *For every σ, there exists a multiset $U_\sigma \subset V$ of size $\mathcal{O}\left(\frac{n\log(n)}{\lambda}\right)$, such that on input $i \in [|U_\sigma|]$ in binary, $\mathcal{U}(1^n, \lambda, \sigma, \varepsilon, i)$ outputs the i^{th}-vertex of U_σ in time $\mathsf{poly}(\log(n))$ and space $\mathcal{O}(\log(n))$.*
- *With probability at least 0.99 over a uniformly random choice of σ, for every $u, v \in V$, if u is connected to v, then there exists an ℓ and a sequence of vertices $\tilde{\rho}_{u,v} = (u, s_1, s_2, \ldots, s_\ell, v)$, such that 1) $s_i \in U_\sigma$ for every $i \in [\ell]$, and 2) for each consecutive pair in $\tilde{\rho}_{u,v}$, there exists a path in G of length at most λ connecting them.*

In order to utilize any multiset $U \coloneqq U_\sigma$ given by Theorem 18, we will need to have access to connectivity at distance at most $2^{\Theta(\sqrt{\log n})}$ in G. Note that Lemma 17 counts walks of length *exactly* 2^ℓ for the given ℓ, but this can be easily rectified by adding self-loops, at the cost of obtaining a value which is not the exact count of short walks in the original graph.

▶ **Definition 19.** *Given graph $G = (V, E)$ on $n \in \mathbb{N}$ vertices, let G' be the graph obtained by adding self-loops edges to all vertices, i.e. $V' = V$ and $E' = E \cup \{(v, v)\}_{v \in V'}$.*

We construct $H = (V_U, E_U)$ to be the graph whose vertex set V_U is U with multiplicity plus the vertices s and t (for convenience we assume $|V_U|$ is a power of two, and otherwise we add dummy vertices to make it such) and whose edge set E_U is all $e = (v_{in}, v_{out}) \in V_U^2$ such that there exists a walk of length λ from v_{in} to v_{out} in G'. We define the weight function $\mu(e)$ for all $e \in E_U$ to be the number of walks of length λ from v_{in} to v_{out} in G'.[4]

[4] We alert the reader to the fact that for any $\mu(u, v)$ is *not* the number of u–v paths in G of length at most λ; it is at least as large as this value and can be much greater. This will not cause an issue to our analysis, since all we need is a non-zero lower bound for connected vertices as well as a cap on the total size of any weight.

We can now compute our long walks algorithm, which will be our global algorithm modulo a few cleanup steps:

▶ **Lemma 20.** *Assume we have access to graph $G = (V, E)$ on $n \in \mathbb{N}$ vertices, prime $p = \mathsf{poly}(n)$, and seed σ. Let $\mathbb{F} := \mathbb{F}_p$, define $\lambda := 2^{\sqrt{(\epsilon/2.1)\log n}}$, and let $U := U_\sigma$ be as given in Theorem 18. Define walk matrix $\mathcal{W}$ with respect to graph $H = (V_U, E_U)$ and weight function μ as defined in Definition 19.*

Assume we have access to a catalytic subroutine $\mathsf{SHORT}(V_{in}, V_{out}; \overrightarrow{r_{in}}, \overrightarrow{r_{out}})$, where we have $V_{in}, V_{out} \subseteq V_U$ and $\overrightarrow{r_{in}} \in \mathbb{F}^{V_{in}}, \overrightarrow{r_{out}} \in \mathbb{F}^{V_{out}}$ are any sections of the catalytic tape, whose effect is to add $\overrightarrow{r_{in}} \cdot \mathcal{W}_0$ to $\overrightarrow{r_{out}}$.

Then there exists a catalytic subroutine $\mathsf{LONG}(V_{in}, V_{out}; \overrightarrow{r_{in}}, \overrightarrow{r_{out}})$, where $V_{in}, V_{out} \subseteq V_U$ and $\overrightarrow{r_{in}} \in \mathbb{F}^{V_{in}}, \overrightarrow{r_{out}} \in \mathbb{F}^{V_{out}}$ are any sections of the catalytic tape, whose effect is to add $\overrightarrow{r_{in}} \cdot \mathcal{W}_\ell$ to $\overrightarrow{r_{out}}$, where $\ell := \log |V_U|$.

If SHORT uses time T, free space S, and catalytic space R, then LONG uses time $T \cdot \mathcal{O}(n^2)$, free space $S + \mathcal{O}(\log n)$, and catalytic memory $|V_{in}| + |V_{out}| + 3|V_U| \cdot \mathcal{O}(\log p)$, namely the vectors $\overrightarrow{r_{in}}, \overrightarrow{r_{out}}$ plus three additional catalytic vectors $\vec{r_1}, \vec{r_2}, \vec{r_3} \in \mathbb{F}^{V_U}$.

4.2 Short Walks via Color Classes

Our goal now is to compute SHORT. By the definition of the weights of H, we have that computing $\mathcal{W}_0$ with respect H is equivalent to computing $\mathcal{W}_{\log \lambda}$ with respect to G' plus accounting for multiplicities; thus we utilize Lemma 17 with this goal in mind.

▶ **Lemma 21.** *Assume we have access to graph $G = (V, E)$ on $n \in \mathbb{N}$ vertices, prime $p = \mathsf{poly}(n)$, and seed σ. Fix $\lambda := 2^{\sqrt{(\epsilon/2.1)\log n}}$, define graphs G' and H as per Definition 19, and define $\mathbb{F} := \mathbb{F}_p$ and $C := \lambda$.*

Then there exists a catalytic subroutine $\mathsf{SHORT}(V_{in}, V_{out}; \overrightarrow{r_{in}}, \overrightarrow{r_{out}})$, where $V_{in}, V_{out} \subseteq V_U$ are any sets of vertices of H and $\overrightarrow{r_{in}} \in \mathbb{F}^{V_{in}}, \overrightarrow{r_{out}} \in \mathbb{F}^{V_{out}}$ are any sections of the catalytic tape, whose effect is to add $\overrightarrow{r_{in}} \cdot \mathcal{W}_0$ to $\overrightarrow{r_{out}}$, where $\mathcal{W}$ is defined with respect to H.

SHORT runs in time $\mathcal{O}(|E| \cdot n^\epsilon)$, uses free space $\mathcal{O}(\log n)$, and uses catalytic memory $|V_{in}| + |V_{out}| + \frac{3n}{2^{\sqrt{(\epsilon/2.1)\log n}}} \cdot \mathcal{O}(\log p)$, namely the vectors $\overrightarrow{r_{in}}, \overrightarrow{r_{out}}$ plus three additional catalytic vectors $\vec{r_1}, \vec{r_2}, \vec{r_3} \in \mathbb{F}^{n/C}$.

4.3 Final Algorithm

Our LONG algorithm, using SHORT as a subroutine, gives us everything we need to compute STCONN, with the remaining issues being to choose U and p so as to ensure correctness and efficiency. We leave the proof details to the full version.

5 Edit Distance, Longest Common Subsequence and Discrete Fréchet Distance

In this section, we focus on computing edit distance, longest common subsequence and discrete Fréchet distance.

▶ **Theorem 22.** *The length of the longest common subsequence and edit distance of two strings of length n can be computed in time $\mathsf{poly}(n)$, using $\mathcal{O}(\log n)$ workspace and $\frac{n}{2^{\Omega(\sqrt{\log n})}}$ catalytic space.*

▶ **Theorem 23.** *Discrete Fréchet distance of two point sequences of length n with distance oracle whose values are in the range $\{0, \ldots, \mathsf{poly}(n)\}$ can be computed in time $\mathcal{O}(n^{4+\varepsilon})$, using $\mathcal{O}(\log n)$ workspace and $\frac{n}{2^{\Omega(\sqrt{\log n})}}$ catalytic space.*

All of these results follow from the same procedure for computing (weighted) reachability on a directed grid graph.

5.1 Grid graphs reachability

For the remainder of this section, we consider only fields of the form $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ for some prime p.

For the purpose of our algorithm, we extend the grid-like graphs used for computing the metrics to be layered, denoted by $\mathsf{Grid}_{n,n}$. Formally, for $n \in \mathbb{N}$ we define $\mathsf{Grid}_{n,n}$ as a directed graph such that

$$\begin{aligned} V(\mathsf{Grid}_{n,n}) &= \{(i,j) \colon 0 \le i \le n, 0 \le j \le n\}, \\ E(\mathsf{Grid}_{n,n}) &= \{((i,j),(i+1,j)) \colon 0 \le i \le n-1, 0 \le j \le n\} \cup \\ &\quad \{((i,j),(i+1,j+1)) \colon 0 \le i \le n-1, 0 \le j \le n-1\} \cup \\ &\quad \{((i,j),(i+1,j-1)) \colon 0 \le i \le n-1, 1 \le j \le n\}. \end{aligned}$$

Given a directed acyclic layered graph $G = (V, E)$ with weighted edges $\mu : V \times V \to \mathbb{N}$, we recall that the weight of an s,t-walk $W = (s = v_0, v_1, \ldots, v_k = t)$ is $\mu(W) = \prod_{i=0}^{k-1} \mu(v_i, v_{i+1})$. In fact, as we consider only layered graphs, and in particular $\mathsf{Grid}_{n,n}$, all walks from any u to any v are paths of the same length.

We thus define the total path weight[5] from s to t as $\mathcal{W}_{s,t} = \sum_{P\, s,t\text{-path}} \mu(P)$. In line with the flow-based algorithm of [43], our goal is to catalytically compute the total path weight from $(0,0)$ to (n,n) in a weighted Grid. To save space, we split each layer of the graph into a few color classes (which will be intervals in each of the layers), and then compute the total path weights recursively on the layers within the color classes up to the base case, where we will use only a single color class on which we will recurse slightly differently.

▶ **Theorem 24.** *There exists a catalytic algorithm that, given a prime $p = \mathsf{poly}(n)$, a graph $\mathsf{Grid}_{n,n}$ with weights $\mu : E(\mathsf{Grid}_{n,n}) \to \mathbb{N}$ given as an oracle $\mathcal{D}$ that returns the weights modulo p, and two vertices $u, v \in V(\mathsf{Grid}_{n,n})$ described by their indices in the layer such that the distance between u and v is a power of two, computes the value $\mathcal{W}_{u,v} = \sum_{P\, u,v\text{-path}} \mu(P) \bmod p$.*

The algorithm runs in time $\mathcal{O}(n^{3+\varepsilon})$ using free space $\mathcal{O}(\log n)$, catalytic space $\frac{n}{2^{\Omega(\sqrt{\log n})}}$ and $\mathcal{O}(\frac{n^{3+\varepsilon}}{2^{3\sqrt{\log n}}})$ weight oracle queries.

We remark that the assumption on the distance between u and v being a power of two is only made for simplicity. It is possible to support arbitrary distances at the cost of $\mathcal{O}(1)$ bits of free space per level of recursion, which would amount to $\mathcal{O}(\log n)$ additional free space.

Using Theorem 24 and well-known reductions to (weighted) reachability [16, 35, 49], Theorems 22 and 23 follow. Detailed proofs are available in the full version.

[5] One can also view this as $\overrightarrow{\tau_{in}}\mathcal{W}_\ell$ for a particular ℓ in Definition 14 with $\overrightarrow{\tau_{in}}$ set to be the unit vector corresponding to s.

References

1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 59–78, 2015. doi:10.1109/FOCS.2015.14.

2 Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made. In *48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 375–388, 2016. doi:10.1145/2897518.2897653.

3 Amir Abboud and Aviad Rubinstein. Fast and deterministic constant factor approximation algorithms for LCS imply new circuit lower bounds. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:14, 2018. doi:10.4230/LIPIcs.ITCS.2018.35.

4 Pankaj K. Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. Computing the discrete Fréchet distance in subquadratic time. *SIAM Journal on Computing*, 43(2):429–449, 2014. doi:10.1137/130920526.

5 Aryan Agarwala, Yaroslav Alekseev, and Antoine Vinciguerra. Linear matroid intersection is in catalytic logspace. In *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:23, 2026. doi:10.4230/LIPIcs.ITCS.2026.3.

6 Aryan Agarwala and Ian Mertz. Bipartite matching is in catalytic logspace. In *66th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2025)*, 2025.

7 Miklós Ajtai, János Komlós, and Endre Szemerédi. Deterministic simulation in LOGSPACE. In *19th Annual ACM Symposium on Theory of Computing (STOC 1987)*, pages 132–140, 1987. doi:10.1145/28395.28410.

8 Yaroslav Alekseev, Yuval Filmus, Ian Mertz, Alexander Smal, and Antoine Vinciguerra. Catalytic computing and register programs beyond log-depth. In *50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*, volume 345 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:18, 2025. doi:10.4230/LIPIcs.MFCS.2025.6.

9 Eric Allender, David A. Mix Barrington, Tanmoy Chakraborty, Samir Datta, and Sambuddha Roy. Grid graph reachability problems. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006)*, pages 299–313, 2006. doi:10.1109/CCC.2006.22.

10 Eric Allender and Klaus-Jörn Lange. RUSPACE($\log n$) $\subseteq$ DSPACE($\log^2 n/\log\log n$). *Theory of Computing Systems*, 31(5):539–550, 1998. doi:10.1007/S002240000102.

11 Carlos E. R. Alves, Edson N. Caceres, and Siang Wun Song. A coarse-grained parallel algorithm for the all-substrings longest common subsequence problem. *Algorithmica*, 45(3):301–335, 2006. doi:10.1007/s00453-006-1216-z.

12 Alexandr Andoni, Michel Deza, Anupam Gupta, Piotr Indyk, and Sofya Raskhodnikova. Lower bounds for embedding edit distance into normed spaces. In *14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003)*, pages 523–526, 2003. URL: http://dl.acm.org/citation.cfm?id=644108.644196.

13 Alexandr Andoni, Assaf Goldberger, Andrew McGregor, and Ely Porat. Homomorphic fingerprints under misalignments: Sketching edit and shift distances. In *45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 931–940, 2013. doi:10.1145/2488608.2488726.

14 Alexandr Andoni and Robert Krauthgamer. The computational hardness of estimating edit distance [Extended abstract]. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 667–676, 2007. doi:10.1109/FOCS.2007.60.

15 Alexandr Andoni and Robert Krauthgamer. The smoothed complexity of edit distance. In *35th International Colloquium on Automata, Languages, and Programming (ICALP 2008)*, volume 5125 of *Lecture Notes in Computer Science*, pages 357–369, 2008. doi:10.1007/978-3-540-70575-8_30.

16 Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Polylogarithmic approximation for edit distance and the asymmetric query complexity. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 377–386, 2010. doi:10.1109/FOCS.2010.43.

17 Alexandr Andoni and Krzysztof Onak. Approximating edit distance in near-linear time. In *41st Annual ACM Symposium on Theory of Computing (STOC 2009)*, pages 199–204, 2009. doi:10.1145/1536414.1536444.

18 Arturs Backurs and Piotr Indyk. Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). In *47th Annual ACM Symposium on Theory of Computing (STOC 2015)*, pages 51–58, 2015. doi:10.1145/2746539.2746612.

19 Nikhil Bansal, Moshe Lewenstein, Bin Ma, and Kaizhong Zhang. On the longest common rigid subsequence problem. *Algorithmica*, 56(2):270–280, 2010. doi:10.1007/s00453-008-9175-1.

20 Greg Barnes, Jonathan F. Buss, Walter L. Ruzzo, and Baruch Schieber. A sublinear space, polynomial time algorithm for directed *st*-connectivity. *SIAM Journal on Computing*, 27(5):1273–1282, 1998. doi:10.1137/S0097539793283151.

21 Tuğkan Batu, Funda Ergun, and Cenk Sahinalp. Oblivious string embeddings and edit distance approximations. In *17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006)*, pages 792–801, 2006. URL: https://dl.acm.org/doi/10.5555/1109557.1109644.

22 Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing*, 21(1):54–58, 1992. doi:10.1137/0221006.

23 Sagar Bisoyi, Krishnamoorthy Dinesh, and Jayalal Sarma. On pure space vs catalytic space. *Theoretical Computer Science*, 921:112–126, 2022. doi:10.1016/J.TCS.2022.04.005.

24 Sagar Bisoyi, Krishnamoothy Dinesh, Bhabya Deep Rai, and Jayalal Sarma. Almost-catalytic computation. In *14th International Conference on Algorithms and Complexity (CIAC 2025)*, volume 15680 of *Lecture Notes in Computer Science*, pages 35–51, 2025. doi:10.1007/978-3-031-92935-9_3.

25 Mahdi Boroujeni, Soheil Ehsani, Mohammad Ghodsi, MohammadTaghi HajiAghayi, and Saeed Seddighin. Approximating edit distance in truly subquadratic time: Quantum and MapReduce. In *29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*, 2018. doi:10.1137/1.9781611975031.76.

26 Mahdi Boroujeni and Saeed Seddighin. Improved MPC algorithms for edit distance and ulam distance. In *31st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2019)*, pages 31–40, 2019. doi:10.1145/3323165.3323205.

27 Karl Bringmann, Fabrizio Grandoni, Barna Saha, and Virginia Vassilevska Williams. Truly sub-cubic algorithms for language edit distance and RNA-folding via fast bounded-difference min-plus product. In *57th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2016)*, pages 375–384, 2016. doi:10.1109/FOCS.2016.48.

28 Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 79–97, 2015. doi:10.1109/FOCS.2015.15.

29 Karl Bringmann and Marvin Künnemann. Multivariate fine-grained complexity of longest common subsequence. In *29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*, pages 1216–1235, 2018. doi:10.1137/1.9781611975031.79.

30 Kevin Buchin, Maike Buchin, and Joachim Gudmundsson. Detecting single file movement. In *16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS 2008)*, pages 1–10, 2008. doi:10.1145/1463434.1463476.

31 Kevin Buchin, Maike Buchin, Joachim Gudmundsson, Maarten Löffler, and Jun Luo. Detecting commuting patterns by clustering subtrajectories. *International Journal of Computational Geometry & Applications*, 21(3):253–282, 2011. doi:10.1142/S0218195911003652.

32 Harry Buhrman, Richard Cleve, Michal Koucký, Bruno Loff, and Florian Speelman. Computing with a full memory: Catalytic space. In *46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 857–866, 2014. doi:10.1145/2591796.2591874.

33 Harry Buhrman, Marten Folkertsma, Ian Mertz, Florian Speelman, Sergii Strelchuk, Sathyawageeswar Subramanian, and Quinten Tupker. Quantum catalytic space. In *20th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2025)*, volume 350 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:24, 2025. doi:10.4230/LIPIcs.TQC.2025.11.

34 Harry Buhrman, Michal Koucký, Bruno Loff, and Florian Speelman. Catalytic space: Nondeterminism and hierarchy. *Theory of Computing Systems*, 62(1):116–135, 2018. doi:10.1007/S00224-017-9784-7.

35 Diptarka Chakraborty, Elazar Goldenberg, and Michal Koucký. Streaming algorithms for embedding and computing edit distance in the low distance regime. In *48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 712–725, 2016. doi:10.1145/2897518.2897577.

36 Diptarka Chakraborty and Raghunath Tewari. An $O(n^{\varepsilon})$ space and polynomial time algorithm for reachability in directed layered planar graphs. *ACM Transactions on Computation Theory*, 9(4):1–11, 2017. doi:10.1145/3154857.

37 Kuan Cheng, Alireza Farhadi, MohammadTaghi Hajiaghayi, Zhengzhong Jin, Xin Li, Aviad Rubinstein, Saeed Seddighin, and Yu Zheng. Streaming and small space approximation algorithms for edit distance and longest common subsequence. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 54:1–54:20, 2021. doi:10.4230/LIPIcs.ICALP.2021.54.

38 James Cook, Surendra Ghentiyala, Ian Mertz, Edward Pyne, and Nathan S. Sheffield. The structure of in-place space-bounded computation. *arXiv preprint*, arXiv:2510.12005, 2025. doi:10.48550/arXiv.2510.12005.

39 James Cook, Jiatu Li, Ian Mertz, and Edward Pyne. The structure of catalytic space: Capturing randomness and time via compression. In *57th Annual ACM Symposium on Theory of Computing (STOC 2025)*, pages 554–564, 2025. doi:10.1145/3717823.3718112.

40 James Cook and Ian Mertz. Catalytic approaches to the tree evaluation problem. In *52nd Annual ACM Symposium on Theory of Computing (STOC 2020)*, pages 752–760, 2020. doi:10.1145/3357713.3384316.

41 James Cook and Ian Mertz. Trading time and space in catalytic branching programs. In *37th Computational Complexity Conference (CCC 2022)*, volume 234 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:21, 2022. doi:10.4230/LIPIcs.CCC.2022.8.

42 James Cook and Ian Mertz. Tree evaluation is in space $O(\log n \cdot \log\log n)$. In *56th Annual ACM Symposium on Theory of Computing (STOC 2024)*, pages 1268–1278, 2024. doi:10.1145/3618260.3649664.

43 James Cook and Edward Pyne. Efficient catalytic graph algorithms. In *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:22, 2026. doi:10.4230/LIPIcs.ITCS.2026.43.

44 Samir Datta, Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Randomized and symmetric catalytic computation. In *15th International Computer Science Symposium in Russia (CSR 2020)*, volume 12159 of *Lecture Notes in Computer Science*, pages 211–223, 2020. doi:10.1007/978-3-030-50026-9_15.

45 Dean Doron, Edward Pyne, and Roei Tell. Opening up the distinguisher: A hardness to randomness approach for BPL=L that uses properties of BPL. In *56th Annual ACM Symposium on Theory of Computing (STOC 2024)*, pages 2039–2049, 2024. doi:10.1145/3618260.3649772.

46 Dean Doron, Edward Pyne, Roei Tell, and R. Ryan Williams. When connectivity is hard, random walks are easy with non-determinism. In *57th Annual ACM Symposium on Theory of Computing (STOC 2025)*, pages 1108–1117, 2025. doi:10.1145/3717823.3718303.

47 Roman Edenhofer. A space-space trade-off for directed *st*-connectivity. *arXiv preprint*, arXiv:2602.21088, 2026. doi:10.48550/arXiv.2602.21088.

48 Jeff Edmonds, Chung Keung Poon, and Dimitris Achlioptas. Tight lower bounds for *st*-connectivity on the NNJAG model. *SIAM Journal on Computing*, 28(6):2257–2284, 1999. doi:10.1137/S0097539795295948.

49 Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, 1994.

50 Marten Folkertsma, Ian Mertz, Florian Speelman, and Quinten Tupker. Fully characterizing lossy catalytic computation. In *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 50:1–50:13, 2025. doi:10.4230/LIPIcs.ITCS.2025.50.

51 Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Unambiguous catalytic computation. In *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019)*, volume 150 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:13, 2019. doi:10.4230/LIPIcs.FSTTCS.2019.16.

52 Chetan Gupta, Rahul Jain, Vimal Raj Sharma, and Raghunath Tewari. Lossy catalytic computation. *arXiv preprint*, arXiv:2408.14670, 2024. doi:10.48550/arXiv.2408.14670.

53 Chetan Gupta, Raghunath Tewari, and Vimal Raj Sharma. Efficient isolation of perfect matching in $o(\log n)$ genus bipartite graphs. *arXiv preprint*, arXiv:2511.21217, 2025. doi:10.48550/arXiv.2511.21217.

54 William Hesse, Eric Allender, and David A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65(4):695–716, 2002. doi:10.1016/S0022-0000(02)00025-9.

55 Sampath Kannan, Sanjeev Khanna, and Sudeepa Roy. STCON in directed unique-path graphs. In *28th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2008)*, volume 2 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 256–267, 2008. doi:10.4230/LIPIcs.FSTTCS.2008.1758.

56 Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping to massive datasets. In *3rd European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 1999)*, volume 1704 of *Lecture Notes in Computer Science*, pages 1–11, 1999. doi:10.1007/978-3-540-48247-5_1.

57 Man-Soon Kim, Sang-Wook Kim, and Miyoung Shin. Optimization of subsequence matching under time warping in time-series databases. In *2005 ACM Symposium on Applied Computing (SAC 2005)*, pages 581–586, 2005. doi:10.1145/1066677.1066814.

58 Masashi Kiyomi, Takashi Horiyama, and Yota Otachi. Longest common subsequence in sublinear space. *Information Processing Letters*, 168:106084, 2021. doi:10.1016/j.ipl.2020.106084.

59 Michal Koucký. Catalytic computation. *Bulletin of the EATCS*, 118, 2016. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/400.

60 Michal Koucký, Ian Mertz, Edward Pyne, and Sasha Sami. Collapsing catalytic classes. In *66th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2025)*, 2025.

61 Jiatu Li, Edward Pyne, and Roei Tell. Distinguishing, predicting, and certifying: On the long reach of partial notions of pseudorandomness. In *65th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2024)*, pages 1–13, 2024. doi:10.1109/FOCS61266.2024.00095.

62 Ian Mertz. Reusing space: Techniques and open problems. *Bulletin of the EATCS*, 141, 2023. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/780.

63 Edward Pyne. Derandomizing logspace with a small shared hard drive. *Computational Complexity*, 34:13, 2025. doi:10.1007/S00037-025-00275-6.

64 Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4):17:1–17:24, 2008. doi:10.1145/1391289.1391291.

65 Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom walks on regular digraphs and the RL vs. L problem. In *38th Annual ACM Symposium on Theory of Computing (STOC 2006)*, pages 457–466, 2006. doi:10.1145/1132516.1132583.

66 Michael Saks and C. Seshadhri. Space efficient streaming algorithms for the distance to monotonicity and asymmetric edit distance. In *24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2013)*, pages 1698–1709, 2013. doi:10.1137/1.9781611973105.122.

67 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970. doi:10.1016/S0022-0000(70)80006-X.

68 Yakov Shalunov. Improved bounds on the space complexity of circuit evaluation. *arXiv preprint*, arXiv:2504.20950, 2025. doi:10.48550/arXiv.2504.20950.

69 Salil Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1–3):1–336, 2012. doi:10.1561/0400000010.

70 R. Ryan Williams. Simulating time with square-root space. In *57th Annual ACM Symposium on Theory of Computing (STOC 2025)*, pages 13–23, 2025. doi:10.1145/3717823.3718225.

Bounds for Hardness Condensation in the Query Model

Chandrima Kayal
Université Paris Cité, CNRS, IRIF, Paris, France

Rajat Mittal
Indian Institute of Technology Kanpur, India

Sai Soumya Nalli
Microsoft Research India, Bangalore, India

Manaswi Paraashar
Indian Institute of Technology Hyderabad, India

Karthikeya Polisetty
Indian Institute of Technology Madras, India

Jayalal Sarma
Indian Institute of Technology Madras, India

Nitin Saurabh
Indian Institute of Technology Hyderabad, India

Abstract

For any Boolean function $f : \{0,1\}^n \to \{0,1\}$ with a complexity measure having value $k \ll n$, is it possible to restrict the function f to $\Theta(k)$ variables while keeping the complexity preserved at $\Theta(k)$? Instantiation of this question for the measure of circuit complexity of the Boolean function was shown to be related to circuit lower bounds (Buresh-Oppenheim and Santhanam, 2006). Variants of the above question were also shown to have connections to the log-rank conjecture in communication complexity (Hrubeš, 2024) and lower bounds in proof complexity (Razborov, 2016). In the context of communication and query complexity, this question was recently studied by Göös, Newman, Riazanov and Sokolov (2024). They showed, among other results, that query complexity cannot be condensed losslessly.

In this work, we show that there exists a Boolean function f such that any restriction of f to $O(\mathcal{M}(f))$ variables has $\mathcal{M}(\cdot)$-complexity at most $\tilde{O}(\mathcal{M}(f)^{2/3})$, where $\mathcal{M}$ is one of block sensitivity (bs), fractional block sensitivity (fbs), certificate complexity (C), deterministic query complexity (D), zero-error randomized query complexity (R_0), and AND (and OR)-decision tree query complexity. This improves upon the results of Göös, Newman, Riazanov, and Sokolov (2024) for D and R_0, and in particular answers their open question about the condensation of block sensitivity.

We complement the negative results on lossless condensation with positive results about lossy condensation. In particular, we show that for every Boolean function f there exists a restriction of f to $O(\mathcal{M}(f))$ variables such that its $\mathcal{M}(\cdot)$-complexity is at least $\Omega(\mathcal{M}(f)^{1/2})$, where $\mathcal{M} \in \{\mathsf{bs}, \mathsf{fbs}, \mathsf{C}, \mathsf{UC}_{\min}, \mathsf{UC}_1, \mathsf{UC}, \mathsf{D}, \widetilde{\deg}, \lambda\}$. In addition, we show lossy condensation for randomized and quantum query complexity with a slightly smaller exponent.

2012 ACM Subject Classification Theory of computation → Oracles and decision trees

Keywords and phrases Query Complexity, Decision Trees, Hardness Condensation

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.30

Related Version *Full Version*: https://arxiv.org/abs/2602.00754v2

Funding *Chandrima Kayal*: Funded by French PEPR integrated project EPiQ (ANR-22-PETQ-0007) and ANR Grant FLITTLA (ANR-21-CE48-0023).
Nitin Saurabh: Supported by ANRF ARG-MATRICS grant ANRF/ARGM/2025/003191/MTR.

Acknowledgements CK and NS would like to thank Meena Mahajan and Gaurav Sood for insightful discussions during the initial stages of the work. We would also like to thank anonymous reviewers whose careful reading and thoughtful comments greatly improved the presentation of this paper. We would also like to thank the audience in STCS seminar at TIFR, Mumbai for helpful discussions.



© Chandrima Kayal, Rajat Mittal, Sai Soumya Nalli, Manaswi Paraashar, Karthikeya Polisetty, Jayalal Sarma, and Nitin Saurabh;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 30; pp. 30:1–30:20



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The problem of proving circuit lower bounds is one of the most challenging problems in computational complexity theory. Buresh-Oppenheim and Santhanam [9] introduced the concept of *hardness condensation* as a way to prove better circuit lower bounds. Their idea was to design a procedure that takes as input a function f that is hard to compute and outputs another function f', possibly on fewer input bits, such that the hardness of f' with respect to its input length is greater than that of f. In particular, they showed hardness condensation for the complexity measure circuit size. A decade later, this strategy was successfully implemented by Razborov [27] to establish strong lower bounds in proof complexity. Since then, the technique of hardness condensation has found many applications in proof complexity [28, 29, 6, 14, 10, 12, 13].

Clearly, the paradigm of hardness condensation is quite general and one could ask the following question for any complexity measure $\mathcal{M}(\cdot)$:

> Given a function f on n variables with complexity $\mathcal{M}(f)$ (a growing function of n), can we transform f explicitly into another function f' on a smaller number of variables t such that $\mathcal{M}(f') = \Theta(\mathcal{M}(f))$ while also being maximal over all functions on t variables?

The aforementioned question has also been explored recently in the context of communication complexity and matrix ranks [17, 19, 16], and query complexity [16]. In communication complexity, we are given a $2^n \times 2^n$ Boolean matrix M and the task is to solve the following two-player game: Alice is given a row $x \in [2^n]$ of M, Bob is given a column $y \in [2^n]$ of M, and their goal is to compute $M(x, y)$ while minimizing communication between them. Let $\mathsf{cc}(M)$ denote the deterministic communication complexity of this game. We say that M *condenses* to k variables if M contains a $2^k \times 2^k$ submatrix N such that $\mathsf{cc}(N) = \Theta(k)$. Further, the deterministic communication complexity *condenses losslessly* if every M condenses to $\Theta(\mathsf{cc}(M))$ variables. The hardness condensation question then asks whether $\mathsf{cc}(\cdot)$ condenses losslessly. Apart from being an interesting question about witnessing intrinsic hardness, it also has connections to the famous log-rank conjecture [24]. In particular, the log-rank conjecture implies that there exists a universal constant $\alpha \geq 1$ such that every M condenses to $\Omega(\mathsf{cc}(M)^{1/\alpha})$ variables (see also [19, 16]). In a recent work, Hrubeš [19] proved this implication unconditionally by showing that every M condenses to $\Omega(\sqrt{\mathsf{cc}(M)})$ variables.

In the context of query complexity, it was explored recently by Göös, Newman, Riazanov and Sokolov [16]. They studied whether hardness condensation can be achieved using the simple and natural transformation – *restrictions* – that is, substituting variables by constants. Typically, to reduce the number of variables, one does *variable substitutions* and/or composes the variables of f with *small gadgets* (e.g., XOR, OR, Indexing functions).

Let $\mathcal{M}$ be any decision tree based complexity measure of Boolean functions (for example, sensitivity, block sensitivity, certificate complexity, query complexity, unambiguous certificate complexity, etc). We say that a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ *condenses* to k variables *with respect to* the measure $\mathcal{M}$, if there exists a restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = k$, such that the restricted function $f|_\rho$ on k variables has large $\mathcal{M}$, i.e., $\mathcal{M}(f|_\rho) = \Theta(k)$. We further say that f *condenses losslessly* with respect to $\mathcal{M}$ if it condenses to $\Theta(\mathcal{M}(f))$ variables. We also say that $\mathcal{M}$ condenses losslessly if every function f condenses losslessly w.r.t. $\mathcal{M}$.

For example, consider the measure of query complexity of Boolean functions, which is studied by [16]. For $f\colon \{0,1\}^n \to \{0,1\}$, the query complexity $\mathsf{D}(f)$ of f is the minimum number of queries made by a deterministic decision tree algorithm computing f in the worst-case over all inputs $x \in \{0,1\}^n$. Following the above definition (also from [16]), we say that f condenses to k variables with respect to the measure of deterministic query

complexity if there exists a partial assignment $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = k$, such that the restricted function $f|_\rho$ on k variables has large query complexity, i.e., $\mathsf{D}(f|_\rho) = \Theta(k)$. The hardness condensation question, w.r.t. $\mathsf{D}(\cdot)$, then asks if $\mathsf{D}(\cdot)$ condenses losslessly.

The authors of [16] answered the aforementioned question in the negative. That is, they showed a function f such that for every restriction ρ that fixes all but at most $O(\mathsf{D}(f))$ variables, the query complexity $\mathsf{D}(f|_\rho)$ of the restricted function $f|_\rho$ is $\tilde{O}(\mathsf{D}(f)^{3/4})$ (where $\tilde{O}$ hides polylog factors in $\mathsf{D}(f)$). In other words, deterministic (and even randomized) query complexity cannot be condensed losslessly in general. The authors wondered if there is a function that witnesses greater loss, i.e., for which the hardness is even less condensible. Observe that the best possible bound cannot be better than $O(\mathsf{D}(f)^{1/3})$. Indeed, it follows from the fact that every f condenses to $\deg(f)$ variables[1] and $\deg(f) \geq \mathsf{D}(f)^{1/3}$ [25]. In [16], the authors left open, among other things, the problem of closing this gap and verifying lossless condensation for other decision tree based measures like, block sensitivity, certificate complexity, unambiguous certificate complexity, etc.

In this article, we delve deeper into the question of hardness condensation for $\mathsf{D}(\cdot)$ and initiate the study of hardness condensation for almost all other decision tree based measures. In particular, we close the gap for D from both sides, showing that it is not possible to condense even with an exponent of $2/3 + \varepsilon$ (improving from $3/4$) and it is always possible to condense with an exponent of $1/2$ (improving from $1/3$). We show similar results for many other well-known complexity measures like block sensitivity, certificate complexity, fractional block sensitivity, AND-decision tree complexity, etc. (Theorems 1, 2, 3 and 5).

Unfortunately, except for approximate degree and spectral sensitivity, there are gaps in terms of what can and cannot be achieved for these complexity measures in terms of hardness condensation (see Table 1). These gaps leave many intriguing questions open; we hope that future research will be able to resolve these questions.

Before going into the details of our results and proof techniques, we present some possible applications of hardness condensation by restrictions.

Applications of Hardness Condensation by Restrictions

Log-Rank Conjecture. We begin with an application to the log-rank conjecture highlighted by [18]. Let the AND-decision tree query complexity $\mathsf{D}_\wedge(f)$ of a Boolean function f be the minimum number of AND queries to the input made by a deterministic query algorithm computing f in the worst case (see Definition 14).

For any $f\colon \{0,1\}^n \to \{0,1\}$, let $\mathsf{mon}(f)$ denote the number of monomials with non-zero coefficients in the unique multilinear polynomial representing f. Further, define the communication problem $f_\wedge\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ as follows $f_\wedge(x,y) = f(x_1 \wedge y_1, \ldots, x_n \wedge y_n)$. It then follows that the rank of the communication matrix of $f_\wedge$ equals $\mathsf{mon}(f)$ [7]. Moreover, it is known that

$$\mathsf{cc}(f_\wedge) \leq 2\mathsf{D}_\wedge(f) = O(\log^5 \mathsf{mon}(f) \cdot \log n) = O(\log^5 \mathsf{rank}(f_\wedge) \cdot \log n),$$

where the second inequality was shown by [22]. That is, the log-rank conjecture holds for $f_\wedge$ as long as $\log\log \mathsf{mon}(f) = \Omega(\log\log n)$. In other words, the log-rank conjecture remains unsolved for $f_\wedge$ when f has a very sparse representation as a polynomial. To resolve this, [18] observed that a hardness condensation of the following form suffices.

[1] Consider a monomial $\mathfrak{m}$ of maximum degree $\deg(f)$ in the unique real polynomial representing f. Let ρ be a restriction that fixes all variables except the ones in $\mathfrak{m}$. Then $\mathsf{D}(f|_\rho) = \deg(f|_\rho) = \deg(f)$.

▶ **Conjecture** ([18]). *There exist universal constants $\alpha, \beta > 0$ such that for every $f\colon \{0,1\}^n \to \{0,1\}$ there exists a restriction ρ with $|\rho^{-1}(*)| = 2^{O(\log^{\alpha} \mathsf{mon}(f))}$ and $\mathsf{D}_{\wedge}(f|_\rho) = \Omega(\mathsf{D}_{\wedge}(f)^{\beta})$.*

Separations between complexity measures. We now illustrate an application of hardness condensation to witness improved separations between complexity measures. We illustrate the argument with a simple example to highlight the applicability of this approach.

Consider the modified Rubinstein function f given in Definition 23. It can be easily shown that $\mathsf{D}(f) = k^2$ and $\mathsf{bs}(f) = k^{1.5}$ (see also Lemma 26). Thus we have an example where $\mathsf{D}(f) = \Omega(\mathsf{bs}(f)^{4/3})$ and $\mathsf{bs}(f) = \Omega(\mathsf{s}(f)^{3/2})$. Using hardness condensation, we now amplify the gap between D and bs. We know from Theorem 1 that f witnesses the incondensability of bs with exponent $2/3$. That is, for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(\mathsf{bs}(f))$ we have $\mathsf{bs}(f|_\rho) = O(k) = O(\mathsf{bs}(f)^{2/3})$. Now consider a restriction γ with $|\gamma^{-1}(*)| = \Theta(\mathsf{bs}(f))$ such that $\mathsf{D}(f|_\gamma) = |\gamma^{-1}(*)| = \Theta(\mathsf{bs}(f))$. Such a restriction is easily seen to exist since $\mathsf{D}(f)$ is full to begin with. We then have by Theorem 1 that $\mathsf{bs}(f|_\gamma) = O(k) = O(\mathsf{bs}(f)^{2/3})$. We therefore have obtained a function $h := f|_\gamma$ such that $\mathsf{D}(h) = \Omega(\mathsf{bs}(h)^{3/2})$ improving over $4/3$ separation witnessed by f. In general, such an approach can be used to achieve better separations.

A candidate application of the approach outlined above could be the gap between D and R. The best known separation between D and R is quadratic [3]. Note that we know of a function that witness more than a quadratic gap between R and s [5]. The approach outlined above exploits this gap with respect to sensitivity to obtain improved separations between complexity measures.

We further note that the positive condensation result in Theorem 5 also shows a limit to amplification using the approach outlined above. For example, we have shown in Theorem 5 that for every f there exists a restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(\mathsf{bs}(f))$ such that $\mathsf{bs}(f|_\rho) = \Omega(\sqrt{\mathsf{bs}(f)})$. Therefore, our approach cannot show a gap of more than $1/2$ between D and bs.

1.1 Our Results

In this work, we further explore the phenomenon of hardness condensation in the setting of query complexity and other Boolean function parameters using restrictions. We begin by recalling that sensitivity and degree are two such measures that condense losslessly. As mentioned above, [16] shows that deterministic query complexity cannot be condensed losslessly. Our first result (proved in Section 3) shows that block sensitivity, fractional block sensitivity and certificate complexity cannot be condensed losslessly.

▶ **Theorem 1.** *There exists a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with $\mathsf{bs}(f) = \mathsf{fbs}(f) = \mathsf{C}(f) = n^{3/4}$, such that for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(n^{3/4})$ we have $\mathsf{bs}(f|_\rho) \leq \mathsf{fbs}(f|_\rho) \leq \mathsf{C}(f|_\rho) = O(\sqrt{n}) = O(\mathsf{bs}(f)^{2/3})$.*

Our second result (proved in Section 4) shows that deterministic query complexity is even less condensible than shown in [16]. We also show that AND-decision tree complexity and 0-decision tree complexity cannot be condensed losslessly.

▶ **Theorem 2.** *There exists a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with $\tilde{\Omega}(n^{3/5}) = \mathsf{D}_0(f) \leq \mathsf{D}_{\wedge}(f) \leq \mathsf{D}(f) = O(n^{3/5})$, such that for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(n^{3/5})$ we have $\mathsf{D}_0(f|_\rho) \leq \mathsf{D}_{\wedge}(f|_\rho) \leq \mathsf{D}(f|_\rho) = \tilde{O}(n^{2/5}) = \tilde{O}(\mathsf{D}_0(f)^{2/3})$.*

Our proof technique for Theorem 2 works even in the dual setting and therefore we also obtain the following incondensability of OR-decision tree complexity.

▶ **Theorem 3.** *There exists a Boolean function* $f\colon \{0,1\}^n \to \{0,1\}$ *with* $\tilde{\Omega}(n^{3/5}) = \mathsf{D}_1(f) \leq \mathsf{D}_\vee(f) = O(n^{3/5})$*, such that for every restriction* $\rho\colon [n] \to \{0,1,*\}$ *with* $|\rho^{-1}(*)| = O(n^{3/5})$ *we have* $\mathsf{D}_1(f|_\rho) \leq \mathsf{D}_\vee(f|_\rho) = \tilde{O}(n^{2/5}) = \tilde{O}(\mathsf{D}_1(f)^{2/3})$.

We further observe that the zero-error randomized query complexity of the example in Theorem 2 is also high $\tilde{\Omega}(n^{3/5})$, thereby obtaining a similar incondensability result for R_0. For a proof of the following Corollary, we refer to the full version.

▶ **Corollary 4.** *There exists a Boolean function* $f\colon \{0,1\}^n \to \{0,1\}$ *with* $\mathsf{R}_0(f) = \tilde{\Theta}(n^{3/5})$*, such that for every restriction* $\rho\colon [n] \to \{0,1,*\}$ *with* $|\rho^{-1}(*)| = O(\mathsf{R}_0(f))$ *we have* $\mathsf{R}_0(f) = \tilde{O}(\mathsf{R}_0(f)^{2/3})$.

We note that Theorems 1 and 2 answer (or, partially answer) Problems 1 and 2 in [16]. Theorem 2 also improves upon (one of) the results in [16]. It is natural to wonder if the gaps in Theorems 1 and 2 are quantitatively optimal. That is, does there exist a function for which a measure is even less condensible than established here?

In particular, for deterministic query complexity, it asks if there exists a function f such that for all restrictions ρ with $|\rho^{-1}(*)| = O(\mathsf{D}(f))$, we have $\mathsf{D}(f|_\rho) = O(\mathsf{D}(f)^\alpha)$ where $\alpha < 2/3$? Observe that $\alpha \geq 1/3$, since $\mathsf{D}(f) \leq \deg(f)^3$ [25] and $\deg(f)$ condenses losslessly. Similarly, for block sensitivity, the exponent $\alpha \geq 1/4$, since $\mathsf{bs}(f) \leq \mathsf{s}(f)^4$ [20] and $\mathsf{s}(f)$ condenses losslessly.

Observe that the lower bound on the exponent α can be improved if conditionally based on other relationships between the parameters. For example, if $\mathsf{D}(f) \leq \deg(f)^2$, or $\mathsf{bs}(f) \leq \mathsf{s}(f)^2$, then such an improvement is possible. Our third result (proved in Section 5) establishes the same lower bound on the exponent α unconditionally.

▶ **Theorem 5.** *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a Boolean function and* $\mathcal{M} \in \{\mathsf{bs}, \mathsf{fbs}, \mathsf{C}, \mathsf{UC}_{\min}, \mathsf{UC}_1, \mathsf{UC}, \mathsf{D}, \widetilde{\deg}, \lambda, \mathsf{R}, \mathsf{R}_0, \mathsf{Q}, \mathsf{Q}_E\}$ *be a complexity measure. Then, the following holds:*

(a) *For* $\mathcal{M} \notin \{\mathsf{R}, \mathsf{R}_0, \mathsf{Q}, \mathsf{Q}_E\}$*, there exists a restriction* $\rho \in \{0,1,*\}^n$ *with* $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ *such that* $\mathcal{M}(f|_\rho) = \Omega(\sqrt{\mathcal{M}(f)})$.

(b) *For* $\mathcal{M} \in \{\mathsf{R}, \mathsf{R}_0, \mathsf{Q}_E\}$*, there exists a restriction* $\rho \in \{0,1,*\}^n$ *with* $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ *such that* $\mathcal{M}(f|_\rho) = \Omega(\mathcal{M}(f)^{1/3})$.

(c) *For* $\mathcal{M} = \mathsf{Q}$*, there exists a restriction* $\rho \in \{0,1,*\}^n$ *with* $|\rho^{-1}(*)| = \Theta(\mathsf{Q}(f))$ *such that* $\mathsf{Q}(f|_\rho) = \Omega(\mathsf{Q}(f)^{1/4})$.

Table 1 lists the known bounds on condensation for many decision tree measures. Note that for approximate degree ($\widetilde{\deg}$) and spectral sensitivity (λ), we have tight bounds on lossy condensation that can be achieved. For the other measures, there is a gap in the negative result and the positive result. Thus, the understanding for many of them is not complete, leaving many interesting open questions.

1.2 Proof Ideas

In this subsection, we explain the ideas and techniques used to obtain our results.

Incondensability of bs and C. Observe that since $\mathsf{s} \leq \mathsf{bs} \leq \mathsf{C}$ and sensitivity condenses losselessly, a counter-example $f\colon \{0,1\}^n \to \{0,1\}$ to lossless condensation of block sensitivity must witness a gap between bs and s, i.e., $\mathsf{bs}(f) = \omega(\mathsf{s}(f))$. Simultaneously, there should also be a gap between the number of variables n and $\mathsf{bs}(f)$, so that every restriction ρ with $|\rho^{-1}(*)| = O(\mathsf{bs}(f))$ must set some variables to constants. Thus the Rubinstein function (and rather a modification of it to introduce a gap between n and bs) becomes a natural choice.

Table 1 Hardness condensation by variable restrictions.

- A "$*$" entry : we do not know whether the measure labeling the row condenses losslessly or not.
- An entry γ in the Negative Result column (labelled by measure $\mathcal{M}$) : $\exists f$ such that for every restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = O(\mathcal{M}(f))$ we have $\mathcal{M}(f|_\rho) = O(\mathcal{M}(f)^\gamma)$.
- An entry ξ in the Positive Result column (labelled by measure $\mathcal{M}$) : $\forall f$ there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ such that $\mathcal{M}(f|_\rho) = \Omega(\mathcal{M}(f)^\xi)$.

Complexity Measure	Negative Result	Positive Result
D	2/3 (Theorem 2)	1/2 (Theorem 5)
D_0, D_1, $\mathsf{D}_\wedge$, $\mathsf{D}_\vee$	2/3 (Theorems 2 & 3)	$*$
R_0	2/3 (Corollary 4)	1/3 (Theorem 5)
R	3/4 ([16])	1/3 (Theorem 5)
$\mathsf{C}, \mathsf{fbs}, \mathsf{bs}$	2/3 (Theorem 1)	1/2 (Theorem 5)
$\mathsf{UC}, \mathsf{UC}_{\min}$	$*$	1/2 (Theorem 5)
Q_E	$*$	1/3 (Theorem 5)
Q	1/2 [OR_n]	1/4 (Theorem 5)
$\widetilde{\deg}$, λ	1/2 [OR_n]	1/2 (Theorem 5)
$\mathsf{s}, \deg$	–	condenses losslessly (see also Lemma 36)

We then observe that the Rubinstein function also has a nice property that only a particular kind of inputs (all 0's) have high block sensitivity and certificate complexity. We use this structure to argue that the certificate complexity, and in turn the block sensitivity, must reduce after restriction.

Incondensability of D and its variants. Incondensability of D was shown by [16] using a cheat-sheet version of Tribes (AND of ORs). Our improved example is also a cheat-sheet version f_{CS} (Equation (1)) of Tribes f (Definition 28). However, we choose different arities for AND and ORs. In particular, we choose $f = \mathsf{AND}_k \circ \mathsf{OR}_{\sqrt{k}}$. This turns out to be a crucial insight that helps design an improved query algorithm for $f_{\mathrm{CS}}|_\rho$; thereby obtaining an improvement.

Our query algorithm for $f_{\mathrm{CS}}|_\rho$, like in [16], has two parts. In the first part our algorithm finds a string $z \in \{0,1\}^{\log t}$, where t is the number of cheat-sheet cells, that the address part $f(x_1|_\rho)f(x_2|_\rho)\cdots f(x_{\log t}|_\rho)$ can take if the restricted function $f_{\mathrm{CS}}|_\rho$ were to evaluate to 1. In the second part, the algorithm simply reads the cell pointed to by z and verifies it.

In the first part, the algorithm makes queries with an aim to find each bit of $z = z_1 z_2 \cdots z_{\log t} \in \{0,1\}^{\log t}$, while satisfying the following property: either we know the value of $z_j = f(x_j|_\rho)$ correctly or there are no valid cheat-sheet cells with $z_j = 1$. For each $j \in [\log t]$, the query algorithm maintains the following two sets: (a) the set $\mathcal{O}_0$ of ORs in the j-th copy of f that can possibly evaluate to 0, and (b) the set $\mathcal{V}_1$ of cheat-sheet cells with $z_j = 1$ and not found to be inconsistent with the queries made so far. Note that initially $|\mathcal{O}_0| = k$ and $|\mathcal{V}_1| = t/2$. Now the algorithm makes queries in such a way that each query reduces either $|\mathcal{O}_0|$ by 1 or $|\mathcal{V}_1|$ by a factor of $(1 - \frac{1}{2\sqrt{k}})$. Finally when the algorithm stops, after making at most $O(k + \sqrt{k}\log t)$ queries, we either know the value of $f(x_j|_\rho)$ or more crucially are guaranteed to be in one of the following two cases: (i) $|\mathcal{O}_0| = O(\sqrt{k})$ or (ii) $|\mathcal{V}_1| = O(k)$. In case (i) the algorithm can query all remaining ORs in $\mathcal{O}_0$ to find out the value of $f(x_j|_\rho)$, while in case (ii) it uses $O(\log k)$ queries to find inconsistency to remove a cell from $\mathcal{V}_1$ (Claim 31), and finally the last remaining cheat-sheet cell in $\mathcal{V}_1$ is verified for 1-certificate that it claims to contain for $f(x_j|_\rho)$.

To find a query that allows us to reduce either $|\mathcal{O}_0|$ by 1 or $|\mathcal{V}_1|$ by a factor of $(1 - \frac{1}{2\sqrt{k}})$, we use the fact that $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}}) = \tilde{O}(k^{1.5})$ and thus there must be a claimed 1-certificate that is completely revealed in at least half the fraction of $\mathcal{V}_1$.

Lossy condensation. We complement our negative results on lossless condensation with positive results about lossy condensation. Our proofs follow an analysis of different cases based on whether $\mathsf{s}(f)$ is large or $\deg(f)$ is large or neither. In each case, we find a restriction that gives the desired lossy condensation. In the first two cases, since sensitivity and degree condense losslessly (Lemma 36), we easily find a restriction witnessing lossy condensation. The last and the interesting case is when neither of them is large; we then argue that it must be the case that block sensitivity is large, which then helps us find a restriction that witnesses lossy condensation. For example, consider the case of the deterministic query complexity. As alluded to before, the interesting case is when neither sensitivity nor degree is large, i.e., at least $\sqrt{\mathsf{D}(f)}$. Then, from $\mathsf{D}(f) \leq \deg(f)\mathsf{bs}(f)$ [25], it follows that $\mathsf{bs}(f) \geq \sqrt{\mathsf{D}(f)}$. Thus, we can consider the following restriction ρ based on an input z with the maximum block sensitivity: leave $\sqrt{\mathsf{D}(f)}$ disjoint (minimal) sensitive blocks unset while setting the rest of the variables according to z. Since $\mathsf{s}(f) < \sqrt{\mathsf{D}(f)}$, $|\rho^{-1}(*)| = O(\mathsf{D}(f))$. At the same time, by construction, we also have $\mathsf{D}(f|_\rho) \geq \mathsf{bs}(f|_\rho) \geq \sqrt{\mathsf{D}(f)}$.

2 Preliminaries

Notations: We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

We recall the definitions of the complexity measures we will use throughout. We refer the reader to the survey [8] for an introduction to the complexity of Boolean functions and complexity measures. Several additional complexity measures and their relationships can also be found in [5] and [2].

In the following $f\colon \{0,1\}^n \to \{0,1\}$ is a Boolean function on n variables. For an input $x \in \{0,1\}^n$ and $i \in [n]$, let x^i represent the string in $\{0,1\}^n$ that is obtained from x by flipping (or, negating) the i-th bit. We say that the i-th bit is *sensitive* at x if $f(x) \neq f(x^i)$.

▶ **Definition 6** (Sensitivity). *The* sensitivity *of f on an input x, $\mathsf{s}(f,x)$, is defined as the number of sensitive bits at x with respect to f. Then, the* sensitivity *of f, denoted $\mathsf{s}(f)$, is defined as $\mathsf{s}(f) = \max\{\mathsf{s}(f,x) \mid x \in \{0,1\}^n\}$.*

We also define 0-sensitivity *of f as $\mathsf{s}_0(f) = \max\{\mathsf{s}(f,x) \mid x \in \{0,1\}^n, f(x) = 0\}$ and* 1-sensitivity *of f as $\mathsf{s}_1(f) = \max\{\mathsf{s}(f,x) \mid x \in \{0,1\}^n, f(x) = 1\}$.*

Similarly for $B \subseteq [n]$, let x^B denote the string obtained from x by flipping the bits indexed by the block B. We say that the block B is *sensitive* at x if $f(x) \neq f(x^B)$.

▶ **Definition 7** (Block Sensitivity). *The* block sensitivity *of a function f on an input x, $\mathsf{bs}(f,x)$, is the maximum number of disjoint subsets $B_1, B_2, \ldots, B_r$ of $[n]$ such that for all $j \in [r]$, $f(x) \neq f(x^{B_j})$. Then, the* block sensitivity *of f, denoted $\mathsf{bs}(f)$, is $\max\{\mathsf{bs}(f,x) \mid x \in \{0,1\}^n\}$. Similarly, define $\mathsf{bs}_0(f) = \max\{\mathsf{bs}(f,x) \mid x \in \{0,1\}^n, f(x) = 0\}$ and $\mathsf{bs}_1(f) = \max\{\mathsf{bs}(f,x) \mid x \in \{0,1\}^n, f(x) = 1\}$.*

A sensitive block at x is called *minimal* if no proper subset of it as a sensitive block at x.

We also need the following generalization of block sensitivity based on linear programming.

▶ **Definition 8** (Fractional Block Sensitivity). *Let* $\mathcal{W}(f,x) := \{B \subseteq [n] \mid f(x^B) \neq f(x)\}$ *denote the set of all sensitive blocks for an input* $x \in \{0,1\}^n$*. The* fractional block sensitivity *of* f *at* x*, denoted* $\mathsf{fbs}(f,x)$*, is the value of the linear program,*

$$\begin{aligned} \textit{maximize} \quad & \sum_{B \in \mathcal{W}(f,x)} y_B, \\ \textit{subject to} \quad & \sum_{B\colon i \in B} y_B \leq 1, \quad \forall\, i \in [n], \\ & 0 \leq y_B \leq 1, \quad \forall\, B \in \mathcal{W}(f,x). \end{aligned}$$

The fractional block sensitivity $\mathsf{fbs}(f)$ *of* f *is then defined as* $\mathsf{fbs}(f) := \max_{x \in \{0,1\}^n} \mathsf{fbs}(f,x)$.

A *partial assignment* is a function $\rho\colon \{0,1\}^n \to \{0,1,*\}$ where the size of ρ is $n - |\rho^{-1}(*)|$. Let $S \subseteq [n]$ denote the set of indices where ρ is not $*$. We say that an input $x \in \{0,1\}^n$ is *consistent* with ρ iff $x_i = \rho(i)$ for all $i \in S$. For $b \in \{0,1\}$, a *b-certificate* for the Boolean function f is a partial assignment ρ such that for all inputs x and y consistent with ρ we have $f(x) = f(y) = b$.

▶ **Definition 9** (Certificate Complexity). *The* certificate complexity *of a function* f *on an input* x*, denoted* $\mathsf{C}(x,f)$*, is the size of the smallest* $f(x)$*-certificate that is consistent with* x*. The* certificate complexity *of* f*, denoted* $\mathsf{C}(f)$*, is* $\max_x\{\mathsf{C}(f,x)\}$*. Further, define* $\mathsf{C}_0(f) = \max_{x\colon f(x)=0}\{\mathsf{C}(f,x)\}$ *and* $\mathsf{C}_1(f) = \max_{x\colon f(x)=1}\{\mathsf{C}(f,x)\}$.

We now note the following relation between these measures.

▶ **Lemma 10** ([26, 31]). $s(f,x) \leq \mathsf{bs}(f,x) \leq \mathsf{fbs}(f,x) \leq \mathsf{C}(f,x) \leq \mathsf{bs}(f,x)\mathsf{s}(f)$.

We also require the following generalization of certificate complexity.

▶ **Definition 11** (Unambiguous Certificate Complexity). *Fix* $b \in \{0,1\}$*. A set* U *of partial assignments is said to form an* unambiguous *collection of b-certificates for* f *if*

- *each partial assignment in* U *is a b-certificate for* f*,*
- *for each* $x \in f^{-1}(b)$ *there is some* $p \in U$ *that is consistent with* x*, and*
- *no two partial assignments in* U *are consistent.*

We then define $\mathsf{UC}_b(f)$ *to be the minimum value of* $\max_{p\in U}|p|$ *over all choices of such collections* U*. Further, define* $\mathsf{UC}(f) := \max\{\mathsf{UC}_0(f), \mathsf{UC}_1(f)\}$ *and* $\mathsf{UC}_{\min}(f) := \min\{\mathsf{UC}_0(f), \mathsf{UC}_1(f)\}$.

We now define the query model of computation. In this model, the input bits can be accessed by queries to an input oracle and the complexity of computing a Boolean function f is the number of queries made to the oracle.

▶ **Definition 12** (Deterministic Decision Trees). *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a Boolean function. Let* A *be a deterministic algorithm that computes* f *by making queries to the bits of an unknown input* x*. The deterministic decision tree complexity (also known as query complexity) of* f*, denoted* $\mathsf{D}(f)$*, is* $\min_A\{\max_x \mathit{cost}(A,x)\}$*, where the minimum is over all deterministic algorithm computing* f *and* $\mathit{cost}(A,x)$ *is the number of queries made by* A *on the input* x*.*

We note the following relations, which are easy to observe.

▶ **Fact 13.** $\deg(f) \leq \mathsf{UC}_{\min}(f) \leq \mathsf{UC}_1(f) \leq \mathsf{UC}(f) \leq \mathsf{D}(f)$*, and* $\mathsf{bs}(f) \leq \mathsf{C}(f) \leq \mathsf{UC}(f)$.

We also need the following variants of the deterministic decision tree.

▶ **Definition 14** (AND-Decision Trees). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let A be a deterministic algorithm that computes f by querying* AND *of a subset $S \subseteq [n]$ of the bits of an unknown input x. Further, let $cost(A,x)$ be the number of queries made by A on the input x, and $cost(A) = \max_x cost(A,x)$. Then, the* AND*-decision tree complexity of f, denoted $\mathsf{D}_\wedge(f)$, is $\min_A\{cost(A)\}$, where the minimum is over all deterministic* AND*-decision tree algorithm computing f.*

We can similarly define OR-decision trees, which are allowed to query OR of a subset of input variables, and OR-decision tree complexity ($\mathsf{D}_\vee(f)$). We also need the notion of decision tree algorithms that minimize the number of 0-queries (or, 1-queries) while computing f.

▶ **Definition 15** (0-Decision Trees). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let A be a deterministic algorithm that computes f by making queries to the bits of an unknown input x. Further, let $cost(A,x)$ be the number of queries made by A on the input x that are answered as 0, and $cost(A) = \max_x cost(A,x)$. Then, the deterministic 0-decision tree complexity of f, denoted $\mathsf{D}_0(f)$, is $\min_A\{cost(A)\}$, where the minimum is over all deterministic 0-decision tree algorithm computing f.*

Similarly, we can define deterministic 1-decision tree complexity ($\mathsf{D}_1(f)$). The following relations between these variants (see also [23]) are easy to observe.

▶ **Fact 16.** *$\mathsf{D}_0(f) \le \mathsf{D}_\wedge(f) \le \mathsf{D}(f)$, and $\mathsf{D}_1(f) \le \mathsf{D}_\vee(f) \le \mathsf{D}(f)$.*

For the definitions of well-studied complexity measures, such as randomized decision trees (R), zero-error randomized decision tree (R_0) , exact quantum query complexity (Q_E) and bounded error quantum query complexity (Q), we refer the reader to [8].

We note the following relations that are easy to observe.

▶ **Fact 17.** *$\mathsf{Q}(f) \le \mathsf{R}(f) \le \mathsf{R}_0(f) \le \mathsf{D}(f)$, and $\mathsf{Q}(f) \le \mathsf{Q}_E(f) \le \mathsf{D}(f)$.*

A real multilinear polynomial $p(x_1,\ldots,x_n)$ is said to *represent* a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, if $p(x) = f(x)$ for all $x \in \{0,1\}^n$. It is easily seen that a Boolean function f has a unique multilinear polynomial p representing it. Let $\deg(f)$ denote the degree of this unique multilinear polynomial representing f. We also need the following notion of approximating polynomials.

▶ **Definition 18.** *We say that a polynomial p* approximates *a Boolean function f, if $|p(x) - f(x)| \le 1/3$ for all $x \in \{0,1\}^n$. Let $\widetilde{\deg}(f)$ denote the minimal possible degree of a real polynomial approximating f.*

We again note relations that are easy to observe.

▶ **Fact 19.** *$\widetilde{\deg}(f) \le \deg(f) \le \mathsf{D}(f)$, and $\widetilde{\deg}(f) \le \mathsf{R}(f)$.*

We also need the following measure that played the key role in the proof of the sensitivity conjecture [20].

▶ **Definition 20.** *The* sensitivity graph *of f, $G_f = (V,E)$, is the graph where $V = \{0,1\}^n$ and $E = \{(x,x^i) \mid f(x) \ne f(x^i), i \in [n], x \in V\}$. Let A_f be the adjacency matrix of the graph G_f. Then, the* spectral sensitivity *of f, denoted $\lambda(f)$, is the spectral norm $\|A_f\|$ of the adjacency matrix A_f.*

We also describe the (disjoint) composition of two Boolean functions.

▶ **Definition 21.** *For any two Boolean functions* $f\colon \{0,1\}^n \to \{0,1\}$ *and* $g\colon \{0,1\}^m \to \{0,1\}$, *define the* composed *function* $f \circ g\colon \{0,1\}^{nm} \to \{0,1\}$ *as follows*

$$f \circ g(x_{11}, \ldots, x_{1m}, \ldots\ldots, x_{n1}, \ldots, x_{nm}) = f(g(x_1), \ldots, g(x_n)),$$

where $x_i = (x_{i1}, \ldots, x_{im}) \in \{0,1\}^m$ *for* $i \in [n]$.

▶ **Definition 22** (Rubinstein's function [30]). *Let* $g\colon \{0,1\}^k \to \{0,1\}$ *be such that* $g(x) = 1$ *iff* x *contains two consecutive ones and the rest of the bits are zeroes. The Rubinstein's function, denoted* $\mathsf{RUB}\colon \{0,1\}^{k^2} \to \{0,1\}$, *is defined to be* $\mathsf{RUB} = \mathsf{OR}_k \circ g$.

We also define a modified version of the Rubinstein function, which will be the witness function for our negative results.

▶ **Definition 23** (Modified Rubinstein function). *Define* $g\colon \{0,1\}^k \to \{0,1\}$ *to be* 1 *iff the input contains* $\sqrt{k}$ *consecutive ones and the rest of the bits are zeroes. Then, define* $f\colon \{0,1\}^{k^2} \to \{0,1\}$ *to be the function* $(\mathsf{OR}_k \circ g)(x)$ *where* $x \in \{0,1\}^{k^2}$.

Our negative results for some of the query measures require cheatsheet functions.

▶ **Definition 24** (Cheat-sheet version of a function). *Fix* $t > 0$. *Let* $f\colon \{0,1\}^n \to \{0,1\}$ *be a function such that the* description *of any (minimal) certificate takes at most* m *bits. We then define*

$$f_{\mathrm{CS}}\colon \{0,1\}^{n \log t + mt \log t} \to \{0,1\}$$

the cheat-sheet version of f *as follows. The input to* f_{CS} *is* $X = (x_1, \ldots, x_{\log t}, Y_0, \ldots, Y_{t-1})$ *where* $x_i \in \{0,1\}^n$ *are inputs to* $\log t$ *copies of* f, *and* $Y_j \in \{0,1\}^{m \log t}$ *are cheat-sheet cells.*

Define $f_{\mathrm{CS}}(X) = 1$ *iff the cheat-sheet cell* Y_ℓ, *given by the positive integer* ℓ *corresponding to the binary string* $f(x_1)f(x_2)\cdots f(x_{\log t})$, *describes* $\log t$ *certificates, one for each* x_i *using* m *bits. That is, it contains a* description *of* $f(x_i)$*-certificate for each* x_i.

For more details on the cheat-sheet framework, we refer to [1, 4].

3 Negative Results I: Incondensability of Block Sensitivity and Certificate Complexity

In this section, we prove Theorem 1, i.e., block sensitivity, fractional block sensitivity and certificate complexity do not condense losslessly.

▶ **Theorem 1.** *There exists a Boolean function* $f\colon \{0,1\}^n \to \{0,1\}$ *with* $\mathsf{bs}(f) = \mathsf{fbs}(f) = \mathsf{C}(f) = n^{3/4}$, *such that for every restriction* $\rho\colon [n] \to \{0,1,*\}$ *with* $|\rho^{-1}(*)| = O(n^{3/4})$ *we have* $\mathsf{bs}(f|_\rho) \le \mathsf{fbs}(f|_\rho) \le \mathsf{C}(f|_\rho) = O(\sqrt{n}) = O(\mathsf{bs}(f)^{2/3})$.

In fact, we will use the same function, Modified Rubinstein function (Definition 23), for all three measures. More formally, we establish the following bounds for its complexity measures.

▶ **Theorem 25.** *There exists a Boolean function* $f\colon \{0,1\}^{k^2} \to \{0,1\}$ *with* $\mathsf{bs}(f) = \mathsf{fbs}(f) = \mathsf{C}(f) = k^{1.5}$, *such that for every partial assignment* $\rho\colon [k^2] \to \{0,1,*\}$ *with* $|\rho^{-1}(*)| = O(k^{1.5})$ *we have* $\mathsf{bs}(f|_\rho) \le \mathsf{fbs}(f|_\rho) \le \mathsf{C}(f|_\rho) = O(k)$.

Proof. Since for every f and input x, $\mathsf{bs}(f,x) \le \mathsf{fbs}(f,x) \le \mathsf{C}(f,x)$, it suffices to show the following two properties hold,

- **(i)** $\mathsf{bs}(f) = \mathsf{C}(f) = k^{1.5}$, and
- **(ii)** for every partial assignment $\rho\colon [k^2] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(k^{1.5})$ we have $\mathsf{C}(f|_\rho) = O(k)$.

Let f be the modified Rubinstein function as given in Definition 23. Then, Item i follows from Lemma 26 and Item ii is shown in Lemma 27. Thus, the theorem follows. ◀

We now prove the two lemmas to complete the proof.

▶ **Lemma 26.** *Let f be the function defined in Definition 23. Then, $\mathsf{C}_0(f) = k^{1.5}$, $\mathsf{bs}_0(f) = k^{1.5}$, $\mathsf{bs}_1(f) = k$, and $\mathsf{C}_1(f) = k$.*

Proof. Recall the k^2 variate function $f = \mathsf{OR}_k \circ g$, where $g\colon \{0,1\}^k \to \{0,1\}$ evaluates to 1 iff the input to g contains $\sqrt{k}$ consecutive ones and the rest of the bits are zeroes. Let $x = (x_{11}, x_{12}, \ldots, x_{1k}, x_{21}, \ldots, x_{2k}, \ldots, x_{k1}, \ldots, x_{kk})$ be an input to f. For $1 \le i \le k$, the i-th copy of g is defined over the set of variables $\{x_{i1}, \ldots, x_{ik}\}$.

Clearly, $\mathsf{C}_1(f) = \mathsf{C}_1(g) \le k$ and $\mathsf{bs}_1(f) = \mathsf{bs}_1(g) \ge k$, which in turn implies that both are equal to k. Note a 0-certificate for f is a collection of 0-certificates, one for each copy of g. Thus, $\mathsf{C}_0(f) \le k \cdot \mathsf{C}_0(g)$. We now upper bound $\mathsf{C}_0(g)$. A 0-input $y \in \{0,1\}^k$ of g falls under (at least) one of the following types:

(1) It contains two 1-bits that are at least $\sqrt{k}$ apart. That is, there exists $i, j \in [k]$ such that $y_i = y_j = 1$ and $|i - j| \ge \sqrt{k}$. For such 0-inputs, the certificate complexity is 2.

(2) It contains three input bits y_i, y_j, y_ℓ such that $i < j < \ell$ and $y_i = 1, y_j = 0$, and $y_\ell = 1$. For such 0-inputs, the certificate complexity is at most 3.

(3) It either (I) contains the substring $01^\ell 0$, or (II) equals to $0^{k-\ell}1^\ell$ or $1^\ell 0^{k-\ell}$ for some ℓ such that $1 \le \ell < \sqrt{k}$. For 0-inputs of kind (I), the certificate complexity is at most 3, namely the first two bits (01) and the last bit (0) of the substring suffice as a 0-certificate. For 0-inputs of kind (II), just the two bits on the boundary suffices, $y_{k-\ell} = 0$ and $y_{k-\ell+1} = 1$ (or, $y_\ell = 1$ and $y_{\ell+1} = 0$). Therefore, the certificate complexity is again at most 3.

(4) The input $y = 0^k$. The certificate complexity for this input is $\sqrt{k}$.

We therefore have $\mathsf{C}_0(f) \le k \cdot \mathsf{C}_0(g) \le k^{1.5}$.

It is now sufficient to show that $\mathsf{bs}_0(f) \ge k^{1.5}$. Consider the input 0^{k^2}. It is easily seen that $\mathsf{bs}(f, 0^{k^2}) = k \cdot \mathsf{bs}(g, 0^k) = k\sqrt{k}$. ◀

▶ **Lemma 27.** *Let f be the function defined in Definition 23. Then, for every partial assignment $\rho\colon [k^2] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(k^{1.5})$, we have $\mathsf{C}(f|_\rho) = O(k)$.*

Proof. Recall $f = \mathsf{OR}_k \circ g$, where $g\colon \{0,1\}^k \to \{0,1\}$ evaluates to 1 iff the input to g contains $\sqrt{k}$ consecutive ones and the rest of the bits are zeroes. Let $x = (x_{11}, \ldots, x_{1k}, \ldots, x_{k1}, \ldots, x_{kk})$ be an input to f. For $1 \le i \le k$, let $g_i = g(x_{i1}, \ldots, x_{ik})$ be the i-th copy of g.

Since $\mathsf{C}_1(f) = k$, it suffices to show $\mathsf{C}_0(f|_\rho) = O(k)$ for every $\rho\colon [k^2] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(k^{1.5})$.

A 0-certificate for $f|_\rho$ comprises of at most k 0-certificates, one for each copy of $g|_\rho$ that remains non-zero. That is, $\mathsf{C}_0(f|_\rho) \le \sum_{i=1}^{k} \mathsf{C}_0(g_i|_\rho)$. We therefore now bound $\mathsf{C}_0(g_i|_\rho)$ when $g_i|_\rho$ is not equal to the constant function 0. We assume, wlog, that $g_1|_\rho$ is non-constant. Based on the restriction ρ, we have two cases to consider.

Case 1: ρ sets some variable of g_1 to 1. That is, there exists $1 \le j \le k$, such that $\rho(x_{1j}) = 1$. In this case, $\mathsf{C}_0(g_1|_\rho) \le 3$.

This is essentially because we only need to certify that the sequence of 1s in a 0-input to $g_1|_\rho$ does not satisfy the required property. In particular, any 0-input of $g_1|_\rho$ is of type (1), (2), or (3).

Case 2: ρ sets no variable of g_1 to 1. That is, ρ may set some variables in $\{x_{11}, \ldots, x_{1k}\}$ to 0 and leaves the rest unset. Let η_1 be the number of variables of $g_1|_\rho$. That is, $|\rho^{-1}(*) \cap \{x_{11}, \ldots, x_{1k}\}| = \eta_1$.

In such a case, observe that the all-zero input to $g_1|_\rho$ remains the hardest to certify. Thus, $\mathsf{C}_0(g_1|_\rho) = \mathsf{C}(g_1|_\rho, 0^{\eta_1}) \leq \eta_1/\sqrt{k}$. The inequality follows because there is a certificate that reveals a 0-bit for every consecutive $\sqrt{k}$ positions and, hence its size is at most $\eta_1/\sqrt{k}$.

Let $\eta_i = |\rho^{-1}(*) \cap \{x_{i1}, \ldots, x_{ik}\}|$ be the number of variables of $g_i|_\rho$ for $1 \leq i \leq k$. We are now ready to bound $\mathsf{C}_0(f|_\rho)$. Let t be the number of g_i's such that $g_i|_\rho$ falls in **Case 2**. Further assume, wlog, that these are $\{g_1, g_2, \ldots, g_t\}$. Then, we have

$$\begin{aligned}\mathsf{C}_0(f|_\rho) &\leq \sum_{i=1}^{k} \mathsf{C}_0(g_i|_\rho) \leq 3(k-t) + \sum_{i=1}^{t} \mathsf{C}_0(g_i|_\rho) \leq 3(k-t) + \sum_{i=1}^{t} \left(\eta_i/\sqrt{k}\right) \\ &\leq 3(k-t) + |\rho^{-1}(*)|/\sqrt{k} = O(k).\end{aligned}$$

◀

▶ Remark. We note that the modified Rubinstein function, even with different parameters, cannot give a better counter-example for lossless condensation than Theorem 25. For a proof, please check the full version.

4 Negative Results II: Incondensability of Decision Tree Complexity

In this section, we prove Theorem 2, i.e., query complexity and its variants do not condense losslessly. In particular, we show that D_0-query and $\mathsf{D}_\wedge$-query complexities do not condense losslessly and also present an improved example for the fact that decision tree query complexity does not condense losslessly. In fact, we will use the same function to establish all three claims.

▶ **Theorem 2.** *There exists a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with $\tilde{\Omega}(n^{3/5}) = \mathsf{D}_0(f) \leq \mathsf{D}_\wedge(f) \leq \mathsf{D}(f) = O(n^{3/5})$, such that for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(n^{3/5})$ we have $\mathsf{D}_0(f|_\rho) \leq \mathsf{D}_\wedge(f|_\rho) \leq \mathsf{D}(f|_\rho) = \tilde{O}(n^{2/5}) = \tilde{O}(\mathsf{D}_0(f)^{2/3})$.*

It was shown in [16] that there exists a function h such that for every restriction ρ that fixes all but at most $O(\mathsf{D}(h))$ variables, the query complexity $\mathsf{D}(h|_\rho)$ of the restricted function $h|_\rho$ is $\widetilde{O}(\mathsf{D}(h)^{3/4})$. We give an improved example by exhibiting a function h' such that $\mathsf{D}(h'|_\rho) = \widetilde{O}(\mathsf{D}(h')^{2/3})$. Our improved example is a variant of the function considered in [16]. In particular, our function is also a cheat-sheet version of Tribes (see Definition 24 for cheatsheet functions), but we use different arities for the basic Tribes function as well as keep the number of cheat-sheet cells much lower. Furthermore, we will use the same function to show that D_0 and $\mathsf{D}_\wedge$ also do not condense losslessly.

We consider the following Tribes function as the base function.

▶ **Definition 28** (Tribes function)**.** *Define $f\colon \{0,1\}^{k^{1.5}} \to \{0,1\}$ to be the function $(\mathsf{AND}_k \circ \mathsf{OR}_{\sqrt{k}})(x)$ where $x \in \{0,1\}^{k^{1.5}}$.*

We note a few well-known properties of the Tribes function.

▶ **Proposition 29.** *For the Tribes function f as defined in Definition 28, it is known that $\mathsf{C}_0(f) = \sqrt{k}$, $\mathsf{C}_1(f) = k$, $\mathsf{D}(f) = k\sqrt{k}$ and $\mathsf{D}_0(f) \geq k\sqrt{k} - k + 1$.*

Proof. Though the bounds on C_0, C_1, and D are well-known, the bound on D_0 is not often encountered. We sketch a proof here for completeness.

Consider the following (usual) adversary algorithm for the Tribes: On a query by a decision tree algorithm, the adversary responds with 0 if the OR containing the queried variable has variables that have not been queried yet. Otherwise it's the last variable to be queried in that OR, in such a case the adversary responds with 1 if there are other ORs whose output is not yet fixed. Finally, the last variable in the last OR is queried, which implies the lower bound. ◀

We consider the *cheat-sheet* version f_{CS} of the Tribes function f (Definition 28) where certificates in the cheat-sheet cells are described as follows. A cell Y_ℓ, where ℓ in binary is given by $\ell_1\ell_2\cdots\ell_{\log t} \in \{0,1\}^{\log t}$, describes a ℓ_i-certificate for x_i with respect to f for $i \in [\log t]$.

Describing 1-certificates: A 1-certificate for $f = \mathsf{AND}_k \circ \mathsf{OR}_{\sqrt{k}}$ contains a 1-certificate for each copy of OR. For each 1-certificate, the description is given by the label of a variable which is set to 1. This requires $\log k^{1.5}$ bits per OR. Therefore, the full description of a 1-certificate for f requires $k(\log k^{1.5})$ bits.

Describing 0-certificates: A 0-certificate for $f = \mathsf{AND}_k \circ \mathsf{OR}_{\sqrt{k}}$ contains a 0-certificate for one of the copies of OR. Hence, we describe a 0-certificate for f by just writing the label of a copy of OR. Thus, the full description of a 0-certificate for f requires $\log k$ bits.

Therefore, the cheat-sheet version f_{CS} of the Tribes function is

$$f_{\mathrm{CS}} \colon \{0,1\}^{k^{1.5}(\log t)+(t\log t)k(\log k^{1.5})} \to \{0,1\}, \tag{1}$$

where its input is $X = (x_1,\ldots,x_{\log t}, Y_0,\ldots,Y_{t-1})$, and $f_{\mathrm{CS}}(X) = 1$ iff the cheat-sheet cell $Y_\ell \in \{0,1\}^{(\log t)k(\log k^{1.5})}$, given by the positive integer ℓ corresponding to the binary string $f(x_1)f(x_2)\cdots f(x_{\log t})$, describes $f(x_i)$-certificate for each x_i as given by the description above. Note that 0-certificates are much smaller than 1-certificates, but we will use the same amount of space, $k(\log k^{1.5})$, for each, with the notation that in the case of 0-certificates first $\log k$ bits describe them.

We now show that for the choice of $t = k^{1.5}$, both $\mathsf{D}(f_{\mathrm{CS}})$ and $\mathsf{D}_0(f_{\mathrm{CS}})$ equals $\widetilde{\Theta}(k^{1.5})$, but for every restriction ρ that fixes all but at most $O(\mathsf{D}(f_{\mathrm{CS}}))$ variables, the query complexity $\mathsf{D}(f_{\mathrm{CS}}|_\rho)$ (and hence $\mathsf{D}_0(f_{\mathrm{CS}}|_\rho)$) of the restricted function $f_{\mathrm{CS}}|_\rho$ is $\widetilde{O}(k)$. We now prove each of these assertions one by one. In the following fix $t = k^{1.5}$. Then the number of variables for f_{CS} is $\Theta(k^{2.5}\log^2 k)$.

▶ **Lemma 30.** $\mathsf{D}(f_{\mathrm{CS}}) \geq t = k^{1.5}$ *and* $\mathsf{D}_0(f_{\mathrm{CS}}) \geq t - t^{2/3}$.

Proof. We will give an adversary argument for f_{CS} based on the adversary for f mentioned in Proposition 29.

On a query, the adversary for f_{CS} will respond as follows: If the queried variable is from the address part, i.e., $x_1,\ldots,x_{\log t}$, then the response is in accordance with the adversary of f given in Proposition 29. Otherwise, the queried variable is from the cheat-sheet cells, i.e., $Y_0,\ldots,Y_{t-1}$, and then the adversary responds with 0.

Now if an algorithm makes only $t-1$ queries and they are answered according to the adversary defined above, then the following observations hold:

- No $f(x_i)$ has been fixed yet,
- there is at least one cheat-sheet cell, say Y_ℓ, which has not been queried at all, and
- the number of 0-queries made until now is at least $t-1-k$.

We can therefore extend x_i's in a way that the address points to the cell Y_ℓ and, furthermore, the cell Y_ℓ can be set as we wish to give values 0 or 1 to contradict the algorithm's answer; thus finishing the adversary argument. ◀

We now present a claim that will be used to construct a query algorithm for f_{CS} as well as its restricted version.

▷ Claim 31. Given two distinct indices $\ell, p \in \{0,\ldots,t-1\}$, at least one of the cells Y_ℓ and Y_p can be discarded with $O(\log k)$ queries.

Proof. Since $\ell \neq p$, there exists $i \in [\log t]$ such that $\ell_i \neq p_i$. Assume, wlog, $\ell_i = 0$ and $p_i = 1$.

The algorithm queries Y_ℓ to know the label of OR, which is a supposed 0-certificate for $f(x_i)$. Let OR_j, $j \in [k]$, be the copy of OR within the i-th copy of f that Y_ℓ has listed as 0-certificate for x_i. Observe that Y_p is supposed to have a 1-certificate for OR_j. Recall a 1-certificate for OR_j is the label of a variable of OR_j. Query the variables in Y_p and the input x_i to verify the 1-certificate for OR_j. Now, if the claimed 1-certificate for OR_j is valid, then we can remove Y_ℓ; else, Y_p can be removed.

The total number of queries made is at most $O(\log k)$. ◁

We are now ready to show a nearly tight upper bound on $\mathsf{D}(f_{\mathrm{CS}})$. Using Claim 31, we prove that the bound of Lemma 30 is tight up to logarithmic factors. By a straightforward algorithm that solves each copy of f and then verifies the contents of the cheat-sheet cell pointed to by the evaluations of f, we get the following. In the full version of the paper, we give another algorithm for f_{CS} that is instructive when designing an algorithm for the restricted version.

▶ **Lemma 32.** $\mathsf{D}(f_{\mathrm{CS}}) = O(t \log k \ + \ k \log k \log t)$.

We now finish the last, but the most important, task of showing that indeed the decision tree complexity of the restricted function reduces; thereby finishing the counterexample for the hardness condensation of decision tree complexity and its variants.

▶ **Lemma 33.** *For every partial assignment ρ to the variables of f_{CS} with $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}})) = O(k^{1.5} \log k)$, we have $\mathsf{D}(f_{\mathrm{CS}}|_\rho) = O(k \log^2 k)$.*

Proof. Fix a partial assignment ρ with $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}})) = O(k^{1.5} \log k)$. Our query algorithm for $f_{\mathrm{CS}}|_\rho$ on an unknown input $X|_\rho = (x_1|_\rho, \ldots, x_{\log t}|_\rho, Y_0|_\rho, \ldots, Y_{t-1}|_\rho)$ has two parts like the unrestricted case.

In the first part we identify a cheat-sheet cell (to be verified later) by zeroing in on the string $z \in \{0,1\}^{\log t}$ that the address part $f(x_1|_\rho)f(x_2|_\rho)\cdots f(x_{\log t}|_\rho)$ can take if the restricted function $f_{\mathrm{CS}}|_\rho$ were to evaluate to 1.

In the second part, like the unrestricted case, we verify the certificates described in Y_z by querying the variables in $Y_z|_\rho$ and the corresponding certificate variables in $(x_1|_\rho, \ldots, x_{\log t}|_\rho)$. If the verification passes, the algorithm outputs 1, else it outputs 0. Since the algorithm needs to verify the contents of one cell, the total number of queries made in the second part is $O(k \log k \log t + \mathsf{C}(f) \log t) = O(k \log^2 k)$.

Algorithm for the first part. We now give the details of our algorithm for the first part. Recall that the address part of the input contains $\log t$ copies of the base function f. We will run (at most) $\log t$ many iterations of the algorithm. At the end of each iteration i, we will have a candidate value for $f(x_i|_\rho)$, i.e., either we know the value of $f(x_i|_\rho)$ or there are no valid cheat-sheet cells for $f(x_i|_\rho) = 1$. At the end of the $\log t$ iterations, there will only be one remaining (possibly valid) cheat-sheet cell to be verified. We now give details of an iteration of the algorithm.

Let c be a constant such that $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}})) \leq ck^{1.5} \log k$.

Details of a particular iteration. Fix an $i \in [\log t]$. At any given point of the iteration let $\mathcal{V}$ be the set of "alive" cheat-sheets (cheat-sheets which have not been found inconsistent) and $\mathcal{V}_1$ be the cheat-sheet cells in $\mathcal{V}$ where, supposedly, $f(x_i|_\rho) = 1$, (i.e., the i-th bit of the index of the cell is 1). On the other hand, let $\mathcal{O}_0$ denote the set of OR_j's ($j \in [k]$) in the i-th copy of f which could still possibly be 0. Initially, $|\mathcal{O}_0| = k$ and, in the first iteration, $|\mathcal{V}| = t$ and $|\mathcal{V}_1| = t/2$. Also, note that each cell in $\mathcal{V}_1$ is supposed to contain a 1-certificate for k many ORs.

In each iteration, the algorithm will have two stages, where the first stage (and arguably the more involved one) will get us to a situation that can be handled by previous techniques. More formally, after the first stage, either $|\mathcal{O}_0| \leq c\sqrt{k}\log k$ or $|\mathcal{V}_1| \leq 2k$.

The second stage addresses these two cases. In the first case, we can check all the alive OR's exhaustively. For the second case, using Claim 31, we will get a single cheat-sheet cell in $|\mathcal{V}_1|$ which is then verified to ascertain whether indeed it contains a 1-certificate for $f(x_i|_\rho)$.

We now describe the two stages in detail.

Stage 1: We can assume that $|\mathcal{O}_0| > c\sqrt{k}\log k$ and $|\mathcal{V}_1| > 2k$, otherwise we move to the second stage.

Looking at $\mathcal{V}_1$, for each $\mathsf{OR}_j \in \mathcal{O}_0$, we call an OR_j to be *revealed* if (supposed) 1-certificates for that particular OR_j is completely fixed by ρ (i.e., not a single $*$) in at least $|\mathcal{V}_1|/2$ many cells in $\mathcal{V}_1$. Notice that if there are no revealed OR's, then for more than $c\sqrt{k}\log k$ many OR's there are at least $|\mathcal{V}_1|/2 > k$ many $*$'s. This is a contradiction because the number of $*$'s, by our assumption, is at most $ck^{1.5}\log k$.

For a revealed j, there exists a *popular* variable, say $l \in [\sqrt{k}]$, which appears in at least $|\mathcal{V}_1|/2\sqrt{k}$ many cells in $\mathcal{V}_1$. We query $x_{i,j,l}$ from the input part x_i. If it is 1 then OR_j can be discarded from $\mathcal{O}_0$, otherwise it is 0 and at least $|\mathcal{V}_1|/2\sqrt{k}$ many cells in $\mathcal{V}_1$ can be thrown out.

Essentially, the first stage of the iteration can be described as: find a revealed OR and query the popular variable till $|\mathcal{O}_0| \leq c\sqrt{k}\log k$ or $|\mathcal{V}_1| \leq 2k$. The existence of such a revealed OR follows from the argument described above. We now bound the query complexity in this stage.

Notice that after every query either $|\mathcal{O}_0|$ reduces by 1 (popular variable being 1) or $|\mathcal{V}_1|$ reduces by a factor of $(1 - \frac{1}{2\sqrt{k}})$ (popular variable being 0). So the query complexity of the first stage is bounded by $O(k + \sqrt{k}\log t) = O(k)$.

Stage 2: We know that either $|\mathcal{O}_0| \leq c\sqrt{k}\log k$ or $|\mathcal{V}_1| \leq 2k$. As alluded to before, for the first case, we can query all the OR's in $\mathcal{O}_0$ and find the value of $f(x_i|_\rho)$. Since the arity of each OR is $\sqrt{k}$, this requires $O(k\log k)$ queries.

For the second case, use Claim 31 to reduce the size of $\mathcal{V}_1$ to one using $O(k\log k)$ many queries. Then, in the remaining cell, the claimed 1-certificate for $f(x_i|_\rho)$ can be verified by reading the entire certificate in $O(k\log k)$ queries. In other words, the second stage makes $O(k\log k)$ many queries.

Hence, the total number of queries made in both stages of the iteration is $O(k\log k)$. This finishes one iteration $i \in [\log t]$ of the algorithm.

Query complexity of the entire algorithm: Since there are $\log t$ many iterations, the total queries made in the first part of the algorithm are $O(k\log^2 k)$. After these $\log t$ iterations, in the second part, there is at most one possible valid cheat-sheet, which can be checked in $O(k\log^2 k)$ many queries (as described in the beginning). Therefore, the entire algorithm makes at most $O(k\log^2 k)$ queries.

Correctness: The correctness of the algorithm follows from the fact that a cheat-sheet cell is removed from the set of valid cells if and only if the cell is found to be inconsistent with respect to input $(x_1|_\rho, \ldots, x_{\log t}|_\rho)$ and f, or when we know the value of $f(x_i)$. Further, the only remaining valid cell is verified in the second part. ◀

As a corollary to Lemma 30, Lemma 32, Lemma 33 and Fact 16, we obtain an improved example for the incondensability of query complexity and also show incondensability of D_0 and $\mathsf{D}_\wedge$.

▶ **Theorem 34.** *Let f_{CS} be the Boolean function on $\Theta(k^{2.5}\log^2 k)$ variables as defined in Equation (1). Then, the following holds*

- $\Omega(k^{1.5}) = \mathsf{D}_0(f_{\mathrm{CS}}) \le \mathsf{D}_\wedge(f_{\mathrm{CS}}) \le \mathsf{D}(f_{\mathrm{CS}}) = O(k^{1.5}\log k)$, *and*
- *for all restrictions ρ with $|\rho^{-1}(*)| = O(\mathsf{D}(f_{\mathrm{CS}}))$, we have $\mathsf{D}_0(f_{\mathrm{CS}}|_\rho) \le \mathsf{D}_\wedge(f_{\mathrm{CS}}|_\rho) \le \mathsf{D}(f_{\mathrm{CS}}|_\rho) = O(k\log^2 k) = \widetilde{O}(\mathsf{D}_0(f_{\mathrm{CS}})^{2/3})$.*

This completes the proof of Theorem 2. We take a moment to note that if we consider the function $h = \mathsf{OR}_k \circ \mathsf{AND}_{\sqrt{k}}$ and the cheat-sheet version h_{CS} of h, a dually similar lower bound and algorithm for h_{CS} and its restricted version exists, which implies the following theorem on the incondensability of OR-decision tree complexity.

▶ **Theorem 3.** *There exists a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with $\tilde{\Omega}(n^{3/5}) = \mathsf{D}_1(f) \le \mathsf{D}_\vee(f) = O(n^{3/5})$, such that for every restriction $\rho\colon [n] \to \{0,1,*\}$ with $|\rho^{-1}(*)| = O(n^{3/5})$ we have $\mathsf{D}_1(f|_\rho) \le \mathsf{D}_\vee(f|_\rho) = \tilde{O}(n^{2/5}) = \tilde{O}(\mathsf{D}_1(f)^{2/3})$.*

We further observe that the zero-error randomized query complexity $\mathsf{R}_0(f_{\mathrm{CS}})$ of f_{CS} is also high, i.e., $\Omega(k^{1.5})$, thus obtaining an improved example for the incondensability of R_0 (Corollary 4).

▶ **Theorem 35.** *Let f_{CS} be the Boolean function on $\Theta(k^{2.5}\log^2 k)$ variables as defined in Equation (1). Then, the following holds*

- $\mathsf{R}_0(f_{\mathrm{CS}}) = \tilde{\Theta}(k^{1.5})$, *and*
- *for all restrictions ρ with $|\rho^{-1}(*)| = O(\mathsf{R}_0(f_{\mathrm{CS}}))$, we have $\mathsf{R}_0(f_{\mathrm{CS}}|_\rho) = O(k\log^2 k) = \widetilde{O}(\mathsf{R}_0(f_{\mathrm{CS}})^{2/3})$.*

Proof. Since $\mathsf{R}_0 \le \mathsf{D}$, it suffices to show that $\mathsf{R}_0(f_{\mathrm{CS}}) = \Omega(k^{1.5})$ (Theorem 34 will then complete the proof). To prove the lower bound on $\mathsf{R}_0(f_{\mathrm{CS}})$, note that $\mathsf{R}_0(f) = \Omega(\mathsf{R}(f))$ and $\mathsf{R}(f) = \Theta(k^{1.5})$ [21]. See also [15, 4, 11]. We will show a reduction from f_{CS} to f.

Given an input x for f, the input of f_{CS} (say y) will consist of $\log t$ copies of x in the address part and all 0's in the cheat-sheet part. Whenever the R_0 algorithm for f_{CS} answers correctly (i.e., does not say "don't know"), it has queried a certificate for y (say C_y). Note that the number of cheat-sheets is more than the allowed number of queries to the R_0 algorithm. This implies that the certificate C_y hasn't queried at least one cheat-sheet cell at all.

We claim that this certificate C_y (that hasn't queried any element of a cheat-sheet cell) has a certificate for at least one address bit. To prove the claim by contradiction, if C_y does not have a certificate for x (i.e., does not certify any of the address bits), the address bits of y could be flipped (keeping it consistent with C_y) to point to the cheat-sheet cell which has not been queried. Since C_y does not contain any element of this cheatsheet cell, but an input consistent with C_y could point to this cheatsheet cell, we have a contradiction.

To summarize, the R_0 algorithm for f_{CS} gives a certificate for y which contains a certificate for at least one address bit. Since the certificate for any address bit is a certificate for x, this will finish the R_0 algorithm for f. ◀

5 Positive Results: Lossy Condensation

It is a natural question to ask if the gaps in Theorem 1 and Theorem 2 are optimal. In other words, does there exist a function for which hardness is even less condensable than shown in these theorems?

In particular, for deterministic query complexity, it asks if there exists a function f such that for all restrictions ρ with $|\rho^{-1}(*)| = O(\mathsf{D}(f))$, we have $\mathsf{D}(f|_\rho) = O(\mathsf{D}(f)^\alpha)$ where $\alpha < 2/3$? Observe that $\alpha \geq 1/3$, since $\mathsf{D}(f) \leq \deg(f)^3$ [25] and $\deg(f)$ condenses losslessly. Note that if the improved bound $\mathsf{D}(f) \leq \deg(f)^2$ were known to hold, then we would have a better lower bound for the exponent.

Similarly, for block sensitivity, the exponent $\alpha \geq 1/4$, since $\mathsf{bs}(f) \leq \mathsf{s}(f)^4$ [20] and $\mathsf{s}(f)$ condenses losslessly. Note again that if the improved bound $\mathsf{bs}(f) \leq \mathsf{s}(f)^2$ were known to hold, then we would have a better lower bound for the exponent.

In this section, we will establish an improved lower bound for the exponent for almost all decision tree-based complexity measures.

▶ **Theorem 5.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function and $\mathcal{M} \in \{\mathsf{bs}, \mathsf{fbs}, \mathsf{C}, \mathsf{UC}_{\min}, \mathsf{UC}_1, \mathsf{UC}, \mathsf{D}, \widetilde{\deg}, \lambda, \mathsf{R}, \mathsf{R}_0, \mathsf{Q}, \mathsf{Q}_E\}$ be a complexity measure. Then, the following holds:*

(a) *For $\mathcal{M} \notin \{\mathsf{R}, \mathsf{R}_0, \mathsf{Q}, \mathsf{Q}_E\}$, there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ such that $\mathcal{M}(f|_\rho) = \Omega(\sqrt{\mathcal{M}(f)})$.*

(b) *For $\mathcal{M} \in \{\mathsf{R}, \mathsf{R}_0, \mathsf{Q}_E\}$, there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = \Theta(\mathcal{M}(f))$ such that $\mathcal{M}(f|_\rho) = \Omega(\mathcal{M}(f)^{1/3})$.*

(c) *For $\mathcal{M} = \mathsf{Q}$, there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = \Theta(\mathsf{Q}(f))$ such that $\mathsf{Q}(f|_\rho) = \Omega(\mathsf{Q}(f)^{1/4})$.*

The proof for Theorem 5 is obtained by noting that the two measures, sensitivity and degree, condense losslessly; and then exploiting known relationships between different complexity measures. The technical details (and complete proofs) are given in the full version.

The proof strategy outlined above works for almost all measures except Q, Q_E, $\widetilde{\deg}$ and λ. The best bounds for these measures are obtained by noting that the two measures, sensitivity and degree, indeed follow a slightly stricter form of hardness condensation. We present this stronger hardness condensation here (It shows that sensitivity and degree can be condensed to any parameter below itself). Again, the application of this stricter form (hardness condensation for Q, Q_E, $\widetilde{\deg}$ and λ) can be seen in the full version.

▶ **Lemma 36** (Stronger hardness condensation)**.** *Let $\mathcal{M} \in \{\mathsf{s}, \deg\}$ and $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let $k \leq \mathcal{M}(f)$. Then, there exists a restriction $\rho \in \{0,1,*\}^n$ with $|\rho^{-1}(*)| = k$ such that $\mathcal{M}(f|_\rho) = k$.*

Proof. We argue each case separately.

$\mathcal{M} = \mathsf{s}$. Let $a \in \{0,1\}^n$ be an input to f such that $\mathsf{s}(f,a) = \mathsf{s}(f)$. Given any integer $k \leq \mathsf{s}(f)$, consider a restriction ρ that leaves any k sensitive variables of a unset and sets the rest of the variables according to a. Clearly, $|\rho^{-1}(*)| = k$ and $\mathsf{s}(f|_\rho) = k$ as well.

$\mathcal{M} = \deg$. Let $k \leq \deg(f)$. Let p be the unique multilinear polynomial that represents f. Suppose p contains a monomial $\mathfrak{m}$ of degree k. Consider a restriction ρ that leaves variables in $\mathfrak{m}$ unset and sets the rest of the variables to 0. Clearly, $p|_\rho$ is the unique polynomial of degree k that represents $f|_\rho$ over k variables.

Now suppose p does not contain any monomial of degree k. Let $d > k$ be the smallest positive integer such that there exists a monomial $\mathfrak{m}$ of degree d in p, and there does not exist a monomial of degree $i \in [k, d-1]$ in p. Consider a restriction ρ that leaves any k variables in $\mathfrak{m}$ unset, sets the rest of $d-k$ variables in $\mathfrak{m}$ to 1, and sets the rest of the variables not in $\mathfrak{m}$ to 0. The resulting monomial in the restriction (from $\mathfrak{m}$) will have a non-zero coefficient and cannot be cancelled by any other restricted monomial from the polynomial p. So, such a restriction ρ will give us a polynomial of degree k that represents $f|_\rho$ over k variables. ◀

We show the following additional result on Fourier sparsity (see full version for details).

▶ **Theorem 37.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function with Fourier sparsity k. Then there exists a partial assignment ρ on $O(k^2)$ variables such that the Fourier sparsity of the restricted function f_ρ is k.*

Although the Fourier sparsity of an n-bit Boolean function can be as large as 2^n, Theorem 37 is non-trivial in regimes where sparsity is $O(\sqrt{n})$. In particular, many natural Boolean functions (e.g., parity and juntas) have Fourier sparsity $k \leq \sqrt{n}$, and hence admit condensation to $O(k^2)$ variables without loss of sparsity. An interesting open question suggested by our result is whether *logarithmic Fourier sparsity* admits condensation.

6 Conclusion

Given that hardness condensation has been pretty useful in proving sharper lower bounds in circuit complexity theory and proof complexity, a natural question to ask is: "how much" of hardness condensation is possible for the complexity measures related to decision tree complexity? Taking cue from Göös, Newman, Riazanov and Sokolov [16]; we initiate the study for almost all established complexity measures and prove both negative and positive results about condensation (see Table 1).

In terms of negative results, we show that lossless condensation is not possible for block sensitivity, certificate complexity, AND-decision tree query complexity, and several other complexity measures. We also prove that there exist variable restrictions that yield polynomial condensation gaps for different complexity measures. However, it remains open whether there exist functions that exhibit even stronger condensation barriers than what is established in this work (see Table 1). We list several other open questions for future work.

(1) An important unresolved question is whether deterministic communication complexity condenses losslessly. Since the AND-decision tree complexity does not condense (Theorem 2), is it possible to show that the answer is negative (as conjectured by [16]) by lifting with 2-bit AND gadget?

(2) Closing several gaps between positive and negative condensation for different measures listed in Table 1 remains an interesting question to explore. An intriguing one is whether unambiguous certificate complexity condenses losslessly (see also [16, 19]).

(3) Another avenue to explore is condensation with respect to projections or other similar restricted reductions. Typically, such modifications are used in applications in proof complexity (see, e.g., [28, 29, 6, 14, 10, 12, 13]).

(4) As mentioned in the introduction (Section 1, see also [18]), "cross condensation" between Boolean function parameters is another interesting variant of hardness condensation to explore. For example, if there is a Boolean function f such that for any restriction ρ, on $O(s(f))$ many variables $\mathsf{bs}(f_\rho) = \omega(\sqrt{\mathsf{bs}(f)})$, then f must have a super-quadratic gap between sensitivity and block sensitivity.

References

1 S. Aaronson, S. Ben-David, and R. Kothari. Separations in query complexity using cheat sheets. In *STOC 2016*, pages 863–876, 2016. doi:10.1145/2897518.2897644.

2 S. Aaronson, S. Ben-David, R. Kothari, S. Rao, and A. Tal. Degree vs. approximate degree and quantum implications of Huang's sensitivity theorem. In *STOC 2021*, pages 1330–1342, 2021. doi:10.1145/3406325.3451047.

3 A. Ambainis, K. Balodis, A. Belovs, T. Lee, M. Santha, and J. Smotrovs. Separations in query complexity based on pointer functions. *Journal of the ACM*, 64(5):32:1–32:24, 2017. doi:10.1145/3106234.

4 A. Ambainis, M. Kokainis, and R. Kothari. Nearly optimal separations between communication (or query) complexity and partitions. In *CCC 2016*, volume 50, pages 4:1–4:14, 2016. doi:10.4230/LIPIcs.CCC.2016.4.

5 S. Ben-David, P. Hatami, and A. Tal. Low-sensitivity functions from unambiguous certificates. In *ITCS*, volume 67, pages 28:1–28:23, 2017. doi:10.4230/LIPIcs.ITCS.2017.28.

6 C. Berkholz and J. Nordström. Supercritical space-width trade-offs for resolution. *SIAM J. Comput.*, 49(1):98–118, 2020. doi:10.1137/16M1109072.

7 H. Buhrman and R. de Wolf. Communication complexity lower bounds by polynomials. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity 2001*, pages 120–130, 2001. doi:10.1109/CCC.2001.933879.

8 H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

9 J. Buresh-Oppenheim and R. Santhanam. Making hard problems harder. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006)*, pages 73–87, 2006. doi:10.1109/CCC.2006.26.

10 S. Buss and N. Thapen. A simple supercritical tradeoff between size and height in resolution. *Electron. Colloquium Comput. Complex.*, TR24-001, 2024. URL: https://eccc.weizmann.ac.il/report/2024/001.

11 S. Chakraborty, C. Kayal, R. Mittal, M. Paraashar, S. Sanyal, and N. Saurabh. On the Composition of Randomized Query Complexity and Approximate Degree. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, volume 275, pages 63:1–63:23, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.63.

12 A. Chattopadhyay and P. Dvorak. Super-critical trade-offs in resolution over parities via lifting. *Electron. Colloquium Comput. Complex.*, TR24-132, 2024. URL: https://eccc.weizmann.ac.il/report/2024/132.

13 S. F. de Rezende, N. Fleming, D. A. Janett, J. Nordström, and S. Pang. Truly supercritical trade-offs for resolution, cutting planes, monotone circuits, and weisfeiler-leman. *CoRR*, abs/2411.14267, 2024. doi:10.48550/arXiv.2411.14267.

14 N. Fleming, T. Pitassi, and R. Robere. Extremely deep proofs. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022*, volume 215, pages 70:1–70:23, 2022. doi:10.4230/LIPIcs.ITCS.2022.70.

15 M. Göös, T. S. Jayram, T. Pitassi, and T. Watson. Randomized communication versus partition number. *ACM Trans. Comput. Theory*, 10(1), 2018. doi:10.1145/3170711.

16 M. Göös, I. Newman, A. Riazanov, and D. Sokolov. Hardness condensation by restriction. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024*, pages 2016–2027, 2024. doi:10.1145/3618260.3649711.

17 L. Hambardzumyan, H. Hatami, and P. Hatami. A counter-example to the probabilistic universal graph conjecture via randomized communication complexity. *Discret. Appl. Math.*, 322:117–122, 2022. doi:10.1016/J.DAM.2022.07.023.

18 G. Hart. *Condensing Hardness in Boolean Functions*, 2025. Undergraduate Research Report, University of Rochester.

19 P. Hrubes. Hard submatrices for non-negative rank and communication complexity. In *39th Computational Complexity Conference, CCC 2024*, volume 300, pages 13:1–13:12, 2024. doi:10.4230/LIPIcs.CCC.2024.13.

20 H. Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Annals of Mathematics*, 190(3):949–955, 2019. doi:10.4007/annals.2019.190.3.6.

21 R. Jain and H. Klauck. The partition bound for classical communication complexity and query complexity. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010*, pages 247–258, 2010. doi:10.1109/CCC.2010.31.

22 A. Knop, S. Lovett, S. McGuire, and W. Yuan. Log-rank and lifting for and-functions. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 197–208, 2021. doi:10.1145/3406325.3450999.

23 B. Loff and S. Mukhopadhyay. Lifting theorems for equality. In *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019*, volume 126, pages 50:1–50:19, 2019. doi:10.4230/LIPIcs.STACS.2019.50.

24 L. Lovász and M. Saks. Lattices, möbius functions and communication complexity. In *29th Annual Symposium on Foundations of Computer Science*, pages 81–90, 1988. doi:10.1109/SFCS.1988.21924.

25 G. Midrijanis. Exact quantum query complexity for total Boolean functions. *arXiv:quant-ph/0403168v2*, 2004.

26 N. Nisan. Crew prams and decision trees. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 327–335, 1989.

27 A. A. Razborov. A new kind of tradeoffs in propositional proof complexity. *J. ACM*, 63(2):16:1–16:14, 2016. doi:10.1145/2858790.

28 A. A. Razborov. On the width of semialgebraic proofs and algorithms. *Math. Oper. Res.*, 42(4):1106–1134, 2017. doi:10.1287/MOOR.2016.0840.

29 A. A. Razborov. On space and depth in resolution. *Comput. Complex.*, 27(3):511–559, 2018. doi:10.1007/S00037-017-0163-1.

30 D. Rubinstein. Sensitivity vs. block sensitivity of Boolean functions. *Combinatorica*, 15(2):297–299, 1995. doi:10.1007/BF01200762.

31 A. Tal. Properties and applications of boolean function composition. In *Innovations in Theoretical Computer Science, ITCS '13*, pages 441–454. ACM, 2013. doi:10.1145/2422436.2422485.

Optimal Testing of Reed-Muller Codes with an Online Adversary

Esty Kelman
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA
Boston University, Cambridge, MA, USA

Uri Meir
Blavatnik School of Computer Science, Tel Aviv University, Israel

Kai Zhe Zheng
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

Motivated by applications to property testing in the online-erasure model of Kalemaj, Raskhodnikova, and Varma (ITCS 2022 and Theory of Computing 2023), we define and analyze *semi-sample-based testers* for Reed-Muller codes. The task in Reed-Muller testing is to determine whether an input function $f : \mathbb{F}^n \to \mathbb{F}$ belongs to the Reed-Muller code or is far from it, using as few point queries to f as possible. Reed-Muller testing is a well-studied task with its roots in both the Property Testing and Probabilistically Checkable Proofs literature. The online-erasure model introduces a twist: after each query made, an adversary may erase up to t points of the input function, potentially thwarting any test in which the queries follow a predictable pattern.

Semi-sample-based testers are a hybrid between sample-based testers – which can only make uniformly random queries to the input function – and standard testers, which can choose their queries freely. They are designed with the online-erasure model in mind and operate by first choosing some subset S of the domain and then making their queries uniformly at random inside of S. We describe semi-sample-based testers for the Reed-Muller code and give an optimal analysis of their soundness.

Consequently, we show that semi-sample-based testers are indeed effective in the presence of online erasures, and thereby achieve optimal query complexity for testing the Reed-Muller code in the online-erasure model. This result improves upon prior work of Minzer and Zheng (SODA 2024). As an added bonus, we show that semi-sample-based testers also exist for the lifted affine-invariant codes of Guo, Kopparty, and Sudan (ITCS 2013), thereby providing the first known testers for these codes in the online-erasure model.

2012 ACM Subject Classification Mathematics of computing → Coding theory; Theory of computation → Error-correcting codes; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases Property testing, Low degree testing

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.31

Related Version *Full Version*: https://arxiv.org/abs/2605.22813

Funding *Esty Kelman*: Supported in part by NSF TRIPODS program (award DMS-2022448), NSF Grant No. 2022446, and NSF CCF award 2227876.
Kai Zhe Zheng: Supported by the NSF GRFP DGE-2141064.

Acknowledgements Part of this research was carried out during the authors' visit to the Simons Institute for the Theory of Computing at UCB. The authors wish to thank Pooya Hatami for bringing [14] and the references therein to our attention; Corollary 1.4 there essentially proves Theorem 3 by generalizing the lower from [5, Theorem 1.2] to all fields, showing that our new algorithm is optimal.

© Esty Kelman, Uri Meir, and Kai Zhe Zheng;
licensed under Creative Commons License CC-BY 4.0

1 Introduction

Local testing of the Reed-Muller code is a fundamental problem in property testing and complexity theory. In this problem, we are given oracle access to a multivariate function over a finite field, $f : \mathbb{F}^n \to \mathbb{F}$, and must decide whether f is a polynomial of total degree at most d, or is instead far from every polynomial of degree at most d. The notion of distance used here is the fractional Hamming distance. The goal is to design a randomized algorithm, called a *tester*, that accepts f with probability 1 in the former case and rejects f with probability at least $2/3$ in the latter case, while making as few queries to f as possible. Ideally, the query complexity should be independent of n.

The problem of Reed-Muller testing has been extensively explored over the past few decades. Its study began with the degree-1 case, which was considered in the seminal work of Blum, Luby, and Rubinfeld [11]. Their work introduced the well-known BLR linearity test and, more broadly, initiated the field of property testing. Subsequently, a large body of work gave and optimized testers for the general degree d case, $d > 1$ [1, 21, 10, 17, 32, 20, 27]. Additionally, a related variant of the Reed-Muller testing problem was instrumental in the early construction of Probabilistically Checkable Proofs (PCPs) [2, 31], and has continued to inspire subsequent research, including [30, 6, 26, 18, 7].

More recently, a new model of property testing introduced by Kalemaj, Raskhodnikova, and Varma, called the *online-erasure model* [19], has garnered significant attention. Several works have revisited Reed-Muller testing and related questions in this setting [19, 28, 5, 33, 3, 23, 24]. In this model, an adversary may *erase* up to t inputs after each query $f(x)$ is made, where typically t is assumed to be small relative to the domain size $|\mathbb{F}|^n$. Testers that work in the online-erasure model are called *online-erasure-resilient*.

We continue this line of work and give improved online-erasure-resilient testers for the Reed-Muller code. Our testers have query complexity that improves upon the previous state of the art in [28]. In addition, we give a matching lower bound showing that, for fixed field size and degree parameter, our query complexity is optimal as a function of the erasure rate parameter t.

Toward this result, we find it convenient to define a new class of testers, which we refer to as *semi-sample-based testers*. At a high level, semi-sample-based testers make their queries uniformly at random within some moderately large set of points S. In the case of Reed-Muller testing, the set $S \subset \mathbb{F}^n$ is chosen to have size independent of n, but still sufficiently large as a function of t, d, and $q := |\mathbb{F}|$. By making S large enough relative to the number of queries made, we ensure that our semi-sample-based testers are also online-erasure-resilient. Beyond Reed-Muller codes, we note that our approach extends naturally to give semi-sample-based testers for the lifted affine-invariant codes of [15]. To the best of our knowledge, these are the first testers for these codes that are resilient to online erasures.

We believe that the notion of semi-sample-based testing is conceptually useful and may find further applications in property testing, particularly in adversarial models such as the online-erasure model. To motivate this notion and explain why it arises naturally in this setting, we examine several natural candidate testers for Reed-Muller codes and show why they fail under online erasures. These candidates represent two extremes, which semi-sample-based testers naturally interpolate between. For simplicity, we restrict the discussion below to the case $d = 1$, as these issues already arise in this setting.

The Standard Tester. A first approach one might try is to run the standard tester in the online-erasure model. This tester is the well-known BLR linearity tester, which chooses $x, y \in \mathbb{F}^n$ uniformly at random, queries $f(x)$, $f(y)$, and $f(x + y)$, and then checks whether

$f(x) + f(y) = f(x + y)$. While this tester has optimal query complexity in the standard model, it fails completely in the online-erasure model. Indeed, an adversary can erase the value $f(x + y)$ after the first two queries are made, making it impossible to carry out the test. The issue here is that the queries are *too structured*.

Sample-based testing. At the other extreme, one can design a tester whose queries are as unstructured as possible. For example, one may consider a tester that queries f at uniformly random points $x \in \mathbb{F}^n$. Such testers were introduced in [12] and are called *sample-based testers*. In this case, it is unlikely that an erased point is ever queried (since $t \ll |\mathbb{F}|^n$), but this robustness comes at a cost: the tester requires at least n queries to succeed. Indeed, as shown in [13], with high probability, any set of fewer than n uniformly random points is linearly independent, and therefore there always exists a linear function that agrees with f on those points. Thus, while sample-based testers are robust to online erasures, they are not query efficient.

Semi-sample-based testing. Semi-sample-based testers interpolate between these two extremes. They operate by first choosing a subset of the domain and then taking uniformly random samples from within that subset. Our tester, in particular, chooses a k-dimensional linear subspace U and queries f at uniformly random points from U. Note that the case $k = n$ corresponds to sample-based testing, while the case $k = 1$ closely resembles the BLR tester.

By choosing the dimension of U appropriately, we obtain the best of both worlds. We achieve erasure resilience by taking k sufficiently large relative to t, while retaining query efficiency by keeping k independent of n.

1.1 Main Results

Our main result is the construction and analysis of a new tester for the Reed-Muller code using the semi-sampled-based approach. Along the way, we note that our semi-sample-based testers can also be made to work for other affine-invariant families, such as lifted affine-invariant codes [15]. Throughout this section, we say that a function $f : \mathbb{F}^n \to \mathbb{F}$ is ε-far from a given family of functions if, for every function in that family, f differs from it on at least an ε-fraction of all possible inputs.

Reed-Muller Codes. Fix a degree parameter $d \in \mathbb{N}$ and a finite field $\mathbb{F}$ of size q, where q is a power of a prime p. The starting point for our tester is the simple observation that the restriction of any polynomial of degree at most d to a linear subspace remains a polynomial of degree at most d.

Formally, let $U \subset \mathbb{F}^n$ be a linear subspace of dimension k with basis $v_1, \ldots, v_k$. If $f : \mathbb{F}^n \to \mathbb{F}$ has degree at most d, then the induced k-variate function $f|_U$, defined by

$$f|_U(z_1, \ldots, z_k) := f\left(\sum_{i=1}^{k} z_i v_i\right),$$

is a polynomial of degree at most d in the variables $z_1, \ldots, z_k$. This observation underlies most previous Reed-Muller testers, including those of [1, 10, 17, 20].

In the standard model, the tester proceeds by choosing a subspace U of dimension $k = k_{q,d} + 1$, where

$$k_{q,d} := \left\lceil \frac{d+1}{q - q/p} \right\rceil,$$

and checking whether $f|_U$ has degree at most d by querying $f(x)$ for every point $x \in U$. Our semi-sample-based variant, which is instead parameterized by $k \geq k_{q,d} + 1$ and denoted by SSB_k, queries f on a random subset of s_k points from U, rather than on all of U. The precise choice of s_k is stated in Theorem 23, but crucially it satisfies

$$s_k = O\left(q^{k_{q,d}+1}\left(\frac{e(d+k)}{k}\right)^k\right).$$

The tester $\mathbf{SSB}_{\boldsymbol{k}}$. The degree-$d$ tester with parameter k proceeds as follows:

1. Choose a linear subspace $U \subset \mathbb{F}^n$ of dimension k uniformly at random.
2. Choose s_k points from U uniformly at random, and let S denote the resulting set.
3. Accept if there exists a degree d function that agrees with f on every point in S. Otherwise, reject.

We note SSB_k makes at least as many queries as the $q^{k_{q,d}+1}$ made by the standard tester, and its query complexity increases with k. However, SSB_k has two properties which are important for us: (1) all queries made are uniformly random inside U; and (2) the *fraction* of points queried in the chosen subspace U equals to s_k/q^k which goes to 0 as k increases. These properties turn out to be crucial for our desired application to testing in the online erasure model.

The focus of this paper is thus to analyze SSB_k above, and our main result is the following.

▶ **Theorem 1** (Informal Version of Theorem 23). *The tester* SSB_k *for degree* d *Reed–Muller codes over* $\mathbb{F}^n$ *satisfies the following:*

- Completeness: *If* $\deg(f) \leq d$*, then the tester accepts with probability* 1.
- Soundness: *If* f *is* ε*-far from every degree* d *function, then the tester rejects with probability at least* $\min\{1/128,\ s_k\varepsilon/8\}$.

To obtain a tester for Reed-Muller codes in the online erasure model, we set k sufficiently large with respect to the erasure parameter t, momentarily setting $\varepsilon = \Omega(1)$. Concretely, we choose $k = \Theta(d) + \log_q t$ and observe that for k at least this large, the probability of querying an erased point is negligible. This observation leads to the following result.

▶ **Theorem 2** (Informal Version of Theorem 28). *For any degree parameter* d*, field* $\mathbb{F}$ *of size* q*, dimension* n*, and erasure parameter* $t \leq O(q^{n-\Theta(d)})$*, there exists a tester for the degree* d *Reed-Muller code over* $\mathbb{F}^n$ *in the* t*-online-erasure model with query complexity* $\left(O_q\left(\frac{\log t}{d}\right)\right)^d$ *that satisfies the following:*

- Completeness: *If* $\deg(f) \leq d$*, then the tester accepts with probability* 1.
- Soundness: *If* f *is* $\Omega(1)$*-far from every degree* d *function, then the tester rejects with probability at least* $\Omega(1)$.

Prior to our work, the best-known online-erasure-resilient Reed-Muller testers used $O_q(\log t)^{3d}$ queries over prime fields and $O_q(\log t)^{3d+3q}$ queries over non-prime fields. Our results improve upon these bounds in all cases. We complement this upper bound with an almost matching lower bound, showing that our query complexity is optimal (up to constant factors, for constant q, d) as a function of the erasure parameter for fixed field size and degree.

▶ **Theorem 3** (Informal Version of the Erasure-Resilient Testing Lower Bound for Reed-Muller Codes). *For any degree parameter* d*, field* $\mathbb{F}$ *of size* q*, dimension* n*, and erasure parameter* t*, any tester for the degree* d *Reed-Muller code over* $\mathbb{F}^n$ *must make at least* $\left(\frac{\log_q t}{d}\right)^d$ *queries.*

Previously, this lower bound was shown over $\mathbb{F}_2$ in [5]. Leveraging recent results from [4, 14], we note that the same argument extends to general finite fields. A more detailed discussion, including a formal statement of Theorem 3, is provided in the full version.

Lifted Affine Invariant Codes. In addition to testing the Reed-Muller code, we note that our semi-sample-based testers can also be defined quite naturally for a family of functions called lifted affine-invariant codes [15]. These codes are a generalization of Reed-Muller codes and were previously studied in the context of property testing in [15, 16, 20]. They are defined as follows. Let $\mathcal{C}_j \subseteq \{\mathbb{F}^j \to \mathbb{F}\}$ be an *affine-invariant* family of functions, meaning $f \in \mathcal{C}_j$ if and only if $f \circ T \in \mathcal{C}_j$ for any affine transformation $T : \mathbb{F}^j \to \mathbb{F}^j$. Then, the n-dimensional lift of $\mathcal{C}_j$ is $\mathcal{C}^{j\nearrow n}$ and is the set of functions $f : \mathbb{F}^n \to \mathbb{F}$ such that $f|_U \in \mathcal{C}_j$ for every j-dimensional affine subspace $U \subset \mathbb{F}^n$.

It is straightforward to design a semi-sample-based tester for $\mathcal{C}^{j\nearrow n}$. Indeed, one may set any $k \geq j$ and adjust the number of random queries in step 2 suitably (denoted by Q_k below). Then, in step 3 the tester checks if there is a function from $\mathcal{C}^{j\nearrow k}$ agreeing with the queries. Our main result for lifted affine-invariant codes is fully formalized and proved in the full version, and is stated informally below.

▶ **Theorem 4** (Informal Version of the Semi-sample-based Tester for Lifted Affine-invariant Codes)**.** *Let $\mathcal{C}_n = \mathcal{C}^{j\nearrow n}$ be the n-dimensional lift of the affine invariant code $\mathcal{C}_j \subseteq \{\mathbb{F}^j \to \mathbb{F}\}$ and let δ_0 be the smallest minimum distance of the codes $\mathcal{C}_j, \cdots, \mathcal{C}_n$. Choose dimension $k \geq j$ and set the number of queries to be*

$$Q_k := \left\lceil \frac{100 \ln |\mathcal{C}_k|}{\delta_0} \right\rceil + 1.$$

The semi-sample-based tester for $\mathcal{C}^{j\nearrow n}$ satisfies the following:

- *Completeness: If $f \in \mathcal{C}^{j\nearrow n}$, then the tester accepts with probability* 1.
- *Soundness: If f is ε-far from $\mathcal{C}^{j\nearrow n}$, then the tester rejects with probability at least* $\min\{Q_k\varepsilon/8, 1/128\}$.

Once again, by choosing k suitably, we also get the following online-erasure-resilient tester for lifted affine-invariant codes. We state it informally below, saving the precise upper threshold for the erasure parameter to be formally defined in the full version.

▶ **Theorem 5** (Informal Version of Lifted Affine-invariant Codes Online-resilient Tester)**.** *Let $\mathcal{C}^{j\nearrow n}$ be a lifted affine-invariant code and let the erasure parameter t and dimension parameter k be chosen suitably. Then $\mathcal{C}^{j\nearrow n}$ has an $O(Q_k)$ query tester in the t-online erasure model that satisfies the following*

- *Completeness: If $f \in \mathcal{C}_n$, then the tester accepts with probability* 1.
- *Soundness: If f is $\Omega(1)$-far from $\mathcal{C}^{j\nearrow n}$, the tester rejects with probability at least* 2/3.

Optimal Soundness Testing and Comparison with [28]. In [28], a similar tester is analyzed in the online-erasure model and a version of Theorem 2 is obtained with a higher query complexity[1] of $O_q(\log(t))^{3d}$. We discuss the differences between our result and the main result of [28]. The tester in [28] can be viewed as SSB_k with a modified third step. Instead of checking if there is a degree d-function agreeing with f on all queried points, the tester of [28] finds a structured dual codeword, over domain U, that is supported on the queried

[1] The result of [28] states $3d + 3$ in the exponent, but their analysis easily shows an exponent of $3d$.

points, and checks if $f|_U$ is indeed dual to this codeword. That is, [28] finds a function $h : U \to \mathbb{F}_q$ such that every $f : U \to \mathbb{F}_q$ with $\deg(f) \leq d$ has inner product 0 with h (i.e. $\sum_{x \in U} f(x)h(x) = 0$) and checks whether this is the case. Note that computing this inner product only requires knowing $f(x)$ for x in the support of h, hence the reason for requiring h to be supported on the queried points. From here, [28] rely on the generic soundness result of [22], which guarantees, under certain conditions satisfied here, that a Q-query tester rejects constant-far functions with probability roughly Q^{-2}. Thus, while their base tester makes $O_q(\log(t))^d$ queries, the soundness they obtain for this test requires $O_q(\log(t))^{2d}$ further repetitions to reject with constant probability, and hence leads to a version of Theorem 2 with $O_q(\log(t))^{3d}$.

By adapting the third step of SSB_k to accept if and only if there is a degree d function explaining the queried points, we are able to obtain (in Theorem 1) a better soundness analysis for SSB_k which rejects constant far functions with constant probability (and hence leads to Theorem 2 with the stated query complexity). In fact, our soundness analysis for the SSB_k is *optimal* up to constant factors. As described in [20], a tester making Q queries is called an *optimal tester* if it achieves rejection probability $\min\{\Omega(1), \Omega(Q\varepsilon)\}$ in the soundness case. Such testers are optimal in the sense that this is the best probability with which one can hope to reject an ε-far function with Q queries. Indeed, thinking of an ε-far function as a valid function with ε-fraction of its entries perturbed, the probability of even querying a perturbed point, and hence having any hope of rejecting, is at most $\min\{\Theta(1), Q\varepsilon\}$.

Showing that a fixed tester has such optimal soundness is typically a difficult task. For example, the first optimal Reed-Muller testers were only shown relatively recently: in [10] when $|\mathbb{F}| = 2$, and in [17, 20, 27] when $|\mathbb{F}| > 2$. The works mentioned only show that the specific tester they consider is optimal, and do not easily extend to other testers such as our semi-sample-based ones or the general testers in [22].

Finally, an additional factor of $(\Theta(d))^d$ in the query complexity comes from a precise choice of $k = \log_q t + \Theta(d)$ (rather than $k = \Theta(d \log_q t)$). This results in the main term of $(\Theta(\frac{\log t}{d}))^d$ in the query complexity, matching with our lower bound in Theorem 3.

1.2 Technical Overview

We describe our techniques towards our main theorem for semi-sample-based Reed-Muller testers Theorem 1. The approach for lifted affine-invariant codes is similar. Our analysis follows the inductive approach pioneered by [10]. However, for our purposes, we need to make a few key enhancements, which use techniques from the PCP literature. In particular, we show a hyperplane agreement lemma (in the low error regime), that is motivated by the Plane versus Point analysis of [31], as well as the high-error hyperplane agreement lemmas that were key to recent breakthroughs on 2-to-2 games [25] and optimal alphabet-soundness tradeoff PCPs [29]. Similar ideas were also recently used in the testing of direct-sums [33].

Proof outline. Let s be the number of queries made, and consider f of distance exactly ε from having degree d. Let $f = g + h$ be such that g is the degree d function that is closest to f and h is a function with fractional support size ε. As in [10], we split the soundness analysis into a small distance case, say where $\varepsilon < \frac{1}{10s}$, and a large distance case, where $\varepsilon \geq \frac{1}{10s}$. In the small distance case, one can show directly that with probability roughly $s \cdot \varepsilon$, the tester will only query one input from the support of h and in this case, one can easily argue that the tester will indeed reject.

More difficult to analyze is the large distance case, where one relies on induction on the ambient dimension n and assumes good soundness is guaranteed for all smaller ambient dimensions. Here we want to show that the test rejects with some constant probability, say 1/100. The key piece of the inductive step is a lemma of the following form:

▶ **Lemma 6.** *If there are $M = q^{\Theta(1)} \cdot s$ hyperplanes $W_1, \ldots, W_M$ on which the restriction of f to any W_i is $\frac{1}{100s}$-close to degree d, then f is $\frac{1}{10s}$-close to degree d.*

Observe that we can perform the semi-sample-based tester of Theorem 1 by first choosing a hyperplane W and then performing the tester on the restriction of f to W, denoted by $f|_W$, which is a function over a space of dimension $n-1$. Thus, we can write the rejection probability as $\Pr[f \text{ is rejected}] = \mathbb{E}_W \Pr[f|_W \text{ is rejected}]$. If there are more than M hyperplanes on which $f|_W$ is $\frac{1}{100s}$-close to degree d, then we get that $f|_W$ is $\frac{1}{10s}$-close to degree d and can apply the small distance analysis. On the other hand, if there are fewer than M such hyperplanes, then with probability roughly $1 - M/q^{n-1}$ over the choice of W, the function $f|_W$ is still relatively far from degree d, and by the inductive hypothesis we reject with some constant probability. In order to ensure that we still get constant soundness overall though, we must ensure that M/q^{n-1} is a small constant, and in particular, this means we must start our induction at dimension roughly $n = \log_q(M)$. Fortunately, Lemma 6 gives a strong enough bound on M above for the Reed-Muller code and take n roughly $\Theta(k)$ to get Theorem 1.

We note that similar statements to Lemma 6 are shown in [10, 17], but neither of these is sufficient for our purposes. The reason is because [10] only works over field size $q = 2$ while [17] requires obtains $M = c(q) \cdot s$ for some $c(q)$ with tower type dependence on q (rather than polynomial in q as in Lemma 6), due to the use of the density Hales-Jewett theorem. Additionally, these arguments do not apply beyond Reed-Muller codes, whereas we ultimately want our techniques to generalize to other affine-invariant codes and function families.

Our proof of Lemma 6 instead employs techniques from agreement testing and PCPs. In [31], for example, a similar statement is shown except with planes in place of hyperplanes and non-trivial agreement with a degree d function on the planes, instead of being close to degree d on the planes. Similarly [25, 29] show a similar statement to Lemma 6, but with non-trivial agreement on the hyperplanes. Thus, Lemma 6 can be viewed as a low-error version of the lemmas in [31, 25, 29], and at a high level, the proof proceeds in the same way. We consider the *consistency graph* over the M hyperplanes, $W_1, \ldots, W_M$, and say that two vertices $i, j \in [M]$ are adjacent if the local degree d functions on W_i, W_j, call them f_i, f_j, agree on the intersection $W_i \cap W_j$. By showing that this graph is "nearly transitive" we find a large clique, which corresponds to many of the f_i's agreeing in pairs. From here we arbitrarily extrapolate the f_i's from the large clique to some global function F and show that 1) F is close to f; and 2) F is close to degree d. By the triangle inequality, this establishes Lemma 6.

While Lemma 6 on its own right is not hard to establish with existing techniques, we believe its statement and the way we use to obtain testing results are generic enough and could find more applications, making it one novel aspect of our work. In particular, using the same outline we extend our results to families of affine-invariant properties. The main bulk of our strategy follows through, save for step 2).

In order to accomplish 2), we rely on known local testing results to show that F passes a local test with high probability and is thus close to the desired family of functions. Indeed, if all the tester's queries fall within one of the W_i's above in the clique, then the test surely passes, as here F is consistent with the legitimate function f_i. If M is sufficiently large, then this probability is high. That is, step 2) relies on a known (typically structured) test, and leads the way to obtaining unstructured tests for the same property.

Currently, we require some lower bound on the soundness of known testers for the analysis of our tester to work (specifically in step 2). For future work, one could hope to perform step 2) algebraically for some fixed properties, as was done for Reed Muller codes in [10, 17]. For non-linear affine-invariant properties, such as some of those considered in [9, 8], such an approach could yield significantly improved soundness.

2 Preliminaries

Notation. In general, we let $\mathbb{F}$ denote a finite field with characteristic p and cardinality $|\mathbb{F}| = q = p^\ell$. At times we will use a subscript such as $\mathbb{F}_q$ or $\mathbb{F}_2$ to specify a specific field size, but we omit the subscript when it is clear from context. For a function $f : \mathbb{F}^n \to \mathbb{F}$, it is well known that there is an n-variate polynomial with degree in each variable at most $|\mathbb{F}| - 1$ whose evaluation over $\mathbb{F}^n$ is equal to f. We let $\deg(f)$ denote the total degree of this polynomial. Through the paper, for the notion of distance, we will use the fractional Hamming distance. For two functions $f, g : \mathbb{F}^n \to \mathbb{F}$, the fractional Hamming distance is

$$\mathsf{dist}(f, g) := \frac{|\{x \in \mathbb{F}^n : f(x) \neq g(x)\}|}{q^n}.$$

We extend the definition to a family of functions $\mathcal{F} \subseteq \{\mathbb{F}^n \to \mathbb{F}\}$ by

$$\mathsf{dist}(f, \mathcal{F}) := \min_{g \in \mathcal{F}} \mathsf{dist}(f, g) \ .$$

The function families that we consider typically form an error-correcting code, meaning any two members of the family are far apart in fractional hamming distance. For a function family $\mathcal{F}$, we define its distance as

$$\delta(\mathcal{F}) = \min_{f,g \in \mathcal{F}, f \neq g} \mathsf{dist}(f, g).$$

We let $\mathcal{G}_k$ be the set of linear subspaces of dimension k inside of $\mathbb{F}^n$, where the ambient space $\mathbb{F}^n$ will always be clear from context. We denote the number of distinct linear subspaces of dimension k inside a linear space of dimension n by

$$\begin{bmatrix} n \\ k \end{bmatrix}_q := |\mathcal{G}_k| = \prod_{i=0}^{k-1} \frac{q^n - q^i}{q^k - q^i}.$$

We will also work with affine subspaces of dimension k, which are sets of the form $x + U = \{x + y \mid y \in U\}$, where $U \in \mathcal{G}_k$. We use $\dim(U)$ to denote the dimension of U when it is either an affine or linear subspace. It will always be clear from context which kind of subspace U is.

Affine Invariance. A transformation $T : \mathbb{F}^n \to \mathbb{F}^n$ is called an affine transformation if it can be expressed as $T(x) = Mx + b$, for some matrix $M \in \mathbb{F}^{n \times n}$ and affine shift $b \in \mathbb{F}^n$.

A family of functions $\mathcal{C}_n \subseteq \{\mathbb{F}^n \to \mathbb{F}\}$ is called affine invariant if for any affine transformation, even non-invertible, $T : \mathbb{F}^n \to \mathbb{F}^n$ the following holds: $f \in \mathcal{C}_n \implies f \circ T \in \mathcal{C}_n$, where $f \circ T$ is the function given by $f \circ T(x) = f(T(x))$. Note that we require the above to hold even for T non-invertible.

For a k-dimensional subspace U (either linear or affine), we let $f|_U : \mathbb{F}^k \to \mathbb{F}$ denote the restriction of f to U. For our purposes, we will not care about the parametrization of U, so we can define $f|_U = f \circ T$ by taking any affine transformation $T : \mathbb{F}^k \to \mathbb{F}^n$ whose image is U. For every dimension k, let $\mathcal{C}_k = \{f|_U \mid f \in \mathcal{C}_n, U \in \mathcal{G}_k\}$.

One can check that if $\mathcal{C}_n$ is affine invariant, then so is $\mathcal{C}_k$. A common local test for $\mathcal{C}_n$, which we call the k-space test, goes as follows:

- Choose $U \in \mathcal{G}_k$ uniformly at random.
- Accept if $f|_U \in \mathcal{C}_k$. Otherwise, reject.

Note that since $\mathcal{C}_k$ is affine invariant, it does not matter what parametrization we choose for $f|_U$ when performing the test. We remark that typically $\mathcal{C}_k$ is defined using affine-subspaces (or flats), instead of only linear subspaces as we do above, and the local test considered is usually based on restrictions to flats, rather than to linear subspaces. This test is typically called the $(k-1)$-flat test, but note that we can simulate a $(k-1)$-flat test using a k dimensional linear subspace test. Indeed, instead of checking $f|_{U'}$ for a random $(k-1)$-flat U', we can instead sample a random linear subspace U of dimension k containing U'. Over random U', U is uniformly random in $\mathcal{G}_k$, and hence performing the k-space test above with $f|_U$ is just as good as the $(k-1)$-flat test.

The Reed-Muller Code. Given a finite field vector space $\mathbb{F}_q^n$, we refer to the set of degree at most d polynomials as the degree d Reed-Muller code and denote this by $\mathsf{RM}[n,q,d]$.

We will need the following well-known facts about the rate, distance and local testability of Reed-Muller codes.

▶ **Fact 1.** *The code* $\mathsf{RM}[n,q,d]$ *has distance* $q^{-\frac{d}{q-1}}$ *and rate at most* $\binom{n+d}{d}/q^n$*. That is, for any two distinct functions* $f, g \in \mathsf{RM}[n,q,d]$*,*

$$\Pr_{x\in\mathbb{F}_q^n}[f(x) \neq g(x)] \geq q^{-\frac{d}{q-1}}, \quad \text{and} \quad \frac{\log_q |\mathsf{RM}[n,q,d]|}{q^n} \leq \frac{\binom{n+d}{d}}{q^n} .$$

For the rate we are using the fact that $\mathsf{RM}[n,q,d]$ *is spanned by all total degree at most* d *monomials over* n *variables and there are at most* $\binom{n+d}{d}$ *such monomials.*

Finally, given field size q and degree d, we define the testing dimension of $\mathsf{RM}[n,q,d]$ to be

$$t_{q,d} = \left\lceil \frac{d+1}{q-q/p} \right\rceil + 1.$$

We will use $t_{q,d}$ to refer to the above quantity throughout the paper. The following result from [20] shows the following soundness for the local test which restricts to a random $t_{q,d}-1$ affine subspace.

▶ **Theorem 7** ([20, Theorem 1.3])**.** *There exists a constant* c_{KM} *such that if* f *is* ε*-far from* $\mathsf{RM}[n,q,d]$*, then* $\Pr_{U'\subset\mathbb{F}_q^n}[\deg(f|_{U'}) > d] \geq q^{-c_{\mathsf{KM}}} \cdot \min\{1, q^{t_{q,d}}\varepsilon\}$*, where* U' *is a random* affine *subspace of dimension* $t_{q,d}-1$*.*

As discussed above, we can also simulate the test above by restricting to a linear subspace of one higher dimension. Its soundness is an immediate consequence of Theorem 7, and we give the short proof below.

▶ **Theorem 8.** *For the same constant* c_{KM} *from Theorem 7 it holds that if* f *is* ε*-far from* $\mathsf{RM}[n,q,d]$*, then* $\Pr_{U\subset\mathbb{F}_q^n}[\deg(f|_U) > d] \geq q^{-c_{\mathsf{KM}}} \cdot \min\{1, q^{t_{q,d}}\varepsilon\}$*, where* U *is a random* linear *subspace of dimension* $t_{q,d}$*.*

Proof. We can sample a random $t_{q,d}$-space by first choosing a $(t_{q,d}-1)$-flat U', and then choosing a random linear subspace U containing U'. Note that U is a uniformly random $t_{q,d}$-space. It follows that,

$$\begin{aligned}\Pr_{U\subset\mathbb{F}_q^n}[\deg(f|_U)>d] &= \mathop{\mathbb{E}}_{U'\subset\mathbb{F}_q^n}\left[\Pr_{U\supset U'}[\deg(f|_{U'})>d]\right]\\ &= \Pr_{U'\subset\mathbb{F}_q^n}[\deg(f|_{U'})>d] \geq q^{-c_{\mathsf{KM}}}\cdot\min\{1,q^{t_{q,d}}\varepsilon\}.\end{aligned}$$ ◀

Lifted Affine-Invariant Codes. Lifted affine-invariant codes were introduced by Guo, Kopparty, and Sudan [15]. We formally define the codes below.

▶ **Definition 9** ([15, Definition 1.1]). *Let $\mathcal{C}\subseteq\{\mathbb{F}^r\to\mathbb{F}\}$ be a linear affine-invariant family of functions. Then the lifted code $\mathcal{C}^{r\nearrow n}$ is the following set of functions:*

$$\{f:\mathbb{F}^n\to\mathbb{F}\mid f|_U\in\mathcal{C}_r,\forall U \text{ affine },\dim(U)=r\}.$$

When $\mathcal{C}$ in the above definition is clear, we will often use $\mathcal{C}_n$ to refer to the code $\mathcal{C}^{r\nearrow n}$. The code $\mathcal{C}$ is often referred to as the base code of $\mathcal{C}^{r\nearrow n}$ and $\mathcal{C}^{r\nearrow n}$ is referred to as the n-dimensional lift of $\mathcal{C}$. Lifted affine-invariant codes generalize Reed-Muller codes, indeed it can be seen that $\mathsf{RM}[n,q,d]$ is the n-dimensional lift of $\mathsf{RM}[t_{q,d}-1,q,d]$. In [20], it was shown that a natural test based on restrictions to r-dimensional affine subspaces is a local test for $\mathcal{C}^{r\nearrow n}$ with optimal soundness. Their result also implies similar soundness for a test based on restricting to $(r+1)$-dimensional linear subspaces. We state this version of their result below as it is more convenient for us.

▶ **Theorem 10** ([20, Theorem 1.4]). *Given $f:\mathbb{F}^n\to\mathbb{F}$, the local test obtained by choosing a uniformly random linear subspace of dimension $r+1$ has the following guarantees:*

- *Completeness: If $f\in\mathcal{C}_n$ then $\Pr_U[f|_U\in\mathcal{C}_{r+1}]=1$.*
- *Soundness: If f is ε-far from $\mathcal{C}_n$, then, $\Pr_U[f|_U\notin\mathcal{C}_{r+1}]\geq q^{-O(1)}\min\{1,q^r\varepsilon\}$.*

Distinguishing all members of a function family.

▷ Claim 11. Fix $f:\mathbb{F}_q^n\to\mathbb{F}_q$ and let $\mathcal{F}\subseteq\{\mathbb{F}_q^n\to\mathbb{F}_q\}$ be a family of functions with size $|\mathcal{F}|=N$ and distance $\delta(\mathcal{F})=\delta$. Then choosing a set $S\subseteq\mathbb{F}_q^n$ of size M uniformly at random with replacement, with probability at least, $1-(1-\delta)^M\cdot\binom{N}{2}\geq 1-e^{-\delta M}\cdot N^2$. every pair of distinct functions in $\mathcal{F}$ disagree on at least one point in $\mathcal{F}$. As a consequence, there is at most one $F\in\mathcal{F}$ such that $f(x)=F(x),\forall x\in S$.

Proof. Fix a pair of functions $F_1,F_2\in\mathcal{F}$. Then the probability that they agree on all points in S is at most $(1-\delta)^M$. As $|\mathcal{F}|=N$, we can union bound over all pairs of functions in $\mathcal{F}$ to get that with probability at most $(1-\delta)^M\cdot\binom{N}{2}$, there exist two distinct functions in $\mathcal{F}$ that agree on S. The result follows. ◁

We conclude that whenever $M>>\frac{\log N}{\delta}$ the probability of S not satisfying Claim 11 is negligible. For example, let $\mathcal{F}\subset\mathbb{F}_2^n\to\mathbb{F}_2$ is the class of degree at most d functions over $\mathbb{F}_2$ then, $\delta=\delta_d=2^{-d}$ and $N=2^{\binom{n}{\leq d}}$ to have a good set with probability at least 0.9 it is enough to take $M\approx 2^d\binom{n}{\leq d}$ samples.

3 The Hyperplane Agreement Lemma

In this section, we show, for any affine-invariant family of functions $\mathcal{C}_n$, if f is close to some function in $\mathcal{C}_n$ on sufficiently many hyperplanes, then f must in fact be close to $\mathcal{C}_n$. The quality of our bound on how many hyperplanes are needed depends on the distance of the code as well as the local testability of $\mathcal{C}_n$.

Before stating the main lemma of the section, let us describe some necessary notation regarding the family $\mathcal{C}_n$. Recall that $\mathcal{G}_j$ denotes the set of j-dimensional subspaces of $\mathbb{F}_q^n$. Fix $\mathcal{C}_n \subseteq \{\mathbb{F}^n \to \mathbb{F}\}$ to be some affine-invariant family of functions and for each dimension k, let $\mathcal{C}_k \subseteq \{\mathbb{F}^n \to \mathbb{F}\}$ be the subset of functions given by, $\mathcal{C}_k = \{f|_U \mid U \subseteq \mathbb{F}^n, U \in \mathcal{G}_k, f \in \mathcal{C}_n\}$. Suppose that there is a dimension t with the following properties:

▶ **Definition 12.** *We say that a code $\mathcal{C}_n$ is locally testable via the t-space test with soundness function $c(\cdot)$, if for any function $f : \mathbb{F}^n \to \mathbb{F}$ it holds that $\Pr_{U\in\mathcal{G}_t}[f|_U \notin \mathcal{C}_t] \geq c(\mathsf{dist}(f, \mathcal{C}_n))$, for some function $c : \mathbb{R} \to \mathbb{R}$ that is continuous, non-decreasing, and satisfies $c(0) = 0$.*

We fix this t for the remainder of the section and let δ_0 be the smallest minimum distance of the codes $\mathcal{C}_t, \mathcal{C}_{t+1}, \ldots, \mathcal{C}_n$. That is,

$$\delta_0 := \min_{k:t\leq k\leq n} \delta(\mathcal{C}_k). \tag{1}$$

The main lemma of this section states that if f is far from $\mathcal{C}_n$, then $f|_W$ must also be far from $\mathcal{C}_{n-1}$ on almost all hyperplanes of $W \subset \mathbb{F}^n$. Formally,

▶ **Lemma 13** (Hyperplane Agreement)**.** *Let $\mathcal{C}_n$ be an affine invariant family of functions over $\mathbb{F}_q^n$ that is locally testable via the t-space test with soundness function $c(\cdot)$ and let δ_0 as in (1). Set $\varepsilon < \delta_0/6$ and let $M \geq \max\left\{\frac{10^5 q^4}{\varepsilon}, \frac{20q^t}{c(\varepsilon)}\right\}$. If there are M distinct hyperplanes $W_1, \ldots, W_M \subset \mathbb{F}^n$, such that for each $i = 1, \ldots, M$, the restriction of f to the hyperplane W_i, denoted by $f|_{W_i}$, is ε-close to $\mathcal{C}_{n-1}$ on W_i, then f is 4ε-close to $\mathcal{C}_n$.*

In the lemma above, M is set sufficiently large with respect to the soundness function, $c(\cdot)$, of the t-space test on $\mathcal{C}_n$, so that $\frac{20q^t}{M} \leq c(\varepsilon)$. In other words, M is sufficiently large so that if $\mathsf{dist}(f, \mathcal{C}_n) > \varepsilon$, then the testing guarantee of Definition 12 gives $\Pr_{U\in\mathcal{G}_t}[f|_U \notin \mathcal{C}_t] \geq \frac{20q^t}{M}$.

The remainder of the section is dedicated to showing Lemma 13. We follow the strategy of [31] in the plane versus point analysis. Suppose that there are many hyperplanes W_i with local degree d functions $f_i : W_i \to \mathbb{F}_q$ such that f_i and f are close. We will first show that many – in fact at least a constant fraction – of these f_i's are actually consistent with each other. Using these consistent f_i's we can then define a function F on the whole space and argue that 1) F is close to f and 2) F passes the $t_{q,d}$-space test with high probability. For both 1) and 2) we are crucially using the fact that F is defined using many consistent local functions on hyperplanes, so for nearly every point x in the domain, there are, in fact, many of these local degree d functions defined on x.

3.1 The Sampling Lemma

In this section, we show a sampling lemma regarding hyperplanes and points. The proof is similar to that of [29, Lemma 5.1], but we strengthen the analysis. In particular, Lemma 14 below includes an additive "error" term of $\sim q/M$, which improves upon a similar factor with $\sqrt{M}$ in the denominator in [29, Lemma 5.1].

We next describe the sampling process and the lemma. Let $\mathcal{W} = \{W_1, \ldots, W_M\}$ be a collection of hyperplanes in $\mathbb{F}_q^n$. Denote by μ the uniform measure over $x \in \mathbb{F}_q^n$, i.e., $\mu(x) = q^{-n}$, and by $\nu_{\mathcal{W}}$ the probability measure over $\mathbb{F}_q^n$ generated by the following sampling procedure:

1. Choose $W_i \in \mathcal{W}$ uniformly at random.
2. Choose a point $x \in W_i$ uniformly at random.

We will omit the subscript $\mathcal{W}$ and use ν when it is clear from the context. Furthermore, we naturally extend the measure on point to sets by $\mu(S) = \sum_{x \in S} \mu(x)$ and similarly for ν. Let us define for $x \in \mathbb{F}_q^n$ the set of hyperplanes containing it

$$N(x) := |\{W_i \in \mathcal{W} \mid W_i \ni x\}|. \tag{2}$$

Since Item 1 samples $W_i \ni x$ with probability $\frac{N(x)}{M}$, and conditioned on this event, Item 2 chooses x with probability $\frac{1}{|W_i|} = q^{-(n-1)}$, we get that by definition

$$\nu(x) = \frac{N(x)}{M} \cdot q^{-(n-1)}. \tag{3}$$

We turn to show the following lemma, which asserts that the measures μ and $\nu_{\mathcal{W}}$ are close whenever $\mathcal{W}$ contains sufficiently many hyperplanes.

▶ **Lemma 14** (Sampling). *For any collection $\mathcal{W} = \{W_1, \ldots, W_M\}$ of M distinct hyperplanes in $\mathbb{F}_q^n$, and any set of points $S \subseteq \mathbb{F}_q^n$, the probability measures $\mu, \nu_{\mathcal{W}}$ satisfy*

$$\frac{1}{2}\left(\mu(S) - \frac{4q}{M}\right) \leq \nu_{\mathcal{W}}(S) \leq 2\mu(S) + \frac{8q}{M}.$$

The key to proving the sampling lemma is analyzing the concentration of $N(x)$.

▷ Claim 15. For any $c > 0$ it holds that $\Pr_{x \in \mathbb{F}_q^n}\left[\left|N(x) - \frac{M}{q}\right| \geq \frac{c \cdot M}{q}\right] \leq \frac{q}{c^2 M}$.

Proof. We analyze the random variable $N(x)$ where x is uniformly chosen from $\mathbb{F}_q^n$, showing that $\mathbb{E}[N(x)] = \frac{M}{q}$ and $\mathrm{var}[N(x)] \leq \frac{M}{q}$. The lemma then follows by applying Chebyshev's inequality.

For the expectation, we write $N(x)$ as a sum of indicators, and use linearity of expectation

$$\mathbb{E}_x[N(x)] = \sum_{i=1}^{M} \mathbb{E}_x[\mathbb{1}_{x \in W_i}] = \sum_{i=1}^{M} \Pr_x[x \in W_i] = \frac{M}{q},$$

where the last equality follows from the cardinality of a hyperplane in $\mathbb{F}_q^n$.

For the variance, writing $N(x)$ as a sum, we get

$$\mathrm{var}[N(x)] = \sum_{i,j \in [M]} cov(\mathbb{1}_{x \in W_i}, \mathbb{1}_{x \in W_j}) = \sum_{i \in [M]} \mathrm{var}[\mathbb{1}_{x \in W_i}] + \sum_{i \neq j \in [M]} cov(\mathbb{1}_{x \in W_i}, \mathbb{1}_{x \in W_j}).$$

The first summation is at most M/q, as for each $i \in [M]$ we have $\mathrm{var}[\mathbb{1}_{x \in W_i}] = 1/q - 1/q^2 \leq 1/q$. We next show the second summation equals 0, using the premise that all hyperplanes are distinct. Indeed, for any $i \neq j$ the intersection of $W_i \neq W_j$ is a subspace of dimension $n-2$ and cardinality M/q^2. Thus, for any $i \neq j$ we get

$$\begin{aligned} cov(\mathbb{1}_{x \in W_i}, \mathbb{1}_{x \in W_j}) &= \mathbb{E}[\mathbb{1}_{x \in W_i} \cdot \mathbb{1}_{x \in W_j}] - \mathbb{E}[\mathbb{1}_{x \in W}] \cdot \mathbb{E}[\mathbb{1}_{x \in W'}] \\ &= \Pr_x[x \in W_i \cap W_j] - \Pr_x[x \in W_i] \cdot \Pr_x[x \in W_j] = 1/q^2 - (1/q)^2 = 0. \end{aligned}$$

Overall, we get that $\mathrm{var}[N(x)] \leq M/q$, as required. ◁

We are now ready to prove the sampling lemma.

Proof of Lemma 14. For each integer i, let

$$m_i := \left| \left\{ x \in \mathbb{F}_q^n \mid \frac{2^i \cdot M}{q} \leq N(x) < \frac{2^{i+1} \cdot M}{q} \right\} \right|.$$

By Claim 15, we get that $\frac{m_i}{q^n} \leq \frac{q}{(2^i-1)^2 \cdot M}$.

To complete the picture, let $m_0 = \left|\left\{x \in \mathbb{F}_q^n \mid N(x) \leq M/q\right\}\right|$. Towards the upper bound, we expand $\nu_{\mathcal{W}}(S)$ and perform a dyadic partitioning with $m_0 \leq |S|$:

$$\nu_{\mathcal{W}}(S) = \sum_{x \in S} \nu_{\mathcal{W}}(x) = \sum_{x \in S} \frac{N(x)}{M \cdot q^{n-1}} \leq \frac{1}{M \cdot q^{n-1}} \left(\frac{2 \cdot M \cdot |S|}{q} + \sum_{i=1}^{\infty} \frac{m_i \cdot 2^{i+1} \cdot M}{q} \right).$$

Once we multiply, the first term in the parenthesis contributes $2\mu(S)$, whereas the contribution of the second summand can be upper bounded by

$$\frac{1}{q^n} \cdot \sum_{i=1}^{\infty} m_i \cdot 2^{i+1} \leq \frac{q}{M} \cdot \sum_{i=1}^{\infty} \frac{2^{i+1}}{(2^i-1)^2} \leq \frac{8q}{M},$$

where the last inequality is since for all $i \geq 1$, it holds that $\frac{2^{i+1}}{(2^i-1)^2} \leq \frac{8}{2^i}$, and the infinite sum of the latter is at most 8.

For the lower bound, recall $\nu_{\mathcal{W}}(x) = \frac{N(x)}{M \cdot q^{n-1}}$ from (3). We expand

$$\nu_{\mathcal{W}}(S) = \sum_{x \in S} \nu_{\mathcal{W}}(x) \geq \sum_{\{x \in S : N(x) \geq M/(2q)\}} \nu_{\mathcal{W}}(x) \geq |\{x \in S \mid N(x) \geq M/(2q)\}| \cdot \frac{1}{2q^n}.$$

We next use Claim 15 with the parameters above, which gives $\Pr_x \left[N(x) < \frac{M}{2q} \right] \leq \frac{4q}{M}$.

It follows that $|\{x \in S \mid N_{\mathcal{W}}(x) \geq M/(2q)\}| \geq q^n \left((\mu(S) - \frac{4q}{M} \right)$. Putting everything together, we get the desired lower bound: $\nu_{\mathcal{W}}(S) \geq \frac{1}{2} \left(\mu(S) - \frac{4q}{M} \right)$. ◀

3.2 Proof of Lemma 13

Suppose $\mathcal{W} = \{W_1, \ldots, W_M\}$ is the set of distinct hyperplanes on which f is ε-close $\mathcal{C}_{n-1}$. For each W_i, let $f_i \in \mathcal{C}_{n-1}$ be this function, so $\mathsf{dist}(f_i, f|_{W_i}) \leq \varepsilon$, for each $1 \leq i \leq M$. Note that if $\varepsilon \geq 1/4$ then the result is trivial, so for the remainder of the proof we suppose $\varepsilon < \frac{1}{4}$.

We now construct the following graph $G = (V, E)$, typically referred to as the *consistency graph*. The vertex set is $V = \{1, \ldots, M\}$ while the set of edges E consists of all pairs (i, j) such that the functions f_i and f_j are consistent: $f_i|_{W_i \cap W_j} = f_j|_{W_i \cap W_j}$.

If $W_i \cap W_j$ is empty, then we also consider f_i and f_j consistent and have $(i, j) \in E$. We will first show that G contains a large clique. Call a graph *transitive* if it is an edge disjoint union of cliques. Define the following "measure" of non-transitivity:

$$\beta(G) = \max_{(i,j) \notin E} \Pr_{k \in V} [(i,k), (j,k) \in E].$$

Clearly, a graph G is transitive if and only if $\beta(G) = 0$. The following lemma from [31] gives an approximate version of this, quantifying how close is G to be transitive in terms of $\beta(G)$.

▶ **Lemma 16** ([31, Lemma 2]). *A graph G can be made transitive by removing at most $3\sqrt{\beta(G)}|V|^2$ edges.*

In our next two lemmas we show that G has many edges and then bound $\beta(G)$. Along with Lemma 16, these steps will establish that G contains a large clique. The large clique corresponds to many local degree d functions f_i that are consistent with one another, and using these functions we will extrapolate a degree d function that is close to f.

▶ **Lemma 17.** *The graph G has at least $0.4M^2$ edges.*

Proof. For each W_i, let $S_i \subseteq W_i$ be the set of points on which f_i and f differ. We have that $\frac{|S_i|}{|W_i|} \leq \varepsilon$, by assumption.

Now fix a W_i. We will apply Lemma 14 inside of W_i. For each j, let $W_j' = W_i \cap W_j$. Let $\mathcal{W}' = \left\{W_j' \mid \frac{W_j' \cap S_i}{W_j'} \geq 3\varepsilon\right\}$, let $\mathcal{I} = \{j \in [M] \mid W_j' \in \mathcal{W}'\}$, and say $M' = |\mathcal{W}'|$. Now consider the size of S_i inside of W_i under the measure $\nu_{\mathcal{W}'}$. We have, $\nu_{\mathcal{W}'}(S_i) \geq 3\varepsilon$.

By Lemma 14 $\nu_{\mathcal{W}'}(S_i) \leq 2\varepsilon + 5q/M'$.

Combining both bounds, it follows that, $M' \leq \frac{8q}{\varepsilon}$.

Finally, we can bound $|\mathcal{I}| \leq 8q^3/\varepsilon$ by noting that for any $W' \in \mathcal{W}'$, there can be at most q^2 hyperplanes W such that $W \cap W_i = W'$.

Thus, for a fixed i,

$$\Pr_{W_j}\left[\frac{|W_i \cap W_j \cap S_i|}{|W_i \cap W_j|} \geq 3\varepsilon\right] \leq \Pr_{j \in [M]}[j \in \mathcal{I}] \leq \frac{8q^3}{\varepsilon M}.$$

Now choose W_i, W_j uniformly at random. By a union bound, it follows that, with probability at least $1 - \frac{16q^3}{\varepsilon M}$, we have that both

$$\frac{|W_j \cap W_i \cap S_i|}{|W_j \cap W_i|} \leq 3\varepsilon \quad \text{and} \quad \frac{|W_j \cap W_i \cap S_j|}{|W_j \cap W_i|} \leq 3\varepsilon.$$

For such i, j, we get that

$$\Pr_{x \in W_i \cap W_j}[f_i(x) = f_j(x)] \geq 1 - \Pr_{x \in W_i \cap W_j}[x \in S_i \cup S_j] \geq 1 - 6\varepsilon > 1 - \delta_0,$$

so $f_i|_{W_i \cap W_j} = f_j|_{W_i \cap W_j}$ by (1). Thus, choosing i, j randomly, we get that they are adjacent in G with probability at least $1 - \frac{16q^3}{\varepsilon M} \geq 0.9$. ◀

▶ **Lemma 18.** *Suppose W_i, W_j are distinct hyperplanes such that $f_i|_{W_i \cap W_j} \neq f_j|_{W_i \cap W_j}$. Then,*

$$\Pr_{W_k \in \mathcal{W}}\left[f_i|_{W_i \cap W_k} \equiv f_k|_{W_i \cap W_k} \bigwedge f_j|_{W_j \cap W_k} \equiv f_k|_{W_j \cap W_k}\right] \leq \frac{1}{100}.$$

Proof. Let $W = W_i \cap W_j$. For each k let $W_k' = W_k \cap W$, let $f_i' = f_i|_W$, and let $f_j' = f_j|_W$. Also let

$$\mathcal{W}' = \{W \cap W_k \mid W_k \in \mathcal{W} \setminus \{W_i, W_j\}, f_i'|_{W_k'} = f_j'|_{W_k'}\},$$

and let $\mathcal{I} = \{k \in [M] \mid W \cap W_k \in \mathcal{W}'\}$. Note that the probability of interest is at most,

$$\Pr_{W_k \in \mathcal{W}}[f_i|_{W_i \cap W_j \cap W_k} = f_j|_{W_i \cap W_j \cap W_k}] = \frac{|\mathcal{W}'|}{|\mathcal{W}|} = \frac{|\mathcal{I}|}{M} \leq \frac{q^3|\mathcal{W}'|}{M},$$

and therefore we may focus on bounding $M' := |\mathcal{W}'|$. Above we are using the fact that for each $W' \in \mathcal{W}'$, there can be at most q^3 values k such that $W' = W \cap W_k$. This follows by noting that the dual space of W' has size q^3 and the one-dimensional dual space of W_k must be inside this space.

To bound M', we will consider W as our ambient space and apply Lemma 14 inside of W, where the set of sampling hyperplanes is $\mathcal{W}'$. Let $S \subseteq W$ be $S = \{x \in W \mid f_i'(x) \neq f_j'(x)\}$.

The key observation is that sampling $W' \in \mathcal{W}'$ and then a point $x \in W'$, the probability that $x \in S$ is actually 0. On the other hand, S is rather large since f_i', f_j' are both from a code with non-trivial distance, so together with Lemma 14, these two observations give an upper bound on the size of S.

Indeed, by the definition of $\mathcal{W}'$, we have that $\nu_{\mathcal{W}'}(S) = 0$, while on the other hand, $|S|/|W| \geq \delta_0$ by the assumption that $f'_i \neq f'_j$ combined with the definition of δ_0 from (1). Now by Lemma 14, we have $0 \geq \frac{1}{2}\left(\delta_0 - \frac{4q}{M'}\right)$. Thus, $M' \leq \frac{4q}{\delta_0}$. Since $\varepsilon \leq \delta_0/6$ and $M \geq 10^5 q^4/\varepsilon \geq 10^5 q^4/\delta_0$, we conclude that $q^3 M'/M \leq 1/100$ and the lemma follows. ◀

Lemma 18 shows that $\beta(G) \leq 1/100$, and applying Lemma 16, G can be made transitive by removing at most $0.3M^2$ edges. Combined with Lemma 17, the remaining transitive subgraph of G has at least $0.1M^2$ edges. Let $\mathcal{K}_1, \ldots, \mathcal{K}_J$ be the edge disjoint union cliques of this transitive subgraph and say $\mathcal{K}_1$ is the largest one. It follows that, $0.1M^2 \leq \sum_{I=1}^{J} |\mathcal{K}_i|^2/2 \leq |\mathcal{K}_1| \cdot M/2$, and G contains a clique of size $M_1 := |\mathcal{K}_1| = 0.2M$. From now on we let $\mathcal{K}$ denote this clique and without loss of generality let us suppose the members are $\mathcal{K} = \{1, \ldots, M_1\}$. We define the following function F using the functions $f_1, \ldots, f_{M_1}$: set $F(x) = f_j(x)$ if there is some $j \in \mathcal{K}$ such that $x \in W_j$, and $F(x) = 0$ otherwise.

We are now ready to prove the main lemma of this section.

Proof of Lemma 13. We will show that

1. F is close to f.
2. F is close to a degree d function.

Using the triangle inequality, this will establish that f is close to a degree d function.

Item 1: Proof that F is 3ε-close to f. Let $S = \{x \in \mathbb{F}^n \mid f(x) \neq F(x)\}$. Since f is ε-close to F on all hyperplanes in $\mathcal{H} := \{W_i\}_{i \in \mathcal{K}}$, we have $\nu_{\mathcal{H}}(S) \leq \varepsilon$.

By Lemma 14, it follows that

$$\varepsilon \geq \frac{1}{2}\left(\mu(S) - \frac{4q}{0.6M}\right),$$

and by our assumption on M $\mu(S) \leq 3\varepsilon$.

Item 2: Proof that F is ε-close to $\mathcal{C}_n$. We will show that F passes the t-space test with high probability. Recall that this test operates by choosing a random subspace L of dimension t and accepting if and only if $F|_L \in \mathcal{C}_t$. By Definition 12 we have,

$$\Pr_{L \in \mathcal{G}_t}[F|_L \notin \mathcal{C}_t] \geq c(\mathsf{dist}(F, \mathcal{C}_n)). \tag{4}$$

Next, we upper bound the rejection probability to bound the distance of F from the family of functions $\mathcal{C}_n$. Let $\mathcal{L} = \{L \in \mathcal{G}_t \mid \nexists i \in \mathcal{K}, W_i \supseteq L\}$.

We have

$$\Pr_{L \in \mathcal{G}_t}[F|_L \notin \mathcal{C}_t] \leq \Pr_{L \in \mathcal{G}_t}[\nexists i \in \mathcal{K}, W_i \supseteq L].$$

We will follow similar calculations to those of Claim 15. Here, for any $L \in \mathcal{G}_t$, we let

$$N(L) = |\{i \in \mathcal{K} \mid W_i \supseteq L\}|.$$

We can now express the upper bound using $N(\cdot)$.

$$\Pr_{L \in \mathcal{G}_t}[\nexists i \in \mathcal{K}, W_i \supseteq L] = \Pr_{L \in \mathcal{G}_t}[N(L) = 0].$$

To further bound the probability, we next bound the first and second moments of the random variable $N(L)$ for uniform random $L \in \mathcal{G}_t$. Let $p_1 := \Pr_{L \in \mathcal{G}_t}[L \subseteq W]$. For the expectation, note that by the linearity of expectation $\mathbb{E}_{L \in \mathcal{G}_t}[N(L)] = \sum_{i=1}^{M} \Pr_{L \in \mathcal{G}_t}[L \subseteq W_i] = p_1 M$.

For the variance analysis, similarly to p_1 we define the probability of a uniform subspace to be contained in a fixed $n-2$ dimensional subspace, which we will use as an intersection of two distinct hyperplanes W, W'.

$$p_2 := \Pr_{L \in \mathcal{G}_t}[L \subseteq W \cap W'].$$

We next bound the $\mathbb{E}_{L \in \mathcal{G}_t}\left[N(L)^2\right]$:

$$\mathbb{E}_{L \in \mathcal{G}_t}\left[N(L)^2\right] = \mathbb{E}_{L \in \mathcal{G}_t}\left[\left(\sum_{i=1}^{M} \mathbb{1}_{L \subseteq W_i}\right)^2\right]$$
$$\leq M \cdot \Pr_{L \in \mathcal{G}_t}[L \subseteq W_i] + M^2 \cdot \Pr_{L \in \mathcal{G}_t}[L \subseteq W_i \cap W_{i'}] = p_1 M + p_2 M^2 \leq p_1 M + p_1^2 M^2.$$

The last inequality follows from

$$p_2 = \frac{\begin{bmatrix} n-2 \\ t \end{bmatrix}_q}{\begin{bmatrix} n \\ t \end{bmatrix}_q} \leq \left(\frac{\begin{bmatrix} n-1 \\ t \end{bmatrix}_q}{\begin{bmatrix} n \\ t \end{bmatrix}_q}\right)^2 = p_1^2.$$

Now,

$$\Pr_{L \in \mathcal{G}_t}[N(L) = 0] \leq \frac{\mathrm{var}(N(L))}{\mathbb{E}[N(L)^2]} = 1 - \frac{\mathbb{E}[N(L)]^2}{\mathbb{E}[N(L)^2]} \leq 1 - \frac{p_1^2 M^2}{p_1 M + p_1^2 M^2} \leq \frac{1}{1 + p_1 M} \leq \frac{1}{p_1 M}.$$

Note that,

$$p_1 = \frac{\begin{bmatrix} n-1 \\ t \end{bmatrix}_q}{\begin{bmatrix} n \\ t \end{bmatrix}_q} = \frac{\prod_{i=0}^{t-1}(q^{n-1} - q^i)}{\prod_{i=0}^{t-1}(q^n - q^i)} = q^{-t}\frac{\prod_{i=1}^{t}(q^n - q^i)}{\prod_{i=0}^{t-1}(q^n - q^i)} = q^{-t}\frac{q^n - q^t}{q^n - 1} \geq \frac{1}{2q^t}.$$

By our assumption on the size of M, $\frac{1}{p_1 \cdot M} \leq \frac{2q^t}{M} \leq c(\varepsilon)$.

Combining with (4), and recall that $c(\cdot)$ is non-decreasing, we conclude $\mathsf{dist}(f, \mathcal{C}_n) \leq \varepsilon$. ◀

4 Sample-Based Testers for All Affine Invariant Families

In this section, we describe our sample-based tester that works for any fixed family of functions $\mathcal{C}_n \subseteq \{\mathbb{F}^n \to \mathbb{F}\}$. The tester samples a certain number of points and makes a decision based on this view. Since we wish the tester to have a one-sided error (always accept inputs from the family $\mathcal{C}_n$), the algorithm does the only thing possible. It rejects only when there is no codeword that can explain the sampled view.

The $\mathcal{T}$ for $\mathcal{C}_n$.

1. Sample s points from $\mathbb{F}^n$ uniformly with replacement. Let S denote the set of points sampled.
2. Accept if there is a function $g \in \mathcal{C}_n$ satisfying $g|_S = f|_S$. Otherwise, Reject.

▶ **Lemma 19.** *Let f be ε-far from $\mathcal{C}_n$, if $s \geq \frac{\ln(2|\mathcal{C}_n|)}{\varepsilon}$ then $\mathcal{T}$ rejects with probability at least $1/4$.*

Proof. For any $g \in \mathcal{C}_n$, we have $\mathsf{dist}(f, g) \geq \varepsilon$ and therefore the probability g is consistent with f on s_n uniform random samples is at most $(1-\varepsilon)^{s_n}$. Union bounding over all members of $\mathcal{C}_n$, the probability that there exists $g \in \mathcal{C}_n$ such that $f|_S \equiv g|_S$, which is the case of acceptance, is at most $(1-\varepsilon)^s|\mathcal{C}_n| \leq \exp(-\varepsilon s)|\mathcal{C}_n| \leq 1/2$. The last inequality is by the bound on s. ◀

Though the result of Lemma 19 is easy and quite general, it suits well when the tester receives the proximity parameter ε as an input and can decide how many samples to make based on the proximity parameter.

Since we are going to apply the sample-based tester on functions with varying distances, and yet we need to squeeze all the probability of rejecting non-codewords. We will use an oblivious (to the proximity parameter) algorithm; the only difference is the setting of the number of samples it uses. We show that such a tester is optimal in the sense discussed in Section 1.1.

Our approach to the analysis now is based on the *learn-and-test* paradigm. The tester operates by choosing twice the amount of points needed to interpolate a unique function from $\mathcal{C}_n$; crucially, this amount is independent of the proximity parameter ε. The first half of the points are used to reduce to one candidate function $g \in \mathcal{C}_n$, and the second half of the points are used to check if the tested function is indeed g.

▶ **Lemma 20.** *Let f be ε-far from $\mathcal{C}_n$, if $s \geq \frac{10 \ln |\mathcal{C}_n|}{\delta}$, where $\delta = \delta(\mathcal{C}_n)$ is the minimal distance of the code $\mathcal{C}_n$, then $\mathcal{T}$ rejects with probability at least* $\min\{1/4, s\varepsilon/8\}$.

Proof. Let S_1 be the first $\frac{5 \ln |\mathcal{C}_n|}{\delta}$ points sampled in S, and let S_2 be the remaining points. Let E be the event that there is at most one $g \in \mathcal{C}_n$ satisfying $g|_{S_1} = f|_{S_1}$. Then the probability that the tester rejects is at least $\Pr[E] \cdot \Pr_S[g|_{S_2} \neq f|_{S_2} \mid E]$, where in the probability g refers to the unique function in $\mathcal{C}_n$ agreeing with f on S_1. By Claim 11, $\Pr_S[\overline{E}] \leq \exp(-\delta|S_1|)|\mathcal{C}_n|^2 \leq 1/2$. Furthermore, by the assumption that f is ε-far from $\mathcal{C}_n$, we have, $\Pr_S[g|_{S_2} \neq f|_{S_2} \mid E] \geq 1 - (1-\varepsilon)^{s/2} \geq \min\left\{\frac{1}{2}, \frac{s\varepsilon}{4}\right\}$. The last inequality is proved in the proof of [5, Lemma 3.1]. The result follows. ◀

We will use Lemma 20 in order to build our semi-sample-based tester. Here we note a quick application of it, improving on the sample complexity given by Lemma 19 whenever ε is much smaller than δ_0.

▶ **Theorem 21.** *There is a sample-based tester for $\mathcal{C}_n$ with a one-sided error that uses at most $O(s + 1/\varepsilon)$ samples. Where $s = O(\ln |\mathcal{C}_n|/\delta)$ and $\delta = \delta(\mathcal{C}_n)$ is the minimal distance between two distinct codewords in $\mathcal{C}_n$.*

Proof. We claim that repeating $\mathcal{T}$ (with $s = O(\ln |\mathcal{C}_n|/\delta)$) enough times and rejecting if any of the instances rejects is a valid tester. Indeed, if f is a codeword, all samples will be consistent with f, and we always accept. If f is ε-far, then by Lemma 20 each instance accepts with probability at most $1 - \min\{1/4, s\varepsilon/8\}$. Since the instances are independent, by repeating $O\left(\frac{1}{s\varepsilon} + 1\right)$ times, the probability that they all accept is at most $1/3$. For the sample complexity, we have any instance use s samples, which in total become $O\left(s\left(\frac{1}{s\varepsilon} + 1\right)\right) = O(s + 1/\varepsilon)$. ◀

5 Semi-Sample-Based Testers

The semi-sample-based tester for $\mathcal{C}_n$ on input function $f : \mathbb{F}_q^n \to \mathbb{F}_q$ works by first choosing a random k-dimensional subspace, A, and then running the sample-based tester on the smaller, restricted function $f|_A$.

The tester $\mathcal{T}_k(Q, \mathcal{C}_n)$.

1. Choose a k-dimensional subspace $A \subset \mathbb{F}^n$ uniformly at random.
2. Sample Q points from A uniformly with replacement. Let S denote the set of points sampled.
3. Accept if there is a function $g \in \mathcal{C}_k$ satisfying $g|_S = f|_S$. Otherwise, Reject.

We will show that with large enough k, choosing Q sufficiently large with respect to the rate and distance of $\mathcal{C}_k$, the tester $\mathcal{T}_k(Q)$ is indeed a valid tester with perfect completeness and non-trivial soundness (or even optimal soundness for certain families).

▶ Remark 22. We remark that while our focus is on query complexity rather than time complexity, the tester $\mathcal{T}_k(Q, \mathcal{C}_n)$ does have good running time when $\mathcal{C}_n$ is a linear code. In this case, checking if there exists $g \in \mathcal{C}_k$ such that $g|_S = f|_S$ can be done by solving a system of $|S|$ linear equations with $\log_q |\mathcal{C}_k| \leq |S|$ variables. In particular, the time complexity is $O(|S|^3)$.

In this section, we show our main theorem regarding semi-sample-based testers for affine invariant families of functions. At a high level, our tester for *every* affine invariant family is simply the tester $\mathcal{T}_k(Q)$ except one must choose the quantity k as well as the number of queries Q depending on the specific code. For families of functions such as Reed-Muller codes or lifted affine-invariant codes, we are able to give explicit values of k and Q, and show that the semi-sample-based tester achieves optimal soundness. For general affine invariant families however, the known soundness of current subspace based testers is not strong enough for us to give explicit values of k and Q for a valid semi-sample-based tester. We state our main theorems for testing Reed-Muller codes, lifted affine-invariant codes, and finally general affine invariant families below.

We state our main result for Reed-Muller codes and subsequently prove it in Section 5.1.

▶ **Theorem 23.** *Let $f : \mathbb{F}^n \to \mathbb{F}$, let $k \geq 8d + 3c_{\mathsf{KM}} + 150$ be a dimension parameter, let $\delta_0 := q^{-d/(q-1)}$ and set*

$$s_k := \left\lceil \frac{100 \ln |\mathsf{RM}[k, q, d]|}{\delta_0} \right\rceil + 1$$

Then $\mathcal{T}_k^f(s_k, \mathsf{RM}[n, q, d])$ is a valid tester for the degree d Reed-Muller code over $\mathbb{F}^n$. Specifically, the tester satisfies the following:

- *Completeness: If $f \in \mathsf{RM}[n, q, d]$, then the tester accepts with probability 1.*
- *Soundness: If f is ε-far from $\mathsf{RM}[n, q, d]$ then the tester rejects with probability at least $\min\{s_k\varepsilon/8, 1/128\}$*

Furthermore, the tester has query complexity s_k, which is bounded by

$$s_k = O\left(q^{t_{q,d}} \cdot \left(\frac{e(d+k)}{k}\right)^k\right) . \tag{5}$$

5.1 Reed-Muller Codes: Proof of Theorem 23

As is usual in property testing, the completeness is clear. To bound the query complexity, s_k, we recall that $t_{q,d} = \lceil \frac{d}{q-q/p} \rceil + 1$, and the bounds from Fact 1, which in particular imply $\ln |\mathsf{RM}[k, q, d]| \leq \binom{d+k}{k} \cdot \ln q$, and $1/\delta_0 \leq q^{t_{q,d}-1} \leq q^{t_{q,d}}/\ln q$. Overall, this bounds the query complexity by

$$s_k = \left\lceil \frac{100 \ln |\mathsf{RM}[k, q, d]|}{\delta_0} \right\rceil + 1 \leq 100 q^{t_{q,d}} \cdot \binom{d+k}{k}. \tag{6}$$

Equation (5) follows from standard bounds on binomial coefficients.

From here on we focus on the proof of soundness. Fix any dimension parameter k such that

$$k \geq 8d + 3c_{\mathsf{KM}} + 150 \tag{7}$$

as in Theorem 23. To simplify notation, we also drop the subscript from s_k, writing $s = s_k$ instead, and write $\mathcal{T}_k^f$ in place of $\mathcal{T}_k^f(s_k, \mathsf{RM}[n,q,d])$. Also let $\varepsilon_f := \mathsf{dist}(f, \mathsf{RM}[n,q,d])$ be the actual distance from f to $\mathsf{RM}[n,q,d]$ and we assume that f is ε-far from $\mathsf{RM}[n,q,d]$, meaning $\varepsilon_f \geq \varepsilon$. Finally, we remark that by Fact 1, δ_0 is the minimum distance of $\mathsf{RM}[t,q,d]$ for all $t_{q,d} \leq t \leq n$.

Similar to the soundness analysis of the standard subspace test for Reed-Muller codes in [10], we split our soundness analysis into two cases based on if the magnitude of ε_f is small or large. The analyses of these two cases together will show the desired soundness for Theorem 23.

Before going into the proof, we state and prove two key claims. The first claim shows that s_k is sufficiently smaller than q^k.

▷ Claim 24. Fix any constant $c > 0$. For any $k \geq 8d + 3c + 24$, it holds that $\frac{s_k}{q^k} \leq q^{-c}$.

Proof. Define $B_k := 100q^{t_{q,d}} \cdot \binom{d+k}{d}$, the upper bound on s_k from (6). We focus on the expression B_k/q^k instead, bound it for a proper dimension k_0, and show it decreases multiplicatively by a significant factor as k increases until for certain $k(c)$ the whole quotient is small enough, immediately implying the lemma. One "step" changes the quotient by a factor of

$$\frac{\frac{B_{k+1}}{q^{k+1}}}{\frac{B_k}{q^k}} = \frac{1}{q} \cdot \frac{B_{k+1}}{B_k} = \frac{1}{q} \cdot \left(1 + \frac{d}{k+1}\right) , \tag{8}$$

where the second equality is due to quotient of "consecutive" binomial coefficients.

From here on, We split into two cases.

The case $q = 2$. Here, it holds that $t_{q,d} = d + 1$. We start from dimension $k_0 = 2d$ and plug in q and $t_{q,d}$ to get:

$$\frac{B_{2d}}{q^{2d}} \leq 100q^{t_{q,d}-2d} \cdot \binom{3d}{d} \leq 100q^{1-d} \cdot 2^{3d} \leq 2^{2d+8} .$$

By (8), for any choice of $k \geq 2d$ the quotient diminishes by a factor of at most

$$\frac{\frac{B_{k+1}}{q^{k+1}}}{\frac{B_k}{q^k}} = \frac{1}{q} \cdot \left(1 + \frac{d}{k+1}\right) < 2^{-1/3} .$$

Applying the above for $z \geq 3(2d + c + 8)$ steps, we get

$$\frac{B_{2d+z}}{q^{2d+z}} \leq \frac{B_{2d}}{q^{2d}} \cdot \left(2^{-1/3}\right)^z \leq 2^{2d+8-z/3} \leq 2^{-c} .$$

That is, for any dimension $k = 2d + z \geq 8d + 3c + 24$ the assertion holds for $q = 2$.

The case $q > 2$. Here we have $t_{q,d} \leq d$. Plugging this and $q \geq 3$, we directly calculate for $k_0 = d$:

We next bound the quotient for $k = d$ (for $q > 2$ we have $t_{q,d} \leq d$):

$$\frac{B_d}{q^d} \leq 100q^{t_{q,d}-d} \cdot \binom{2d}{d} \leq 100 \cdot 4^d \leq q^{2d+5} .$$

By (8), for any $k \geq d$, the quotient diminishes by a factor of

$$\frac{\frac{B_{k+1}}{q^{k+1}}}{\frac{B_k}{q^k}} = \frac{1}{q} \cdot \left(1 + \frac{d}{k+1}\right) < \frac{2}{q} < q^{-1/3} \ .$$

Applying the above iteratively for $z \geq 3(2d + c + 5)$ steps, we get $\frac{B_{d+z}}{q^{d+z}} \leq \frac{B_d}{q^d} \cdot q^{-z/3} \leq q^{2d+5-z/3} \leq q^{-c}$, concluding that for any dimension $k = d + z \geq 7d + 3c + 15$ the assertion holds for $q > 2$. ◁

The second claim is a specialization of Lemma 13 to degree d Reed-Muller codes with the parameters that we will use.

▷ Claim 25. Suppose there are $M = 10^6 q^{\max\{4, c_{\text{KM}}\}} \cdot s$ distinct hyperplanes $W_1, \ldots, W_M \subset \mathbb{F}^n$ such that for each $i = 1, \ldots, M$, $f|_{W_i}$ is $\frac{1}{8s}$-close to a degree d function on W_i. Then f is $\frac{1}{2s}$-close to a degree d function.

Proof. We apply Lemma 13 with the ε therein set to $\frac{1}{8s}$ (which indeed satisfies $\frac{1}{8s} \leq \frac{\delta_0}{6}$). The conclusion suffices for our proof. It therefore remains to verify that M is large enough to satisfy the premise of Lemma 13. In particular, we check that

$$M \overset{?}{\geq} \max\left\{\frac{10^5 q^4}{1/(8s)}, \frac{20 q^{t_{q,d}}}{c\left(\frac{1}{8s}\right)}\right\}, \tag{9}$$

where $c(\cdot)$ is the soundness function of Definition 12 for Reed-Muller codes.

Clearly, we have $M \geq \frac{10^5 q^4}{1/(8s)}$, which leaves the need to verify M is larger than the second term. By Theorem 8, the $t_{q,d}$-flat test has soundness function $c(\varepsilon) = q^{-c_{\text{KM}}} \cdot \min\{1, q^{t_{q,d}}\varepsilon\}$ for Reed-Muller codes, and in particular $c\left(\frac{1}{8s}\right) = q^{-c_{\text{KM}}} \cdot \frac{q^{t_{q,d}}}{8s}$, since $8s \geq q^{t_{q,d}}$. This implies

$$\frac{20 q^{t_{q,d}}}{c\left(\frac{1}{8s}\right)} = 20 q^{t_{q,d}} \cdot q^{c_{\text{KM}}} \cdot \frac{8s}{q^{t_{q,d}}} = 160 q^{c_{\text{KM}}} \cdot s \leq M \ ,$$

which shows our choice of M indeed satisfies (9), concluding the proof. ◁

5.1.1 Small distance

▶ **Lemma 26.** *If $\varepsilon_f \leq \frac{1}{2s}$ then $\Pr\left[\mathcal{T}_k^f \text{ rejects}\right] \geq \frac{s\varepsilon_f}{8} \geq \frac{s\varepsilon}{8}$.*

Proof. Let $g^* \in \mathsf{RM}[n, q, d]$ be such that $\mathsf{dist}(f, g) = \varepsilon_f$. Let S_1 be the first $1/10$ of the points chosen in S and S_2 be the remaining $9/10$ of the points chosen in S, so $|S_2| = 9s/10$.

Let $\Delta = \{x \in \mathbb{F}^n \mid f(x) \neq g^*(x)\}$. The tester proceeds by choosing a linear subspace $U \subset \mathbb{F}^n$ of dimension k and then sampling $S_1, S_2 \subseteq U$.

For our analysis, it will be convenient to think of S as being chosen as follows.

1. Fix $U \subseteq \mathbb{F}^n$ to be a generic k-dimensional subspace of dimension U and choose $S_1' \subseteq U$ of size $|S_1|$.
2. Choose $S_2' \subseteq U$ of size $|S_2|$.
3. Choose a random, full-rank linear transformation $T : \mathbb{F}^n \to \mathbb{F}^n$.
4. Set $S_i = \{T(x) \mid x \in S_i\}$ for $i = 1, 2$.
5. Set $S = S_1 \cup S_2$.

In the above view of $\mathcal{T}_k^f$, the k-dimensional subspace fixed by the tester is $T \circ U$, and the points queried are $S \subseteq T \circ U$. To analyze the rejection probability, we define the following events:

- E_1: the event that every pair of degree d functions on U (these correspond to $\mathsf{RM}[k,q,d]$) disagree on at least one point in S_1'.
- E_2: the event that f and g^* agree on S_1, i.e. $f|_{S_1} \equiv g^*|_{S_1}$.
- E_3: the event that $S_2 \cap \Delta \neq \emptyset$.

One can check that if all three events above occur, then the tester rejects. Indeed, if E_1 and E_2 both occur, then $g^*|_{T\circ U} \in \mathsf{RM}[k,q,d]$ is the unique degree d function over $T \circ U$ that agrees with f on S_1. The tester will then reject if g^* and f disagree on any point on S_2 – in other words, if E_3 occurs. Thus,

$$\Pr[\mathcal{T}_k^f \text{ rejects}] \geq \Pr_{S_1,S_2}[E_1 \wedge E_2 \wedge E_3] = \Pr_{S_1'}[E_1] \cdot \Pr_{S_1',S_2',T}[E_2 \wedge E_3 \mid E_1].$$

We now bound each term appearing above. First note that E_1 does not depend on the identity of the subspace U. By Claim 11, we have

$$\Pr_{S_1'}[E_1] \geq 1 - e^{-\delta_0|S_1|}|\mathsf{RM}[k,q,d]|^2 \geq \frac{1}{2}.$$

For the second term, we have

$$\Pr_{S_1',S_2',T}[E_2 \wedge E_3 \mid E_1] \geq \Pr_{S_1',S_2',T}[E_3 \mid E_1] - \Pr_{S_1',S_2',T}[\overline{E_2} \mid E_1].$$

We will bound each term on the right-hand side. Let us suppose S_1' and S_2' have already been sampled, and now bound the probabilities over random T.

For the first term, we use the principle of inclusion-exclusion to get a lower bound:

$$\begin{aligned}\Pr_T[E_3 \mid E_1] &\geq \sum_{x\in S_2'\setminus\{0\}} \Pr_T[T(x)\in\Delta] - \sum_{x,y\in S_2'\setminus\{0\}, x\neq y,} \Pr_T[T(x),T(y)\in\Delta] \\ &\geq \frac{9s/10-1}{2}\varepsilon_f - 2\frac{(9s/10-1)^2}{4}\varepsilon_f^2 \geq \frac{3s\varepsilon_f}{8}\ .\end{aligned}$$

In the first transition we are using the fact that $T(x)$ is uniformly random over random T and the fact that $T(x), T(y)$ are nearly independent for $x \neq y$. In the last transition we are using the fact that $s\varepsilon_f/4 \leq 1/8$ by assumption.

To upper bound the second term (again as a probability over T, fixing S_1', S_2'), we use a union bound:

$$\Pr_{S_1',S_2',T}[\overline{E_2} \mid E_1] \leq \sum_{x\in S_1'} \Pr_T[T(x)\in\Delta] \leq \frac{s\varepsilon_f}{10}.$$

Putting everything together, we get

$$\Pr[\mathcal{T}_k^f \text{ rejects}] \geq \Pr_{S_1'}[E_1] \cdot \Pr_{S_1',S_2',T}[E_2 \wedge E_3 \mid E_1] \geq \frac{1}{2}\cdot\frac{s\varepsilon_f}{4} \geq \frac{s\varepsilon_f}{8}\ .$$ ◀

5.1.2 Large distance

We have already shown that the tester has the desired soundness when $\varepsilon_f \leq \frac{1}{2s}$. To complete the proof of Theorem 23, it remains to show the desired soundness when $\varepsilon_f > \frac{1}{2s}$. In particular, the following lemma suffices.

▶ **Lemma 27.** *Set $\eta := \frac{1}{128}$ and $\gamma := 10^7 q^{\max\{4,c_{\mathsf{KM}}\}} \cdot s$. The following holds for every $n \geq k$. Let $f\colon \mathbb{F}_q^n \to \mathbb{F}_q$ be such that $\mathsf{dist}(f, \mathsf{RM}[n,q,d]) = \varepsilon_f \geq 16\eta/s$. Then*

$$\Pr\left[\mathcal{T}_k^f \text{ rejects}\right] \geq \eta + \frac{\gamma}{q^n}.$$

We will analyze the large distance case by induction of n. Consider the following tester $\mathcal{T}'_k$: the tester first picks a (linear) hyperplane W and then simulates $\mathcal{T}_k^{f|_W}$. That is, the tester chooses a random k-dimensional subspace $U \subseteq W$ and then performs the sample-based tester inside of U. We note that U is still a uniformly random k-dimensional subspace in $\mathbb{F}^n$, so it follows that

$$\Pr[\mathcal{T}_k^f \text{ rejects}] = \mathbb{E}_W \left[\Pr[\mathcal{T}_k^{f|_W} \text{ rejects}]\right]. \tag{10}$$

Therefore, in order to conclude that the rejection probability is high, we will invoke Claim 25 to say that for nearly all of the hyperplanes W, the distance of $f|_W$ to $\mathcal{C}_{n-1}$ is still within a constant factor of ε_f. As $f|_W$ and $\mathcal{C}_{n-1}$ are one dimension down, we can conclude that $\mathcal{T}_k^{f|_W}$ rejects with good probability. Since Claim 25 allows us to conclude $\mathsf{dist}(f|_W, \mathcal{C}_{n-1})$ can only increase by more than a constant factor for a number of W that is independent of the dimension n, the above argument is sufficient for the inductive step.

Proof of Lemma 27. The proof is by induction on the ambient dimension, n. In the base case, $n = k$, and $\mathcal{T}_k^f$ is simply the sample-based tester of Lemma 20. Therefore, $\mathcal{T}_k^f$ rejects with probability at least $\min\{1/4, s\varepsilon_f/8\} \geq 2\eta$, using the premise and the definition of η. Next, apply Claim 24 with $c = \max\{4, c_{\mathsf{KM}}\} + 31$, noting that by (7), we have large enough k, yielding

$$\frac{\gamma}{q^k} = \frac{10^7 q^{\max\{4,c_{\mathsf{KM}}\}} \cdot s_k}{q^k} \leq 10^7 q^{\max\{4,c_{\mathsf{KM}}\}} q^{-c} < \frac{1}{128} = \eta,$$

which implies rejection probability of at least $2\eta \geq \eta + \gamma/q^n$, concluding the base case.

For the induction step, suppose that the lemma holds for testing all functions over $\mathbb{F}_q^{n'}$ that are sufficiently far from $\mathsf{RM}[n', q, d]$, for all $k \leq n' < n$. We will show that the lemma also holds for $f : \mathbb{F}_q^n \to \mathbb{F}_q$ with $\varepsilon_f \geq 16\eta/s$.

Let $\mathcal{H}$ denote the set of all hyperplanes in $\mathbb{F}^n$ and let $N = |\mathcal{H}| = (q^n - 1)/(q - 1)$. Define

$$\mathcal{H}^* = \{W \in \mathcal{H} \ : \ \mathsf{dist}(f|_W, \mathcal{C}_{n-1}) < 16\eta/s\},$$

and let $K = |\mathcal{H}^*|$. Using (10) and the induction hypothesis, we can decompose the rejection probability as follows:

$$\begin{aligned}\Pr[\mathcal{T}_k^f \text{ rejects}] &= \mathbb{E}_{W\in\mathcal{H}} \left[\Pr[\mathcal{T}_k^{f|_W} \text{ rejects}]\right] \\ &\geq \mathbb{E}_{W\in\mathcal{H}\setminus\mathcal{H}^*} \left[\Pr[\mathcal{T}_k^{f|_W} \text{ rejects}]\right] - \Pr_{W\in\mathcal{H}}[W \in \mathcal{H}^*] \geq \eta + \frac{\gamma}{q^{n-1}} - \frac{K}{N},\end{aligned}$$

where we are applying the induction hypothesis on $f|_W$ in the third transition.

Case 1: $K \leq \gamma \cdot \frac{q^n-1}{q^n}$. In this case, $\frac{K}{N} \leq \gamma \cdot \frac{q-1}{q^n}$. Then, $\Pr[\mathcal{T}_k^f \text{ rejects}] \geq \eta + \frac{\gamma}{q^{n-1}} - \frac{K}{N} \geq \eta + \frac{\gamma}{q^n}$, and we are done.

Case 2: $K > \gamma \cdot \frac{q^n-1}{q^n}$. In this case $K > \gamma/2 > 10^6 q^{\max\{4,c_{\mathsf{KM}}\}} \cdot s$, which means there are at least this many hyperplanes $W \in \mathcal{H}$ on which $f|_W$ is $\frac{1}{8s}$-close to degree d (by the definition of $\mathcal{H}^*$ and $\eta = 1/128$). This allows us to invoke Claim 25 and deduce that $\varepsilon_f \leq 1/(2s)$. Finally, applying Lemma 26 shows that $\Pr[\mathcal{T}_k^f \text{ rejects}] \geq \frac{s\varepsilon_f}{8} \geq \eta + \frac{\gamma}{q^n}$, where the second inequality uses the same argument as in the induction base case. ◀

6 Application to Testing with an Online-Adversary

We show that Theorem 23 lead to a tester for Reed-Muller codes in the online-adversary model. In the full version, we give a formal statement and proof of Theorem 4 and then show how it leads to a tester for lifted affine-invariant codes. Also in the full version, for more general affine-invariant families, we state and prove an analogue of Theorems 1 and 4, making the requisite modifications to handle general affine-invariant families that admit an efficient flat tester of bounded dimension. We further show that this result also gives a tester in the online adversary model. However, we are not aware of any such families for which the currently known testers have strong enough soundness for our theorem to apply.

The online-manipulation-resilient testing model was originally proposed by Kalemaj, Raskhodnikova, and Varma [19], and with extended variants was later formally defined by Ben-Eliezer, Kelman, Meir, and Raskhodnikova [5]. We follow the definitions of the latter.What matters for our discussion is that the model provides oracle access to the data via an online adversarial oracle. Specifically, given an erasure-rate parameter t, an adversary is allowed to erase (or, in a more challenging variant of the model, corrupt) up to t data points in the input data *after* each query is answered.

6.1 Optimal Online-Manipulation-Resilient Tester for Reed-Muller

By setting the dimension of the semi-sample-based tester large enough relative to the erasure rate t, we obtain testers for Reed-Muller codes in the t-online erasure model.

▶ **Theorem 28.** *Let ε be a distance parameter, for any erasure rate $t \le \varepsilon^2 q^{n-O(d)}$ and set $k_{\mathsf{adv}} = O(d) + \log_q(t/\varepsilon^2)$ sufficiently large. Then given an input function $f : \mathbb{F}^n \to \mathbb{F}$, there is a tester for $\mathsf{RM}[n,q,d]$ with the following parameters in the t-online erasure model satisfying:*

- *Completeness: If $f \in \mathsf{RM}[n,q,d]$, the tester always accepts.*
- *Soundness: If f is ε-far from $\mathsf{RM}[n,q,d]$, then $\Pr[\mathcal{T}_k^f \text{ rejects}] \ge \frac{2}{3}$.*
- *Query Complexity:*

$$Q = \max\left\{\frac{1}{\varepsilon}, 128 s_{k_{\mathsf{adv}}}\right\} = O\left(\frac{1}{\varepsilon} + q^{2d}\left(1 + O\left(\frac{\log_q(t/\varepsilon)}{d}\right)\right)^d\right)$$

Proof. The tester is the semi-sample-based tester $\mathcal{T}_k^f(Q, \mathsf{RM}[n,q,d])$ from Section 5 with $k = k_{\mathsf{adv}}$. For the online erasure model, we also make the following slight modification: if any of the queried points is erased (i.e., answered with $\perp$), the tester will stop and accept. This modification maintains completeness while degrading the soundness by an amount that we will see is negligible.

By Theorem 23, we can repeat the tester $O\left(\frac{1}{s_k \varepsilon} + 1\right)$ times and reject with probability at least $9/10$ if no erasure is ever queried, so it remains for us to upper bound the probability of querying an erasure. The total number of queries made is $Q = O(1/\varepsilon + s_k)$ queries.

It remains to bound the probability that the tester queries a point that is erased. We show that this probability is at most $1/5$ completing the proof. Indeed, since the erasure rate is t, the total number of erasure made during the run of the algorithm is at most tQ. Since every query is uniform within a k-space. the probability for each query to be erased is at most tQ/q^k, by a union bound, the probability that any of the Q queries is erased is at most

$$tQ^2/q^k = O\left(\frac{t}{q^k}\left(\frac{1}{\varepsilon^2} + s_k^2\right)\right). \tag{11}$$

Note that for $k > d$ we have $s_k \leq q^{2d}(10k/d)^d$. For large enough $k_{\mathsf{adv}} = O(d + \log_q(t/\varepsilon))$ we have that this quantity is at most,

$$O\left(\frac{t}{q^k}\left(\frac{1}{\varepsilon^2} + s_k^2\right)\right) \leq O\left(\frac{t}{q^k\varepsilon^2}\right) + O\left(\frac{t}{q^k}\left(\frac{10qk}{d}\right)^{4d}\right) < 1/5.$$

It follows that an ε-far function is accepted with probability at most $1/10 + 1/5 \leq 1/3$, as required. ◀

Using an observation by [19], we get a similar theorem for the online corruption model, wherein the adversary can change the value of f at a point rather than just erase the value at that point. Now, with a two-sided error, an error may occur whenever the tester queries a manipulated point in either case. The formal statement and its proof are provided in the full version.

7 Further Directions

Our work leaves a number of directions open for future work. We list these below.

- Improved Soundness for [22]: As mentioned in our introduction, there is a general result which shows any local characterization for an affine invariant family also yields a local tester [22]. The soundness shown for these testers is sub-optimal however, so it would be interesting to see if our techniques could be useful towards improving the soundness of the general testers of [22].
- Testing Other Affine Invariant Families: It would be interesting to see if our techniques can yield new testing results for other affine invariant families of functions, particularly the nonlinear ones considered in [8]. As mentioned, our analysis falls short in that the hyperplane agreement lemma currently relies on having an r-space tester with decent soundness, as opposed to the inexplicit soundness given in [8].

References

1 Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing Reed-Muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005. doi:10.1109/TIT.2005.856958.

2 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs; A new characterization of NP. In *33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, October 24-27, 1992*, pages 2–13. IEEE Computer Society, 1992. doi:10.1109/SFCS.1992.267824.

3 Vipul Arora, Esty Kelman, and Uri Meir. On optimal testing of linearity. In Ioana Oriana Bercea and Rasmus Pagh, editors, *2025 Symposium on Simplicity in Algorithms, SOSA 2025, New Orleans, LA, USA, January 13-15, 2025*, pages 65–76. SIAM, 2025. doi:10.1137/1.9781611978315.5.

4 Paul Beame, Shayan Oveis Gharan, and Xin Yang. On the bias of reed-muller codes over odd prime fields, 2018. arXiv:1806.06973.

5 Omri Ben-Eliezer, Esty Kelman, Uri Meir, and Sofya Raskhodnikova. Property testing with online adversaries. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ITCS.2024.11.

6 Amey Bhangale, Irit Dinur, and Inbal Livni Navon. Cube vs. cube low degree test. *CoRR*, abs/1612.07491, 2016. arXiv:1612.07491.

7 Amey Bhangale and Silas Richelson. Plane vs. plane low degree test. Technical Report 126, Electronic Colloquium on Computational Complexity (ECCC), 2025.

8 Arnab Bhattacharyya, Eldar Fischer, Hamed Hatami, Pooya Hatami, and Shachar Lovett. Every locally characterized affine-invariant property is testable. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 429–436. ACM, 2013. doi:10.1145/2488608.2488662.

9 Arnab Bhattacharyya, Eldar Fischer, and Shachar Lovett. Testing low complexity affine-invariant properties. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1337–1355. SIAM, 2013. doi:10.1137/1.9781611973105.97.

10 Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of Reed-Muller codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, Las Vegas, Nevada, USA, October 23-26, 2010*, pages 488–497, 2010. doi:10.1109/FOCS.2010.54.

11 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993. doi:10.1016/0022-0000(93)90044-W.

12 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

13 Oded Goldreich and Dana Ron. On sample-based testers. *ACM Trans. Comput. Theory*, 8(2), April 2016. doi:10.1145/2898355.

14 Alexander Golovnev, Zeyu Guo, Pooya Hatami, Satyajeet Nagargoje, and Chao Yan. Hilbert functions and low-degree randomness extractors, 2024. doi:10.48550/arXiv.2405.10277.

15 Alan Guo, Swastik Kopparty, and Madhu Sudan. New affine-invariant codes from lifting. In Robert D. Kleinberg, editor, *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 529–540. ACM, 2013. doi:10.1145/2422436.2422494.

16 Elad Haramaty, Noga Ron-Zewi, and Madhu Sudan. Absolutely sound testing of lifted codes. *Theory Comput.*, 11:299–338, 2015. doi:10.4086/toc.2015.v011a012.

17 Elad Haramaty, Amir Shpilka, and Madhu Sudan. Optimal testing of multivariate polynomials over small prime fields. *SIAM Journal on Computing*, 42(2):536–562, 2013. doi:10.1137/120879257.

18 Prahladh Harsha, Mrinal Kumar, Ramprasad Saptharishi, and Madhu Sudan. An improved line-point low-degree test. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 1883–1892. IEEE, 2024. doi:10.1109/FOCS61266.2024.00113.

19 Iden Kalemaj, Sofya Raskhodnikova, and Nithin Varma. Sublinear-time computation in the presence of online erasures. *Theory of Computing*, 19(1):1–48, 2023. doi:10.4086/TOC.2023.V019A001.

20 Tali Kaufman and Dor Minzer. Improved optimal testing results from global hypercontractivity. *SIAM Journal on Computing*, 54(3):625–663, 2025. doi:10.1137/23M1571022.

21 Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM Journal on Computing*, 36(3):779–802, 2006. doi:10.1137/S0097539704445615.

22 Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 403–412, 2008. doi:10.1145/1374376.1374434.

23 Esty Kelman, Ephraim Linder, and Sofya Raskhodnikova. Online versus offline adversaries in property testing. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference, ITCS 2025, January 7-10, 2025, Columbia University, New York, NY, USA*, volume 325 of *LIPIcs*, pages 65:1–65:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ITCS.2025.65.

24 Esty Kelman, Uri Meir, Debanuj Nayak, and Sofya Raskhodnikova. Homomorphism testing with resilience to online manipulations. *arXiv preprint arXiv:2511.23363*, 2025. doi:10.48550/arXiv.2511.23363.

25 Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. *Annals of Mathematics*, 198(1):1–92, 2023.

26 Dor Minzer and Kai Zheng. Approaching the soundness barrier: A near optimal analysis of the cube versus cube test. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 2761–2776. SIAM, 2023. doi:10.1137/1.9781611977554.CH104.

27 Dor Minzer and Kai Zhe Zheng. Optimal testing of generalized reed-muller codes in fewer queries. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 206–233. IEEE, 2023. doi:10.1109/FOCS57990.2023.00022.

28 Dor Minzer and Kai Zhe Zheng. Adversarial low degree testing. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 4395–4409. SIAM, 2024. doi:10.1137/1.9781611977912.154.

29 Dor Minzer and Kai Zhe Zheng. Near optimal alphabet-soundness tradeoff pcps. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 15–23. ACM, 2024. doi:10.1145/3618260.3649606.

30 Dana Moshkovitz and Ran Raz. Sub-constant error low degree test of almost-linear size. *SIAM Journal on Computing*, 38(1):140–180, 2008. doi:10.1137/060656838.

31 Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 475–484. ACM, 1997. doi:10.1145/258533.258641.

32 Noga Ron-Zewi and Madhu Sudan. A New Upper Bound on the Query Complexity of Testing Generalized Reed-Muller Codes. *Theory of Computing*, 9(25):783–807, 2013. doi:10.4086/TOC.2013.V009A025.

33 Alek Westover, Edward Yu, and Kai Zheng. New direct sum tests. *CoRR*, abs/2409.10464, 2024. doi:10.48550/arXiv.2409.10464.

Fixed-Parameter Degree Bounds and Complexity of the Orbit Closure Intersection Problem for Tensors

M. Levent Doğan
LMU Munich, Germany

John Maar
Technical University Berlin, Germany

Rafael Oliveira
University of Waterloo, Canada

Youming Qiao
University of Technology Sydney, Australia

Abstract

The orbit closure intersection problem for a reductive group action is a geometric relaxation of the orbit equality problem. These problems capture a range of isomorphism, degeneration and identity-testing problems in computational complexity. We study this problem for the tensor action of $G = \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_m(\mathbb{K})$ on $\mathcal{V} = \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$ (equivalently, on m-tuples of $n \times n$-matrices) where the base field is algebraically closed with characteristic zero.

We focus on the fixed-parameter regime where m is constant and n is allowed to grow. The case of $m = 3$ is already interesting in the context of tensor rank and matrix multiplication. Prior to our work, only a special case of this problem, namely when one of the input tensors is the zero-tensor (this corresponds to the null cone problem for the tensor action), was known to be solvable in polynomial time (Bürgisser–Franks–Garg–Oliveira–Walter–Wigderson, *FOCS*'19).

Our main result is to show that the orbit closure intersection problem for the above action can be solved in randomized polynomial time. This is achieved by the following new ingredients:

1. We prove an explicit fixed-parameter bound on the degrees needed to generate the invariant ring for the tensor action: for every fixed m, these bounds are polynomial in n.
2. We prove that there is a succinct encoding of the invariants: we construct a uniform, polynomial-sized arithmetic circuit generating all invariants up to the required degree.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Computing methodologies → Algebraic algorithms

Keywords and phrases computational invariant theory, geometric complexity theory, orbit closure intersection problem

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.32

Funding *M. Levent Doğan*: The author was funded by European Union (ERC Grant SYMOPTIC, 101040907) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, 556164098).
John Maar: The author was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, 556164098).
Youming Qiao: Youming's research is partly supported by ARC LP220100332.

Acknowledgements The authors are grateful to Peter Bürgisser for valuable discussions during the Simons Program. This work started while the authors were supported by the Simons Institute for the Theory of Computing, and conducted when the authors were visiting the Institute. The authors would like to thank the generosity, hospitality, and excellent research environment provided by the Simons Institute.

© M. Levent Doğan, John Maar, Rafael Oliveira, and Youming Qiao;
licensed under Creative Commons License CC-BY 4.0

1 Introduction

Several decision problems in complexity theory ask whether two given objects are equivalent up to a given set of transformations. For many of these problems, these transformations are captured by the action of a group G on a set $\mathcal{V}$, and the equivalence question becomes the orbit problem for this action, which asks whether $u, v \in \mathcal{V}$ are *equivalent* under the action of the group G, that is, whether u and v are in the same orbit. For instance, a flagship problem in this family is the graph isomorphism problem, which asks whether two graphs are the same up to permuting the vertices. Other examples of orbit problems are the equivalence of quiver representations [16, 17], classification of quantum states under LOCC and SLOCC operations [3, 4, 13], ring/algebra/module isomorphism [38, 6, 25], tensor isomorphism [27, 28, 29], linear code equivalence [47] and group isomorphism [26].

When the group G and the set $\mathcal{V}$ admit a natural geometric structure, a geometric version of the orbit problem is the orbit closure intersection problem, which asks whether the topological (Zariski) closures of the orbits of $u, v \in \mathcal{V}$ intersect. Orbit closure intersection, together with its important special case – the null cone problem with one of the vectors is taken to be the zero vector – has its origins in invariant theory, tracing back to the celebrated work of Hilbert on the finite generation of invariant rings for reductive groups [31, 32], and Mumford's development of geometric invariant theory [43].

In computational complexity, the algorithmic study of orbit closure intersection problems was largely motivated by algebraic complexity theory. In [33], Hrubeš and Wigderson discovered a close relation between non-commutative rational identity testing and the null cone problem for the left-right action on matrix tuples. This connection has led to the development and rigorous analysis of polynomial time algorithms, via invariant theory, for the rational identity testing problem [23, 36]. Another connection comes from a closely related problem, the orbit closure containment problem, which asks whether the orbit closure of u is contained in the orbit closure of v. Orbit closure containment problems encompass (geometric strengthenings of) major problems in algebraic complexity theory, including the algebraic complexity of matrix multiplication [12] and the VP vs VNP problem [42]. In [41], Mulmuley proposed the succinct encoding framework for invariant rings and presented a general methodology for using polynomial identity testing to devise algorithms for orbit closure intersection problems.

Before proceeding further, we formally introduce the orbit closure intersection problem.

▶ **Problem 1** (Orbit Closure Intersection). *The **Orbit Closure Intersection (OCI)** problem is defined as follows: given as input an action of a linearly reductive group G on the vector space $\mathcal{V}$[1] and two vectors $u, v \in \mathcal{V}$, decide whether $\overline{G \cdot u} \cap \overline{G \cdot v} \neq \emptyset$.*

▶ Remark 2. When $v = 0$ in Problem 1, the Orbit Closure Intersection problem is known as the **null cone** problem: decide if $0 \in \overline{G \cdot u}$.

In the past decade, progress was made on the various instances of the orbit closure intersection problem. Notable examples where a polynomial time algorithm is available are the orbit closure intersection problem for matrix tuples under simultaneous conjugation [21, 17], under the left-right action [2, 17, 34], as well as the action of a product of general linear groups on quiver representations, the actions of tori [9, 10], and when the group G is fixed [41].

[1] We will assume in this paper that the group action belongs to a parameterized family of group actions, such as the left-right action of $\mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K})$ on $\mathrm{Mat}(n, \mathbb{K})^{\oplus m}$, with the action is specified by the input parameter, e.g. by the pair (n, m) for the left-right action.

Since they play an important role in this work, we recall that the left-right action of $(g, h) \in \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K})$ sends a matrix tuple $(A_1, \ldots, A_m) \in \mathrm{Mat}(n, \mathbb{K})^m$ to $(gA_1h^T, \ldots, gA_mh^T)$. Here, and in the rest of the paper, $\mathbb{K}$ denotes an algebraically closed field with characteristic zero. The orbit closure intersection problem for this action has intriguing applications in areas such as identity testing non-commutative rational formulas [33] and geometric complexity theory [41], and it naturally generalizes classical combinatorial problems such as the bipartite perfect matching problem and the linear matroid intersection problem [23, 35]. The null cone problem for the left-right action admits three deterministic polynomial-time solutions [23, 36, 30], and so does its orbit closure intersection problem [2, 17, 34]. These results and the algorithmic techniques developed for solving them demonstrate connections between different areas including invariant theory, analysis, optimization, and non-commutative algebra, as seen in the surveys by Bürgisser [7] and Wigderson [49].

After these positive results for various group actions, the next natural and important target is to study tensors with the natural action by direct products of reductive algebraic groups. For example, we consider the action of $\mathrm{SL}_a(\mathbb{K}) \times \mathrm{SL}_b(\mathbb{K}) \times \mathrm{SL}_c(\mathbb{K})$ on the space of tensors $\mathbb{K}^a \otimes \mathbb{K}^b \otimes \mathbb{K}^c$, with its action on the standard basis given by

$$(g, h, k) \cdot e_i \otimes e_j \otimes e_l = (ge_i) \otimes (he_j) \otimes (ke_l), \quad (g, h, k) \in \mathrm{SL}_a(\mathbb{K}) \times \mathrm{SL}_b(\mathbb{K}) \times \mathrm{SL}_c(\mathbb{K}),$$

or the action of $\mathrm{GL}_n(\mathbb{K})$ on a space $\bigoplus_{i=1}^{a} (\mathbb{K}^n)^{\otimes c_i} \oplus \bigoplus_{i=1}^{b} ((\mathbb{K}^n)^*)^{\otimes d_i}$ of tensors.[2] Such actions are important due to their connections and applications to other areas, such as tensor networks in quantum information [1], tensor rank and matrix multiplication [48, 5, 8], cryptography [37], and isomorphism problems for groups and polynomials [27]. Interestingly, the orbit closure intersection problem for these actions appears to be much harder, with evidence from invariant theory [18], algebraic complexity [24], and optimization [22]. The recent result [40, Theorem 1.5] gives a partial explanation, namely the orbit closure intersection problem for the natural $\mathrm{GL}_n(\mathbb{K})$ action on $(\mathbb{K}^n)^{\otimes 2} \oplus (\mathbb{K}^n)^{\otimes 3} \oplus ((\mathbb{K}^n)^*)^{\otimes 2}$, as well as for certain other natural tensor actions, is graph isomorphism hard.

Given these difficulties, it is intriguing to see that the null cone problem for $\mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$ where $m \in O(1)$ is a constant can be solved in deterministic polynomial time [11, Corollary 6.24]. This case is also of great interest in tensor rank and matrix multiplication even for $m = 3$: Strassen proved that certain invariants in $\mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^3$ can be elevated via the substitution method to prove lower bounds for the border rank of the matrix multiplication [48, 5], see also [11, Section 1.7.1]. However, before this work, the orbit closure intersection problem was only known to be in exponential time. Recently, Bürgisser asked if the orbit closure intersection problem admits a deterministic polynomial-time algorithm; see Problem 2.4 in the open problems list from the recent Workshop on Randomness, Invariants, and Complexity at the Simons Institute in 2025. Note that the distinction between null cone and orbit closure intersection could be significant, as already seen in the case of the left-right action on matrix tuples: while the scaling algorithm for the null cone problem does not require invariant-theoretic degree bounds [23], a polynomial degree bound for matrix semi-invariants [16] is necessary for the scaling algorithm for orbit closure intersection [2].

In this paper, our main result is to put the orbit closure intersection problem for the action of $\mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_m(\mathbb{K})$ on $\mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$ where $m = O(1)$ in coRP.

[2] It is important to restrict to unit determinant matrices for the first action, as the orbit closures of two tensors will always intersect at zero under the action of the product of general linear groups.

▶ **Theorem 3.** *The orbit closure intersection problem for the tensor action of* $G := \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_m(\mathbb{K})$ *on* $\mathcal{V} := \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$ *with* $m = O(1)$ *is in* coRP. *More explicitly, given* n *and two vectors* $u, v \in \mathbb{Q}[i]^n \otimes \mathbb{Q}[i]^n \otimes \mathbb{Q}[i]^m$*, there exists a randomized algorithm with one-sided bounded error that decides whether the orbit closures of* u *and* v *intersect in time polynomial in* n *and the bit-lengths of* u *and* v*.*

The above theorem is the main application of our two technical theorems, which we present in Section 1.2, along with a technical overview.

▶ Remark 4. In fact, our results and techniques apply to more general settings, which include the important setting of higher dimensional *rectangular* tensors – i.e., tensors of constantly many sides where all but two dimensions of the tensor factors are constant. However, for ease of exposition, we have decided to present only the above theorem as the main application of our technical results. We provide a more detailed discussion on the generality of our results at the end of Section 1.2.

We now review the literature and previous works.

1.1 Related Work

The orbit closure intersection problem has been studied in certain algorithmically tractable group actions, but the tensor action has so far remained the main frontier for this problem. A key positive case is the left-right action of $\mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K})$ on matrix tuples (equivalently, on $\mathcal{V} = \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$ where the third factor is not acted upon). In this setting, polynomial degree bounds for the generators of the invariant ring are known [16], and they are critical for obtaining efficient algorithms for the orbit closure intersection problem [2, 36, 17].

Another positive regime is when the group $\mathrm{SL}_m(\mathbb{K})$, with constant m, acts on a vector space. In geometric complexity theory, Mulmuley [41] developed the idea of "succinct encoding" framework for invariant rings, and constructed explicit generating invariants for $\mathrm{SL}_m(\mathbb{K})$-actions that enable randomized polynomial time algorithms for the orbit closure intersection problem.

In contrast, the complexity of OCI seems to drastically change when we consider the action of $\mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K})$ on $\mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^n$, where n varies. While no explicit hardness result for OCI is known, several lines of work show limitations of the current techniques. An indication of hardness from invariant theory comes from [18], where the authors proved exponential lower bounds for the degrees of the generators for the action of $\mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K})$ on $(\mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^n)^{\oplus 9}$. This result highlights that the generation in low degree may fail once all dimensions are allowed to vary.

Even in the setting where degree bounds are not the main obstacle, succinct encoding of the invariants may fail: [24] showed (under standard complexity theoretic assumptions) that for certain (torus) actions on $\mathbb{C}^n \otimes \mathbb{C}^n \otimes \mathbb{C}^n$, no polynomial-sized encoding of the invariants exists, placing a further limitation in this setting.

The optimization approach also seems to have important barriers, as we now explain. The non-commutative optimization framework of Bürgisser et al. [11] provides a natural gradient descent algorithm for the null cone problem. The iteration complexity of their algorithm depends inversely on a parameter known as the *weight margin* of the action. For the 3-tensor action where all dimensions are allowed to vary, this parameter can be exponentially small as shown by Franks and Reichenbach [22], and the resulting algorithm is only guaranteed to run in exponential-time.

Our work studies the intermediate fixed-parameter regime for the action of $G = \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_m(\mathbb{K})$ on $\mathcal{V} = \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$ where m is constant and n is allowed to grow. On the one hand, the group $\mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K})$ acts by the left-right action on $\mathcal{V}$ but on the other

hand the $\mathrm{SL}_m(\mathbb{K})$ factor is constant and acts only by shuffling the m slices. Conceptually, our approach exploits this decomposition of the group, albeit in a very different way than the work [11] exploits this product structure in their polynomial-time algorithm for the null cone problem. The null cone algorithm for [11, Corollary 6.24] is based on the scaling technique, and the key there is to bound the weight margins of the representation, and they use the product structure of the group to obtain such a bound. On the other hand, we manage to first obtain a degree bound for generating the invariant ring, and then utilize an effective version of the Cayley's Omega process to obtain a succinct encoding of the invariant ring. Combining these with the polynomial identity lemma (also known as Schwartz-Zippel lemma), the randomized polynomial-time algorithm follows.

1.2 Technical Overview

In this subsection we provide a proof overview of Theorem 3.

1.2.1 Geometric invariant theory

Consider the natural action of $G := \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_m(\mathbb{K})$ on the space of tensors $\mathcal{V} := \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$. We are given two tensors $u, v \in \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$, each given to us by n^2m many entries in $\mathbb{K}$. Our goal is to decide if their orbit closures intersect (OCI).

We consider the ring $\mathbb{K}[\mathcal{V}]$ of polynomial functions on $\mathcal{V}$. This ring is isomorphic to the polynomial ring with n^2m variables with coefficients in $\mathbb{K}$. A polynomial $f \in \mathbb{K}[\mathcal{V}]$ is called an *invariant* if $f(g \cdot v) = f(v)$ for every $g \in G$ and $v \in \mathcal{V}$. The set of invariants forms a $\mathbb{K}$-subalgebra of $\mathbb{K}[\mathcal{V}]$ that we denote by $\mathbb{K}[\mathcal{V}]^G$.

The conceptual bridge between invariant theory and OCI is Mumford's theorem: For linearly reductive groups, orbit closures are disjoint if and only if there is an invariant polynomial that separates them.

▶ **Theorem 5** (Mumford's Theorem [43]). *Let $\mathbb{K}$ be an algebraically closed field and let G be a reductive group acting linearly on a vector space $\mathcal{V}$. Given two elements $u, v \in \mathcal{V}$, we have $\overline{G \cdot u} \cap \overline{G \cdot v} \neq \emptyset \iff p(u) = p(v)$ for all $p \in \mathbb{K}[\mathcal{V}]^G$.*

In an ideal world, Mumford's theorem reduces OCI to computing and evaluating invariants. In practice, however, this approach fails due to two phenomena: first, even when invariants are generated in low degree for some special actions, the general upper bounds are exponential and there are instances where they must be exponential [18]. Second, even if a generating set exists in low degrees, it may be algorithmically infeasible to evaluate them all. We shall now discuss how we address these two issues.

1.2.2 Degree bounds for the invariant ring

As the invariant ring is usually infinite, we can only hope to capture it by a finite generating set. Indeed, the finite generation of this ring is ensured by Hilbert's celebrated results [31, 32]. The first difficulty that one needs to overcome is to bound the degrees of generators. Our first main contribution is a polynomial degree bound for the generators of the invariant ring.

▶ **Theorem 6** (Theorem 21 – informal). *Consider the action of $G := \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_m(\mathbb{K})$ on $\mathcal{V} := \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$. When $m = O(1)$, $\mathbb{K}[\mathcal{V}]^G$ is generated by homogeneous invariants of degree* $\mathrm{poly}(n)$.

To the best of our knowledge, only an upper bound of the form $O(\exp(n^2))$ was known for the degrees of the generators in this regime, see [15, Prop. 4.7.16] or [11, Prop. 7.2].

1.2.3 Succinct encoding of invariant rings

Even though the invariant ring is finitely generated, the number of generators can be exponential in size and they may be infeasible to evaluate.

To overcome this issue, Mulmuley proposed the notion of succinct encodings of invariant rings [41]. The basic idea is to introduce auxiliary variables so a set of generators can be obtained by interpolating over these auxiliary variables. This leads to the following definition from [41], which we state in adapted form from [24, Definition 1.3].

▶ **Definition 7.** *Let a group G act on a vector space $\mathcal{V} \cong \mathbb{K}^N$. We say that an arithmetic circuit $C = C(x_1, \ldots, x_N, y_1, \ldots, y_r)$* succinctly encodes *$\mathbb{K}[\mathcal{V}]^G$, if the set*

$$\{\, C(x_1, \ldots, x_N, a_1, \ldots, a_r) \mid (a_1, \ldots, a_r) \in \mathbb{K}^r \,\}$$

generates $\mathbb{K}[\mathcal{V}]^G$ as a $\mathbb{K}$-algebra. The size *of a succinct encoding is defined to be the size of the circuit C, measured by the number of gates and the bit-lengths of the constants used in the computation of C. In particular, we require that the constants are rational numbers.*

A succinct encoding for the action of G on $\mathcal{V}$ allows one to obtain a randomized algorithm for the orbit closure problem of this action as follows. By Theorem 5, the orbit closures of $u, v \in V$ intersect if and only if for any $f \in \mathbb{K}[V]^G$, $f(u) = f(v)$. Therefore, once a succinct encoding $C(x_1, \ldots, x_N, y_1, \ldots, y_r)$ of $\mathbb{K}[\mathcal{V}]^G$ is given, testing whether the orbit closures of u and v intersect reduces to testing if the two circuits in y-variables, $C(u_1, \ldots, u_N, y_1, \ldots, y_r)$ and $C(v_1, \ldots, v_N, y_1, \ldots, y_r)$, compute the same polynomial. This is an instance of the polynomial identity testing problem, which, by the polynomial identity testing lemma, admits a randomized polynomial-time algorithm by specializing $y_1, y_2, \ldots, y_r$ randomly to integers of bit-length $\mathrm{poly}(\mathrm{size}(C))$.

Thus, to reduce the OCI problem to an instance of PIT, we need to prove that we can efficiently compute a succinct generator for $\mathbb{K}[\mathcal{V}]^G$. This is our second main contribution: an efficient succinct encoding of the invariant ring $\mathbb{K}[\mathcal{V}]^G$.

▶ **Theorem 8.** *Assume $m = O(1)$. Then, there is a uniform family of $\mathrm{poly}(n)$-sized arithmetic circuit C that succinctly encodes $\mathbb{K}[\mathcal{V}]^G$.*

We prove Theorem 8 in Section 4. We now give an overview of how we prove degree bounds and also on how we efficiently construct such a succinct generator.

1.2.4 The basic strategy to prove Theorems 6 and 8

Our basic strategy is to "split" the action of $G = \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_m(\mathbb{K})$ into two steps as $G = H \times K$, where $H = \mathrm{SL}_n \times \mathrm{SL}_n$ and $K = \mathrm{SL}_m$. This decomposition encodes the invariant ring $R = \mathbb{K}[\mathcal{V}]^G$ as $R = (\mathbb{K}[\mathcal{V}]^H)^K$. Note that $\mathbb{K}[\mathcal{V}]^H$ is the invariant ring for the left-right action on matrix tuples, for which closed forms of invariant polynomials are known (Theorem 14), and polynomial degree upper bounds are also known [16]. In particular, these yield a succinct encoding of the invariant ring $\mathbb{K}[\mathcal{V}]^H$.

Given these, going from $\mathbb{K}[\mathcal{V}]^H$ to $(\mathbb{K}[\mathcal{V}]^H)^K$ still requires some work. Note that by the polynomial degree bound of the H-action, we can focus on K acting only on those invariants of polynomially bounded degrees, but the issue that we need to be careful about is that those invariants span a vector space of exponential dimension. This creates obstacles for proving a degree bound of the form $\mathrm{poly}(n, m)^{f(m)}$ where a function just depending on m stands on the exponent, as the generation degree bounds such as in [14] always depend polynomially in the dimension of the base vector space. And on the succinct generation question, even from

the succinct generation for the left-right action, we still need to obtain generators for the tensor action, thus we cannot naively construct the succinct generator based on the vector space of generators of the left-right action.

1.2.5 Addressing the technical difficulties

We overcome the above issues by (1) presenting a generation degree upper bound for the invariant ring R in the form of $\mathrm{poly}(n,m)^{f(m)}$, and (2) making an effective use of the Cayley Omega process to produce a succinct encoding.

For the degree upper bounds, we mentioned the problem of K acting on an exponential-dimension vector space. To get around this problem, the main idea is to resort to the nullcone defining bound as much as possible rather than on the generation bound. Indeed, in [14, Proposition 1.2], the nullcone defining bound has the intriguing property that it *does not depend on the ambient vector space dimension*. This allows us to get around the issue of working over a vector space of an exponentially large dimension by showing a suitable nullcone defining bound first, and then using the relation between generation and nullcone bounds [14, Theorem 1.1] to obtain the generation bound for the full invariant ring $\mathbb{K}[\mathcal{V}]^G$. Details on how we execute this approach are in Section 3.

Given the degree generation bound, we resort to the Cayley Omega process to obtain a succinct generator of $(\mathbb{K}[\mathcal{V}]^H)^K$ from the succinct generator of $\mathbb{K}[\mathcal{V}]^H$. The Cayley Omega process is a generic procedure to obtain a generating set, and we need to apply this process to the succinct encoding setting where sequences of partial derivatives need to be computed efficiently. At first sight, efficiently computing the partial derivative operator arising from the Omega process seems to be an impossible task, due to the complexity of the Omega process. However, the key here is to note that we are only applying the Omega process in constantly many variables (when $m \in O(1)$), which allows us to be inefficient in terms of m, while still preserving the efficiency of the succinct generator for the left-right action, as we operate on it to transform it into a succinct generator for the tensor action. This is analogous to how Mulmuley was able to compute his succinct generator [41]. Our efficient construction of the succinct generator appears in Section 4.

1.2.6 Generality of Our Techniques

While the above results are only stated in terms of the natural 3-tensor action, a quick look into our techniques and our proofs show that our results apply in much more generality. In particular, our degree bounds and efficient generation results need only the following general assumptions on G and $\mathcal{V}$: that the group G factors as a product of two groups $G = H \times K$, for which K is a product of special linear groups of constant total dimension – that is, $K = \prod_{i=1}^{k} \mathrm{SL}_{m_i}$ where $\prod_{i=1}^{k} m_i \in O(1)$ – and H is any group such that the corresponding action of H on $\mathcal{V}$ has polynomial degree bounds and an efficiently computable succinct generator. For instance, this setup also captures more general tensor actions where we have constantly many tensor factors where all but 2 of the factors have bounded dimension.

2 Preliminaries

Throughout this paper, we write $[\ell] := \{1, 2, \ldots, \ell\}$ for $\ell \in \mathbb{N}$. We take $\mathbb{K}$ to be an algebraically closed field of characteristic zero. For a $\mathbb{K}$-vector space $\mathcal{V}$, we denote by $\mathbb{K}[\mathcal{V}]$ the ring of polynomial functions on $\mathcal{V}$. If $\dim \mathcal{V} = N$, then we have an isomorphism $\mathbb{K}[\mathcal{V}] \cong \mathbb{K}[x_1, x_2, \ldots, x_N]$ with the latter denoting the polynomial ring with N variables over $\mathbb{K}$.

The zero locus of a set of polynomials $h_1, h_2, \ldots, h_\ell \in \mathbb{K}[\mathcal{V}]$, denoted $\mathcal{Z} = \mathcal{Z}(h_1, h_2, \ldots, h_\ell) := \{v \in \mathcal{V} \mid h_1(v) = \ldots = h_\ell(v) = 0\}$, is called an *affine algebraic set*. The ring of regular functions on $\mathcal{Z}$ is denoted by $\mathbb{K}[\mathcal{Z}] := \mathbb{K}[\mathcal{V}]/I$ where I is the radical of the ideal generated by $h_1, h_2, \ldots, h_\ell$. If $S \subset \mathcal{V}$ is an arbitrary set, we denote by $\overline{S}$ its *Zariski closure*, the smallest affine algebraic set that contains S.

2.1 Groups, Representations, and Invariant Theory

We will briefly introduce the main results in invariant theory that are relevant for the next sections. We refer to [15, 39, 43] for a more detailed treatment of the subject.

For a natural number $s \in \mathbb{N}$, we denote by $\mathrm{GL}_s(\mathbb{K})$ (resp. $\mathrm{SL}_s(\mathbb{K})$) the set of invertible (resp. with unit determinant) $s \times s$-matrices with entries in $\mathbb{K}$. A Zariski closed subgroup $G \leq \mathrm{GL}_s(\mathbb{K})$ is called an *algebraic group*. For simplicity of the presentation, we will assume that $G \leq \mathrm{SL}_s(\mathbb{K})$, since this is true for the groups that we will consider throughout the paper. In this case, G is an affine algebraic set in $\mathrm{Mat}(s, \mathbb{K})$, the vector space of $s \times s$ matrices over $\mathbb{K}$, and it is the zero locus of some polynomials $h_1, h_2, \ldots, h_\ell \in \mathbb{K}[g_{1,1}, \ldots, g_{s,s}]$.

Suppose $\mathcal{V}$ is an N-dimensional vector space over $\mathbb{K}$. A group homomorphism $\pi : G \to \mathrm{GL}(\mathcal{V})$ is called a *rational representation* if matrix entries of $\pi(g)$ are regular functions on G. That is,

$$\pi(g) = \begin{pmatrix} a_{1,1}(g) & a_{1,2}(g) & \ldots & a_{1,N}(g) \\ a_{2,1}(g) & \ddots & & \\ \vdots & & \ddots & \\ a_{N,1}(g) & & & a_{N,N}(g) \end{pmatrix}, \quad g \in G \tag{1}$$

for some polynomials $a_{1,1}, \ldots, a_{N,N} \in \mathbb{K}[g_{1,1}, \ldots, g_{s,s}]$.

Let $\pi : G \to \mathrm{GL}(\mathcal{V})$ be a rational representation. A subspace $\mathcal{U} \subset \mathcal{V}$ is called *G-stable* if $\pi(g)u \in \mathcal{U}$ for every $g \in G$ and $u \in \mathcal{U}$. In this case, the map $\gamma : G \to \mathrm{GL}(\mathcal{U})$ defined by $\gamma(g) = \pi(g)|_{\mathcal{U}}$ is also a rational representation of G. We call $\mathcal{V}$ *irreducible* if the only G-stable subspaces are $\{0\}$ and $\mathcal{V}$. An algebraic subgroup $G \leq \mathrm{GL}_s(\mathbb{K})$ is called *reductive* if $\mathcal{V}$ is a direct sum of its irreducible G-stable subspaces for every rational representation $\pi : G \to \mathrm{GL}(\mathcal{V})$. Important examples of reductive algebraic groups are $\mathrm{GL}_s(\mathbb{K}), \mathrm{SL}_s(\mathbb{K}), \mathrm{O}_s(\mathbb{K})$ (the group of orthogonal matrices), $\mathrm{Sp}_s(\mathbb{K})$ (the group of symplectic matrices), the tori $(\mathbb{K}^\times)^s$, all finite groups and their direct products.

A rational representation $\pi : G \to \mathrm{GL}(\mathcal{V})$ naturally defines a group action of G on $\mathcal{V}$ via

$$g \cdot v = \pi(g)v, \quad g \in G, v \in \mathcal{V}.$$

The action of G on $\mathcal{V}$ also induces an action on $\mathbb{K}[\mathcal{V}]$ by defining

$$(g \cdot f)(v) = f(g^{-1} \cdot v), \quad g \in G, v \in \mathcal{V}, f \in \mathbb{K}[\mathcal{V}].$$

A polynomial f is called an *invariant* if $g \cdot f = f$ for every $g \in G$. We denote by $\mathbb{K}[\mathcal{V}]^G$ the ring of invariants, which is a $\mathbb{K}$-subalgebra of $\mathbb{K}[\mathcal{V}]$. A celebrated result due to Hilbert [31, 32] states that the invariant ring $\mathbb{K}[\mathcal{V}]^G$ is finitely generated as a $\mathbb{K}$-algebra for every rational representation $\pi : G \to \mathrm{GL}(\mathcal{V})$ of a reductive group G. We note that since polynomials are continuous functions, invariant polynomials must take the same value on an orbit closure $\overline{G \cdot u}$. In particular, the invariant polynomials cannot distinguish two orbit closures that intersect. Mumford's Theorem 5 provides a converse statement, namely, two orbit closures $\overline{G \cdot u}$ and $\overline{G \cdot v}$ do not intersect if and only if there exists an invariant polynomial that distinguishes them.

Two important numerical invariants of a rational representation are: the minimum degree of polynomials that generate the invariant ring, that we denote by $\beta_\pi(\mathcal{V})$, and the minimum degree $\sigma_\pi(\mathcal{V})$ of invariants $f_1, f_2, \ldots, f_\ell$ such that the invariant ring $\mathbb{K}[\mathcal{V}]^G$ is integral over the $\mathbb{K}$-algebra $\mathbb{K}[f_1, f_2, \ldots, f_\ell]$. Namely,

$$\beta_\pi(\mathcal{V}) := \min\left\{d \in \mathbb{N} \mid \mathbb{K}[\mathcal{V}]^G \text{ is generated by invariants of degree } \leq d\right\} \tag{2}$$

and

$$\sigma_\pi(\mathcal{V}) := \min\left\{d \in \mathbb{N} \mid \mathbb{K}[\mathcal{V}]^G \text{ is integral over } \mathbb{K}[\mathcal{V}]^G_{\leq d}\right\}. \tag{3}$$

The *null cone* $\mathcal{N}(\mathcal{V}) \subset \mathcal{V}$ of π is defined as the zero locus of all homogeneous, non-constant invariants. By [15, Remark 1.7.2], $\sigma_\pi(\mathcal{V})$ is also equal to the minimum degree of *null cone definers*, i.e.,

$$\sigma_\pi(\mathcal{V}) = \min\left\{d \mid \mathcal{N}(\mathcal{V}) = \mathcal{Z}(f_1, f_2, \ldots, f_\ell) \text{ for some } f_1, f_2, \ldots, f_\ell \in \mathbb{K}[\mathcal{V}]^G_{\leq d}\right\}.$$

We note that by definition, $\beta_\pi(\mathcal{V})$ and $\sigma_\pi(\mathcal{V})$ only depend on the invariant ring $\mathbb{K}[\mathcal{V}]^G$, i.e., if $\mathbb{K}[\mathcal{V}]^G \cong \mathbb{K}[\mathcal{W}]^H$ as filtered $\mathbb{K}$-algebras for two rational representations $\pi : G \to \mathrm{GL}(\mathcal{V})$ and $\gamma : H \to \mathrm{GL}(\mathcal{W})$, then $\beta_\pi(\mathcal{V}) = \beta_\gamma(\mathcal{W})$ and $\sigma_\pi(\mathcal{V}) = \sigma_\gamma(\mathcal{W})$.

The following important result appears in [14, Proposition 1.2]. An important consequence of it is that $\sigma_\pi(\mathcal{V})$ admits an upper bound independent of the dimension of $\mathcal{V}$.

▶ **Proposition 9** ([14, Proposition 1.2]). *In the setting above, define $m := \dim(G)$. Recall that $G \leq \mathrm{GL}_s(\mathbb{K})$ is an algebraic group. Additionally, let $H := \max\{\deg(h_i) \mid i \in [\ell]\}$ and $A := \max\{\deg(a_{j,k}) \mid 1 \leq j, k \leq \dim(\mathcal{V})\}$ where $a_{i,j}$ are the defining polynomials of π, see Equation (1). Then*

$$\sigma_\pi(\mathcal{V}) \leq H^{s^2-m} \cdot A^m.$$

Let's consider the case $G = \mathrm{SL}_s(\mathbb{K})$. The special linear group is cut out by the polynomial $\det(g) - 1$, which has degree $H = s$, and the dimension of $\mathrm{SL}_s(\mathbb{K})$ is $m = s^2 - 1$.

▶ **Corollary 10.** *Let $\pi : \mathrm{SL}_s(\mathbb{K}) \to \mathrm{GL}(\mathcal{V})$ be a rational representation with $\pi(g) = (a_{i,j}(g))_{i,j\in[N]}$. Set $A := \max\limits_{1\leq i,j\leq N} \deg(a_{i,j})$. Then,*

$$\sigma_\pi(\mathcal{V}) \leq s\, A^{s^2-1}.$$

The above degree bounds on the null cone can be used to derive degree bounds for the generators of the invariant ring, as the following remarkable result shows.

▶ **Theorem 11** (Degree bounds for generators, [14, Theorem 1.1]). *In the setting of Proposition 9,*

$$\beta_\pi(\mathcal{V}) \leq \max\left\{2,\ \frac{3}{8} \cdot \dim(\mathbb{K}[\mathcal{V}]^G) \cdot \sigma_\pi(\mathcal{V})^2\right\},$$

where $\dim \mathbb{K}[\mathcal{V}]^G$ is the Krull dimension *of $\mathbb{K}[\mathcal{V}]^G$, and thus $\dim \mathbb{K}[\mathcal{V}]^G \leq \dim \mathcal{V}$.*

Suppose $\mathcal{X} \subset \mathcal{V}$ and $\mathcal{Z} \subset \mathcal{W}$ are two G-stable affine algebraic sets. A morphism $f : \mathcal{X} \to \mathcal{Z}$ that satisfies $f(g \cdot x) = g \cdot f(x)$ is called a G-equivariant map. There is an induced ring homomorphism $f^\sharp : \mathbb{K}[\mathcal{Z}] \to \mathbb{K}[\mathcal{X}]$ defined by $f^\sharp(F)(x) = F(f(x))$. We say that f has degree at most D if

$$\deg(f^\sharp(F)) \leq D \cdot \deg(F), \quad \text{for every } F \in \mathbb{K}[\mathcal{Z}].$$

If $f^\sharp$ is surjective, then we say that f is an *embedding*.

▶ **Proposition 12** ([15, Corollary 2.2.9]). *Let $f : \mathcal{X} \to \mathcal{Z}$ be a G-equivariant embedding. Then*

$$f^\sharp(\mathbb{K}[\mathcal{Z}]^G) = \mathbb{K}[\mathcal{X}]^G.$$

Proposition 12 allows us to pull-back the degree bounds for $\mathbb{K}[\mathcal{Z}]^G$ to $\mathbb{K}[\mathcal{X}]^G$.

▶ **Corollary 13.** *Let $\pi : G \to \mathrm{GL}(\mathcal{V})$, $\gamma : G \to \mathrm{GL}(\mathcal{W})$ be rational representations, $\mathcal{X} \subseteq \mathcal{V}, \mathcal{Z} \subseteq \mathcal{W}$ be G-stable algebraic sets and $f : \mathcal{X} \to \mathcal{Z}$ be a G-equivariant embedding of degree D. Then,*

$$\beta_\pi(\mathcal{X}) \leq D \cdot \beta_\gamma(\mathcal{Z}) \quad \textit{and} \quad \sigma_\pi(\mathcal{X}) \leq D \cdot \sigma_\gamma(\mathcal{Z}).$$

2.2 The Left-Right Action on Matrix Tuples

For certain families of actions, we have more structure on the set of generators for the invariant ring, as well as a better control on the degree bounds for the null cone and for the generators of the invariant ring. An important example is the left-right action, which we now define.

Let $m, n \in \mathbb{N}$, $H := \mathrm{SL}_n \times \mathrm{SL}_n$ and $\mathcal{V} := \mathrm{Mat}(n, \mathbb{K})^m$ where $\mathrm{Mat}(n, \mathbb{K})$ denotes the space of $n \times n$-matrices over $\mathbb{K}$. The left-right action of H on $\mathcal{V}$ is given by

$$\pi_H(X, Y) \cdot (A_1, \ldots, A_m) := (XA_1Y^t, \ldots, XA_mY^t).$$

The following theorem on the generators of the ring of invariants for the left-right action appears in [19, Corollary 3], [20] and [46].

▶ **Theorem 14.** *The space $\mathbb{K}[\mathcal{V}]^H$ is spanned as a vector space by all the forms*

$$f_T(A) := \det\left(\sum_{i=1}^{m} T_i \otimes A_i\right) \tag{4}$$

where $T = (T_1, \ldots, T_m) \in \mathrm{Mat}(d, \mathbb{K})^m$ and $d \in \mathbb{N}$.

If we apply the general bounds in Proposition 9 and Theorem 11 from [14], we would get a degree bound for $\mathbb{K}[\mathcal{V}]^H$ that is exponential in n, see also [11, Prop. 7.2]. The result [16, Theorem 1.8] (see also [36, Theorem 1]), which we now state, gives much sharper degree bounds for equations defining the null cone of the left-right action.

▶ **Theorem 15** ([16, Theorem 1.8]). *The following bound holds: $\sigma_{\pi_H}(\mathcal{V}) \leq n(n-1)$.*

We note that since $f_T(X)$ has degree dn, Theorem 15 implies that the null cone is cut out by the polynomials $f_T(X)$ for $T = (T_1, \ldots, T_m) \in \mathrm{Mat}(d, \mathbb{K})^m$ with $d \leq n - 1$. This bound, when combined with Popov's theorem [44, 45] yields the following generating degree bounds for the left-right action, obtained in [16, Theorem 1.2].

▶ **Theorem 16** ([16, Theorem 1.2]). *The following bound holds: $\beta_{\pi_H}(\mathcal{V}) \leq m \cdot n^4$.*

2.3 Reynolds Operator & Cayley's Omega Process

In the general setting of a rational representation $\pi : G \to \mathrm{GL}(\mathcal{V})$ of a reductive group G, there exists a linear operator $\mathcal{R} : \mathbb{K}[\mathcal{V}] \to \mathbb{K}[\mathcal{V}]^G$ with the following properties

1. $\mathcal{R}(f) = f$ for all invariants $f \in \mathbb{K}[\mathcal{V}]^G$.
2. $\mathcal{R}(g \cdot h) = \mathcal{R}(h)$ for all $h \in \mathbb{K}[\mathcal{V}]$ and $g \in G$.

This operator is uniquely determined by the properties above and is called the Reynolds operator [15, Theorem 2.2.5]. The following lemma gives an important factoring property of the Reynolds operator, and appears in [15, Lemma 4.5.4].

▶ **Lemma 17.** *Consider a rational representation $\pi : G \to \mathrm{GL}(\mathcal{V})$ and suppose that G has a normal subgroup H. Let $\mathcal{R}_G : \mathbb{K}[\mathcal{V}] \to \mathbb{K}[\mathcal{V}]^G$ be the Reynolds operator with respect to G, $\mathcal{R}_H : \mathbb{K}[\mathcal{V}] \to \mathbb{K}[\mathcal{V}]^H$ the Reynolds operator with respect to H and $\mathcal{R}_{G/H} : \mathbb{K}[\mathcal{V}]^H \to \mathbb{K}[\mathcal{V}]^G$ be the Reynolds operator with respect to G/H. Then $\mathcal{R}_G = \mathcal{R}_{G/H} \circ \mathcal{R}_H$.*

For homogeneous actions of $G = \mathrm{SL}_m(\mathbb{K})$, there is an explicit algebraic construction of the Reynolds operator, given by Cayley's Omega process, which we now briefly present. We will use this construction in Section 4 together with other results from algebraic complexity theory to give uniform succinct circuits that compute generators for the invariant ring of the tensor action. For more background and proofs of the results that we invoke in this subsection, we refer the reader to [15, 11] and references therein.

Consider a homogeneous degree D representation $\pi : \mathrm{SL}_m(\mathbb{K}) \to \mathrm{GL}(\mathcal{W})$, along with its induced action on the polynomials in $\mathbb{K}[\mathcal{W}]$, denoted by $\tilde{\pi}$. That is, $(\tilde{\pi}(g)f)(x) := f(\pi^{-1}(g)x)$. After fixing a basis (e_i) of $\mathcal{W}$, the entries of $\pi(Z)$ are homogeneous degree D polynomials in the variables $Z := (Z_{i,j})_{i,j\in[m]}$ and the entries of $\tilde{\pi}$, when acting on the space $\mathbb{K}[\mathcal{W}]_r$ of degree r forms, are homogeneous polynomials of degree $\tilde{D}_r := rD(m-1)$.[3] Our goal is to describe the invariant ring $\mathbb{K}[\mathcal{W}]^{\mathrm{SL}_m}$ corresponding to the above action.

Define $\partial := (\partial_{Z_{i,j}})_{i,j\in[m]}$ to be the $m \times m$ matrix of partial derivatives of Z. Let $\mathbb{K}[\partial]$ be the ring of differential operators on $\mathbb{K}[Z]$. For any $p \in \mathbb{K}[Z]$, let $\partial_p := p(\partial)$. Cayley's Ω-operator is defined as follows

$$\Omega : \mathbb{K}[Z] \to \mathbb{K}[Z], \qquad \Omega := \partial_{\det} = \sum_{\sigma\in S_m} \mathrm{sgn}(\sigma) \prod_{k=1}^{m} \partial_{Z_{k,\sigma(k)}}. \tag{5}$$

Applying Ω to a homogeneous polynomial q reduces its degree by m, thus if q has degree rm, $\Omega^r(q)$ is a constant. In order to state the invariance properties of the Ω-operator we introduce the left and right multiplication maps for $A \in \mathrm{Mat}(m, \mathbb{K})$.

$$R_A : \mathrm{Mat}(m,\mathbb{K}) \to \mathrm{Mat}(m,\mathbb{K}), \quad X \mapsto XA$$
$$L_A : \mathrm{Mat}(m,\mathbb{K}) \to \mathrm{Mat}(m,\mathbb{K}), \quad X \mapsto AX.$$

With the above definitions, we have the following properties:

▶ **Lemma 18** ([11, Proposition 7.8])**.** *In the setting above, the following properties hold:*

1. *For any $q \in \mathbb{K}[Z]$ and $A \in \mathrm{Mat}(m,\mathbb{K})$ we have*

$$\Omega(q \circ L_A) = \det(A) \cdot (\Omega(q) \circ L_A), \qquad \Omega(q \circ R_A) = \det(A) \cdot (\Omega(q) \circ R_A).$$

2. *If $q \in \mathbb{K}[Z]_{rm}$ and $A \in \mathrm{SL}_m$, then $\Omega^r(q) \in \mathbb{K}$ and $\Omega^r(q) = \Omega^r(q \circ L_A) = \Omega^r(q \circ R_A)$.*
3. *$c_{r,m} := \Omega^r(\det(Z)^r) \in \mathbb{K}^\times$, that is, $c_{r,m}$ is a nonzero element of $\mathbb{K}$.*

[3] We note that for $g \in \mathrm{SL}_m(\mathbb{K})$ we have $g^{-1} = \mathrm{Adj}(g)$ where $\mathrm{Adj}(g)$ denotes the *adjugate matrix*, which has degree $m-1$ as a polynomial function of g.

We are now ready to define the Reynolds operator $\mathcal{R} : \mathbb{K}[\mathcal{W}] \to \mathbb{K}[\mathcal{W}]$ for the representation $\pi : \mathrm{SL}_m(\mathbb{K}) \to \mathrm{GL}(\mathcal{W})$. For $f \in \mathbb{K}[\mathcal{W}]_r$, we have:

$$\mathcal{R}(f) := \begin{cases} \dfrac{\Omega^{\tilde{D}_r/m} \circ \tilde{\pi}(Z) \cdot f}{c_{\tilde{D}_r/m,m}} & \text{if } m \mid \tilde{D}_r, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $\tilde{D}_r = rD(m-1)$, $c_{\tilde{D}_r/m,m}$ is defined in Lemma 18 (3), and $\Omega^{\tilde{D}_r/m} \circ \tilde{\pi}(Z) \cdot f$ means applying $\Omega^{\tilde{D}_r/m}$ to the Z coefficients of $\tilde{\pi}(Z) \cdot f$.

The following implies that $\mathcal{R}$ indeed has the properties characterizing the Reynolds operator.

▶ **Proposition 19** ([11, Proposition 7.9 and Corollary 7.10]). *The map $\mathcal{R} : \mathbb{K}[\mathcal{W}] \to \mathbb{K}[\mathcal{W}]$ defined in Equation (6) satisfies the following conditions.*

1. *If $f \in \mathbb{K}[\mathcal{W}]^G$, then $\mathcal{R}(f) = f$.*
2. *If $f \in \mathbb{K}[\mathcal{W}]$ and $g \in G$, then $\mathcal{R}(g \cdot f) = \mathcal{R}(f)$.*
3. *$\mathcal{R}(\mathbb{K}[\mathcal{W}]) = \mathbb{K}[\mathcal{W}]^G$.*

3 Fixed-Parameter Polynomial Degree Bounds

In this section we prove our first main technical theorem: Theorem 21, which is the formal statement of Theorem 6.

3.1 Setup

Let $n, m \in \mathbb{N}$ be natural numbers and consider the natural action of the product group $G := \mathrm{SL}_n \times \mathrm{SL}_n \times \mathrm{SL}_m$ on the tensor space $\mathcal{V} := \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$, when m is fixed and n becomes large. In the following, we will identify a tensor $T \in \mathcal{V}$ with the matrix tuple $(A_1, A_2, \ldots, A_m)$, where $A_i \in \mathrm{Mat}(n, \mathbb{K})$ is the i^{th} 3-*slice* of T.

The action of G on $\mathcal{V}$ is given by the rational representation $\pi_G : G \to \mathrm{GL}(\mathcal{V})$, which defines the action of G on $\mathcal{V}$ as

$$(X, Y, Z) \cdot (A_1, \ldots, A_m) = \left(X \left(\sum_{j=1}^{m} Z_{1j} A_j \right) Y^t,\ X \left(\sum_{j=1}^{m} Z_{2j} A_j \right) Y^t,\ \ldots,\ X \left(\sum_{j=1}^{m} Z_{mj} A_j \right) Y^t \right).$$

In other words, $\mathrm{SL}_n \times \mathrm{SL}_n$ acts on $\mathcal{V}$ by the *left-right action* and SL_m acts by *shuffling* the 3-slices. In representation theoretic terms, we have

$$\pi_G(X, Y, Z) = X \otimes Y \otimes Z \in \mathrm{GL}(\mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m).$$

The group G is a direct product of its subgroups $H := \mathrm{SL}_n \times \mathrm{SL}_n$ and $K := \mathrm{SL}_m$. We denote the restriction of π_G to H (resp. K) by π_H (resp. π_K). Then, the induced action of H and K on $\mathcal{V}$ are given by

$$(X, Y) \cdot (A_1, \ldots, A_m) = (XA_1Y^t, \ldots, XA_mY^t),$$

$$Z \cdot (A_1, \ldots, A_m) = \left(\sum_{j=1}^{m} Z_{1j} A_j, \ldots, \sum_{j=1}^{m} Z_{mj} A_j \right),$$

and we have $\pi_H(X, Y) = X \otimes Y \otimes I_m$ and $\pi_K(Z) = I_n \otimes I_n \otimes Z$.

Since H and K commute, we have the following identification of the invariant rings:

$$\mathbb{K}[\mathcal{V}]^G = \left(\mathbb{K}[\mathcal{V}]^H\right)^K \tag{7}$$

In the following, we will obtain degree bounds on $\mathbb{K}$-algebra generators of $\mathbb{K}[\mathcal{V}]^G$ by making use of this identification. As π_H is simply the left-right action, by Theorem 16, the invariant ring $\mathbb{K}[\mathcal{V}]^H$ is generated by invariants of degree $\beta_H \leq mn^4$, where we denote $\beta_H := \beta_{\pi_H}(\mathcal{V})$ from now on and similarly for β_G and β_K.

3.2 Using the left-right action

Consider the vector space $\mathcal{W} := \bigoplus_{d=1}^{\beta_H} \mathbb{K}[\mathcal{V}]_d^H \subset \mathbb{K}[\mathcal{V}]^H$. In plain English, $\mathcal{W}$ is the space of non-constant H-invariants up to degree β_H. We denote the dimension of $\mathcal{W}$ by $N := \dim \mathcal{W}$ and we choose a linear basis $\{f_1, \ldots, f_N\}$ of $\mathcal{W}$ consisting of homogeneous invariants. Note that f_i's may be of different degrees.

We now show that there is a polynomial ring $\mathbb{K}[y_1, y_2, \ldots, y_N]$, on which K acts, such that $\mathbb{K}[y_1, \ldots, y_N]^K$ surjects onto $\mathbb{K}[\mathcal{V}]^G$. We define the *categorical quotient* $\mathcal{V} /\!\!/ H$ of $\mathcal{V}$ by the action of H as the affine variety $\operatorname{maxSpec}(\mathbb{K}[\mathcal{V}]^H)$, corresponding to the ring $\mathbb{K}[\mathcal{V}]^H$. Explicitly, we may assume that $\mathcal{V} /\!\!/ H$ is the image of the polynomial map $\mathcal{V} \to \mathbb{K}^N$, $v \mapsto (f_1(v), f_2(v), \ldots, f_N(v))$. We denote the image of a vector $v \in \mathcal{V}$ under this map by $[v]$. Since the actions of K and H commute, there is a natural K-action on $\mathcal{V} /\!\!/ H$ via

$$k \cdot [v] = [k \cdot v].$$

Note that this is well-defined since K sends H-invariants to H-invariants.

There is a map from the categorical quotient $\mathcal{V} /\!\!/ H$ to the dual vector space $\mathcal{W}^*$, defined by

$$\Phi : \mathcal{V} /\!\!/ H \to \mathcal{W}^*, \qquad \Phi([v]) = \mathrm{ev}_v \text{ where } \mathrm{ev}_v(f) := f(v).$$

The corresponding ring map is the $\mathbb{K}$-algebra homomorphism $\Phi^\sharp : \mathbb{K}[\mathcal{W}^*] \to \mathbb{K}[\mathcal{V}]^H$, defined by $F \mapsto F \circ \Phi$. Since $\mathcal{W}$ is N-dimensional, we can identify $\mathbb{K}[\mathcal{W}^*] = \mathbb{K}[y_1, y_2, \ldots, y_N]$ and $\Phi^\sharp$ becomes

$$\Phi^\sharp : \mathbb{K}[y_1, y_2, \ldots, y_N] \to \mathbb{K}[\mathcal{V}]^H, \quad \Phi^\sharp(y_i) = f_i \text{ for } i = 1, 2, \ldots, N.$$

We note that if $F(y_1, \ldots, y_N) \in \mathbb{K}[y_1, \ldots, y_N]$ is a polynomial of degree D, then $\Phi^\sharp(F)$ has degree at most $D \cdot \max_i \deg(f_i) = D \cdot \beta_H$.

Since the subgroups H and K commute, K acts on $\mathcal{W}$ by a restriction of its action on $\mathbb{K}[\mathcal{V}]$. More concretely, $k \in K$ acts by $(k \cdot f)(v) = f(k^{-1} \cdot v) = (f \circ \pi_K(k^{-1}))(v)$ for $f \in \mathcal{W}$ and $v \in \mathcal{V}$. This action induces an action of K on $\mathcal{W}^*$ by $(k \cdot \varphi)(f) = \varphi(k^{-1} \cdot f) = \varphi(f \circ \pi_K(k))$. More concretely $\mathcal{W}^* \cong \bigoplus_{d=1}^{\beta_H} S^d(\mathcal{V})^H$ as representations of K, where $S^d(\cdot)$ denotes the d-th symmetric power. Hence, for $v \in \mathcal{V}$ we have $(k \cdot \mathrm{ev}_v)(f) = \mathrm{ev}_v(k^{-1} \cdot f) = f(k \cdot v) = \mathrm{ev}_{k \cdot v}(f)$. This implies that

$$k \cdot \mathrm{ev}_v = \mathrm{ev}_{k \cdot v} \tag{8}$$

▶ **Lemma 20.** *The map Φ is a K-equivariant embedding and we have $\Phi^\sharp(\mathbb{K}[y_1, \ldots, y_N]^K) = \mathbb{K}[\mathcal{V}]^G$.*

Proof. For $[v] \in \mathcal{V} /\!\!/ H$ and $k \in K$ we have

$$\Phi(k \cdot [v]) = \Phi([k \cdot v]) = \mathrm{ev}_{k \cdot v} = k \cdot \mathrm{ev}_v = k \cdot \Phi([v]),$$

where we use Equation (8) in the third equality. This shows that Φ is K-equivariant.

The ring map $\Phi^\sharp : \mathbb{K}[\mathcal{W}^*] \to \mathbb{K}[\mathcal{V} /\!\!/ H] = \mathbb{K}[\mathcal{V}]^H$ is surjective since $\Phi^\sharp(y_i) = f_i$ and $f_1, f_2, \ldots, f_N$ generate the invariant ring $\mathbb{K}[\mathcal{V}]^H$. Thus, Φ is an embedding, and the second claims follows from Proposition 12. ◀

We are now ready to prove our main technical theorem: a fixed-parameter degree bound for $\beta_G(\mathcal{V})$.

▶ **Theorem 21** (Fixed-parameter degree bounds). *In the setup of this section, we have*

$$\sigma_G(\mathcal{V}) \leq m \cdot (\beta_H)^{m^2} \quad \text{and} \quad \beta_G(\mathcal{V}) \leq \frac{3}{8} \cdot m^3 n^2 \cdot (\beta_H)^{2m^2}.$$

In particular, $\beta_G(\mathcal{V}) = \mathrm{poly}(n)$ when m is a constant.

Proof. Note that the ring map $\Phi^\sharp$ has degree $\max_i \deg(f_i) = \beta_H$ since $\Phi^\sharp(y_i) = f_i$ and $\mathbb{K}[\mathcal{W}^*] = \mathbb{K}[y_1, \ldots, y_N]$. Hence, Lemma 20 and Corollary 13 imply that

$$\sigma_G(\mathcal{V}) \leq \beta_H \cdot \sigma_K(\mathcal{W}^*).$$

We now claim that $\sigma_K(\mathcal{W}^*) \leq m \cdot (\beta_H)^{m^2-1}$. Let $\rho : K \to \mathrm{GL}(\mathcal{W}^*)$ denote the representation of $K = \mathrm{SL}_m$ that is induced from its action on $\mathcal{W}^*$. The action of K on $\mathcal{W}$ is a restriction of its action on $\mathbb{K}[\mathcal{V}] = \bigoplus_{\ell \geq 0} S^\ell(\mathcal{V}^*)$. Since $\mathcal{W}$ consists of polynomials of degree at most β_H, $\mathcal{W}^*$ is a K-stable subspace of $\bigoplus_{\ell=1}^{\beta_H} S^\ell(\mathcal{V})$. In particular, as matrices we have for $k \in K = \mathrm{SL}_m$

$$\rho(k) = \left(\bigoplus_{\ell=1}^{\beta_H} S^\ell(\pi_K(k)) \right) \Bigg|_{\mathcal{W}^*}, \tag{9}$$

where $S^\ell(\pi_K(k))$ denotes the ℓ-th symmetric matrix power. By the definition of π_K, the degrees of the matrix coefficients $\pi_K(k)$ have degree 1. Hence, the degrees of the matrix coefficients of $\bigoplus_{\ell=1}^{\beta_H} S^\ell(\pi_K(k))$ are bounded by β_H. Since restriction does not increase the degree, we deduce that the degrees of the matrix coefficients of $\rho(k)$ are bounded by β_H.

Now, by Corollary 10 we have

$$\sigma_K(\mathcal{W}^*) \leq m \cdot (\beta_H)^{m^2-1}.$$

Here a key property of Corollary 10 is that $\dim(\mathcal{W}^*)$, which is exponential in n, does not play a role in this bound. This finishes the proof of the first claim. The second claim simply follows from Theorem 11 using the upper bound for $\sigma_G(\mathcal{V})$. ◀

▶ Remark 22. In the above theorem, it is important to upper bound $\sigma_K(\mathcal{W})$ first and use it to upper bound $\sigma_G(\mathcal{V})$ and then deduce an upper bound for $\beta_G(\mathcal{V})$ using Theorem 11. If we instead upper bound $\beta_K(\mathcal{W})$ first and then pull-back the bound along $\Phi^\sharp$ to get an upper bound for $\beta_G(\mathcal{V})$, we get the worse upper bound $\beta_G(\mathcal{V}) \leq \frac{3}{8} N \cdot m^2 \cdot (\beta_H)^{2m^2-1}$, which is exponentially large in n.

4 Efficient Succinct Computation of Generating Invariants

In this section we will show that, in the setting where $m = O(1)$, we can efficiently compute a succinct circuit that computes the generating invariants for the tensor action of $G := \mathrm{SL}_n \times \mathrm{SL}_n \times \mathrm{SL}_m$ on $\mathcal{V} := \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$. Once we have this succinct circuit, we will be able to give a randomized algorithm for the Orbit Closure Intersection problem. Before we show this, we will need some elementary results which are folklore in the literature, and thus we present them here with full proofs. For a vector $\mathbf{u} \in \mathbb{K}^t$, let $V(\mathbf{u}) \in \mathrm{Mat}(t, \mathbb{K})$ be the (square) Vandermonde matrix given by $V(\mathbf{u})_{i,j} := \mathbf{u}_i^j$, where $j \in \{0, \ldots, t-1\}$. The Vandermonde matrix is invertible whenever the entries of $\mathbf{u}$ are pairwise distinct. In this section, given a tuple of variables $\mathbf{y} := (y_1, \ldots, y_m)$ and $\mathbf{e} \in \mathbb{Z}^m$, we define

$$\mathbf{y}^{\mathbf{e}} := \begin{cases} \prod_{i \in [m]} y_i^{e_i}, & \text{if } \mathbf{e} \in \mathbb{N}^m, \\ 0, & \text{otherwise.} \end{cases}$$

▶ **Proposition 23** (Multivariate Vandermonde). *Let $m, D \in \mathbb{N}$ and $\mathcal{S} := \{0, \dots, D\}$. In the space of square matrices $\mathrm{Mat}((D+1)^m, \mathbb{K})$, index the rows and columns by tuples from $\mathcal{S}^m$. Lastly, let $M_{\mathcal{S}} \in \mathrm{Mat}((D+1)^m, \mathbb{K})$ be the matrix defined as $M_{\mathcal{S}}(\mathbf{a}, \mathbf{b}) := \mathbf{a}^{\mathbf{b}}$ for $\mathbf{a}, \mathbf{b} \in \mathcal{S}^m$. Then, $M_{\mathcal{S}} = V(0, \dots, D)^{\boxtimes m}$, and thus, $M_{\mathcal{S}}$ is invertible.*

Proof. Let $V := V(0, \dots, D)$ and $M := V^{\boxtimes m}$. Since $M \in \mathrm{Mat}((D+1)^m, \mathbb{K})$, let us compute $M(\mathbf{a}, \mathbf{b})$. By definition, we have $M(\mathbf{a}, \mathbf{b}) = \prod_{i \in [m]} V_{\mathbf{a}_i, \mathbf{b}_i} = \prod_{i \in [m]} \mathbf{a}_i^{\mathbf{b}_i} = \mathbf{a}^{\mathbf{b}} = M_{\mathcal{S}}(\mathbf{a}, \mathbf{b})$, as we wanted. ◀

▶ **Proposition 24** (Coefficients and evaluations). *Let $D \in \mathbb{N}$, $\mathcal{S} := \{0, 1, \dots, D\}$, R be a graded $\mathbb{K}$-algebra with $R_0 = \mathbb{K}$, $\mathbf{y} := (y_1, \dots, y_m)$ and $P \in R[\mathbf{y}]$ be such that $\deg_{y_i} P \leq D$ for all $i \in [m]$. Thus, we can write $P = \sum_{\mathbf{a} \in \mathcal{S}^m} P_{\mathbf{a}} \cdot \mathbf{y}^{\mathbf{a}}$ for some $P_{\mathbf{a}} \in R$. Let $\mathbf{v}_P \in R^{\mathcal{S}^m}$ be defined as $(\mathbf{v}_P)_{\mathbf{a}} := P_{\mathbf{a}}$ and $\mathbf{w}_P \in R^{\mathcal{S}^m}$ be defined as $(\mathbf{w}_P)_{\mathbf{a}} := P(\mathbf{a})$. Then, we have that $\mathbf{w}_P = M_{\mathcal{S}} \cdot \mathbf{v}_P$, where $M_{\mathcal{S}}$ is the matrix from Proposition 23.*

Proof. Since $(\mathbf{w}_P)_{\mathbf{a}} = P(\mathbf{a}) = \sum_{\mathbf{b} \in \mathcal{S}^m} P_{\mathbf{b}} \cdot \mathbf{a}^{\mathbf{b}} = \sum_{\mathbf{b} \in \mathcal{S}^m} M_{\mathcal{S}}(\mathbf{a}, \mathbf{b}) \cdot P_{\mathbf{b}}$, which is the $\mathbf{a}^{th}$ entry of the matrix-vector product $M_{\mathcal{S}} \mathbf{v}_P$, this proves the above proposition. ◀

With the above facts, we can give a uniform algorithm to compute any homogeneous partial derivative operator (of a given individual degree) of a given circuit.

▶ **Proposition 25.** *Let $\mathbf{x} := (x_1, \dots, x_r)$ and $\mathbf{z} := (z_1, \dots, z_m)$ be tuples of variables and let $P \in \mathbb{K}[\mathbf{x}, \mathbf{z}]$ be a polynomial with $\deg_{z_i}(P) \leq D$ for all $i \in [m]$. Let $\partial_{\mathbf{z}} := (\partial_{z_1}, \dots, \partial_{z_m})$ and $Q \in \mathbb{K}[\partial_{\mathbf{z}}]$ be a partial differential operator with $\deg_{\partial_{z_i}}(Q) \leq D$ for all $i \in [m]$. If P can be computed by a uniform algebraic circuit Ψ of size $s \geq m$, then $(Q \circ P)\,|_{\mathbf{z}=\mathbf{0}}$ can be computed by a uniform algebraic circuit Φ of size $O((D+1)^{3m+3} \cdot s)$.*

Proof. Take $\mathcal{S} := \{0, \dots, D\}$ and $M := M_{\mathcal{S}}$ as in Proposition 23. Since $\deg_{z_i}(P) \leq D$ for all $i \in [m]$, we can write $P = \sum_{\mathbf{a} \in \mathcal{S}^m} P_{\mathbf{a}}(\mathbf{x}) \cdot \mathbf{z}^{\mathbf{a}}$ where $P_{\mathbf{a}} \in \mathbb{K}[\mathbf{x}]$. Similarly, we can write $Q(\partial_{\mathbf{z}}) = \sum_{\mathbf{b} \in \mathcal{S}^m} Q_{\mathbf{b}} \cdot \partial_{\mathbf{z}}^{\mathbf{b}}$ with $Q_{\mathbf{b}} \in \mathbb{K}$. Thus, we get that

$$(Q \circ P)\,|_{\mathbf{z}=\mathbf{0}} = \sum_{\mathbf{a} \in \mathcal{S}^m} P_{\mathbf{a}}(\mathbf{x}) \cdot (Q(\partial_{\mathbf{z}}) \circ \mathbf{z}^{\mathbf{a}})\,|_{\mathbf{z}=\mathbf{0}} = \sum_{\mathbf{a} \in \mathcal{S}^m} P_{\mathbf{a}}(\mathbf{x}) \cdot (\partial_{\mathbf{z}}^{\mathbf{a}} \circ Q(\mathbf{z}))\,|_{\mathbf{z}=\mathbf{0}}$$

where $Q(\mathbf{z}) := \sum_{\mathbf{b} \in \mathcal{S}^m} Q_{\mathbf{b}} \cdot \mathbf{z}^{\mathbf{b}}$.

Let $\mathbf{v}_Q, \mathbf{w}_Q \in \mathbb{K}^{\mathcal{S}^m}$ and $\mathbf{v}_P, \mathbf{w}_P \in \mathbb{K}[\mathbf{x}]^{\mathcal{S}^m}$ be defined as in Proposition 24. With these definitions, Proposition 24 and the fact that $(\partial_{\mathbf{z}}^{\mathbf{a}} \circ Q(\mathbf{z}))\,|_{\mathbf{z}=\mathbf{0}} = Q_{\mathbf{a}}$ tell us that the above equation can be written in matrix-vector product format as:

$$(Q \circ P)\,|_{\mathbf{z}=\mathbf{0}} = \mathbf{v}_P^T \cdot \mathbf{v}_Q = (M^{-1}\mathbf{w}_P)^T \cdot \mathbf{v}_Q = \mathbf{w}_P \cdot (M^{-1})^T \cdot \mathbf{v}_Q.$$

Since $(\mathbf{w}_P)_{\mathbf{b}} = \Psi(\mathbf{b})$, we can compute $\mathbf{w}_P$ by $(D+1)^m$ distinct evaluations to Ψ. As Q is a $(D+1)^m$-sparse polynomial, we can (uniformly) compute $\mathbf{v}_Q$ simply by writing all coefficients of Q in its sparse representation. By the definition of M we can (uniformly) compute it with $(D+1)^m \cdot D^3 \cdot m$ algebraic operations. In particular, we can (uniformly) compute $(M^{-1})^T$ by the naive algorithm using $O((D+1)^{3m})$ algebraic operations. This implies that $(Q \circ P)|_{\mathbf{z}=\mathbf{0}}$ can be computed by $O((D+1)^{3m+3} \cdot s)$ operations. ◀

We can now construct an efficient succinct circuit for a generating set of invariants for the tensor action. We recall the setting of Section 4, that is, $\mathcal{V} := \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$, with $G := \mathrm{SL}_n \times \mathrm{SL}_n \times \mathrm{SL}_m$ acting on $\mathcal{V}$ via the natural tensor action. We denote $H := \mathrm{SL}_n \times \mathrm{SL}_n$ and $K := \mathrm{SL}_m$ so G is the direct product of its subgroups H and K.

▶ **Theorem 26** (Efficient Succinct Generation of Invariants)**.** *There is a universal constant $c > 0$ such that the following holds. Given any $D \in \mathbb{N}$, there exists a uniform algebraic circuit Φ of size $\leq c \cdot (Dnm^3)^{3m^2+6}$ that computes a succinct generator for the vector space $\mathbb{K}[\mathcal{V}]^G_{\leq D}$. Moreover, the circuit Φ has $1 + m \cdot D^2$ auxiliary variables and computes a polynomial with individual degree upper bounded by D.*

Proof. By Equation (7), we have $\mathbb{K}[\mathcal{V}]^G = (\mathbb{K}[\mathcal{V}]^H)^K$ as graded $\mathbb{K}$-algebras. Thus, we have $\mathbb{K}[\mathcal{V}]^G_{\leq D} = \left(\mathbb{K}[\mathcal{V}]^H_{\leq D}\right)^K$. By Theorem 14, we have that $\mathbb{K}[\mathcal{V}]^H$ is spanned by the homogeneous polynomials

$$f_T(X) := \det\left(\sum_{i=1}^{m} T_i \otimes X_i\right),$$

where $T = (T_1, \ldots, T_m) \in \mathrm{Mat}(d, \mathbb{K})^m$ and $d \in \mathbb{N}$. In particular, if $N := \dim \mathbb{K}[V]^H_{\leq D}$, there are N tuples of matrices $\left\{T^{(e)}\right\}_{e\in[N]}$, where $T^{(e)} \in \mathrm{Mat}(d_e, \mathbb{K})^m$ for some $d_e \leq D$ such that, denoting by $f_e(X) := f_{T^{(e)}}(X)$, we have that $\mathbb{K}[\mathcal{V}]^H_{\leq D} = \mathrm{span}_{\mathbb{K}} \{f_e\}_{e\in[N]}$. Thus, we have

$$\mathbb{K}[\mathcal{V}]^G_{\leq D} = \mathrm{span}_{\mathbb{K}} \{\mathcal{R}_G(f_e)\}_{e\in[N]} = \mathrm{span}_{\mathbb{K}} \{\mathcal{R}_K(f_e)\}_{e\in[N]}, \tag{10}$$

where the last equality above comes from the fact that $\mathcal{R}_G = \mathcal{R}_K \circ \mathcal{R}_H$, by Lemma 17, together with the fact that $\mathcal{R}_H(f_e) = f_e$ since f_e is an H-invariant.

Now, let $Y := (Y_1, \ldots, Y_m)$ be a tuple of symbolic $D \times D$ matrices, and for each $r \in [D]$ and $i \in [m]$, let $Y_i^{(r)} := (Y_i)_{[r]\times[r]}$ be the top-left $r \times r$ principal submatrix of Y_i. Define

$$F_r(Y, X) := f_{Y^{(r)}}(X) = \det\left(\sum_{i=1}^{m} Y_i^{(r)} \otimes X_i\right).$$

Letting u be a formal variable and using the Lagrange interpolation polynomials $L_r(u) := \prod_{\substack{i\in[D]\\ i\neq r}} \frac{u-i}{r-i}$, we have that the following polynomial is a succinct generator for the non-constant polynomials in $\mathbb{K}[V]^H_{\leq D}$

$$F(u, Y, X) := \sum_{r=1}^{D} L_r(u) \cdot F_r(Y, X). \tag{11}$$

To see that the above polynomial is a succinct generator, note that for any $e \in [N]$, if we set $u = d_e$ and $\widehat{Y}_i = T^{(e)} \oplus 0_{D-d_e}$, we have that $F(d_e, \widehat{Y}, X) = L_{d_e}(d_e) \cdot F_{d_e}(\widehat{Y}, X) = f_{T^{(e)}}(X) = f_e$.

The above, together with Equation (10) implies that $\mathcal{R}_K(F(u, Y, X))$ is a succinct generator for $\mathbb{K}[V]^G_{\leq D}$. Using Equation (6) to express $\mathcal{R}_K$ in terms of Ω, we have that the action of $\mathcal{R}_K$ on $F(u, Y, X)$ is given by

$$\begin{aligned}
\mathcal{R}_K(F(u, Y, X)) &= \sum_{r=1}^{D} L_r(u) \cdot \mathcal{R}_K(F_r(Y, X)) = \sum_{r=1}^{D} L_r(u) \cdot \mathcal{R}_K(f_{Y^{(r)}}(X)) \\
&= \sum_{r=1}^{D} L_r(u) \cdot c^{-1}_{rn(m-1)/m,m} \cdot \Omega^{rn(m-1)/m} \left(\widetilde{\pi}_K(Z) \circ f_{Y^{(r)}}(X)\right)) \\
&= \sum_{r=1}^{D} L_r(u) \cdot c^{-1}_{rn(m-1)/m,m} \cdot \Omega^{rn(m-1)/m} \left(f_{Y^{(r)}}(\pi_K(Z)^{-1} \circ X))\right). \qquad (12)
\end{aligned}$$

By definition of the Ω-process (see Equation (5)), we have that $\Omega^{rn(m-1)/m}$ is a polynomial derivative operator over m^2 variables, with integral coefficients, and of individual degree $\leq rn(m-1)/m$. In particular, its sparse representation has $\leq (rn(m-1)/m)^{m^2}$ terms, each monomial having an integral coefficient, and they can all be simultaneously computed uniformly with $(rn(m-1)/m)^{2m^2}$ algebraic operations by the naive algorithm. Note that $f_{Y^{(r)}}(\pi_K(Z)^{-1} \circ X)$ can be computed by a (uniform) circuit Ψ_r of size $O((rn \cdot m^3)^3)$, as $f_{Y^{(r)}}$ is the determinant of size $rn \times rn$ symbolic matrix whose entries involve determinants of size m matrices.[4] By Proposition 25 there is a universal constant $C > 0$ and a (uniform) circuit Φ_r of size $C \cdot (rn)^{3m^2+3} \cdot (rnm^3)^3$ such that Φ_r computes $\Omega^{rn(m-1)/m}\left(f_{Y^{(r)}}(\pi_K(Z)^{-1} \circ X)\right)$.[5] Note that the Ω-process constants can also be computed by a uniform circuit with the same upper bound as above.

Combining the circuits Φ_r with $L_r(u)$ and the Ω-process constants, we obtain a uniform circuit Φ of size $\leq 3 \cdot C \cdot (Dnm^3)^{3m^2+6}$ computing $\mathcal{R}_K(F(u, Y, X))$. Taking $c := 3C$ finishes the proof. It is easy to see from the above construction the bound on the number of auxiliary variables as well as on the individual degree of the succinct generator. ◀

We can now prove our second main result: Theorem 8, which gives a uniform, polynomial-sized succinct generating circuit for the natural tensor action, whenever the third tensor factor is of constant dimension.

▶ **Theorem 8.** *Assume $m = O(1)$. Then, there is a uniform family of $\mathrm{poly}(n)$-sized arithmetic circuit C that succinctly encodes $\mathbb{K}[\mathcal{V}]^G$.*

Proof of Theorem 8. By Theorem 21, the invariant ring $\mathbb{K}[\mathcal{V}]^G$ is generated by invariants of degree at most $D := \frac{3}{8} \cdot m^3 n^2 \cdot (\beta_H)^{2m^2} \leq \frac{3}{8} \cdot m^{2m^2+3} \cdot n^{4m^2+2}$, where the inequality follows from the bound $\beta_H \leq m \cdot n^4$ (Theorem 16).

By Theorem 26, there is an algebraic circuit of size

$$O\left((Dnm^3)^{3m^2+6}\right) = O(m^{6m^4+30m^2+36} \cdot n^{12m^4+33m^2+18}),$$

that succinctly encodes the invariants up to degree D. Since $\mathbb{K}[\mathcal{V}]^G$ is generated by forms of degree at most D, this circuit is a succinct encoding of the invariant ring. When $m = O(1)$, the size of the circuit is $\mathrm{poly}(n)$ and this finishes the proof. ◀

Given the above results, we can now prove our main application: a randomized, polynomial-time algorithm for the Orbit Closure Intersection problem for the natural tensor action, whenever the third factor is of constant dimension.

▶ **Theorem 3.** *The orbit closure intersection problem for the tensor action of $G := \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_n(\mathbb{K}) \times \mathrm{SL}_m(\mathbb{K})$ on $\mathcal{V} := \mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^m$ with $m = O(1)$ is in coRP. More explicitly, given n and two vectors $u, v \in \mathbb{Q}[i]^n \otimes \mathbb{Q}[i]^n \otimes \mathbb{Q}[i]^m$, there exists a randomized algorithm with one-sided bounded error that decides whether the orbit closures of u and v intersect in time polynomial in n and the bit-lengths of u and v.*

[4] As Z is from SL_m, $\pi_K(Z)^{-1}$ can be computed by taking the adjugate matrix.

[5] We can apply Proposition 25, since the derivative operator has the same Z-degree as the polynomial $f_{Y^{(r)}}(\pi_K(Z)^{-1} \circ X))$, and thus

$$\Omega^{rn(m-1)/m}\left(f_{Y^{(r)}}(\pi_K(Z)^{-1} \circ X))\right) = \left(\Omega^{rn(m-1)/m}\left(f_{Y^{(r)}}(\pi_K(Z)^{-1} \circ X))\right)\right)|_{Z=0}.$$

Proof. By Theorem 8, there is a uniform algebraic circuit $C(x_1, x_2, \ldots, x_{n^2m}, y_1, y_2, \ldots, y_r)$ of size $\mathrm{poly}(n)$ that succinctly encodes $\mathbb{K}[\mathcal{V}]^G$. In particular, the number of auxiliary variables r and the bit-lengths of the constants used in C are bounded by $\mathrm{poly}(n)$. By Theorem 26, the degree of C, i.e., the degree of the polynomial that C computes, is also bounded by $\mathrm{poly}(n)$.

Mumford's Theorem 5 implies that

$$\overline{G \cdot u} \cap \overline{G \cdot v} \neq \varnothing \quad \iff \quad C(u_1, \ldots, u_{n^2m}, a_1, \ldots, a_r) = C(v_1, \ldots, v_{n^2m}, a_1, \ldots, a_r)$$

for every $\mathbf{a} \in \mathbb{K}^r$. The latter is an instance of polynomial identity testing. By the Polynomial Identity Lemma, choosing each $a_1, a_2, \ldots, a_r$ uniformly at random from $[3 \deg C]$ and returning the result of the equality check $C(u_1, \ldots, u_{n^2m}, a_1, \ldots, a_r) = C(v_1, \ldots, v_{n^2m}, a_1, \ldots, a_r)$ gives us an algorithm with error probability bounded by $1/3$. Since $\mathrm{size}(C), \deg C \in \mathrm{poly}(n)$, we can compute $C(u, a)$ and $C(v, a)$ and test whether they are equal in $\mathrm{poly}(n, \langle u \rangle, \langle v \rangle)$-time where $\langle \cdot \rangle$ denotes the number of bits in the binary expansion of a vector. This finishes the proof. ◀

References

1. Arturo Acuaviva, Visu Makam, Harold Nieuwboer, David Pérez-Garcia, Friedrich Sittner, Michael Walter, and Freek Witteveen. The minimal canonical form of a tensor network. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*, pages 328–362. IEEE, 2023. doi:10.1109/FOCS57990.2023.00027.
2. Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Mendes de Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 172–181. ACM, 2018. doi:10.1145/3188745.3188942.
3. Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Phys. Rev. Lett.*, 70:1895–1899, March 1993. doi:10.1103/PhysRevLett.70.1895.
4. Charles H. Bennett, Sandu Popescu, Daniel Rohrlich, John A. Smolin, and Ashish V. Thapliyal. Exact and asymptotic measures of multipartite pure-state entanglement. *Physical Review A*, 63:012307, December 2000. doi:10.1103/PhysRevA.63.012307.
5. Markus Bläser. A $5/2n^2$-lower bound for the multiplicative complexity of $n \times n$-matrix multiplication. In *STACS 2001*, pages 99–109. Springer, 2001.
6. Peter A. Brooksbank and Eugene M. Luks. Testing isomorphism of modules. *Journal of Algebra*, 320(11):4020–4029, 2008. Computational Algebra. doi:10.1016/j.jalgebra.2008.07.014.
7. Peter Bürgisser. Optimization, complexity and invariant theory (invited talk). In *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021*, volume 187, pages 1:1–1:20. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.STACS.2021.1.
8. Peter Bürgisser, Michael Clausen, and Mohammad A Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013.
9. Peter Bürgisser, Mahmut Levent Doğan, Visu Makam, Michael Walter, and Avi Wigderson. Polynomial Time Algorithms in Invariant Theory for Torus Actions. In *36th Computational Complexity Conference (CCC 2021)*, pages 32:1–32:30. Schloss Dagstuhl, 2021. doi:10.4230/LIPIcs.CCC.2021.32.
10. Peter Bürgisser, Mahmut Levent Doğan, Visu Makam, Michael Walter, and Avi Wigderson. Complexity of Robust Orbit Problems for Torus Actions and the abc-Conjecture. In *39th Computational Complexity Conference (CCC 2024)*, volume 300, pages 14:1–14:48. Schloss Dagstuhl, 2024. doi:10.4230/LIPIcs.CCC.2024.14.

11 Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Towards a theory of non-commutative optimization: Geodesic 1st and 2nd order methods for moment maps and polytopes. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 845–861. IEEE, 2019. arXiv:1910.12375, version 3. arXiv:1910.12375.

12 Peter Bürgisser and Christian Ikenmeyer. Explicit lower bounds via geometric complexity theory. In *Symposium on Theory of Computing Conference, STOC 2013*, pages 141–150. ACM, 2013. doi:10.1145/2488608.2488627.

13 Zhili Chen, Joshua A. Grochow, Youming Qiao, Gang Tang, and Chuanqi Zhang. On the complexity of isomorphism problems for tensors, groups, and polynomials III: actions by classical groups. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, Berkeley, CA, USA, January 30 - February 2, 2024*, volume 287 of *LIPIcs*, pages 31:1–31:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ITCS.2024.31.

14 Harm Derksen. Polynomial bounds for rings of invariants. *Proceedings of the American Mathematical Society*, 129(4):955–963, 2001.

15 Harm Derksen and Gregor Kemper. *Computational Invariant Theory*. Springer, 2015.

16 Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants. *Advances in Mathematics*, 310:44–63, 2017.

17 Harm Derksen and Visu Makam. Algorithms for orbit closure separation for invariants and semi-invariants of matrices. *Algebra & Number Theory*, 14(10):2791–2813, 2020.

18 Harm Derksen and Visu Makam. An exponential lower bound for the degrees of invariants of cubic forms and tensor actions. *Advances in Mathematics*, 368:107136, 2020.

19 Harm Derksen and Jerzy Weyman. Semi-invariants of quivers and saturation for littlewood-richardson coefficients. *Journal of the American Mathematical Society*, 13(3):467–479, 2000.

20 Mátyás Domokos and Alexander N Zubkov. Semi-invariants of quivers as determinants. *Transformation groups*, 6(1):9–24, 2001.

21 Michael A. Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 527–542, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-40328-6_37.

22 William Cole Franks and Philipp Reichenbach. Barriers for recent methods in geodesic optimization. In *36th Computational Complexity Conference, CCC 2021*, volume 200, pages 13:1–13:54. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.13.

23 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Operator scaling: theory and applications. *Foundations of Computational Mathematics*, 20(2):223–290, 2020. doi:10.1007/S10208-019-09417-Z.

24 Ankit Garg, Christian Ikenmeyer, Visu Makam, Rafael Mendes de Oliveira, Michael Walter, and Avi Wigderson. Search problems in algebraic complexity, gct, and hardness of generators for invariant rings. In *35th Computational Complexity Conference, CCC 2020*, volume 169, pages 12:1–12:17. Schloss Dagstuhl, 2020. doi:10.4230/LIPIcs.CCC.2020.12.

25 Joshua A. Grochow. Matrix Isomorphism of Matrix Lie Algebras. In *2012 IEEE 27th Conference on Computational Complexity*, pages 203–213, 2012. doi:10.1109/CCC.2012.34.

26 Joshua A. Grochow and Youming Qiao. On p-Group Isomorphism: Search-To-Decision, Counting-To-Decision, and Nilpotency Class Reductions via Tensors. In *36th Computational Complexity Conference (CCC 2021)*, volume 200, pages 16:1–16:38. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.16.

27 Joshua A. Grochow and Youming Qiao. On the Complexity of Isomorphism Problems for Tensors, Groups, and Polynomials I: Tensor Isomorphism-Completeness. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185, pages 31:1–31:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.31.

28 Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials IV: linear-length reductions and their applications. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 766–776. ACM, 2025. doi:10.1145/3717823.3718282.

29 Joshua A. Grochow, Youming Qiao, Katherine E. Stange, and Xiaorui Sun. On the complexity of isomorphism problems for tensors, groups, and polynomials V: over commutative rings. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 777–784. ACM, 2025. doi:10.1145/3717823.3718286.

30 Masaki Hamada and Hiroshi Hirai. Computing the nc-rank via discrete convex optimization on cat (0) spaces. *SIAM Journal on Applied Algebra and Geometry*, 5(3):455–478, 2021. doi:10.1137/20M138836X.

31 David Hilbert. Über die Theorie der algebraischen Formen. *Math. Ann.*, 36(4):473–534, 1890. doi:10.1007/BF01208503.

32 David Hilbert. Über die vollen Invariantensysteme. *Math. Ann.*, 42:313–373, 1893. URL: http://eudml.org/doc/157652.

33 Pavel Hrubeš and Avi Wigderson. Non-commutative arithmetic circuits with division. *Theory Comput.*, 11:357–393, 2015. doi:10.4086/TOC.2015.V011A014.

34 Gábor Ivanyos and Youming Qiao. On the orbit closure intersection problems for matrix tuples under conjugation and left-right actions. In *2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023*, pages 4115–4126. SIAM, 2023. doi:10.1137/1.9781611977554.CH158.

35 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative edmonds' problem and matrix semi-invariants. *Comput. Complex.*, 26(3):717–763, 2017. doi:10.1007/S00037-016-0143-X.

36 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *Comput. Complex.*, 27(4):561–593, 2018. doi:10.1007/S00037-018-0165-7.

37 Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In *Theory of Cryptography - 17th International Conference, TCC 2019*, volume 11891 of *Lecture Notes in Computer Science*, pages 251–281. Springer, 2019. doi:10.1007/978-3-030-36030-6_11.

38 Neeraj Kayal and Nitin Saxena. Complexity of Ring Morphism Problems. *computational complexity*, 15(4):342–390, December 2006. doi:10.1007/s00037-007-0219-8.

39 Hanspeter Kraft. *Geometrische Methoden in der Invariantentheorie*. Aspects of Mathematics, D1. Friedr. Vieweg & Sohn, Braunschweig, 1984. doi:10.1007/978-3-322-83813-1.

40 Vladimir Lysikov and Michael Walter. Complexity theory of orbit closure intersection for tensors: reductions, completeness, and graph isomorphism hardness, 2025. doi:10.48550/arXiv.2411.04639.

41 Ketan Mulmuley. Geometric complexity theory v: Efficient algorithms for noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.

42 Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory I: an approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526, 2001. doi:10.1137/S009753970038715X.

43 David Mumford, John Fogarty, and Frances Kirwan. *Geometric invariant theory*, volume 34. Springer Science & Business Media, 1994.

44 Vladimir L Popov. Constructive invariant-theory. *Astérisque*, pages 303–334, 1981.

45 Vladimir L Popov. The constructive theory of invariants. *Mathematics of the USSR-Izvestiya*, 19(2):359, 1982.

46 Aidan Schofield and Michel Van den Bergh. Semi-invariants of quivers for arbitrary dimension vectors. *Indagationes Mathematicae*, 12(1):125–138, 2001.

47 Nicolas Sendrier and Dimitris E. Simos. The Hardness of Code Equivalence over $\mathbb{F}_q$ and Its Application to Code-Based Cryptography. In *Post-Quantum Cryptography*, pages 203–216, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-38616-9_14.

48 V Strassen. Relative bilinear complexity and matrix multiplication. *Journal für die reine und angewandte Mathematik*, 374:406–443, 1987.

49 Avi Wigderson. Non-commutative optimization - where algebra, analysis and computational complexity meet. In *ISSAC '22: International Symposium on Symbolic and Algebraic Computation, 2022*, pages 13–19. ACM, 2022. doi:10.1145/3476446.3535489.

Multilinear Formula Lower Bounds for Sparse Determinants

Pruthvi Boyapati
CSTAR, International Institute of Information Technology, Hyderabad, India

Suryajith Chillara
CSTAR, International Institute of Information Technology, Hyderabad, India

Pratyush Vempati
CSTAR, International Institute of Information Technology, Hyderabad, India

Abstract

Raz (2009) proved that multilinear formulas computing the determinant of a generic $n \times n$ matrix require size $n^{\Omega(\log n)}$. A fundamental question in understanding this lower bound is identifying which structural properties of the determinant drive this hardness. Is it the quadratic number of variables? The dense connectivity? Specific algebraic symmetries?

We establish that *density is not essential*. We prove the existence of $n \times n$ symbolic matrices with only $\Theta(n \log^6 n)$ nonzero entries – reducing the variable count by a factor of $n/\log^6 n$ – such that any multilinear formula computing their determinants still requires size $n^{\Omega(\log n)}$. Our construction uses rectangle sampling from the complete bipartite graph to generate sparse matrices that simultaneously maintain perfect matchings (ensuring nonzero determinant) while exhibiting *diagonal imbalance* under random vertex permutations – a geometric property we identify as the key driver of factor imbalance in Raz's framework.

This demonstrates that Raz's partial derivatives method is remarkably robust to sparsification, and suggests that the fundamental source of multilinear hardness for determinant lies in expansion-like combinatorial structure rather than density. Our techniques combine concentration inequalities for dependent random variables with insights from random graph theory.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Determinants, Multilinear polynomials, Formula lower bounds

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.33

Acknowledgements We would like to thank the anonymous referees of CCC 2026 for their careful readings and for their valuable comments and insights. SC thanks Srikanth Srinivasan for informative discussions about the multilinear model of computation over the years and in particular, pointing out the VBP-hardness of sparse determinants in this work.

1 Introduction

The computational complexity of the determinant and permanent of a generic $n \times n$ matrix has been a central question in algebraic complexity theory. Valiant [17] established fundamental completeness results in his algebraic complexity framework: the permanent is VNP-complete under p-projections, while the determinant lies in the more restricted class VBP (polynomial-size algebraic branching programs) and it characterizes VP through quasi-polynomial projections. Furthermore, Valiant [18] showed that computing the permanent of $\{0,1\}$ matrices is #P-complete, establishing a direct connection between algebraic and Boolean complexity. The separation between determinant and permanent complexity parallels the P versus NP question in Boolean complexity.

While the determinant can be computed in polynomial time via Gaussian elimination over fields, computing determinants over general commutative rings or in parallel complexity classes requires more sophisticated techniques. Berkowitz [1] developed an elegant algorithm

© Pruthvi Boyapati, Suryajith Chillara, and Pratyush Vempati;
licensed under Creative Commons License CC-BY 4.0

that computes the characteristic polynomial (and hence the determinant) of an $n \times n$ matrix using $O(\log^2 n)$ parallel time with polynomially many processors, based on iterated matrix multiplication and the Cayley-Hamilton theorem. Mahajan and Vinay [8] provided a combinatorial characterization of the determinant through *closed walk sequences* (clow sequences), leading to an alternative iterated matrix multiplication approach that places the determinant computation in the complexity class GapL. Both algorithms demonstrate that despite the apparent sequential nature of Gaussian elimination, the determinant can be computed through carefully structured iterated matrix products, in parallel. In general, iterated matrix products are also referred to as *Algebraic Branching Programs* (ABPs) and they characterize the complexity class VBP.

A polynomial $f(x_1, \ldots, x_n)$ is *multilinear* if the degree of each variable x_i in every monomial is at most one. A formula is *syntactically multilinear* if every gate in the formula computes a multilinear polynomial. Both the determinant and permanent are multilinear polynomials, yet classical algorithms for computing the determinant (such as Mahajan-Vinay or Berkowitz's algorithm) crucially rely on intermediate computations that involve non-multilinear terms, which eventually cancel to produce the final multilinear result. This suggests that restricting to purely multilinear computation models might make these polynomials harder to compute. Indeed, Raz [11, 13] established the first superpolynomial lower bounds for multilinear formulas, proving that any multilinear formula computing the determinant or permanent of an $n \times n$ matrix requires size $n^{\Omega(\log n)}$. Raz [12] then proved that multilinear circuits are superpolynomially more powerful than multilinear formulas (see [14] for a simplified exposition). The lower bounds of Raz and subsequent work apply to syntactically multilinear formulas, where the multilinearity constraint is enforced at every gate.

While the determinant has a polynomial-sized ABP in the general setting, it is not known whether determinant can be computed by polynomial-sized multilinear ABPs. Raz and Yehudayoff [14] conjectured that multilinear ABPs might be exponentially more powerful than multilinear formulas. This conjecture was resolved by Dvir, Malod, Perifel, and Yehudayoff [6], who constructed an explicit polynomial family computable by polynomial-size multilinear ABPs but requiring superpolynomial multilinear formula size. Raz and Yehudayoff [14] proved that for every constant k, there exist polynomials computable by multilinear formulas of depth $k+1$ that require superpolynomial size when restricted to depth k, establishing a hierarchy for constant depths. This hierarchy was made *near-optimal* by Chillara, Engels, Limaye, and Srinivasan [4].

Shpilka and Yehudayoff [16, Open Problem 14] raised the following fundamental question – are general formulas strictly more powerful than multilinear formulas? Over large fields, Chillara, Limaye and Srinivasan [5] showed that the general formulas outperform multilinear formulas of depth $o(\log n)$. Extending this to $O(\log n)$ depth would resolve Shpilka and Yehudayoff's question as [5] also showed that the depth reduction of formulas to $O(\log n)$ depth ([3, 2]) is optimal. It is important to note that [5] provides evidence that the multilinearity restriction fundamentally limits computational power, and that even natural algebraic problems like iterated matrix multiplication over small matrices become significantly harder when intermediate cancellations of non-multilinear terms are forbidden.

1.1 Motivation and Our Results

A central challenge in algebraic complexity is understanding *why* certain restricted computation models admit lower bounds while others remain intractable. Raz's groundbreaking proof [11, 13] that multilinear formulas require size $n^{\Omega(\log n)}$ to compute the determinant is

our strongest lower bound for an explicit polynomial family in formula complexity. However, the proof's reliance on multiple properties of the determinant – dense connectivity, quadratic variable count, permutation-based symmetries – makes it unclear which aspects are essential and which are incidental.

Understanding the essential properties that drive multilinear hardness is valuable for two reasons. First, it guides the search for new lower bounds: if we can isolate the minimal structural requirements, we can identify other polynomials that might admit similar techniques. Second, it provides insight into the broader question of whether general formulas are fundamentally more powerful than multilinear formulas – an important open problem raised by Shpilka and Yehudayoff [16]. While recent work by Chillara, Limaye, and Srinivasan [5] shows that general formulas can outperform shallow multilinear formulas, the question remains open for logarithmic-depth formulas.

Our work takes the following approach: we systematically stress-test Raz's technique by removing structural properties to determine which are essential. Specifically, we investigate whether *density* – the fact that the determinant involves n^2 variables in a fully connected matrix – is necessary for the lower bound. If hardness persists for sparse determinants with only $O(n \operatorname{polylog} n)$ variables, this would demonstrate that density is not the core driver, and would suggest that Raz's method captures deeper combinatorial properties.

▶ **Theorem 1** (Lower Bound for Sparse Determinants)**.** *There exists a family of $n \times n$ sparse symbolic matrices $(X_G)_{n\in\mathbb{N}}$ (constructed via probabilistic sampling) with the following properties:*

1. *Each row and column has $\Theta(\log^6 n)$ nonzero entries,*
2. *The underlying bipartite graph G has a perfect matching, ensuring $\mathsf{Det}(X_G) \neq 0$,*
3. *Any multilinear formula computing $\mathsf{Det}(X_G)$ requires size $s = n^{\Omega(\log n)}$.*

Our result demonstrates that Raz's technique is remarkably robust: reducing the variable count from n^2 to $\Theta(n \log^6 n)$ – a factor of $n/\log^6 n$ – still yields a superpolynomial lower bound. Despite the weakening, we get the best possible lower bound of $n^{\Omega(\log n)}$.

▶ Remark 2. All results stated for determinant also hold for the permanent.

1.2 Proof overview

Since we build on the proof strategy of Raz [11], we shall present the overview of their proof and then point out the locations where our proof deviates from theirs.

Raz's partial derivative measure

Let f be a polynomial defined on the variable set X. Let $\Gamma : X \mapsto Y \sqcup Z$ be an equi-partition of X into two sets Y and Z. Let $\mathcal{M}_Y$ be the set of all multilinear monomials over the variables $\Gamma^{-1}(Y)$, and $\mathcal{M}_Z$ be the set of all multilinear monomials over the variables $\Gamma^{-1}(Z)$. Let M_f be a matrix whose rows and columns are indexed by elements of $\mathcal{M}_Y$ and $\mathcal{M}_Z$ respectively. An entry of this matrix M_f indexed by row m_Y and column m_Z is the coefficient of the monomial $m_Y \cdot m_Z$ in f. Naturally, rank of this matrix is well defined. Raz's complexity measure of the polynomial f is the rank of the matrix M_f. We shall henceforth denote it as $\mu(f)$. It is clear that $\mu(f) \leq 2^{|X|/2}$. Raz [11] made a fundamental observation that if f has a *multiplicative structure*, then under a random partition Γ, the rank could be much lower than the maximum possible rank with good probability.

Let $f = f_1 \times f_2$ where f_1 and f_2 are defined over the variable disjoint variable sets X_1 and X_2 where $X_1 \sqcup X_2 = X$. Let Y_1, Z_1 and Y_2, Z_2 be the partitions imposed by Γ on X_1 and X_2 respectively. It follows that $M_f = M_{f_1} \otimes M_{f_2}$ and thus $\mu(f) = \mu(f_1) \cdot \mu(f_2)$. If

$|Y_1| \neq |Z_1|$ then it is easy to see that $\mu(f_1)$ is at most $2^{(|X_1|-1)/2}$ and same would be the case for $\mu(f_2)$. Thus, under this structure of f and the partition Γ that imposes inequality in the factors f_1 and f_2, $\mu(f)$ can at most be $2^{\frac{|X|}{2}-1}$. Instead of two factors, if f was expressed as a product of t many factors and if at least ℓ of these t factors had this "discrepancy" then we get that $\mu(f)$ is at most $2^{\frac{|X|-\ell}{2}}$. Instead of asking for $|Y_i| \neq |Z_i|$, we could also ask for a difference of at least k between $|Y_i|$ and $|Z_i|$ for a "discrepancy" parameter k. For this, we need non-trivial guarantees on the size of X_i's depending on k.

It was also shown[1] that every multilinear formula F of size s that computes a polynomial f on n variables admits a product-decomposition of the following form.

$$f = \sum_{i=1}^{s+1} f_i \text{ where } f_i = \prod_{j=1}^{t} f_{ij}\,.$$

where $t = \Theta(\log n)$ and for each $j \in [t]$,

$$\left(\frac{1}{3}\right)^j |X| \leq |X_{ij}| \leq \left(\frac{2}{3}\right)^j |X|\,.$$

That is, at least $\Omega_\delta(\log n)$ factors $X_{i,j}$ are such that $|X_{ij}| \geq |X|^\delta$. This gives us the leeway to work with a suitable discrepancy factor k. To prove a lower bound, we need to show that there is a equi-partition of the variables under which the polynomial of interest attains full rank and the multilinear formula of small size attains rank that is far from full. Raz [12] notes that technique can be applied to a *full-rank polynomial* directly but not to dthe eterminant.

Since the determinant is a polynomial on n^2 many variables, Raz's measure on all n^2 variables would yield a matrix of with $2^{n^2/2}$ many rows and columns. But this matrix can never attain full rank or a rank that is near-full as the number of monomials in the determinant polynomial are at most $2^{n\log(n)}$. Towards this, Raz [11] embeds m 2×2 matrices (with two variables per matrix kept alive and rest are set to 1) along a generalized diagonal (for some $m = o(n)$) at random and sets those variables in the rest of the $(n-2m)\times(n-2m)$ matrix that correspond to the same generalized diagonal to 1, and all the other variables to 0. The determinant of such an embedding would "eventually" be of the form $\prod_{i=1}^{m}(y_i - z_i)$. The rank attained by the matrix of the restricted determinant would be 2^m.

On the formula side, Raz shows that each factor under this random restriction attains an imbalance with a high probability. Further he shows that each summand has rank deficiency with a probability that is inverse quasi-polynomial. Putting all of these together with careful probabilistic arguments, Raz [11] concludes that a multilinear formula of size smaller than $n^{O(\log n)}$ cannot compute determinant (and permanent).

Unlike Raz [13], our polynomial of interest is not explicit but is a determinant of a probabilistically constructed sparse matrix X_G. We construct X_G through rectangle sampling: we independently sample each of the $N = \binom{n}{2}^2$ rectangles (that is, 2×2 sub-matrices) in the complete $n\times n$ generic matrix with probability $p = \frac{\log^6 n}{n^3}$ (equivalently sampling $K_{2,2}$'s from a complete bipartite graph $K_{n,n}$), and retain an entry (edge) if it appears in at least one sampled rectangle. This yields a sparse matrix X_G and the corresponding sparse graph G with $\Theta(n\log^6 n)$ edges where each vertex has degree $\Theta(\log^6 n)$ which is a union of sampled rectangles $\mathcal{R}$.

[1] The current presentation of formulas through their *log-product decomposition* is due to [16]. Though this form of presentation does not appear in [11], it is just a rephrasing of Raz's ideas.

A crucial property we seek from G is that under a random permutation $\sigma = (\sigma_L, \sigma_R)$ of vertices, for any large set S of adversarially chosen edges (with $|S| \geq |E_G|^{0.75}$), a constant fraction of rectangles exhibit *diagonal imbalance* relative to S – meaning exactly one of the two opposite diagonal corners belongs to S. This geometric property, which we prove via McDiarmid's inequality with bounded differences, creates the asymmetry that drives imbalance and thus rank deficiency of a factor in our lower bound proof.

Another crucial property that we need is the non-zeroness of $\mathsf{Det}(X_G)$, which we verify via two approaches. One approach is through rigorous analysis of ensuring Hall's Marriage Theorem holds across various size regimes (small, moderate, and near-full sets) to ensure $\mathsf{Det}(X_G) \neq 0$ with high probability. Through this we establish that a sparse matrix exists with the required structure with a very high probability.

We then apply a carefully designed random restriction to both the determinant $\mathsf{Det}(X_G)$ and any purported multilinear formula computing it – we sample $m = \Theta(n^{1/3})$ disjoint[2] rectangles uniformly from the collection $\mathcal{R}$, and for each sampled rectangle, we randomly choose one of two diagonal configurations: top-left and bottom-right corners become either active variables (y_i, z_i) or (z_i, y_i), while the other diagonal is set to all 1s. This creates a partition of variables into sets Y and Z of size m each, with the remaining $|E_G| - 2m$ variables set according to a perfect matching in the residual graph. The key technical result (Theorem 23) shows that for any factor in the formula decomposition with at least $N^{0.75}$ variables, the number of sampled rectangles contributing diagonal imbalance to that factor is quite substantial with high probability. Since each rectangle's configuration is chosen independently, the (Y, Z) partition for that factor follows an approximate binomial distribution with sufficient concentration, making k-balance (where $||Y| - |Z|| < 2k$ for a suitable parameter k) occur with small probability.

The lower bound emerges from analyzing the formula decomposition structure. Any formula of size s decomposes into $s+1$ summands, each being a product of $\Theta(\log n)$ factors, where crucially $\Theta(\log n)$ factors in each product have at least $N^{0.75}$ variables. Conditioned on the sampled rectangles, the balance events across these large factors form a *read-2 family* with respect to the independent configuration choices, since each rectangle's two active variables can affect at most two factors. Applying the read-2 Chernoff bound (Theorem 16) with the small average balance probability, we show that with probability at least $1 - \exp(-\Omega(\log^2 n))$, at least half of the large factors in any single summand exhibit k-imbalance. By Proposition 11, this multiplicative rank deficiency gives each summand rank at most $2^m \cdot 2^{-\Omega(n^{1/32} \log n)}$. Taking a union bound over all $s+1$ summands, if $s = n^{O(\log n)}$, then with probability $1 - o(1)$, the total rank is bounded by $(s+1) \cdot 2^m \cdot 2^{-\Omega(n^{1/32} \log n)} \ll 2^m$. However, the restricted determinant maintains full rank 2^m after the restriction (since the residual graph retains a perfect matching by Theorem 29), yielding a contradiction. Therefore, any multilinear formula computing $\mathsf{Det}(X_G)$ requires size $s = n^{\Omega(\log n)}$.

Our analysis crucially relies on the read-k tail bounds of [7].

1.3 Discussion

In this section, at the risk of being redundant, we discuss key design choices in our construction.

A natural question is: why not simply use the Erdős-Rényi model $G_{n,n,p}$ for a suitably chosen edge probability $p = \Theta(\frac{\log^6 n}{n})$? After all, we showed in Section 3 that such graphs have perfect matchings with high probability. The critical obstacle is the second-round

[2] Sampled rectangles should not share rows or columns, analogous to [13].

restriction. In Raz's framework, we must sample vertex-disjoint rectangles and assign their corners to variables while ensuring the *residual graph* (after removing sampled vertices) retains a perfect matching. Under the $G_{n,n,p}$ model, the event that a sufficient number of vertex-disjoint rectangles exist as induced subgraphs *and* the residual graph maintains a perfect matching has extremely low probability. Rectangle sampling resolves this. By directly sampling rectangles as atomic units with probability $\frac{\log^6 n}{n^3}$, we create a graph G that is the union of sampled rectangles. This design has two crucial advantages. First, rectangles are guaranteed to exist: the sampled set $\mathcal{R}$ of $\Theta(n \log^6 n)$ rectangles is available for second-round sampling by construction – we simply sample $m = \Theta(n^{1/3})$ rectangles uniformly from $\mathcal{R}$. Second, we have distribution invariance under deletion (Theorem 29): when we remove vertices corresponding to sampled rectangles, the residual graph has a similar distribution as rectangle sampling on the smaller $(n - 2m) \times (n - 2m)$ complete bipartite graph, implying the residual graph has a perfect matching with the same high probability as the original construction. Our approach is directly guided by Raz's method [13], which embeds m disjoint 2×2 block-diagonal matrices along a generalized diagonal of the dense $n \times n$ matrix. We achieve the same structural property – vertex-disjoint rectangles available for restriction – while working in a sparse regime, and the rectangle sampling framework is precisely what enables this.

Another natural approach would be to use structured sparse matrices such as band matrices, block-diagonal matrices, or matrices with support on expander graphs. These constructions have the advantage of being fully explicit and deterministic. However, structured constructions face a fundamental challenge: they provide an adversary with exploitable structure. Multilinear formulas decompose polynomials as $f = \sum_{i=1}^{s+1} \prod_{j=1}^{t} f_{ij}$, where the variable sets X_{ij} are determined by the formula's structure. An adversarial formula could potentially align factors with structured sparsity patterns, ensuring that each factor intersects few edges and thereby avoiding diagonal imbalance, or exploit symmetries in deterministic constructions to achieve balance across many factors simultaneously.

Randomness provides robustness against adversarial decompositions. Our random construction via rectangle sampling ensures that for any large variable set $S \subseteq E_G$ with $|S| \geq N^{0.75}$ (regardless of how the formula chooses S), under a random permutation σ, a constant fraction of rectangles exhibit diagonal imbalance relative to S (Theorem 21). The random permutation $\sigma = (\sigma_L, \sigma_R)$ of vertices is essential for creating diagonal imbalance in the sparse regime. Without it, corner positions (top-left, bottom-right, etc.) would be fixed, allowing an adversarial variable set S to align with the predetermined rectangle structure. Since these sparse matrices have at least 1 but at most $O(\log^6 n)$ rectangles per edge (versus $(n-1)^2$ in the dense case), such alignment could eliminate diagonal imbalance entirely. Random permutation ensures every variable has equal probability of occupying each corner position, symmetrizing the structure so that for any large set S, approximately half the rectangles containing S-edges exhibit diagonal imbalance (Theorem 21), regardless of how the formula chooses S.

The choice of sampling probability $p = \frac{\log^6 n}{n^3}$ yields edge density $\Theta(\frac{\log^6 n}{n})$ and degree $\Theta(\log^6 n)$ per vertex. The exponent 6 is not fundamental – it can be improved. Here any $\omega(\log n)$ parameter would suffice (and still not necessary) for the construction, with appropriate adjustments to the concentration bounds. The exponent 6 provides comfortable margins in our probabilistic arguments and most importantly make way for a cleaner exposition.

We also note that sparse determinants inherit VBP-hardness from general determinants through a natural reduction. Given any $n \times n$ determinant, the Mahajan-Vinay characterization [8] expresses it as the entry of an iterated matrix product, yielding an ABP of polynomial

size s and width w. To create a sparse determinant encoding this ABP, we stagger the computation: replace each high-degree node in the ABP with a binary tree of intermediate nodes, ensuring the resulting graph has maximum degree $O(1)$. This transformation produces a sparse matrix where each row and column has constantly many nonzero entries, with size increasing from s to $(s \cdot w)^{O(1)}$ (polynomial) and width from w to $O(w \log s)$ (logarithmic increase). Since this reduction is a polynomial-size projection, sparse determinants are VBP-hard under polynomial projections. Consequently, if sparse determinants admitted general formulas of size $n^{o(\log n)}$, we would have VF = VBP, contradicting the widely believed hierarchy VF $\subsetneq$ VBP $\subsetneq$ VP.

2 Preliminaries

Throughout this paper, we work with an $n \times n$ symbolic matrix X where entries $x_{i,j}$ are formal variables. We identify this matrix with a complete bipartite graph $K_{n,n}$ where the left partition represents rows and the right partition represents columns.

▶ **Definition 3** (Rectangle). *A rectangle is a 2×2 submatrix of X. Given rows i, i' and columns j, j' with $i < i'$ and $j < j'$, the rectangle R consists of four entries: $(i,j),(i,j'),(i',j),(i',j')$. The complete bipartite graph $K_{n,n}$ contains exactly $N = \binom{n}{2}^2$ distinct rectangles.*

▶ **Definition 4** (Corner Positions). *For a rectangle defined by rows $\{i,i'\}$ (with $i < i'$) and columns $\{j,j'\}$ (with $j < j'$), we define four corner positions. The top-left position is (i,j). The top-right position is (i,j'). The bottom-left position is (i',j). The bottom-right position is (i',j'). These positions depend on an ordering $\sigma = (\sigma_L, \sigma_R)$ of the left and right vertices respectively.*

2.1 Edmonds' Criterion and Determinant Non-vanishing

▶ **Definition 5** (Edmonds' Graph). *The Edmonds' graph of an $n\times n$ symbolic matrix $A = \{x_{i,j}\}$ is the bipartite graph $G_A = (V_1 \sqcup V_2, E)$, where $V_1 = \{u_1, \ldots, u_n\}$, $V_2 = \{v_1, \ldots, v_n\}$, such that $(u_i, v_j) \in E$ if and only if $x_{i,j} \neq 0$.*

▶ **Lemma 6** (Edmonds' Criterion). *Let A be an $n \times n$ symbolic matrix, and let G_A be its Edmonds' graph. Then, $\mathsf{Det}(A) \neq 0$ if and only if G_A has a perfect matching.*

2.2 Algebraic complexity classes

Arithmetic circuits are a standard model of computation in the algebraic complexity theory. An arithmetic circuit is a directed acyclic graph where internal nodes are labeled with arithmetic operations ($+$ or $\times$) and leaves are labeled with variables or field constants, computing a polynomial in the natural way. The size of a circuit is the number of edges, and the fundamental question in algebraic complexity theory is to understand which polynomials require large circuits to compute.

Let $(f_n)_{n\in\mathbb{N}}$ be a polynomial family where f_n has n variables and degree poly(n). We say $(f_n) \in \mathsf{VF}$ if each f_n is computable by a polynomial-size arithmetic formula (tree-shaped circuit with fan-out one at each gate); $(f_n) \in \mathsf{VP}$ if each f_n is computable by a polynomial-size arithmetic circuit of polynomial degree; and $(f_n) \in \mathsf{VNP}$ if $f_n(x) = \sum_{e\in\{0,1\}^m} g_n(x,e)$ for some $(g_n) \in \mathsf{VP}$ with $m = \text{poly}(n)$ variables. The known containments $\mathsf{VF} \subseteq \mathsf{VP} \subseteq \mathsf{VNP}$ parallel the Boolean hierarchy, with both believed to be strict.

Algebraic branching programs (ABPs), also known as algebraic branching programs, compute polynomials through iterated matrix multiplication along paths from source to sink. It is is known that VBP lies between formulas and general circuits: $\mathsf{VF} \subseteq \mathsf{VBP} \subseteq \mathsf{VP}$, where both containments are believed to be strict.

2.3 Raz's Partial Derivatives Measure

▶ **Definition 7** (Variable Partition and Coefficient Matrix). *Let f be a multilinear polynomial over variables X. A partition $\sigma : X \mapsto Y \sqcup Z$ divides X into two disjoint sets with $|Y| = |Z| = |X|/2$ (assuming $|X|$ is even). Let $\mathcal{M}_Y$ denote the set of all multilinear monomials over $\sigma^{-1}(Y)$, and $\mathcal{M}_Z$ the set of all multilinear monomials over $\sigma^{-1}(Z)$. The coefficient matrix M_f^σ has rows indexed by $\mathcal{M}_Y$ and columns indexed by $\mathcal{M}_Z$, where the entry (m_Y, m_Z) equals the coefficient of the monomial $m_Y \cdot m_Z$ in the expansion of f. We define the complexity measure $\mu_\sigma(f) = rank(M_f^\sigma)$.*

▶ **Observation 8.** *For any partition σ of a multilinear polynomial f over $|X|$ variables, we have $\mu_\sigma(f) \leq 2^{|X|/2}$, with equality when f is full-rank under σ.*

▶ **Proposition 9** (Multiplicativity Under Products). *If $f = f_1 \times f_2$ where f_1 and f_2 are multilinear polynomials defined over disjoint variable sets X_1 and X_2 with $X = X_1 \sqcup X_2$, then $M_f^\sigma = M_{f_1}^{\sigma|_{X_1}} \otimes M_{f_2}^{\sigma|_{X_2}}$ and consequently $\mu_\sigma(f) = \mu_{\sigma|_{X_1}}(f_1) \cdot \mu_{\sigma|_{X_2}}(f_2)$.*

2.4 Imbalance and Rank Deficiency

▶ **Definition 10** (k-Imbalance and Balance). *For a multilinear polynomial f over variable set X and partition $\sigma : X \mapsto Y \sqcup Z$, we say that f exhibits k-imbalance under σ if $||Y| - |Z|| \geq 2k$, and that f is k-balanced under σ if $||Y| - |Z|| < 2k$.*

▶ **Proposition 11** (Rank Loss from Imbalance). *If f is a multilinear polynomial over $|X|$ variables exhibiting k-imbalance under partition σ, then $\mu_\sigma(f) \leq 2^{(|X|-2k)/2} = 2^{|X|/2} \cdot 2^{-k}$. If $f = \prod_{i=1}^{t} f_i$ where each f_i is defined over disjoint variable set X_i, and at least ℓ factors exhibit k-imbalance, then $\mu_\sigma(f) \leq 2^{(|X|-2\ell k)/2} = 2^{|X|/2} \cdot 2^{-\ell k}$.*

Proof. For a single factor with k-imbalance, we have $||Y| - |Z|| \geq 2k$. Without loss of generality, assume $|Y| \leq |Z|$, so $|Y| = (|X| - \Delta)/2$ and $|Z| = (|X| + \Delta)/2$ where $\Delta \geq 2k$. The coefficient matrix has dimension $2^{|Y|} \times 2^{|Z|}$, so rank is at most $\min(2^{|Y|}, 2^{|Z|}) = 2^{|Y|} \leq 2^{(|X|-2k)/2}$. For products, by Proposition 9, each imbalanced factor contributes a multiplicative factor of at most 2^{-k} to the rank. ◀

By Proposition 9, the product structure over disjoint variable sets ensures that the rank measure μ_σ behaves multiplicatively: $\mu_\sigma(f_1 \times f_2) = \mu_\sigma(f_1) \cdot \mu_\sigma(f_2)$. Under a random partition Γ, this multiplicativity is decisive: each factor exhibiting k-imbalance contributes a multiplicative factor of 2^{-k} to the rank (Proposition 11), and with ℓ such factors, the rank suffers an exponential deficiency of $2^{-\ell k}$. Our approach therefore requires that multilinear formulas admit decompositions exposing product structure over disjoint variable sets – a structural property already established in the literature [16, 15].

▶ **Theorem 12** (Formula Decomposition [16, 15]). *Every multilinear formula F of size s computing a multilinear polynomial f over N variables admits a decomposition $f = \sum_{i=1}^{s+1} f_i$ where $f_i = \prod_{j=1}^{t} f_{ij}$ with $t = \Theta(\log N)$ and for each level $j \in [t]$, the variable set X_{ij} satisfies $\left(\frac{1}{3}\right)^j N \leq |X_{ij}| \leq \left(\frac{2}{3}\right)^j N$.*

2.5 Concentration Inequalities

We would need the following concentration inequalities in our work.

▶ **Theorem 13** (Chernoff Bound [10]). *Let $X_1, \ldots, X_n$ be independent random variables with $X_i \in \{0,1\}$ and $\Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X] = \sum_{i=1}^{n} p_i$. Then for any $\delta > 0$:*

$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{2+\delta}\right) \leq \exp\left(-\frac{\delta^2\mu}{3}\right) \quad \text{for } 0 < \delta \leq 1,$$

$$\Pr[X \leq (1-\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{2}\right).$$

▶ **Theorem 14** (McDiarmid's Inequality [9]). *Let $X_1, \ldots, X_n$ be independent random variables and let $f : \mathcal{X}^n \to \mathbb{R}$ satisfy the bounded differences property: there exist constants $c_1, \ldots, c_n$ such that for all $i \in [n]$ and all $x_1, \ldots, x_n, x_i' \in \mathcal{X}$,*

$$|f(x_1, \ldots, x_i, \ldots, x_n) - f(x_1, \ldots, x_i', \ldots, x_n)| \leq c_i.$$

Then for any $t > 0$,

$$\Pr\left[|f(X_1, \ldots, X_n) - \mathbb{E}[f(X_1, \ldots, X_n)]| \geq t\right] \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^{n} c_i^2}\right).$$

▶ **Definition 15** (Read-k Family of Events). *Let $Y_1, \ldots, Y_r$ be random events determined by functions of independent random variables $X_1, \ldots, X_m$. We say that the collection forms a read-k family if each X_j influences at most k of the events Y_i.*

▶ **Theorem 16** ([7]). *Let $Y_1, \ldots, Y_r$ be a family of read-k indicator variables with $\Pr[Y_i = 1] = p_i$, and let p be the average of $p_1, \ldots, p_r$. Then for any $\varepsilon > 0$,*

$$\mathbb{P}\left[Y_1 + \ldots + Y_r \geq (p+\varepsilon)r\right] \leq e^{-D(p+\varepsilon \,\|\, p)\cdot r/k}$$

and

$$\mathbb{P}\left[Y_1 + \ldots + Y_r \leq (p-\varepsilon)r\right] \leq e^{-D(p-\varepsilon \,\|\, p)\cdot r/k}.$$

Here, $D(q \,\|\, p)$ is the *Kullback-Leibler divergence* defined as

$$D(q \,\|\, p) = q\log\left(\frac{q}{p}\right) + (1-q)\log\left(\frac{1-q}{1-p}\right),$$

where $D(q \,\|\, p) \geq 2(q-p)^2$.

3 Sparse Matrix Construction via Rectangle Sampling

Let X be a generic $n \times n$ symbolic matrix corresponding to the complete bipartite graph $K_{n,n}$. Through controlled deletion of edges, we generate a sparse bipartite graph G on n vertices in each partition. We define our polynomial of interest as $\mathsf{Det}(X_G)$, the determinant of the matrix obtained by setting $x_{i,j} = 0$ for all edges $(i,j) \notin E_G$ and leaving $x_{i,j}$ as a formal variable for all $(i,j) \in E_G$. This polynomial possesses the following key properties: it is defined over $|E_G| = \Theta(n\log^6 n)$ variables, it is nonzero due to the existence of perfect matchings in G, and the combinatorial structure of G inherited from rectangle sampling creates the necessary asymmetry and thus the hardness for our lower bound technique.

3.1 The Sampling Process

▶ **Definition 17** (Rectangle Sampling). *We sample a set of rectangles $\mathcal{R}$ by including each of the $N = \binom{n}{2}^2$ rectangles independently with probability $p = \frac{\log^6 n}{n^3}$. An edge (i, j) survives in the graph G if and only if it appears in at least one sampled rectangle.*

The expected number of rectangles is given by linearity of expectation as $\mathbb{E}[|\mathcal{R}|] = N \cdot p = \binom{n}{2}^2 \cdot \frac{\log^6 n}{n^3} \approx n \log^6 n$. Since rectangles are sampled independently, standard Chernoff bounds yield the following result.

▶ **Proposition 18** (Concentration of $|\mathcal{R}|$). *With probability at least $1 - \exp(-\Omega(n \log^6 n))$, we have $|\mathcal{R}| = \Theta(n \log^6 n)$.*

3.2 Edge Survival Probabilities

Each edge (i, j) appears in exactly $(n-1)^2$ rectangles (by choosing one additional row and one additional column in $(n-1)$ ways each). The probability that edge (i, j) survives is given by the following calculation. We have

$$\mathbb{P}\left[(i,j) \text{ survives}\right] = 1 - (1-p)^{(n-1)^2} \approx 1 - e^{-(n-1)^2 p} \approx (n-1)^2 p \approx \frac{\log^6 n}{n}.$$

The expected number of edges follows by linearity of expectation as $\mathbb{E}[|E_G|] \approx n^2 \cdot \frac{\log^6 n}{n} = n \log^6 n$.

▶ **Proposition 19** (Concentration of $|E_G|$ via Read-4 Chernoff). *With probability at least $1 - \exp(-\Omega(n \log^6 n))$, we have $|E_G| = \Theta(n \log^6 n)$. Similarly, each vertex degree is $\Theta(\log^6 n)$ with probability at least $1 - \exp(-\Omega(\log^6 n))$.*

Proof. Let $Y_1, \ldots, Y_{n^2}$ be indicator random variables for edges and $X_1, \ldots, X_N$ be indicator random variables for rectangles. Each edge appears in exactly four corner positions across all rectangles it belongs to, which forms a read-4 structure. The edge indicator Y_e depends on the rectangle indicators for rectangles containing e, and each rectangle indicator X_R influences exactly four edge indicators corresponding to the four corners of R. Applying the read-4 Chernoff bound with the expected number of edges $\mathbb{E}[|E_G|] = \Theta(n \log^6 n)$ yields the stated concentration. The vertex degree bound follows similarly by considering the read-2 structure restricted to edges incident to a fixed vertex. ◀

Let N_e be the number of rectangles an edge $e = (i, j) \in [n] \times [n]$ belongs to after the sampling process. Observe then that $\mathbb{E}[N_e] = (n-1)^2 \cdot p \approx \frac{\log^6 n}{n}$ and hence $N_e = \Theta\left(\frac{\log^6 n}{n}\right)$ with a high probability. However, note that $N_e \geq 1$ when edge e survives.

3.3 Non-zero Determinant via Hall's Marriage Theorem

Given a random permutation $\sigma = (\sigma_L, \sigma_R)$ of the left and right vertices, rectangles acquire oriented corners.

▶ **Definition 20** (Diagonal Rectangle Sets). *For a set $S \subseteq E_G$, and a random permutation $\sigma = (\sigma_L, \sigma_R)$ of the left and right vertices, let $\mathcal{R}_{S,diag}(\sigma)$ denote the set of rectangles in $\mathcal{R}$ where exactly one of the two opposite corners (TL and BR, or TR and BL) belongs to S under the ordering σ.*

▶ **Theorem 21** (Concentration of Diagonal Rectangles). *Let $S \subseteq E_G$ with $|S| \geq |E_G|^{0.75}$. Then with probability at least $1 - 2\exp(-\Omega(n^{0.5}\log^{-3} n))$ over the random permutation $\sigma = (\sigma_L, \sigma_R)$, we have $|\mathcal{R}_{S,diag}(\sigma)| \geq (1-\delta)\cdot\frac{|S|}{2}$ for any constant $\delta > 0$.*

Proof. For each rectangle $R \in \mathcal{R}$, define $W_R(\sigma) = \mathbf{1}[\text{exactly one of TL, BR of } R \text{ is in } S]$. Then $|\mathcal{R}_{S,\text{diag}}(\sigma)| = \sum_{R\in\mathcal{R}} W_R(\sigma)$.

For computing the expectation, consider a fixed rectangle R defined by rows $\{i, i'\}$ and columns $\{j, j'\}$, with four edges $e_1 = (i,j)$, $e_2 = (i,j')$, $e_3 = (i',j)$, $e_4 = (i',j')$. The permutation σ randomly determines which pair becomes TL and BR versus TR and BL, with each pairing having probability one half. Therefore, $\mathbb{E}[W_R(\sigma)] = \frac{1}{2}[\mathbf{1}[e_1 \in S \oplus e_4 \in S] + \mathbf{1}[e_2 \in S \oplus e_3 \in S]]$, where $\oplus$ denotes XOR.

Summing over all rectangles and rearranging by edges gives

$$\begin{aligned}\mathbb{E}[|\mathcal{R}_{S,\text{diag}}(\sigma)|] &= \frac{1}{2}\sum_{R\in\mathcal{R}}[\mathbf{1}[e_1 \in S \oplus e_4 \in S] + \mathbf{1}[e_2 \in S \oplus e_3 \in S]] \\ &= \frac{1}{2}\sum_{e\in E_G}\sum_{\substack{R\in\mathcal{R}\\ e\in R}} \mathbf{1}[e \in S \oplus \text{opposite}(e,R) \in S].\end{aligned}$$

For any subset S of edges, let $N_e^{(S)}$ count the number of rectangles in which opposite corner of e is in S. For $e \in S$, this counts rectangles where the opposite corner is not in S, contributing $N_e - N_e^{(S)}$. For $e \notin S$, this counts rectangles where the opposite corner is in S, contributing $N_e^{(\overline{S})}$. Since $\sum_{e\notin S} N_e^{(\overline{S})} = \sum_{e\in S} N_e^{(S)}$, we have

$$\mathbb{E}[|\mathcal{R}_{S,\text{diag}}(\sigma)|] = \frac{1}{2}\left[\sum_{e\in S}(N_e - N_e^{(S)}) + \sum_{e\in S} N_e^{(S)}\right] = \frac{1}{2}\sum_{e\in S} N_e \geq \frac{|S|}{2},$$

using the fact that $N_e \geq 1$ for all surviving edges $e \in S$.

For concentration via McDiarmid's inequality, changing a single coordinate of σ affects at most $c = O(\log^6 n)$ rectangles (bounded by the maximum vertex degree). By McDiarmid's inequality applied to $2n$ coordinates, we have

$$\Pr\left[\left||\mathcal{R}_{S,\text{diag}}(\sigma)| - \mathbb{E}[|\mathcal{R}_{S,\text{diag}}(\sigma)|]\right| > t\right] \leq 2\exp\left(-\frac{2t^2}{2nc^2}\right).$$

Setting $t = \delta\cdot\mathbb{E}[|\mathcal{R}_{S,\text{diag}}(\sigma)|] = \Theta(|S|) = \Theta(n^{0.75}\log^{4.5} n)$ yields

$$\Pr[\text{deviation}] \leq 2\exp\left(-\Omega\left(\frac{n^{1.5}\log^9 n}{n\log^{12} n}\right)\right) = 2\exp(-\Omega(n^{0.5}\log^{-3} n)).$$ ◀

3.4 Sampling Rectangles for Round 2

The preceding theorem establishes concentration for $|\mathcal{R}_{S,\text{diag}}(\sigma)|$ under a fixed permutation σ. We now analyze what happens when we sample m rectangles uniformly at random from $\mathcal{R}$ in the second round. This is the key connection between our construction and the factor imbalance needed for the lower bound.

▶ **Definition 22** (Sampled Diagonal Rectangles). *Let $\mathbf{R} = (R_1, \ldots, R_m)$ denote m rectangles sampled uniformly and independently from $\mathcal{R}$ with replacement. For a set $S \subseteq E_G$ and permutation σ, define $Y_S(\sigma, \mathbf{R}) = |\{i \in [m] : R_i \in \mathcal{R}_{S,diag}(\sigma)\}|$ as the number of sampled rectangles that belong to $\mathcal{R}_{S,diag}(\sigma)$.*

▶ **Theorem 23** (Concentration of Sampled Diagonal Rectangles). *Let $S \subseteq E_G$ with $|S| \geq |E_G|^{0.75}$. Sample $m = \Theta(n^{1/3})$ rectangles uniformly and independently from $\mathcal{R}$ with replacement, and let σ be a uniformly random permutation. Then with probability at least $1 - \exp(-\Omega(n^{1/12}\log^{-1.5} n))$ over both σ and $\mathbf{R}$, we have $Y_S(\sigma, \mathbf{R}) \geq (1-\delta') \cdot \frac{m|S|}{2|\mathcal{R}|}$ for any constant $\delta' > 0$.*

Proof. We condition on σ and analyze the distribution of Y_S given σ. For the conditional distribution, given a fixed permutation σ, the rectangles $R_1, \ldots, R_m$ are sampled independently and uniformly from $\mathcal{R}$. Therefore, $Y_S \mid \sigma = \sum_{i=1}^{m} \mathbf{1}[R_i \in \mathcal{R}_{S,\mathrm{diag}}(\sigma)]$ is a sum of m independent Bernoulli random variables, each with success probability $\frac{|\mathcal{R}_{S,\mathrm{diag}}(\sigma)|}{|\mathcal{R}|}$. Thus, $\mathbb{E}[Y_S \mid \sigma] = m \cdot \frac{|\mathcal{R}_{S,\mathrm{diag}}(\sigma)|}{|\mathcal{R}|}$. By the Chernoff bound, for any $\delta_1 > 0$, we have

$$\Pr\left[Y_S < (1-\delta_1)\cdot \mathbb{E}[Y_S \mid \sigma] \;\middle|\; \sigma\right] \leq \exp\left(-\frac{\delta_1^2 \mathbb{E}[Y_S \mid \sigma]}{2}\right).$$

For the expectation over σ, using the law of total expectation gives $\mathbb{E}_{\sigma,\mathbf{R}}[Y_S] = \mathbb{E}_\sigma[\mathbb{E}_{\mathbf{R}}[Y_S|\sigma]] = m \cdot \frac{\mathbb{E}_\sigma[|\mathcal{R}_{S,\mathrm{diag}}(\sigma)|]}{|\mathcal{R}|} \geq \frac{m|S|}{2|\mathcal{R}|}$, where the last inequality follows from Theorem 21.

For the two-stage concentration argument, from Theorem 21, for any constant $\delta_2 > 0$, with probability at least $1 - 2\exp(-\Omega(n^{0.5}\log^{-3} n))$ over σ, we have $|\mathcal{R}_{S,\mathrm{diag}}(\sigma)| \geq (1-\delta_2)\cdot\frac{|S|}{2}$.

Let $\mathcal{E}_1$ denote the event that $\mathbb{E}[Y_S|\sigma] \geq (1-\delta_2)\cdot\mathbb{E}[Y_S]$, and let $\mathcal{E}_2$ denote the event that $Y_S \geq (1-\delta_1)\cdot\mathbb{E}[Y_S|\sigma]$. We want to show that $Y_S \geq (1-\delta')\cdot\mathbb{E}[Y_S]$ with high probability, which follows from $\mathcal{E}_1 \cap \mathcal{E}_2$ by choosing $\delta' = \delta_1 + \delta_2 - \delta_1\delta_2$.

We decompose the failure probability as $\Pr[(\mathcal{E}_1 \cap \mathcal{E}_2)^c] \leq \Pr[\mathcal{E}_1^c] + \Pr[\mathcal{E}_2^c]$. From Theorem 21, we have $\Pr[\mathcal{E}_1^c] \leq 2\exp(-\Omega(n^{0.5}\log^{-3} n))$. To bound $\Pr[\mathcal{E}_2^c]$, we further decompose as $\Pr[\mathcal{E}_2^c] \leq \Pr[\mathcal{E}_2^c|\mathcal{E}_1] + \Pr[\mathcal{E}_1^c]$.

For bounding $\Pr[\mathcal{E}_2^c|\mathcal{E}_1]$, conditioning on $\mathcal{E}_1$ means we condition on $\sigma \in \mathcal{E}_1$. For any such σ, we have $\mathbb{E}[Y_S|\sigma] \geq (1-\delta_2)\cdot\frac{m|S|}{2|\mathcal{R}|}$. With $|S| \geq |E_G|^{0.75} = \Theta(n^{0.75}\log^{4.5} n)$, $|\mathcal{R}| = \Theta(n\log^6 n)$, and choosing $m = \Theta(n^{1/3})$, we have

$$\mathbb{E}[Y_S|\sigma] \geq \Theta\left(\frac{n^{1/3}\cdot n^{0.75}\log^{4.5} n}{n\log^6 n}\right) = \Theta\left(\frac{n^{13/12}\log^{4.5} n}{n\log^6 n}\right) = \Theta(n^{1/12}\log^{-1.5} n).$$

For large n where this expectation is $\Omega(n^{1/12})$, by the Chernoff bound we have $\Pr[\mathcal{E}_2^c|\sigma] \leq \exp(-\Omega(n^{1/12}\log^{-1.5} n))$. Since this bound holds uniformly for all $\sigma \in \mathcal{E}_1$, we obtain $\Pr[\mathcal{E}_2^c|\mathcal{E}_1] \leq \exp(-\Omega(n^{1/12}\log^{-1.5} n))$.

For the final bound, we have

$$\begin{aligned}\Pr[(\mathcal{E}_1 \cap \mathcal{E}_2)^c] &\leq \Pr[\mathcal{E}_1^c] + Pr[\mathcal{E}_1^c] + \Pr[\mathcal{E}_2^c|\mathcal{E}_1] \\ &\leq 2\exp(-\Omega(n^{0.5}\log^{-3} n)) + \exp(-\Omega(n^{1/12}\log^{-1.5} n)) \\ &= \exp(-\Omega(n^{1/12}\log^{-1.5} n)),\end{aligned}$$

where the dominant term is the last exponential. ◀

3.5 Verification of Hall's Marriage Condition

To rigorously establish that our randomly sampled sparse graphs have perfect matchings with high probability, we verify Hall's marriage condition. For every subset $S \subseteq V_1$, the neighborhood $N(S) = \{v \in V_2 : \exists u \in S, (u,v) \in E_G\}$ must satisfy $|N(S)| \geq |S|$. We partition the analysis by set size into three regimes: small sets ($|S| \leq \frac{n}{\log n}$), moderate sets ($\frac{n}{\log n} < |S| \leq n - \frac{n}{\log n}$), and near-full sets ($n - \frac{n}{\log n} < |S| \leq n$).

3.5.1 Quantifying Neighborhood Coverage

For a fixed set $S \subseteq V_1$ of size k and a set $T \subseteq V_2$ of size ℓ, let $M_{T,S}$ denote the number of rectangles incident on at least one row in S and at least one column in T. Let $C_{i,T}$ be the set of rectangles incident on row i with at least one column in T. Then we have $|C_{i,T}| = (n-1)\left[\binom{n}{2} - \binom{n-\ell}{2}\right]$ and $|C_{i,T} \cap C_{j,T}| = \binom{n}{2} - \binom{n-\ell}{2}$. By inclusion-exclusion, we obtain

$$\begin{aligned} M_{T,S} = \left|\bigcup_{i\in S} C_{i,T}\right| &\geq \sum_{i\in S} |C_{i,T}| - \sum_{\substack{i,j\in S \\ i\neq j}} |C_{i,T} \cap C_{j,T}| \\ &= k(n-1)\left[\binom{n}{2} - \binom{n-\ell}{2}\right] - \binom{k}{2}\left[\binom{n}{2} - \binom{n-\ell}{2}\right] \\ &= \underbrace{k\left[\binom{n}{2} - \binom{n-\ell}{2}\right]\left(n - \frac{k+1}{2}\right)}_{\alpha(n,k,\ell)}. \end{aligned}$$

The probability that $T \cap N(S) = \emptyset$ (no column from T neighbors S) is at most $\Pr[T \cap N(S) = \emptyset] \leq (1-p)^{M_{T,S}} \leq \exp(-p \cdot \alpha(n,k,\ell))$. Hall's condition fails for set S if and only if there exists T with $|T| \geq n-k+1$ such that $T \cap N(S) = \emptyset$. Therefore, $\Pr[|N(S)| < k] \leq \sum_{\ell \geq n-k+1} \binom{n}{\ell} \exp(-p \cdot \alpha(n,k,\ell))$.

3.5.2 Small Sets: $|S| \leq \frac{n}{\log n}$

▶ **Theorem 24** (Hall's Condition for Small Sets). *With probability at least* $1 - \exp(-\Omega(\log^6 n))$, *Hall's condition holds for all sets* S *with* $1 \leq |S| \leq \frac{n}{\log n}$.

Proof. Simplifying the binomial terms gives $\binom{n}{2} - \binom{n-\ell}{2} = \frac{n(n-1)-(n-\ell)(n-\ell-1)}{2} = \frac{\ell(2n-\ell-1)}{2}$. For $\ell \geq n-k+1$, the expression $\ell(2n-\ell-1)$ is minimized at $\ell = n-k+1$, giving $\alpha(n,k,\ell) \geq k \cdot \frac{(n-k+1)(n+k-2)}{2} \cdot \left(n - \frac{k+1}{2}\right)$.

For $k \leq \frac{n}{\log n}$, we have $\alpha(n,k,\ell) \geq k \cdot \left(n - \frac{n}{\log n}\right) \cdot n \cdot \left(n - \frac{n}{2\log n}\right) \geq \frac{kn^3}{8}$. Therefore, the failure probability satisfies:

$$\begin{aligned} \Pr[\exists S : |S| = k \leq \frac{n}{\log n}, |N(S)| < k] &\leq \sum_{k=1}^{n/\log n} \binom{n}{k} \sum_{\ell \geq n-k+1} \binom{n}{\ell} \exp(-p \cdot \alpha(n,k,\ell)) \\ &\leq \sum_{k=1}^{n/\log n} \binom{n}{k} \cdot k \cdot \binom{n}{n-k} \cdot \exp\left(-\Omega\left(\frac{k\log^6 n \cdot n^3}{n^3}\right)\right) \\ &\leq \sum_{k=1}^{n/\log n} k^3 n^{2k} \cdot \exp(-\Omega(k\log^6 n)) \\ &\leq \sum_{k=1}^{n/\log n} \exp(-\Omega(k\log^6 n)) \leq \exp(-\Omega(\log^6 n)). \end{aligned}$$

◀

3.5.3 Moderate Sets: $\frac{n}{\log n} < |S| \leq n - \frac{n}{\log n}$

For the moderate size regime, we analyze sets with $\frac{n}{\log n} < k \leq n - \frac{n}{\log n}$.

▷ Claim 25. For $\frac{n}{\log n} < k \leq n - \frac{n}{\log n}$ and $\ell \geq n-k+1 > \frac{n}{\log n}$, we have $\alpha(n,k,\ell) \geq \Omega\left(\frac{n^4}{\log^2 n}\right)$.

Proof. From the general lower bound, $\alpha(n,k,\ell) \geq k \cdot \frac{(n-k+1)(n+k-2)}{2} \cdot \left(n - \frac{k+1}{2}\right)$. Minimizing each factor in the relevant regime yields $\alpha(n,k,\ell) \geq \frac{n}{\log n} \cdot \frac{n}{\log n} \cdot n \cdot \frac{n}{2} = \Omega\left(\frac{n^4}{\log^2 n}\right)$. ◁

▶ **Theorem 26** (Hall's Condition for Moderate Sets). *With probability at least* $1 - \exp(-\Omega(n \log^4 n))$*, Hall's condition holds for all sets* S *with* $\frac{n}{\log n} < |S| \leq n - \frac{n}{\log n}$.

Proof. By a union bound and Claim 25, we have

$$\begin{aligned}
&\Pr\left[\exists S : \frac{n}{\log n} < |S| \leq n - \frac{n}{\log n}, |N(S)| < |S|\right] \\
&\leq \sum_{k=n/\log n}^{n-n/\log n} \sum_{\ell \geq n-k+1} \binom{n}{k}\binom{n}{\ell} \exp(-p \cdot \alpha(n,k,\ell)) \\
&\leq 4^n \cdot \mathrm{poly}(n) \cdot \exp\left(-\frac{\log^6 n}{n^3} \cdot \Omega\left(\frac{n^4}{\log^2 n}\right)\right) = 4^n \cdot \mathrm{poly}(n) \cdot \exp(-\Omega(n \log^4 n)) \\
&= \exp(-\Omega(n \log^4 n)).
\end{aligned}$$

◀

3.5.4 Near-Full Sets: $n - \frac{n}{\log n} < |S| \leq n$

▶ **Theorem 27** (Hall's Condition for Near-Full Sets). *With probability at least* $1 - \exp(-\Omega(\log^6 n))$*, Hall's condition holds for all sets* S *with* $n - \frac{n}{\log n} < |S| \leq n$.

Proof. For sets $S \subseteq V_1$ with $|S| = k > n - \frac{n}{\log n}$, consider the complementary set $\overline{S} = V_1 \setminus S$ with $|\overline{S}| = n - k < \frac{n}{\log n}$. By the symmetry of the bipartite graph construction, Hall's condition for S is equivalent to verifying that the complementary vertices have sufficiently large neighborhoods.

Specifically, if $|N(\overline{S})| \geq |\overline{S}|$, then the vertices in $V_2 \setminus N(\overline{S})$ can only be matched to vertices in S. Since $|V_2 \setminus N(\overline{S})| \geq n - |N(\overline{S})| \geq n - n + |\overline{S}| = |\overline{S}|$ implies that when $|N(\overline{S})| < n - |\overline{S}| + 1 = k$, we would have Hall's condition violated for S. However, if Hall's condition holds for $\overline{S}$, meaning $|N(\overline{S})| \geq |\overline{S}|$, then Hall's condition is satisfied for S as well.

The key observation is that the probability analysis for small sets in Theorem 24 applies directly to $\overline{S}$, since $|\overline{S}| < \frac{n}{\log n}$. By that theorem, with probability at least $1 - \exp(-\Omega(\log^6 n))$, all sets of size at most $\frac{n}{\log n}$ satisfy Hall's condition. This includes all complementary sets $\overline{S}$ for $|S| > n - \frac{n}{\log n}$.

Therefore, by symmetry and the analysis of small sets, with probability at least $1 - \exp(-\Omega(\log^6 n))$, Hall's condition holds for all near-full sets S with $n - \frac{n}{\log n} < |S| \leq n$. ◀

Combining Theorem 24, Theorem 26, and Theorem 27 via union bound gives the following:

▶ **Theorem 28** (Perfect Matching Existence). *With probability at least* $1 - \exp(-\Omega(\log^6 n))$*, the randomly sampled sparse graph* G *has a perfect matching, and hence* $\mathsf{Det}(X_G) \neq 0$.

▶ **Theorem 29** (Perfect Matching After Random Vertex Deletion). *Let* $m = \Theta(n^{1/3})$*, and let* S_L *and* S_R *denote the random sets of deleted rows and columns. With probability at least* $1 - \exp(-\Omega(\log^6 n))$ *over the construction of* G *and the choice of deletion of* $2m$ *rows and* $2m$ *columns* $\mathcal{D} = (S_L, S_R)$*, the graph* $G_{rest} = G|_{V_1 \setminus S_L, V_2 \setminus S_R}$ *has a perfect matching, and hence the restricted determinant remains non-zero.*

Proof. We apply the law of total expectation, conditioning on the specific deleted vertices. Let $\mathcal{D} = (S_L, S_R)$ denote a deterministic choice of which rows and columns are deleted. We write $\Pr[G_{\text{rest}} \text{ has perfect matching}] = \mathbb{E}_{\mathcal{D}}[\Pr[G_{\text{rest}} \text{ has perfect matching} \mid \mathcal{D}]]$.

For the inner probability conditioned on a fixed deletion pattern $\mathcal{D} = (S_L, S_R)$, by the analysis above, Hall's marriage condition provides the criterion for perfect matching existence. We established in Theorem 24, Theorem 26, and Theorem 27 that Hall's condition holds for all subsets of V_1 with high probability. Specifically, we showed that with probability at least $1 - \exp(-\Omega(\log^6 n))$ over the sparse matrix construction, every subset $S \subseteq V_1$ satisfies $|N(S)| \geq |S|$ in the graph G.

Given a fixed deletion pattern $\mathcal{D}$ with $|S_L| = |S_R| = 2m$ where $m = \Theta(n^{1/3}) \ll n$, consider the restricted graph $G|_{V_1 \setminus S_L, V_2 \setminus S_R}$. For Hall's condition to hold in this restricted graph, we need to verify that for every subset $T \subseteq V_1 \setminus S_L$, the neighborhood $N_{G_{\text{rest}}}(T)$ in the restricted graph satisfies $|N_{G_{\text{rest}}}(T)| \geq |T|$.

The key observation is that the neighborhood in the restricted graph can only be smaller than in the original graph. That is, $N_{G_{\text{rest}}}(T) = N_G(T) \cap (V_2 \setminus S_R)$, so $|N_{G_{\text{rest}}}(T)| = |N_G(T) \setminus S_R|$. For Hall's condition to fail in the restricted graph, we would need $|N_G(T) \setminus S_R| < |T|$, which means $|N_G(T) \cap S_R| > |N_G(T)| - |T|$.

However, our analysis in Section 3.5 established Hall's condition with substantial margin. Specifically, for small sets with $|T| \leq \frac{n}{\log n}$, Theorem 24 showed that the probability of neighborhood deficiency is exponentially small in $\log^6 n$. For moderate sets, Theorem 26 provided even stronger exponential bounds in $n \log^4 n$. For near-full sets, Theorem 27 established the result via symmetry with small sets.

Since the deletion affects at most $2m = \Theta(n^{1/3})$ vertices from each partition, and our Hall's condition analysis holds with high probability uniformly over the structure of G, the restricted graph maintains sufficient neighborhood expansion to satisfy Hall's condition. The margin provided by our concentration bounds in Section 3.5 accommodates the removal of $o(n)$ vertices without violating the matching condition. To see this explicitly, we can invoke Theorem 28 with sets each of size $n - 2m$ and in the analysis with-in, we avoid the rows and columns specified by $\mathcal{D}$.

For the outer expectation over deletion patterns $\mathcal{D}$, we note that the second-round sampling of rectangles determines $\mathcal{D}$ through the union of vertices in sampled rectangles. By the earlier invoking of Theorem 28, regardless of which specific $2m$ rows and columns are involved in the sampled rectangles, the conditional probability that the remaining graph has a perfect matching remains bounded below by $1 - \exp(-\Omega(\log^6 n))$. This follows because the rectangle sampling described earlier creates a graph with robust expansion properties that persist under any deletion pattern removing $o(n)$ vertices.

Therefore, by the law of total expectation,

$$\begin{aligned}\Pr[G_{\text{rest}} \text{ has perfect matching}] &= \mathbb{E}_{\mathcal{D}}[\Pr[G_{\text{rest}} \text{ has perfect matching} \mid \mathcal{D}]] \\ &\geq 1 - \exp(-\Omega(\log^6 n)),\end{aligned}$$

where the bound holds uniformly over all possible deletion patterns $\mathcal{D}$. ◀

3.6 Sparse matrix

We can now formally define the matrix whose determinant complexity we analyze. Given the bipartite graph $G = (V_1 \sqcup V_2, E_G)$ constructed via rectangle sampling, where $|E_G| = \Theta(n \log^6 n)$, define the $n \times n$ *sparse symbolic matrix* X_G by setting $(X_G)_{i,j} = x_{i,j}$ (a formal variable) if $(i, j) \in E_G$, and $(X_G)_{i,j} = 0$ otherwise.

Our construction ensures X_G satisfies three critical properties:

1. **Sparsity:** $\Theta(\log^6 n)$ nonzero entries per row/column (Proposition 19),
2. **Non-vanishing:** $\mathsf{Det}(X_G) \neq 0$ via perfect matching (Corollary 28),

3. **Diagonal imbalance:** For large variable sets S with $|S| \geq |E_G|^{0.75}$, under random vertex permutations σ, at least $(1-\delta)|S|/2$ rectangles exhibit diagonal imbalance relative to S with high probability (Theorem 21).

Property (3) provides the geometric foundation for our lower bound: the rectangle sampling creates an asymmetric structure that, combined with random permutations, guarantees factor imbalance in the analysis of Section 4. The polynomial $\mathsf{Det}(X_G)$ over $\Theta(n \log^6 n)$ variables is our explicit candidate.

4 Lower Bounds Against Multilinear Formulas

▶ Remark 30. Our sparse determinant polynomial $\mathsf{Det}(X_G)$ is defined over the edge variables in E_G. Therefore, the total number of variables is $N = |E_G| = \Theta(n \log^6 n)$.

▶ **Lemma 31** (Count of Large Factors). *In the decomposition of Theorem 12 with $N = |E_G| = \Theta(n \log^6 n)$ total variables, the number of levels j for which $|X_{ij}| \geq N^{0.75} = \Theta(n^{0.75} \log^{4.5} n)$ is at least $\Theta(\log n)$.*

Proof. We require $(1/3)^j \cdot N \geq N^{0.75}$, which gives $(1/3)^j \geq N^{-0.25}$. Taking logarithms yields $j \log(1/3) \geq -0.25 \log N$, so $j \leq \frac{0.25 \log N}{\log 3}$. With $N = \Theta(n \log^6 n)$, we have $\log N = \Theta(\log n + \log \log n) = \Theta(\log n)$, so $j \leq \Theta(\log n)$. Since the total number of levels is $t = \Theta(\log N) = \Theta(\log n)$, and all levels $j \leq \Theta(\log n)$ satisfy $|X_{ij}| \geq N^{0.75}$, we have $\Theta(\log n)$ large factors per product. ◀

4.1 Sampling Rectangles and Concentration

We begin the second-round analysis by studying what happens when we sample rectangles uniformly at random from $\mathcal{R}$. This analysis connects the first-round construction to the factor imbalance needed for the lower bound.

▶ **Definition 32** (Rectangle Sampling Distributions). *We consider two probability distributions over sets of m rectangles from $\mathcal{R}$:*

- *Distribution μ^*: Sample m rectangles uniformly and independently from $\mathcal{R}$ with replacement. Let $\mathbf{R} = (R_1, \ldots, R_m)$ denote the sampled sequence.*
- *Distribution μ: Sample m rectangles uniformly from $\mathcal{R}$ such that no two rectangles share a row or a column (i.e., the rectangles are vertex-disjoint). Let $\mathbf{R} = \{R_1, \ldots, R_m\}$ denote the sampled set.*

For a set $S \subseteq E_G$ and permutation σ, define $Y_S(\sigma, \mathbf{R}) = |\{i \in [m] : R_i \in \mathcal{R}_{S,diag}(\sigma)\}|$ as the number of sampled rectangles that belong to $\mathcal{R}_{S,diag}(\sigma)$.

4.1.1 Choice of Distribution

In our lower bound proof[3], we apply the random restriction to both the determinant polynomial $\mathsf{Det}(X_G)$ and any multilinear formula computing it using the same distribution μ (rectangles with no shared rows or columns). The vertex-disjoint constraint in μ ensures that the random restriction creates a clean partition of variables: each variable in E_G belongs to at most one sampled rectangle, and the remaining graph G_{rest} after removing sampled vertices has a perfect matching with high probability (Theorem 29). This property

[3] We also borrow inspiration for this line of argument from [13, Section 6].

is crucial for arguing that $\mathsf{Det}(X_G)$ maintains full rank after restriction. While our analysis uses distribution μ (vertex-disjoint rectangles), many of our concentration bounds are more naturally analyzed under the simpler distribution μ^* (sampling with replacement). Fortunately, the statistical distance between μ and μ^* is $o(1)$ for our parameter regime. With $m = \Theta(n^{1/3})$ rectangles sampled from $|\mathcal{R}| = \Theta(n \log^6 n)$ rectangles, the probability that any two sampled rectangles share a vertex under μ^* is at most $o(1)$ when $m^2 = o(n)$. Therefore, events that hold with probability q under μ^* also hold with probability $q - o(1)$ under μ. This allows us to freely use concentration results established for μ^* in our proofs while applying the restriction under μ.

4.2 Random Restriction and Variable Assignment

We now apply the random restriction, which operates on the variables from X_G. After sampling $m = \Theta(n^{1/3})$ rectangles uniformly at random from $\mathcal{R}$, we define a random restriction that assigns variables to create the $Y \sqcup Z$ partition (and rest of the variables to $\{0, 1\}$), which proceeds as follows:

- **Sampling rectangles:** Let $\mathbf{R} = (R_1, \ldots, R_m)$ denote the $m = \Theta(n^{1/3})$ rectangles sampled uniformly and independently from $\mathcal{R}$ with replacement. Each rectangle R_i involves four vertices: two rows $\{r_1^i, r_2^i\}$ and two columns $\{c_1^i, c_2^i\}$. Let $S_L = \bigcup_{i=1}^m \{r_1^i, r_2^i\}$ and $S_R = \bigcup_{i=1}^m \{c_1^i, c_2^i\}$ denote the sets of rows and columns involved in the sampled rectangles. We have $|S_L|, |S_R| \leq 2m$, with equality when rectangles are disjoint on vertices.
- **Rectangle variable assignment:** For each rectangle R_i defined by rows $\{r_1^i, r_2^i\}$ and columns $\{c_1^i, c_2^i\}$, we independently choose one of two configurations with equal probability. The configurations are as follows:
  - Configuration 1: Set $x_{r_1^i, c_1^i} \to y_i$ and $x_{r_2^i, c_2^i} \to z_i$, while setting $x_{r_1^i, c_2^i} = x_{r_2^i, c_1^i} = 1$ as constants.
  - Configuration 2: Set $x_{r_1^i, c_1^i} \to z_i$ and $x_{r_2^i, c_2^i} \to y_i$, while setting $x_{r_1^i, c_2^i} = x_{r_2^i, c_1^i} = 1$ as constants.

  This creates two diagonal corners as active variables assigned to y_i and z_i, and sets the other two diagonal corners to constants.
- **Remaining variable assignment:** Consider the submatrix obtained by removing the rows in S_L and columns in S_R from X_G. Let G_{rest} denote the induced bipartite graph on the remaining vertices $V_1 \setminus S_L$ and $V_2 \setminus S_R$. We established earlier that with high probability, regardless of which specific rows and columns are deleted, the graph G_{rest} has a perfect matching. Fix an arbitrary perfect matching M_{rest} in G_{rest}. For all remaining variables (those not in the m rectangles), if edge e belongs to M_{rest}, we set the corresponding variable to 1, and if edge e does not belong to M_{rest}, we set the corresponding variable to 0.
- **Final partition:** Let $Y = \{y_1, \ldots, y_m\}$ and $Z = \{z_1, \ldots, z_m\}$ denote the sets of active variables after restriction. The restriction creates a partition of the original $N = |E_G|$ variables into $|Y| = m$ variables in the Y class, $|Z| = m$ variables in the Z class, and $N - 2m$ variables set to constants.

  For a factor f_{ij} in the formula decomposition over variable set X_{ij}, let $Y_{ij} = Y \cap X_{ij}$ and $Z_{ij} = Z \cap X_{ij}$ denote the y and z variables in this factor respectively.

4.3 Balance Analysis via Binomial Distribution

▶ **Lemma 33** (Binomial Balance Distribution). *Consider a factor f_{ij} with $|X_{ij}| = M \geq N^{0.75}$ variables. Under the random restriction with $m = \Theta(n^{1/3})$ rectangles sampled from $\mathcal{R}$ according to distribution μ (or μ^*), the random variable $|Y_{ij}|$ (the number of y-variables in factor f_{ij}) is distributed as Binomial$(Y_S, 1/2)$ where $Y_S = \Omega(n^{1/12}\log^{-1.5} n)$ with probability at least $1 - \exp(-\Omega(n^{1/12}\log^{-1.5} n))$.*

Proof. By Theorem 23, when $|X_{ij}| \geq N^{0.75}$, with probability at least $1 - \exp(-\Omega(n^{1/12}\log^{-1.5} n))$ over the rectangle sampling and permutation σ, the number of sampled rectangles that contribute to X_{ij} with diagonal imbalance is at least $Y_S = \Omega(n^{1/12}\log^{-1.5} n)$.

Each such rectangle R_ℓ with diagonal imbalance relative to X_{ij} has exactly one corner in X_{ij} that will be assigned to a y-variable and one corner that will be assigned to a z-variable. The assignment depends on whether Configuration 1 or Configuration 2 is chosen for rectangle R_ℓ, and this choice is made independently and uniformly at random for each rectangle.

Therefore, conditioned on having Y_S rectangles with diagonal imbalance relative to X_{ij}, the number of these rectangles that contribute a y-variable to X_{ij} (as opposed to a z-variable) is the sum of Y_S independent Bernoulli random variables, each with success probability $1/2$. This gives $|Y_{ij}| \sim \text{Binomial}(Y_S, 1/2)$ with expectation $Y_S/2$ and variance $Y_S/4$. ◀

▶ **Proposition 34** (Single Factor Balance Probability). *For a factor f_{ij} with $|X_{ij}| = M \geq N^{0.75} = \Theta(n^{0.75}\log^{4.5} n)$ variables and $k = n^{1/32}$, we have $\Pr[$factor f_{ij} is k-balanced$] = O(n^{-1/96}\log^{0.75} n)$.*

Proof. Let $\mathcal{G}_{ij}$ denote the event that the number of sampled rectangles with diagonal imbalance relative to X_{ij} is at least $Y_S = \Omega(n^{1/12}\log^{-1.5} n)$. By Theorem 23, we have $\Pr[\mathcal{G}_{ij}] \geq 1 - \exp(-\Omega(n^{1/12}\log^{-1.5} n))$.

Conditioned on $\mathcal{G}_{ij}$, by Lemma 33, $|Y_{ij}|$ is distributed as Binomial$(Y_S, 1/2)$ with expectation $Y_S/2$ and variance $Y_S/4$. The factor is k-balanced if $||Y_{ij}| - |Z_{ij}|| < 2k$, which is equivalent to $||Y_{ij}| - Y_S/2| < k$ since the Y_S rectangles contribute exactly Y_S variables split between Y_{ij} and Z_{ij}.

By the normal approximation to the binomial distribution, we have

$$\Pr[k\text{-balanced} \mid \mathcal{G}_{ij}] = O\left(\frac{k}{\sqrt{Y_S}}\right) = O\left(\frac{n^{1/32}}{\sqrt{n^{1/12}\log^{-1.5} n}}\right) = O(n^{-1/96}\log^{0.75} n).$$

Therefore, by the law of total probability,

$$\begin{aligned}\Pr[k\text{-balanced}] &= \Pr[k\text{-balanced} \mid \mathcal{G}_{ij}] \cdot \Pr[\mathcal{G}_{ij}] + \Pr[k\text{-balanced} \mid \mathcal{G}_{ij}^c] \cdot \Pr[\mathcal{G}_{ij}^c] \\ &\leq O(n^{-1/96}\log^{0.75} n) \cdot 1 + 1 \cdot \exp(-\Omega(n^{1/12}\log^{-1.5} n)) \\ &= O(n^{-1/96}\log^{0.75} n),\end{aligned}$$

where the dominant term is the first summand since $\exp(-\Omega(n^{1/12}\log^{-1.5} n)) = o(n^{-1/96}\log^{0.75} n)$. ◀

4.4 Read-2 Structure and Dependent Factors

The balance events for different factors depend on two sources of randomness: which rectangles are sampled, and how each sampled rectangle is configured. We analyze these dependencies carefully to establish the read-2 structure.

▶ **Lemma 35** (Read-2 Structure of Balance Events). *Conditioned on the set of sampled rectangles* $\mathbf{R} = (R_1, \ldots, R_m)$*, the balance events for factors in a formula decomposition form a read-2 family with respect to the configuration choices.*

Proof. Given a fixed set of sampled rectangles $\mathbf{R} = (R_1, \ldots, R_m)$, the remaining randomness consists of m independent random variables $C_1, \ldots, C_m$, where $C_i \in \{1, 2\}$ indicates whether rectangle R_i uses Configuration 1 or Configuration 2. Each configuration choice C_i is made independently with equal probability.

The configuration choice C_i for rectangle R_i determines which two of the four edges in R_i become active variables y_i and z_i, and consequently determines how these two variables are assigned to the Y and Z classes. In the multilinear formula decomposition, within a single product (summand), the variable sets of different factors are disjoint. Therefore, each active variable belongs to exactly one factor within the product.

Since rectangle R_i contributes exactly two active variables (one assigned to y_i and one to z_i), and each active variable belongs to at most one factor in the product, the configuration choice C_i can influence the balance of at most two factors. Therefore, conditioned on $\mathbf{R}$, the balance events form a read-2 family with respect to the independent random variables $C_1, \ldots, C_m$. ◀

4.5 Imbalance in Multiple Factors

▶ **Theorem 36** (Multiple Factor Imbalance with High Probability). *Let* $\mathcal{L}$ *denote the set of* $r = \Theta(\log n)$ *levels* j *where* $|X_{ij}| \geq N^{0.75}$*. Consider a fixed summand* $f_i = \prod_{j=1}^{t} f_{ij}$ *in the formula decomposition. Under the random restriction, with probability at least* $1 - \exp(-\Omega(\log^2 n))$ *over both the choice of sampled rectangles* $\mathbf{R}$ *and the configuration choices, at least half of the factors in* $\mathcal{L}$ *exhibit* k*-imbalance with* $k = n^{1/32}$*.*

Proof. The random restriction involves two stages of randomness: sampling rectangles $\mathbf{R} = (R_1, \ldots, R_m)$ uniformly from $\mathcal{R}$ with replacement, then independently choosing configurations for each sampled rectangle. We condition on the first stage and apply concentration bounds for the second.

For each factor $\ell \in \mathcal{L}$, let $B_\ell = \mathbf{1}[\text{factor } \ell \text{ is } k\text{-balanced}]$. By Theorem 23, with probability at least $1 - \exp(-\Omega(n^{1/12} \log^{-1.5} n))$ over the rectangle sampling and permutation, every factor $\ell \in \mathcal{L}$ receives at least $\Omega(n^{1/12} \log^{-1.5} n)$ sampled rectangles with diagonal imbalance. Call this the good event $\mathcal{G}$.

Conditioned on any outcome $\mathbf{R} \in \mathcal{G}$, by Lemma 35, the balance events form a read-2 family with respect to the configuration choices. Moreover, by the binomial analysis from Proposition 34, each factor satisfies $\Pr[B_\ell = 1 \mid \mathbf{R}] = O(n^{-1/96} \log^{0.75} n)$. Therefore, the average conditional probability is $p = O(n^{-1/96} \log^{0.75} n)$.

We apply Theorem 16 with $k = 2$ to bound the probability that at least half the factors are balanced. For $q = 1/2$ and $p = O(n^{-1/96}) \ll 1/2$, the Kullback-Leibler divergence satisfies $D\,(1/2 \,\|\, p) \approx \frac{1}{2} \log \frac{1}{2p} = \Theta(\log n)$. Therefore, for any $\mathbf{R} \in \mathcal{G}$,

$$\Pr\left[\sum_{\ell \in \mathcal{L}} B_\ell \geq \frac{r}{2} \;\middle|\; \mathbf{R}\right] \leq \exp\left(-\frac{r}{k} \cdot D\,(1/2 \,\|\, p)\right) = \exp(-\Omega(\log^2 n)).$$

Since this bound holds uniformly for all $\mathbf{R} \in \mathcal{G}$, and the complement event $\mathcal{G}^c$ has probability at most $\exp(-\Omega(n^{1/12} \log^{-1.5} n))$, the total probability satisfies

$$\Pr\left[\sum_{\ell \in \mathcal{L}} B_\ell \geq \frac{r}{2}\right] \leq \exp(-\Omega(\log^2 n)) + \exp(-\Omega(n^{1/12} \log^{-1.5} n)) = \exp(-\Omega(\log^2 n)).$$

Therefore, with probability at least $1 - \exp(-\Omega(\log^2 n))$, at least half the factors in $\mathcal{L}$ exhibit k-imbalance. ◀

4.6 The Lower Bound via Union Bound

▶ **Theorem 37** (Lower Bound for Sparse Determinants). *Let X_G be the sparse matrix constructed with all the requisite combinatorial properties as constructed above (in 3.6). Then any multilinear formula computing* $\mathsf{Det}(X_G)$ *must have size* $s = \exp(\Omega(\log^2 n))$.

Proof. Suppose a multilinear formula F of size $s = n^{O(\log n)}$ computes $\mathsf{Det}(X_G)$. By Theorem 12, the formula F admits a decomposition $\sum_{i=1}^{s+1} f_i$ where each $f_i = \prod_{j=1}^{t} f_{ij}$ is a product of $t = \Theta(\log N) = \Theta(\log n)$ factors (since $N = \Theta(n \log^6 n)$). By Lemma 31, each product contains $r = \Theta(\log n)$ factors with at least $N^{0.75}$ variables.

Consider a fixed summand f_i. By Theorem 36, under the random restriction of Section 4.2, with probability at least $1 - \exp(-\Omega(\log^2 n))$, at least $\ell = r/2 = \Theta(\log n)$ factors in this summand exhibit k-imbalance with $k = n^{1/32}$.

Applying Proposition 11, on this high-probability event, the summand f_i satisfies

$$\mu_\sigma(f_i) \leq 2^m \cdot 2^{-\ell k} = 2^m \cdot 2^{-\Theta(\log n) \cdot n^{1/32}} = 2^m \cdot 2^{-\Omega(n^{1/32} \log n)},$$

where $m = \Theta(n^{1/3})$ is the number of active variables after the restriction.

Taking a union bound over all $s+1$ summands, if $s = n^{O(\log n)}$, then with probability at least $1-(s+1)\cdot\exp(-\Omega(\log^2 n)) = 1-o(1)$, all summands simultaneously exhibit $\ell = \Theta(\log n)$ imbalanced factors. By subadditivity of rank,

$$\mu_\sigma\left(\sum_{i=1}^{s+1} f_i\right) \leq \sum_{i=1}^{s+1} \mu_\sigma(f_i) \leq (s+1) \cdot 2^m \cdot 2^{-\Omega(n^{1/32} \log n)}.$$

For $s = n^{O(\log n)}$ and $m = \Theta(n^{1/3})$, since $2^{\Omega(n^{1/32} \log n)} \gg n^{O(\log n)}$ for sufficiently large n, we have with probability $1 - o(1)$,

$$\mu_\sigma(F) \leq n^{O(\log n)} \cdot 2^m \cdot 2^{-\Omega(n^{1/32} \log n)} \ll 2^m$$

On the other hand, the restricted determinant $\mathsf{Det}(X_G)$ maintains full rank after the restriction. By Theorem 29, with probability at least $1 - \exp(-\Omega(\log^6 n))$, the residual graph G_{rest} obtained after removing the $2m$ rows and columns involved in sampled rectangles retains a perfect matching. The random restriction assigns variables as follows: for each of the m sampled rectangles, two opposite diagonal corners become active variables (y_i, z_i) while the other diagonal is set to constants; the remaining $|E_G| - 2m$ edges are assigned values according to the perfect matching M_{rest} in G_{rest} (matched edges set to 1, unmatched edges set to 0). Under this assignment, the restricted determinant expands as

$$\mathsf{Det}(X_G)|_{\text{restricted}} = \prod_{i=1}^{m} (y_i z_i - 1)\,.$$

This polynomial is defined over m variables in $Y = \{y_1, \ldots, y_m\}$ and m variables in $Z = \{z_1, \ldots, z_m\}$, yielding a coefficient matrix M_{Det} of dimension $2^m \times 2^m$ with full rank 2^m (since the determinant expansion contains all 2^m monomials over the Y variables paired with all 2^m monomials over the Z variables).

Therefore, $\mu_\sigma(\mathsf{Det}(X_G)) = 2^m$, contradicting the bound $\mu_\sigma(F) \ll 2^m$ established above. This contradiction shows that any multilinear formula computing $\mathsf{Det}(X_G)$ requires size $s = n^{\Omega(\log n)}$. ◀

References

1 Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inf. Process. Lett.*, 18(3):147–150, 1984. doi:10.1016/0020-0190(84)90018-8.

2 Richard P. Brent. The parallel evaluation of general arithmetic expressions. *J. ACM*, 21(2):201–206, 1974. doi:10.1145/321812.321815.

3 Richard P. Brent, David J. Kuck, and Koichi Maruyama. The parallel evaluation of arithmetic expressions without division. *IEEE Trans. Computers*, 22(6):532–534, 1973. doi:10.1109/T-C.1973.223745.

4 Suryajith Chillara, Christian Engels, Nutan Limaye, and Srikanth Srinivasan. A near-optimal depth-hierarchy theorem for small-depth multilinear circuits. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 934–945. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00092.

5 Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. Small-depth multilinear formula lower bounds for iterated matrix multiplication with applications. *SIAM J. Comput.*, 48(1):70–92, 2019. doi:10.1137/18M1191567.

6 Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. *SIAM J. Comput.*, 45(4):1974–2002, 2016. doi:10.1137/130950215.

7 Dmitry Gavinsky, Shachar Lovett, Michael E. Saks, and Srikanth Srinivasan. A tail bound for read-k families of functions. *Random Struct. Algorithms*, 47(1):99–108, 2015. doi:10.1002/RSA.20532.

8 Meena Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chicago J. Theor. Comput. Sci.*, 1997, 1997. URL: http://cjtcs.cs.uchicago.edu/articles/1997/5/contents.html.

9 Colin McDiarmid. On the method of bounded differences. *Surveys in Combinatorics*, 141:148–188, 1989.

10 Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2nd edition, 2017. doi:10.1017/9781316822135.

11 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 633–641. ACM, 2004. doi:10.1145/1007352.1007353.

12 Ran Raz. Separation of multilinear circuit and formula size. *Theory Comput.*, 2(1):121–135, 2006. doi:10.4086/toc.2006.v002a006.

13 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):8:1–8:17, 2009. doi:10.1145/1502793.1502797.

14 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, July 15-18, 2009*, pages 128–139. IEEE Computer Society, 2009. doi:10.1109/CCC.2009.16.

15 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. https://github.com/dasarpmar/lowerbounds-survey/, 2021. Manuscript.

16 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3-4):207–388, 2010. doi:10.1561/0400000039.

17 Leslie G. Valiant. Completeness classes in algebra. In *STOC*, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.

18 Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979. doi:10.1016/0304-3975(79)90044-6.

Quantum Advantage in Tolerant Junta Testing

Avishay Tal
University of California, Berkeley, CA, USA

Weiqiang Yuan
EPFL, Lausanne, Switzerland

Abstract

We establish the first super-polynomial quantum advantage for the tolerant junta testing problem in the adaptive setting. Specifically, we show that within a certain parameter regime, tolerant k-junta testing with high precision can be solved using $\mathrm{poly}(k)$ quantum queries, whereas any classical algorithm requires at least $k^{\Omega(\log k)}$ queries.

The problem of tolerant k-junta testing is as follows: given parameters $(k, \varepsilon_1, \varepsilon_2)$, with $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1/2$, and black-box access to a Boolean function f (defined on n variables), distinguish whether f is ε_1-close to some k-junta or ε_2-far from every k-junta.

We show the quantum advantage for a range of parameters close to $1/2$, for example, $\varepsilon_1 = 1/2 - 1/k$ and $\varepsilon_2 = 1/2 - 1/(2k^2)$. (As such, the problem is more naturally captured using the notion of correlation with closest k-junta.) The (non-adaptive) quantum tester we use was given by a recent work of Bao, Liu, Yao, Ye, and Zhang (SOSA 2026). We slightly adapt their analysis to show that it holds in the above parameter regime. On the other hand, our classical lower bound requires substantial new ideas. Inspired by the lower bound techniques of Chen and Patel (FOCS 2023), we introduce a new hard distribution of "yes" instances (i.e., instances with distance at most ε_1 to k-juntas) that is based on planting an "approximate-junta" as follows: we randomly pick k out of n coordinates, and for each fixing of the k coordinates, the 2^{n-k} values in the restricted subcube are drawn randomly except for an error-correcting code on which we place the same random bit. We show that this distribution is much closer to k-juntas than the uniform distribution, but on the other hand, they are indistinguishable with respect to any classical algorithm making $k^{o(\log k)}$ queries.

2012 ACM Subject Classification Theory of computation → Oracles and decision trees; Theory of computation → Quantum complexity theory

Keywords and phrases Property testing, Quantum query complexity, Error-correcting code, Boolean function analysis

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.34

Funding *Avishay Tal*: Supported by the National Science Foundation CAREER Award CCF-2145474.
Weiqiang Yuan: Supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00026.

1 Introduction

A Boolean function $f : \{0,1\}^n \to \{\pm 1\}$ is said to be a k-junta if it depends on only k of n input variables. Despite representing a structurally simple class of functions, juntas play a foundational role across various areas of theoretical computer science, including computational learning theory [31, 21, 37], hardness amplification [9, 28], and property testing (see references below), which is the focus of this study.

In the standard junta testing problem, we are given oracle access to a hidden Boolean function $f : \{0,1\}^n \to \{\pm 1\}$. The objective is to determine whether f is a k-junta or ε-far from all k-juntas. To be precise, a *randomized* algorithm $\mathcal{A}^f$ is a (k, ε)-junta tester if it accepts every k-junta f with probability at least $2/3$, while rejecting with probability at least

© Avishay Tal and Weiqiang Yuan;
licensed under Creative Commons License CC-BY 4.0

2/3 every f that differs from every k-junta on at least an ε-fraction of inputs. We measure the tester's efficiency by its query complexity, namely, the number of oracle calls to f (while the running time is not considered). Ideally, we want the tester's query complexity to be as small as possible in terms of k and ε, but independent of n (the input length of f).

In the seminal work of Fischer, Kindler, Ron, Safra, and Samorodnitsky [22], the authors proposed the first junta tester with query complexity depending only polynomially on k and $1/\varepsilon$. Since then, a long line of research [19, 25, 2, 6, 7, 36, 1, 35, 18, 8, 30, 29, 4, 10, 20, 27, 33, 15, 13, 32, 14, 16, 17] has been dedicated to developing more efficient testers and sharper lower bounds, leading to a comprehensive understanding of the problem and its variants. In the standard model, Blais [7] provided a tester using $O(k \log k + k/\varepsilon)$ queries, which was later shown to be optimal for constant ε [35]. On the other hand, [18] shows that any non-adaptive tester must make $\tilde{\Omega}(k^{3/2}/\varepsilon)$ queries, nearly matching the $\tilde{O}(k^{3/2}/\varepsilon)$ upper bound established by Blais [6].

Beyond the classical setting, quantum algorithms have been shown to achieve a quadratic speedup: [1] provides a quantum tester making only $\tilde{O}(\sqrt{k/\varepsilon})$ queries, which is nearly optimal [11]. Lastly, recent work has extended these results to more general models, such as distribution-free testing [30, 10] and relative-error testing [17], demonstrating that junta testing remains easy even under these broader distance measures. However, this efficiency relies heavily on the assumption of perfect, noise-free inputs.

Tolerant Testing

A significant limitation of the standard testing model is that there is no guarantee on the performance of the tester on functions close to satisfying the property. A more realistic scenario is that given a black-box function f, one seeks to test whether there are k features that explain the output *in most cases*, or f is far from it. This is because data in the wild are often noisy and almost never perfect. To address this, Parnas, Ron, and Rubinfeld [34] introduce a natural relaxation of the standard testing model called *tolerant testing*.

In the context of junta testing, a $(k, \varepsilon_1, \varepsilon_2)$-tolerant junta tester must decide whether a Boolean function f is ε_1-close to some k-junta or ε_2-far from every k-junta by making oracle calls to f. Standard junta testing naturally emerges as a special case where $\varepsilon_1 = 0$.

In sharp contrast to the standard model, tolerant junta testing appears to be much harder. In the non-adaptive setting, Chen et al. [14] proved that any classical tolerant junta tester requires $2^{\Omega(\sqrt{k})}$ queries. This lower bound was subsequently shown to be tight by Nadimpalli and Patel [32] (building on [20, 27]), who provided a matching non-adaptive upper bound of $2^{\tilde{O}(\sqrt{k \log(1/(\varepsilon_2 - \varepsilon_1))})}$.

In the classical adaptive setting, despite significant effort, our understanding of tolerant junta testing remains limited. The best-known adaptive $(k, \varepsilon_1, \varepsilon_2)$-tolerant junta tester has query complexity $2^{\tilde{O}(k^{1/3})}$ when $\varepsilon_2 - \varepsilon_1$ is constant [16], while the best-known lower bound is $k^{\Omega(\log(1/(\varepsilon_2 - \varepsilon_1)))}$ when $\varepsilon_1, \varepsilon_2$ are bounded away from 0 and 1/2 [15]. In short, even in the parameter regime where $\varepsilon_1 = 0.01$ and $\varepsilon_2 = 0.49$, neither a tester with query complexity $\exp(o(k^{1/3}))$ nor a superpolynomial lower bound is known.

In the quantum world, however, the landscape changes dramatically. Recent work by Bao, Liu, Yao, Ye, and Zhang [3] introduces an elegant, non-adaptive quantum tester with query complexity $O(k \log k)$, provided that $4\varepsilon_1 < \varepsilon_2$. To evaluate this advantage, Bao et al. [3] complement their algorithm with a classical non-adaptive lower bound, demonstrating a surprising exponential quantum speedup over classical *non-adaptive* tolerant junta testers.

This provable advantage highlights the power of quantum in tolerant junta testing. However, [3]'s results do not establish any super-polynomial quantum speedup over classical *adaptive* tolerant junta testers, since the quantum tester in [3] works only in a parameter regime that is mutually exclusive of that of the superpolynomial lower bound by [15].[1] This leads to the following natural question:

Do adaptive quantum tolerant junta testers significantly outperform their classical counterparts?

1.1 Main Result

In this paper, we give an affirmative answer to the above question by showing a superpolynomial quantum speedup against classical adaptive tolerant junta testers.

▶ **Theorem 1** (Main Result). *For every $k \in \mathbb{N}$ and $\varepsilon > 2^{-k^{.99}}$, there exist $0 < \varepsilon_1 < \varepsilon_2 < 1/2$ such that $\varepsilon = \varepsilon_2 - \varepsilon_1$ and the following holds:*

- *There exists a* non-adaptive quantum *$(k, \varepsilon_1, \varepsilon_2)$-tolerant junta tester with query complexity $O(\mathrm{poly}(k, 1/\varepsilon))$.*
- *Any* adaptive classical *$(k, \varepsilon_1, \varepsilon_2)$-tolerant junta tester must make at least $k^{\Omega(\log(1/\varepsilon))}$ queries.*

The key to the proof of Theorem 1 is a stronger classical adaptive lower bound, inspired by [15], which applies to a wider range of parameters for which the quantum tester in [3] is valid. The quantum advantage is perhaps better captured by the notion of correlation with the closest k-junta, which is equivalent to the distance up to a linear transformation. In this notation, we show that for any $\tau > 2^{-k^{.99}}$,

Quantum Tester (Theorem 23) There exists a non-adaptive quantum algorithm, making $\mathrm{poly}(k/\tau)$ queries, that distinguishes between functions with at least $\tau_1 = \tau$ correlation and at most $\tau_2 = \tau^2/2$ correlation to k-juntas.

Classical Lower Bound (Theorem 16) On the other hand, we show that distinguishing $\tau_1 = \tau$ from $\tau_2 = \tau^2/2$ correlation with k-juntas (and even $\tau_1 = \tau$ versus $\tau_2 = \tau^c$ for any constant $c > 1$) requires $k^{\Omega(\log(1/\tau))}$ classical queries, even adaptively.

By picking $\tau = 1/\mathrm{poly}(k)$, we obtain a quantum algorithm with $\mathrm{poly}(k)$ queries, compared to a quasi-polynomial classical lower bound of $k^{\Omega(\log k)}$ queries.

Prior to our work, superpolynomial lower bound for tolerant junta testing was only known for much closer parameters τ_1, τ_2, i.e., with $\tau_1 = \tau_2 \cdot (1 + o(1))$ [15].

We view this as a quantum advantage to a natural tolerant property testing problem: distinguish whether there exist k features that explain at least τ or at most $\tau^2/2$ fraction of the outputs.

1.2 Proof Overview

1.2.1 Quantum Tester

Our quantum tolerant junta tester is a simple adaptation of the one provided in [3].

[1] More precisely, the quantum tester is efficient if either the gap $\varepsilon_2 - \varepsilon_1$ is constant, or both $\varepsilon_1, \varepsilon_2$ are close to 0, while the only known super-polynomial classical adaptive lower bound [15] requires a sub-constant gap between $\varepsilon_1, \varepsilon_2$ and that both $\varepsilon_1, \varepsilon_2$ are bounded away from 0 and 1/2, which we will discuss in more detail in Section 1.2.2.

▶ **Theorem 2** ([3, Theorem 1.3]). *Let $0 < \varepsilon_1 < \varepsilon_2 < 1/2$ be constants such that $4\varepsilon_1 < \varepsilon_2$. Then, there exists a non-adaptive quantum $(k, \varepsilon_1, \varepsilon_2)$-junta tester with query complexity $O(k \log k)$.*

To demonstrate this, Bao et al. [3] first approximate the minimum distance of a Boolean function f and T-juntas by $\mathrm{Inf}_{\overline{T}}[f]$, where $\mathrm{Inf}_{\overline{T}}(f)$ is expressed in terms of the Fourier expansion of f as $\sum_{S\subseteq[n]:S\cap T\neq\emptyset} \widehat{f}_S^2$. Specifically, they show

$$\mathrm{Inf}_{\overline{T}}[f]/4 \leq \mathrm{dist}(f, \mathcal{J}_T) \leq \mathrm{Inf}_{\overline{T}}[f],$$

where $\mathrm{dist}(f, \mathcal{J}_T)$ denotes the minimum distance of a Boolean function f from $\mathcal{J}_T$, the class of juntas on T. Then, they design an "influence tester" that distinguishes between the following two cases:

- Yes: $\mathrm{Inf}_{\overline{T}}[f] \leq \gamma_1$ for some $T \subseteq [n]$ with size $|T| = k$
- No: $\mathrm{Inf}_{\overline{T}}[f] \geq \gamma_2$ for all $T \subseteq [n]$ with size $|T| = k$

for any $\gamma_1 < \gamma_2$ with query complexity $\mathrm{poly}(k, (\gamma_2 - \gamma_1)^{-1})$.

The quantum tester in Theorem 1 follows directly from combining their influence tester with the following improved characterization of $\mathrm{corr}(f, \mathcal{J}_T)$ in terms of $\inf_{\overline{T}}[f]$:

$$1 - \mathrm{Inf}_{\overline{T}}(f) \leq \mathrm{corr}(f, \mathcal{J}_T) \leq \sqrt{1 - \mathrm{Inf}_{\overline{T}}(f)}.$$

We emphasize that this characterization is not new: the upper bound was shown in [3, Lemma 29] as an intermediate step, while the lower bound was provided in [22].

In what follows, we briefly discuss the main idea of the quantum influence tester in [3]. Its efficiency stems from the fact that a single quantum query suffices to sample $S \subseteq [n]$ from the Fourier distribution of f. With a sufficient number of such samples, we are able to estimate $\mathrm{Inf}_{\overline{T}}[f]$ for any $T \subseteq [n]$ up to some small error. We emphasize here that while the query complexity is $\mathrm{poly}(k)$, the running time of the algorithm is exponential in k – although this is not the focus of this paper.

As our goal is to determine if $\min_{T:|T|=k} \mathrm{Inf}_{\overline{T}}[f]$ is smaller than γ_1 or greater than γ_2, we may want a good approximation of $\mathrm{Inf}_{\overline{T}}[f]$ for all T with $|T| = k$. However, since there are $\binom{n}{k}$ choices of subsets of $[n]$ with size k, a naive union bound over all these subsets will incur a $\log n$ factor to the number of queries, which is intolerable.

De, Mossel, and Neeman [20] provide a solution by showing that the tolerant junta testing problem for functions with an arbitrarily large input length n can be reduced to the case where $n = \mathrm{poly}(k)$. Subsequently, Iyer, Tal, and Whitmeyer [27] further reduce n to $O(k)$, while barely changing the distance to the closest k-junta. Bao et al. [3] use Fourier samples to get an even simpler and more efficient reduction, using only $O(k \log k)$ quantum queries.

1.2.2 Classical Lower Bound

Our classical lower bound is inspired by the work of Chen and Patel [15], where they give the first superpolynomial adaptive lower bound for tolerant junta testing. Specifically, they prove the following.

▶ **Theorem 3** (Theorem 1 in [15]). *Let $0.01 \leq \varepsilon_1 < \varepsilon_2 \leq 0.49$ be two parameters such that $\varepsilon_2 - \varepsilon_1 \geq 2^{-O(k^{.99})}$. Then any $(k, \varepsilon_1, \varepsilon_2)$-tolerant junta tester requires $k^{\Omega(\log(1/(\varepsilon_2-\varepsilon_1)))}$ queries.*

Unfortunately, their result together with the quantum tester from [3] does not directly imply Theorem 1. When $\varepsilon_2 - \varepsilon_1 \to 0$, the quantum tester requires that $\frac{1}{2} - \varepsilon < \varepsilon_1 < \varepsilon_2$; while [15]'s lower bound requires $\varepsilon_1, \varepsilon_2$ to be bounded away from $1/2$. To further clarify why the lower bound in [15] does not apply to our setting, we provide a brief overview of the key ideas of their proof – some of which also play important roles in our proof.

Chen-Patel's Proof

The key of [15]'s proof is the construction of two hard distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$, which are *mostly* supported on yes and no instances, respectively. By Yao's minimax lemma, it suffices to show that any deterministic tester making a polynomial number of queries cannot distinguish between the two distributions. Below, we will only focus on the core of the construction in [15], which corresponds to the parameter regime where

$$n = k + \ell, \qquad \ell < k^{0.99}, \qquad \tau_2 = \tau_0 + 2^{-4\ell}, \qquad \tau_1 = \tau_0 + 2^{-2\ell} - 2^{-4\ell},$$

where $\tau_1 \coloneqq 1 - 2\varepsilon_1, \tau_2 \coloneqq 1 - 2\varepsilon_2$ are the correlations between $\mathcal{D}_Y, \mathcal{D}_N$ and the closest k-junta, and $\tau_0 \coloneqq \mathbb{E}_{\boldsymbol{a} \sim \{\pm 1\}^{2^\ell}}[|\sum_{i=1}^{2^\ell} \boldsymbol{a}_i|/2^\ell] \approx \sqrt{2/\pi} \cdot 2^{-\ell/2}$.

The hard distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$ are specified as follows:

- $\mathcal{D}_N$: The uniform distribution over all n-bit Boolean functions.
- $\mathcal{D}_Y$: Sample $\boldsymbol{S} \sim \binom{[n]}{k}$. Then, for each $y \in \{0,1\}^{\boldsymbol{S}}$, sample the values $\{\boldsymbol{f}(y,z)\}_{z \in \{0,1\}^{\overline{\boldsymbol{S}}}}$ uniformly at random from $\{\pm 1\}$, conditioned on the product of these values being -1:

$$\prod_{z \in \{0,1\}^{\overline{\boldsymbol{S}}}} \boldsymbol{f}(y,z) = -1.$$

A straightforward application of Hoeffding's inequality shows $\mathrm{corr}(\boldsymbol{g}, \mathcal{J}_k)$ is concentrated around τ_0 over $\boldsymbol{g} \sim \mathcal{D}_N$, hence at most τ_2 with high probability. On the other hand, by a coupling argument, [15] shows that

$$\mathbb{E}_{\boldsymbol{a}}[|\sum_{i=1}^{2^\ell} \boldsymbol{a}_i|/2^\ell] \geq \tau_0 + 2^{-2\ell},$$

where $\boldsymbol{a}$ is uniformly sampled from $\{\pm 1\}^{2^\ell}$ conditioned on $\prod_{i=1}^{2^\ell} \boldsymbol{a}_i = -1$. Using the same reasoning, we confirm that $\mathrm{corr}(\boldsymbol{f}, \mathcal{J}_k)$ is at least τ_1 with high probability over $\boldsymbol{f} \sim \mathcal{D}_Y$.[2]

To show the two distributions are indistinguishable, Chen and Patel [15] observe the following: conditioned on any fixed $\boldsymbol{S} = S \in \binom{n}{k}$, an algorithm can distinguish the two distributions only if for some $y \in \{0,1\}^S$, all the points in $\{y,z\}_{z \in \{0,1\}^{\overline{S}}}$ are queried. In particular, after querying $x^{(1)}, \ldots, x^{(m)}$, the algorithm can achieve non-zero advantage only if there exists a pair of indices $1 \leq i < j \leq m$, such that $x^{(i)} = (x^{(j)})^{\oplus \overline{S}}$. Since the probability that $x^{(i)} = (x^{(j)})^{\oplus \overline{S}}$ for random $\boldsymbol{S} \sim \binom{[n]}{k}$ is at most $1/\binom{n}{k}$, by union bound, the algorithm distinguishes two distributions with advantage at most $m^2/\binom{n}{k}$. Therefore, any tester that distinguishes between the two distributions with constant advantage requires $\Omega\left(\sqrt{\binom{n}{k}}\right)$ queries.

[2] In [15], the authors did not manage to optimize the bound for the gap $\tau_1 - \tau_2$ under this construction. Indeed, we conjecture that the right answer here is $\Theta(2^{-3\ell/2})$. However, since we can transform the odd-parity distribution into the uniform distribution by flipping at most one bit, the gap cannot exceed $2^{-\ell}$, thus, does not suffice for our purpose.

In the above construction, $\mathbb{E}_{\boldsymbol{g}\sim\mathcal{D}_N}[\mathrm{corr}(\boldsymbol{g},\mathcal{J}_k)]$ equals $(1-o(1))\cdot\mathbb{E}_{\boldsymbol{f}\sim\mathcal{D}_Y}[\mathrm{corr}(\boldsymbol{f},\mathcal{J}_k)]$, whereas we require the former to be quadratic in the latter. Although [15] extends their construction to a wider class of parameters by padding and planting biased output bits, unfortunately, any straightforward adaptation of their proof remains incompatible with the parameter regime required for our quantum tester.

An Unsuccessful Attempt

Let $\mathcal{D}_N$ remain the uniform distribution. Our objective is to find a "pseudorandom" distribution $\mathcal{D}_Y$ with correlation at least $2^{-\ell/4}$ to k-juntas. To achieve this, we still first sample $\boldsymbol{S}\sim\binom{[n]}{k}$, which is presumed to be the support of the nearest k-junta. Then, for any fixed assignment $y\in\{0,1\}^S$, we aim to ensure that $\mathbb{E}[|\sum_{z\in\{0,1\}^{\overline{S}}}\boldsymbol{f}(y,z)|]\geq 2^{3\ell/4}$. This allows us to apply Hoeffding's inequality to show that $\mathrm{corr}(f,\mathcal{J}_{\boldsymbol{S}})=\Omega(2^{-\ell/4})$ holds with high probability. A natural way to implement this is to first sample a uniform sign $\boldsymbol{b}_y\sim\{\pm1\}$. Then generate $\{\boldsymbol{f}(y,z)\}_{z\in\{0,1\}^{\overline{S}}}$ from an independent biased distribution towards $\boldsymbol{b}_y$, that is, for each $z\in\{0,1\}^{\overline{S}}$, let $\boldsymbol{f}(y,z)=\boldsymbol{b}_y$ with probability $1/2+2^{-\ell/4}$, and $\boldsymbol{f}(y,z)=-\boldsymbol{b}_y$ otherwise.

Unfortunately, this strategy is vulnerable to a simple, efficient distinguisher based on estimating influences. While a function $\boldsymbol{g}\sim\mathcal{D}_N$ satisfies $\mathrm{Inf}_i[\boldsymbol{g}]\geq 1/2-\Theta(2^{-n/2}\cdot\sqrt{\log n})$ for all $i\in[n]$ with high probability, a function $\boldsymbol{f}\sim\mathcal{D}_Y$ exhibits significantly lower influence: for any $i\in\overline{\boldsymbol{S}}$, $\mathrm{Inf}_i[\boldsymbol{f}]\leq 1/2-\Theta(2^{-\ell/2})$ with high probability. Consequently, an algorithm can distinguish the two distributions by estimating $\mathrm{Inf}_i[\boldsymbol{f}]$ for each $i\in[n]$ with precision $O(2^{-\ell/2})$ using $\Theta(2^\ell)=\Theta(1/\varepsilon)$ samples, and then simply checking if the minimum of the estimations over $i\in[n]$ is below a threshold of $1/2-\Theta(2^{-\ell/2})$.

Our Strategy

The fundamental flaw in the previous construction is that it introduces local correlations between the values of $\boldsymbol{f}$ at adjacent points. More generally, $\mathcal{D}_Y$ remains distinguishable from $\mathcal{D}_N$ if the marginal distribution of $\boldsymbol{f}$ on any Hamming ball of constant radius deviates from being uniform. To circumvent this, we propose a construction based on error-correcting codes. By planting the bias only on the codewords of a high-distance code, we ensure that correlations only emerge between inputs with large Hamming distance; hence, $\boldsymbol{f}$ is locally uniform.

Formally, our "yes" distribution $\mathcal{D}_Y$ is generated as follows:

1. Sample a support $\boldsymbol{S}\sim\binom{[n]}{k}$.
2. Let $C\subseteq\{0,1\}^{\overline{S}}$ be an arbitrary error-correcting code with rate $R=3/4$ and relative distance $\delta=0.01$, as guaranteed by the Gilbert-Varshamov bound.
3. For each assignment $y\in\{0,1\}^S$, sample a uniform sign $\boldsymbol{b}_y\sim\{\pm1\}$.
4. For each $z\in\{0,1\}^{\overline{S}}$, let

$$\boldsymbol{f}(y,z)\begin{cases}:=\boldsymbol{b}_y & \text{if } z\in C,\\ \sim\{\pm1\} & \text{if } z\notin C.\end{cases}$$

Since $\mathbb{E}_{\boldsymbol{f}}[|\sum_{z\in\{0,1\}^{\overline{S}}}\boldsymbol{f}(y,z)|]\geq|C|-2^{\ell/2}=\Omega(2^{3\ell/4})$, a standard application of Hoeffding's inequality confirms that $\mathrm{corr}(\boldsymbol{f},\mathcal{J}_k)\geq\Omega(2^{-\ell/4})$ happens with high probability.

Indistinguishability

Next, we present the main idea of showing the indistinguishability of $\mathcal{D}_Y$ and $\mathcal{D}_N$ generated as described above. Let $\mathcal{A}^f$ be an arbitrary algorithm making m queries to f, and consider any path $\mathcal{P}$ in its decision tree representation. Let $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ denote the set of points queried along $\mathcal{P}$. Under the uniform distribution $\boldsymbol{g} \sim \mathcal{D}_N$, $\mathcal{A}^{\boldsymbol{g}}$ follows $\mathcal{P}$ with probability exactly 2^{-m}. We show that under $\boldsymbol{f} \sim \mathcal{D}_Y$, this probability remains $(1 \pm o(1))2^{-m}$ as long as $m = k^{O(\ell)}$. The indistinguishability of $\mathcal{D}_Y$ and $\mathcal{D}_N$ follows from a standard coupling argument. Below, we briefly discuss how to show $\mathcal{A}^{\boldsymbol{f}}$ follows $\mathcal{P}$ with probability close to 2^{-m} over $\boldsymbol{f} \sim \mathcal{D}_Y$.

We say $S \in \binom{[n]}{k}$ is good if for every distinct pair of query points $x^{(i)}, x^{(j)}$, at least one of the following holds:

- $x^{(i)}$ and $x^{(j)}$ are inconsistent on at least one of the coordinates in S,
- $x^{(i)}$ and $x^{(j)}$ has Hamming distance less than $\ell/100$.

We claim that conditioned on $\boldsymbol{S} = S$ for a good S, the queried values $\boldsymbol{f}(x^{(1)}), \ldots, \boldsymbol{f}(x^{(m)})$ are independent and uniform. To see this, fix any $y \in \{0,1\}^S$. The "good" property ensures that the subcube $\{x \in \{0,1\}^n \mid x_S = y\}$ contains at most one queried point $x^{(i)}$ whose projection on $\overline{S}$ is a codeword in C, because if there were two such points, they must agree on S while having a Hamming distance of at least $\ell/100$, which contradicts the assumption that S is good. Since each query hits at most one codeword per subcube, and the signs $(\boldsymbol{b}_y)_{y \in \{0,1\}^S}$ associated with the codewords are independent and uniform, we deduce that $\boldsymbol{f}(x^{(1)}), \ldots, \boldsymbol{f}(x^{(m)})$ are independent and uniform. Therefore, conditioned on $\boldsymbol{S}$ being good, $\mathcal{A}^{\boldsymbol{f}}$ traverses $\mathcal{P}$ with probability exactly 2^{-m}.

It remains to show that a random $\boldsymbol{S} \sim \binom{[n]}{k}$ is good with high probability. For any two queried points $x^{(i)}, x^{(j)}$ with Hamming distance at least $\ell/100$, the probability that they coincide on coordinates in $\boldsymbol{S}$ is at most $\binom{n-\ell/100}{k} / \binom{n}{k} = k^{-\Omega(\ell)}$. By applying a union bound over all m^2 pairs of queries, we conclude that $\boldsymbol{S}$ is good with probability at least $1 - m^2 \cdot k^{-\Omega(\ell)}$.

1.3 Organization

The remainder of this paper is structured as follows: In Section 2, we provide the necessary notation and recall the standard tools in Boolean function analysis, coding theory, and concentration inequalities. In Section 3, we establish our new classical lower bound. Finally, in Section 4, we describe our quantum tester and analyze its query complexity and correctness.

2 Preliminaries

Throughout the paper, we use log to denote the base-2 logarithm, and ln to denote the natural logarithm. For any distribution $\mathcal{D}$ over a finite space Ω, we use $\mathcal{D}(x)$ to denote $\Pr_{\boldsymbol{x} \sim \mathcal{D}}[\boldsymbol{x} = x]$ for every $x \in \Omega$. Given two distributions μ, ν over the same space, we use $d_{\mathrm{TV}}(\mu, \nu)$ to denote the *total variation distance* between μ and ν. Given $S \subseteq [n]$, we use $\overline{S} := [n] \setminus S$ to denote the complement of S when n is clear from the context. Given $x \in \{0,1\}^n$, we use $x^{\oplus S} := x \oplus 1^S 0^{\overline{S}}$ to denote the string obtained by flipping all the coordinates in S of x.

Boolean Functions

Every n-bit Boolean function $f : \{0,1\}^n \to \{\pm 1\}$ can be uniquely written as a polynomial in the Fourier basis: $f(x) = \sum_{S \subseteq [n]} \widehat{f}_S \chi_S(x)$, where $\chi_S(x) := (-1)^{\sum_{i \in S} x_i}$, and $\{\widehat{f}_S\}_{S \subseteq [n]}$ are the Fourier coefficients of f.

Parseval's identity states that the sum of the squares of the Fourier coefficients of any Boolean function is exactly 1, that is, $\sum_{S\subseteq[n]} \widehat{f}_S^2 = 1$. Plancherel's theorem generalizes Parseval's identity. It states that for any two functions $f, g : \{0,1\}^n \to \mathbb{R}$, $\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})g(\boldsymbol{x})] = \sum_{S\subseteq[n]} \widehat{f}_S\widehat{g}_S$.

We recall the standard notion of influence for Boolean functions.

▶ **Definition 4** (Influence). *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a Boolean function and $i \in [n]$. The influence of i on f is defined as*

$$\mathrm{Inf}_i[f] \coloneqq \Pr_{\boldsymbol{x}\sim\{0,1\}^n}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus\{i\}})] = \sum_{S\ni i} \widehat{f}_S^2.$$

In this paper, we also work with the following two variants of influence.

▶ **Definition 5** (Level-k Influence). *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a Boolean function and $k, i \in [n]$. The level-k influence of i on f is defined as*

$$\mathrm{Inf}_i^{\leq k}[f] \coloneqq \sum_{S\ni i : |S|\leq k} \widehat{f}_S^2.$$

▶ **Definition 6** (Generalized Influence). *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a Boolean function and $T \subseteq [n]$ be an arbitrary set. The influence of T on f is defined as*

$$\mathrm{Inf}_T(f) \coloneqq \sum_{S : S\cap T\neq\varnothing} \widehat{f}_S^2 = 1 - \sum_{S : S\subseteq\overline{T}} \widehat{f}_S^2.$$

Juntas

For any set $S \subseteq [n]$, we say a Boolean function $f : \{0,1\}^n \to \{\pm 1\}$ is an S-junta if it only depends on variables in S. We denote the set of S-juntas on n variables by $\mathcal{J}_S^n$. Moreover, for any integer $k \in [n]$, we say f is a k-junta if it depends on at most k variables. We denote the set of k-juntas on n variables by $\mathcal{J}_k^n$. Note that $\mathcal{J}_k^n = \bigcup_{S\subseteq[n]:|S|=k} \mathcal{J}_S^n$. We will write $\mathcal{J}_S^n$ and $\mathcal{J}_k^n$ simply as $\mathcal{J}_S$ and $\mathcal{J}_k$ when n is clear from the context.

For any two Boolean functions $f, g : \{0,1\}^n \to \{\pm 1\}$, we use $\mathrm{dist}(f,g) \coloneqq \Pr_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})]$ to denote the distance between f and g. We also define $\mathrm{corr}(f,g) \coloneqq \mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})g(\boldsymbol{x})] = 1 - 2\mathrm{dist}(f,g)$ to denote the correlation between f and g. Let $\mathcal{F}$ be a set of n-bit Boolean functions. We use $\mathrm{dist}(f,\mathcal{F}) \coloneqq \min_{g\in\mathcal{F}} \mathrm{dist}(f,g)$ to denote the distance between f and the closest function in $\mathcal{F}$. Similarly, we use $\mathrm{corr}(f,\mathcal{F}) \coloneqq \max_{g\in\mathcal{F}} \mathrm{corr}(f,g) = 1 - 2\mathrm{dist}(f,\mathcal{F})$ to denote the maximum correlation between f and any function in $\mathcal{F}$.

Given $T \subseteq [n]$, the correlation between f and the closest T-junta can be calculated as follows.

▷ Claim 7 ([27]). Let $f : \{0,1\}^n \to \{\pm 1\}$ be a Boolean function and $T \subseteq [n]$. Define $f_{\mathrm{avg},T} : \{0,1\}^n \to [-1,1]$ where

$$f_{\mathrm{avg},T}(x) \coloneqq \mathbb{E}_{\boldsymbol{y}\sim\{0,1\}^n}[f(\boldsymbol{y}) \mid \boldsymbol{y}_T = \boldsymbol{x}_T].$$

It holds that

$$\mathrm{corr}(f, \mathcal{J}_T) = \mathbb{E}_{\boldsymbol{x}\sim\{0,1\}^n}[|f_{\mathrm{avg},T}(\boldsymbol{x})|].$$

The following lemma states that $\mathrm{corr}(f, \mathcal{J}_k)$ can be approximated by the maximum correlation between f and k-juntas supported only on coordinates with high level-k influence.

▶ **Lemma 8** ([27]). *Let $f : \{0,1\}^n \to \{\pm 1\}$ be any Boolean function and $\tau > 0$ be any parameter. Let $C \coloneqq \{i \in [n] \mid \mathrm{Inf}_i^{\leq k}[f] \geq \tau^2/k\}$. Then $|C| \leq k^2/\tau^2$. Moreover,*

$$\max_{S \subseteq C : |S| \leq k} \mathrm{corr}(f, \mathcal{J}_S) \geq \mathrm{corr}(f, \mathcal{J}_k) - \tau.$$

Quantum Query Algorithms

Let $f : \{0,1\}^n \to \{\pm 1\}$ be a Boolean function with Fourier representation $f(x) = \sum_{S \subseteq [n]} \widehat{f}_S \chi_S(x)$. By Parseval's theorem, one can define a natural distribution $\mathcal{S}_f$ over $2^{[n]}$ where $\mathcal{S}_f(S) = \widehat{f}_S^2$. The following folklore states that $\mathcal{S}_f$ can be sampled by an algorithm that makes a single quantum query to f.

▷ Claim 9 ([5]). There is an algorithm $\textsc{FourierSample}^f$ that makes one quantum query to $f : \{0,1\}^n \to \{\pm 1\}$, and for every $S \subseteq [n]$, it outputs S with probability $\widehat{f}_S^2$.

Error-correcting Codes

Given two n-bit strings $x, y \in \{0,1\}^n$, let $\Delta(x,y) \coloneqq \{i \mid x_i \neq y_i\}$ denote the Hamming distance between x and y. Moreover, the relative distance between x and y is defined as $\delta(x,y) \coloneqq \Delta(x,y)/n$.

▶ **Definition 10.** *A binary code of block length n is a subset $C \subseteq \{0,1\}^n$. The rate of C is defined as $\log|C|/n$, and the relative distance of C is defined as $\min_{c_1 \neq c_2 \in C} \delta(c_1, c_2)$. We say C is an (n, R, δ)-code if it has block length n, rate R, and relative distance δ.*

The following textbook result states that as long as $R + H_2(\delta) \leq 1$, there exists an (n, R, δ)-code, where $H_2(x) \coloneqq x\log(1/x) + (1-x)\log(1/(1-x))$ is the binary entropy function, extended continuously to the boundaries so that $H_2(0) = H_2(1) = 0$.

▶ **Theorem 11** (Gilbert–Varshamov bound [23, 38]). *Let $\delta \in (0, 1/2)$ and $R \in (0,1)$ be parameters such that $R + H_2(\delta) \leq 1$. Then there exists an (n, R, δ)-code for every $n \in \mathbb{N}$.*

We also use $H_2^{-1} : [0,1] \to [0,1/2]$ to denote the inverse binary entropy function, where given $x \in [0,1]$, $H_2^{-1}(x)$ is defined as the unique $y \in [0,1/2]$ such that $H_2(y) = x$.

▶ **Fact 12** ([12, Theorem 2.2]). *For every $x \in [0,1]$, it holds that $H_2^{-1}(x) \geq \frac{x}{2\log(6/x)}$.*

Concentration Bounds

We recall the following standard concentration inequalities.

▶ **Lemma 13** (Hoeffding's inequality [26]). *Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \in [0,1]$ be independent random variables and $\boldsymbol{S} = \sum_{i=1}^n \boldsymbol{X}_i$. Then for every $t > 0$,*

$$\Pr[|\boldsymbol{S} - \mathbb{E}[\boldsymbol{S}]| \geq t] \leq 2\exp(-2t^2/n).$$

▶ **Fact 14** ([24]). *Let n be even and $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n \sim \{\pm 1\}$ be n independent Rademacher variables. Then $\mathbb{E}_{\boldsymbol{a}}[|\sum_{i=1}^n \boldsymbol{a}_i|] \leq \sqrt{2n/\pi}$.*

We will also use the following simple inequality.

▶ **Fact 15.** *Let $0 \leq k, m \leq n$ be integers such that $m + k \leq n$. Then*

$$\binom{n-m}{k} \Big/ \binom{n}{k} \leq \left(\frac{n-k}{n}\right)^m.$$

Proof. Let $\ell := n - k$. Then,

$$\binom{n-m}{k}/\binom{n}{k} = \binom{n-m}{\ell-m}/\binom{n}{\ell} = \prod_{i=0}^{m-1}\frac{\ell-i}{n-i} \le (\ell/n)^m. \qquad \blacktriangleleft$$

3 Classical Lower Bound

In this section, we demonstrate our new classical lower bound for tolerant junta testing.

▶ **Theorem 16.** *Let $k \in \mathbb{N}$, $\tau_2 \in (2^{-k^{0.99}}, 1/2)$, and $\tau_2 < \tau_1 = \tau_2^c$ and for some $c \in (\frac{\log\log(1/\tau_2)}{\log(1/\tau_2)}, 1)$. Then any adaptive classical algorithm which*

- *accepts f with probability at least $2/3$ if f has correlation at least τ_1 to some k-junta;*
- *rejects f with probability at least $2/3$ if f has correlation at most τ_2 to every k-junta*

must make $k^{\Omega(\frac{\log(1/\tau_1)}{\log(2/c)})}$ queries to f.

To prove the theorem, we follow the standard approach for establishing testing lower bound: Using Yao's minimax lemma, it suffices to find two hard distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$ – supported mostly on yes instances and no instances respectively – and prove that they are indistinguishable by any classical algorithm making a polynomial number of queries.

The rest of the section is organized as follows: In Section 3.1, we define the hard distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$ and verify that they satisfy the required correlation properties. Then in Section 3.2, we show the indistinguishability of these two distributions. Finally, in Section 3.3 we put everything together to show Theorem 16.

3.1 Hard Distributions

Given any $k \in \mathbb{N}$, $\tau_2 \in (2^{-k^{0.99}}, 1/2)$, and $\tau_2 < \tau_1 = \tau_2^c$ and for some $c \in (\frac{\log\log(1/\tau_2)}{\log(1/\tau_2)}, 1)$, let $\ell := 2\log(1/\tau_2)$. We will focus only on the special case where $n = k + \ell$. The proof for arbitrary large n follows from a simple padding argument, namely, adding dummy input bits to f.

The two hard distributions $\mathcal{D}_N$ and $\mathcal{D}_Y$ are described as follows.

▶ **Definition 17.** *Let $R := 1 - c/2 + 2/\ell$. For every $S \in \binom{[n]}{k}$, let $C_{\overline{S}} \subseteq \{0,1\}^{\overline{S}}$ be a fixed arbitrary (ℓ, R, δ)-code with $\delta := H_2^{-1}(1-R)$. The distributions $\mathcal{D}_Y$ and $\mathcal{D}_N$ are defined as follows:*

- *$\mathcal{D}_N$: The uniform distribution over all n-bit Boolean functions.*
- *$\mathcal{D}_Y$: Draw $\boldsymbol{S} \sim \binom{[n]}{k}$. For each $y \in \{0,1\}^{\boldsymbol{S}}$, draw $\boldsymbol{b}_y \sim \{\pm 1\}$. Then for each $z \in \{0,1\}^{\overline{\boldsymbol{S}}}$, let*

$$\boldsymbol{f}(y,z) \begin{cases} := \boldsymbol{b}_y & z \in C_{\overline{\boldsymbol{S}}} \\ \sim \{\pm 1\} & z \notin C_{\overline{\boldsymbol{S}}} \end{cases}.$$

We first show $\mathcal{D}_N$ is mostly supported on functions with correlation at most τ_2 with every k-junta.

▷ Claim 18. Let $\boldsymbol{f} \sim \mathcal{D}_N$. With probability at least $1 - \exp(-2^{k/3})$, $\mathrm{corr}(\boldsymbol{f}, \mathcal{J}_k) \le 2^{-\ell/2} = \tau_2$.

Proof. Fix any $S \in \binom{[n]}{k}$. For any assignment $y \in \{0,1\}^S$ over coordinates in S, define $\boldsymbol{a}_y := 2^{-\ell} \cdot |\sum_{z \in \{0,1\}^{\overline{S}}} \boldsymbol{f}(y,z)|$. By Claim 7, it holds that $\mathrm{corr}(\boldsymbol{f}, \mathcal{J}_S) = 2^{-k} \cdot \sum_{y \in \{0,1\}^S} \boldsymbol{a}_y$.

We first bound the expected value of each individual $\boldsymbol{a}_y$: Since $\{\boldsymbol{f}(y,z)\}_{z\in\{0,1\}^{\overline{S}}}$ are independent Rademacher variables, by Fact 14, we have $\mathbb{E}[\boldsymbol{a}_y] \le \sqrt{2/\pi}\cdot 2^{-\ell/2}$. Then using Hoeffding's inequality, we have $\mathrm{corr}(\boldsymbol{f},\mathcal{J}_S) = 2^{-k}\cdot\sum_{y\in\{0,1\}^S}\boldsymbol{a}_y \ge 2^{-\ell/2} = \tau_2$ with probability $1-\exp(-2^{k/2})$. Finally, by applying union bound over all $S\in\binom{[n]}{k}$, we obtain

$$\Pr[\mathrm{corr}(\boldsymbol{f},\mathcal{J}_k)\ge\tau_2] \le \sum_{S:|S|=k}\Pr[\mathrm{corr}(\boldsymbol{f},\mathcal{J}_S)\ge\tau_2] \le \binom{n}{k}\cdot\exp(-2^{k/2}) \le \exp(-2^{k/3}). \quad\triangleleft$$

Then we demonstrate that $\mathcal{D}_Y$ is mostly supported on functions with correlation at least τ_1 with some k-juntas.

▷ Claim 19. Let $\boldsymbol{f}\sim\mathcal{D}_Y$. With probability at least $1-\exp(-2^{k/2})$, $\mathrm{corr}(\boldsymbol{f},\mathcal{J}_k)\ge\tau_1$.

Proof. Fix any $S\in\binom{[n]}{k}$. Sample a uniform $\boldsymbol{f}\sim\mathcal{D}_Y$ conditioned on $\boldsymbol{S}=S$. We similarly define $\boldsymbol{a}_y \coloneqq 2^{-\ell}\cdot|\sum_{z\in\{0,1\}^{\overline{S}}}\boldsymbol{f}(y,z)|$ for each assignment $y\in\{0,1\}^S$. By Claim 7, $\mathrm{corr}(\boldsymbol{f},\mathcal{J}_S) = 2^{-k}\cdot\sum_{y\in\{0,1\}^S}\boldsymbol{a}_y$. However, for this time, we have

$$\boldsymbol{a}_y = 2^{-\ell}\cdot\left|C_{\overline{S}}|\cdot\boldsymbol{b}_y + \sum_{z\notin C_{\overline{S}}}\boldsymbol{f}(y,z)\right|.$$

Using the fact that $\{\boldsymbol{f}(y,z)\}_{z\notin C_{\overline{S}}}$ are independent, an application of Fact 14 yields

$$\mathbb{E}[\boldsymbol{a}_y] \ge |C_{\overline{S}}|/2^{\ell} - 2^{-\ell/2} \ge 2^{-(1-R)\cdot\ell-1}.$$

Consequently, by Hoeffding's inequality, with probability at least $1-\exp(-2^{k/2})$, we have

$$\mathrm{corr}(\boldsymbol{f},\mathcal{J}_S) = 2^{-k}\cdot\sum_{y\in\{0,1\}^S}\boldsymbol{a}_y \ge 2^{-(1-R)\cdot\ell-2} = 2^{-c\cdot\ell/2} = \tau_2^c = \tau_1,$$

where the second equality follows from the definition of $R\coloneqq 1-c/2+2/\ell$. The desired claim follows by averaging over $\boldsymbol{S}\sim\binom{[n]}{k}$. $\triangleleft$

3.2 Indistinguishability

Next, we show that $\mathcal{D}_Y$ and $\mathcal{D}_N$ are indistinguishable by classical algorithms making a polynomial number of queries.

▶ **Lemma 20.** *Let $\mathcal{D}_Y$ and $\mathcal{D}_N$ be defined as in Definition 17. Then for any classical algorithm $\mathcal{A}^f$ that makes m queries to f,*

$$\left|\Pr_{\boldsymbol{f}\sim\mathcal{D}_Y}[\mathcal{A}^{\boldsymbol{f}}=1] - \Pr_{\boldsymbol{g}\sim\mathcal{D}_N}[\mathcal{A}^{\boldsymbol{g}}=1]\right| \le m^2\cdot k^{-\Omega(\delta\cdot\ell)}. \tag{1}$$

Proof of Lemma 20. Consider $\mathcal{A}^f$ as a depth-m decision tree. Without loss of generality, we can assume that every root-to-leaf path in the tree has length exactly m. Let $\mathcal{P}$ denote an arbitrary path. It is straightforward to observe that $\mathcal{A}^{\boldsymbol{f}}$ follows path $\mathcal{P}$ with probability exactly 2^{-m} when $\boldsymbol{f}$ is drawn from the uniform distribution $\mathcal{D}_N$. Our primary task is to show that when $\boldsymbol{f}\sim\mathcal{D}_Y$, $\mathcal{A}^{\boldsymbol{f}}$ follows $\mathcal{P}$ with probability close to 2^{-m}.

Let $x^{(1)},\dots,x^{(m)}$ denote the points queried on $\mathcal{P}$. We say $S\in\binom{[n]}{k}$ is *good* (w.r.t. $\mathcal{P}$) if for every $1\le i<j\le n$, either $x_S^{(i)}\ne x_S^{(j)}$, or $\Delta(x^{(i)},x^{(j)})<\delta\cdot\ell$. We first show that when S is good, the probability that $A^{\boldsymbol{f}}$ follows $\mathcal{P}$ is exactly 2^{-m} over $\boldsymbol{f}\sim\mathcal{D}_Y$ conditioned on $\boldsymbol{S}=S$.

▷ Claim 21. Let $r^{(1)}, \ldots, r^{(m)} \in \{\pm 1\}$ be m arbitrary signs. Suppose S is good, then

$$\Pr_{\boldsymbol{f} \sim \mathcal{D}_Y | \boldsymbol{S}=S}[\boldsymbol{f}(x^{(i)}) = r^{(i)}, \forall i \in [m]] = 2^{-m}.$$

In particular, $\mathcal{A}^{\boldsymbol{f}}$ follows $\mathcal{P}$ with probability exactly 2^{-m} over $\boldsymbol{f} \sim \mathcal{D}_Y$ conditioned on $\boldsymbol{S} = S$.

Proof. For every $i \in [m]$, we can write $x^{(i)} = (y^{(i)}, z^{(i)})$ where $y^{(i)} \in \{0,1\}^S, z^{(i)} \in \{0,1\}^{\overline{S}}$. Since S is good, for any pair of distinct indices i, j such that $y^{(i)} = y^{(j)}$, we must have $\Delta(z^{(i)}, z^{(j)}) < \delta \cdot \ell$. This implies either $z^{(i)} \notin C_{\overline{S}}$, or $z^{(j)} \notin C_{\overline{S}}$. Moreover, since the signs $(\boldsymbol{b}_y)_{y \in \{0,1\}^S}$ associated with the codewords are independent and uniform, we deduce that $\boldsymbol{f}(y^{(1)}, z^{(1)}), \ldots, \boldsymbol{f}(y^{(m)}, z^{(m)})$ are independent uniform bits, from which the desired statement follows. ◁

Next, we show $\boldsymbol{S}$ is good with high probability over $\boldsymbol{S} \sim \binom{[n]}{k}$.

▷ Claim 22. With probability $1 - m^2 \cdot n^{-\Omega(\delta \cdot \ell)}$ over $\boldsymbol{S} \sim \binom{[n]}{k}$, $\boldsymbol{S}$ is good.

Proof. For any fixed $1 \le i < j \le m$, let $T \coloneqq \{r \in [n] \mid x_r^{(i)} \neq x_r^{(j)}\}$. Observe that whenever $|T| \ge \delta \cdot \ell$,

$$\Pr_{\boldsymbol{S} \sim \binom{n}{k}}[\boldsymbol{S} \cap T = \emptyset] \le \binom{n - \delta \cdot \ell}{k} \Big/ \binom{n}{k} \le (\ell/n)^{\delta \cdot \ell} \le k^{0.01 \delta \cdot \ell},$$

where the second inequality follows from Fact 15, and the last inequality follows from the assumption that $\ell \le k^{0.99}$. Therefore,

$$\Pr_{\boldsymbol{S} \sim \binom{n}{k}}[x_{\boldsymbol{S}}^{(i)} = x_{\boldsymbol{S}}^{(j)} \wedge \Delta(x^{(i)}, x^{(j)}) \ge \delta \cdot \ell] \le n^{-\Omega(\delta \cdot \ell)}.$$

Finally, by union bound over all pairs (i, j),

$$\Pr_{\boldsymbol{S}}[\boldsymbol{S} \text{ is not good}] \le \sum_{1 \le i < j \le m} \Pr_{\boldsymbol{S}}[x_{\boldsymbol{S}}^{(i)} = x_{\boldsymbol{S}}^{(j)} \wedge \Delta(x^{(i)}, x^{(j)}) \ge \delta \cdot \ell] \le m^2 \cdot k^{-\Omega(\delta \cdot \ell)}.$$ ◁

With the above two claims in hand, we are ready to show (1). Let μ_Y, μ_N denote the distributions of the path which $\mathcal{A}^{\boldsymbol{f}}$ follows for $\boldsymbol{f} \sim \mathcal{D}_Y$ and $\boldsymbol{f} \sim \mathcal{D}_N$ respectively. Claims 21 and 22 together imply that for every path $\mathcal{P}$, $\mu_Y(\mathcal{P}) \ge (1 - m^2 \cdot k^{-\Omega(\delta \cdot \ell)}) \cdot 2^{-m}$. Hence, the total variation distance between μ_Y and μ_N is

$$d_{\mathrm{TV}}(\mu_Y, \mu_N) = \sum_{\mathcal{P}} \max(0, \mu_N(\mathcal{P}) - \mu_Y(\mathcal{P})) \le \sum_{\mathcal{P}} \mu_N(\mathcal{P}) \cdot (m^2 \cdot k^{-\Omega(\delta \cdot \ell)}) = m^2 \cdot k^{-\Omega(\delta \cdot \ell)}.$$

It follows that

$$\left| \Pr_{\boldsymbol{f} \sim \mathcal{D}_Y}[\mathcal{A}^{\boldsymbol{f}} = 1] - \Pr_{\boldsymbol{g} \sim \mathcal{D}_N}[\mathcal{A}^{\boldsymbol{g}} = 1] \right| \le d_{\mathrm{TV}}(\mu_Y, \mu_N) \le m^2 \cdot k^{-\Omega(\delta \cdot \ell)}.$$ ◀

3.3 Putting Everything Together

We are now ready to prove Theorem 16.

Proof of Theorem 16. Let $\mathcal{D}_Y$ and $\mathcal{D}_N$ be defined as in Definition 17. Let $\mathcal{D}'_Y$ denote the distribution of $\boldsymbol{f} \sim \mathcal{D}_Y$ conditioned on $\mathrm{corr}(\boldsymbol{f}, \mathcal{J}_k) \geq \tau_1$, and $\mathcal{D}'_N$ denote the distribution of $\boldsymbol{g} \sim \mathcal{D}'_N$ conditioned on $\mathrm{corr}(\boldsymbol{g}, \mathcal{J}_k) \leq \tau_2$. Note that $\mathcal{D}'_Y$ is fully supported on yes instances, and $\mathcal{D}'_N$ is fully supported on no instances. Moreover, Claim 19 implies that $d_{\mathrm{TV}}(\mathcal{D}_Y, \mathcal{D}'_Y) \leq \exp(-2^{k/2})$, and similarly, Claim 18 implies that $d_{\mathrm{TV}}(\mathcal{D}_N, \mathcal{D}'_N) \leq \exp(-2^{k/3})$. Combing with Lemma 20, for any m-query algorithm $\mathcal{A}^f$, it holds that

$$\begin{aligned}
&\left| \Pr_{\boldsymbol{f}\sim\mathcal{D}'_Y}[\mathcal{A}^{\boldsymbol{f}} = 1] - \Pr_{\boldsymbol{g}\sim\mathcal{D}'_N}[\mathcal{A}^{\boldsymbol{g}} = 1] \right| \\
\leq & d_{\mathrm{TV}}(\mathcal{D}'_Y, \mathcal{D}_Y) + \left| \Pr_{\boldsymbol{f}\sim\mathcal{D}_Y}[\mathcal{A}^{\boldsymbol{f}} = 1] - \Pr_{\boldsymbol{g}\sim\mathcal{D}_N}[\mathcal{A}^{\boldsymbol{g}} = 1] \right| + d_{\mathrm{TV}}(\mathcal{D}_N, \mathcal{D}'_N) \\
\leq & 2m^2 \cdot k^{-\Omega(\delta\cdot\ell)}.
\end{aligned}$$

Since

$$\delta \cdot \ell = H_2^{-1}(c/2 - 2/\ell) \cdot \ell \geq \Omega(\frac{c}{\log(2/c)} \log \tau_2) = \Omega(\frac{\log(1/\tau_1)}{\log(2/c)}).$$

By Yao's principle, we conclude that any algorithm must make $k^{\frac{\log(1/\tau_1)}{\log(2/c)}}$ queries. ◀

4 Quantum Tolerant Junta Tester

In this section, we introduce the quantum tolerant junta tester.

▶ **Theorem 23.** *Let $k \in \mathbb{N}$ and $0 < \tau_2 < \tau_1^2 < 1$ be parameters. Then there exists a non-adaptive quantum algorithm with query complexity $\mathrm{poly}(k, 1/\tau)$ which*

- *accepts f with probability at least $2/3$ if f has correlation at least τ_1 to some k-junta;*
- *rejects f with probability at least $2/3$ if f has correlation at most τ_2 to every k-junta.*

The quantum tolerant junta tester described in Theorem 23 is implemented by Algorithm 1. It is essentially an adaptation of the tester from [3], tailored to the following refined trade-off between advantage and influence:

▶ **Lemma 24.** *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a Boolean function and $T \subseteq [n]$ be an arbitrary set. Then*

$$1 - \mathrm{Inf}_{\overline{T}}(f) \leq \mathrm{corr}(f, \mathcal{J}_T) \leq \sqrt{1 - \mathrm{Inf}_{\overline{T}}(f)}.$$

The upper bound was shown in [3] as an intermediate step, and the lower bound was shown in [22].

We reprove this lemma here for completeness.

Proof. Let $f_{\mathrm{avg},T}$ be as given in Claim 7. It holds that $\mathrm{corr}(f, \mathcal{J}_T) = \mathbb{E}_{\boldsymbol{x}}[|f_{\mathrm{avg},T}(\boldsymbol{x})|]$. Moreover,

$$f_{\mathrm{avg},T}(x) = \sum_{S\subseteq[n]} \widehat{f}_S \mathbb{E}_{\boldsymbol{y}\sim\{0,1\}^n}[\chi_S(\boldsymbol{y}) \mid \boldsymbol{y}_S = x_S] = \sum_{S\subseteq T} \widehat{f}_S \chi_S(x).$$

Using Parseval's identity, we have $\mathbb{E}_{\boldsymbol{x}}[f_{\mathrm{avg},T}(\boldsymbol{x})^2] = \sum_{S\subseteq T} \widehat{f}_S^2 = 1 - \mathrm{Inf}_{\overline{T}}(f)$. Finally, since $f_{\mathrm{avg},T}(x) \in [-1, 1]$ for all $x \in \{0,1\}^n$, we get

$$\mathbb{E}_{\boldsymbol{x}}[f_{\mathrm{avg},T}(\boldsymbol{x})^2] \leq \mathbb{E}_{\boldsymbol{x}}[|f_{\mathrm{avg},T}(\boldsymbol{x})|] \leq \sqrt{\mathbb{E}_{\boldsymbol{x}}[f_{\mathrm{avg},T}(\boldsymbol{x})^2]}.$$

This completes the proof. ◀

Algorithm 1 Quantum tolerant junta tester.

Input: Quantum oracle access to f, an integer k, and parameters $\varepsilon_1, \varepsilon_2$ with $(1-2\varepsilon_1)^2 > 1-2\varepsilon_2$

Output: "Yes" or "No"

1: $\varepsilon \leftarrow ((1-2\varepsilon_1)^2 - (1-2\varepsilon_2))/2$
2: $\tau \leftarrow \varepsilon/3$
3: $M \leftarrow \frac{100k\log k \cdot \log(1/\tau)}{\tau^2}$
4: Run FourierSample^f M times to obtain $\boldsymbol{S}^{(1)}, \ldots, \boldsymbol{S}^{(M)} \sim \mathcal{S}_f$.
5: $\boldsymbol{p}_j \leftarrow 0$ for all $j \in [n]$
6: **for** $i \in [M]$ **do**
7: **if** $|\boldsymbol{S}^{(i)}| \le k$ **then**
8: $\boldsymbol{p}_j \leftarrow \boldsymbol{p}_j + 1/M$ for all $j \in \boldsymbol{S}^{(i)}$
9: **end if**
10: **end for**
11: $\widetilde{\boldsymbol{C}} \leftarrow \{j \in [n] : \boldsymbol{p}_j \ge \frac{\tau^2}{2k}\}$
12: $\boldsymbol{q}_R = 0$ for all $R \subseteq \widetilde{\boldsymbol{C}}$ with $|R| \le k$
13: **for** $i \in [M]$ **do**
14: $\boldsymbol{q}_R \leftarrow \boldsymbol{q}_R + 1/M$ for all $\boldsymbol{S}^{(i)} \subseteq R \subseteq \widetilde{\boldsymbol{C}}$ with $|R| \le k$
15: **end for**
16: $\widetilde{\mathbf{corr}} \leftarrow \max_{R \subseteq \widetilde{\boldsymbol{C}} : |R| \le k} \boldsymbol{q}_R$
17: **if** $\widetilde{\mathbf{corr}} > ((1-2\varepsilon_1)^2 + (1-2\varepsilon_2))/2$ **then**
18: **return** "Yes"
19: **else**
20: **return** "No"
21: **end if**

Regarding query complexity, the tester only queries f during the execution of FourierSample^f at line 4. Consequently, it makes $M = \frac{100k\log k \cdot \log(1/\tau)}{\tau^2}$ quantum queries in total, as desired. The remainder of this section is dedicated to proving the correctness of the algorithm.

Before presenting the formal proof, we provide a brief overview of how the algorithm proceeds. First, the algorithm draws M random subsets $\boldsymbol{S}^{(1)}, \ldots, \boldsymbol{S}^{(M)} \sim \mathcal{S}_f$. Then from line 5 to 11, the tester utilizes these samples to compute $\boldsymbol{p}_j$ as an estimate $\mathrm{Inf}_j^{\le k}[f]$, and identifies a set $\widetilde{\boldsymbol{C}}$ of coordinates with high estimated level-k influence.

Next, from line 12 to 15, the tester uses the sampled subsets to compute $\boldsymbol{q}_R$ as an estimate $1 - \mathrm{Inf}_R[f]$ for all $R \subseteq C$ with size $|R| = k$. Finally, the algorithm computes $\widetilde{\mathbf{corr}}$ to approximate $\mathrm{corr}(f, \mathcal{J}_k)$ and compares this value with a certain threshold to determine whether to accept or reject f.

▶ **Lemma 25.** *Fix any $k, \varepsilon_1, \varepsilon_2$ with $(1-2\varepsilon_1)^2 > 1-2\varepsilon_2$. Let $\mathcal{A}^f$ denote Algorithm 1. Then:*
- *for every f with $\mathrm{corr}(f, \mathcal{J}_k) > 1-2\varepsilon_1$, with high probability $\mathcal{A}^f$ outputs "YES"; and*
- *for every f with $\mathrm{corr}(f, \mathcal{J}_k) < 1-2\varepsilon_2$, with high probability $\mathcal{A}^f$ outputs "NO".*

Proof. Fix any Boolean function $f : \{0,1\}^n \to \{\pm 1\}$. Recall that Algorithm 1 computes $\widetilde{\mathbf{corr}}$ as an estimation of $\mathrm{corr}(f, \mathcal{J}_k)$ at line 16. It suffices to show that for any Boolean function f, with high probability $\mathrm{corr}^2(f, \mathcal{J}_k) - \varepsilon < \widetilde{\mathbf{corr}} < \mathrm{corr}(f, \mathcal{J}_k) + \varepsilon$ for $\varepsilon := ((1-2\varepsilon_1)^2 - (1-2\varepsilon_2))/2$.

Let $C := \{j \in [n] \mid \mathrm{Inf}_j^{\le k}[f] \ge \tau^2/k\}$ denote the set of coordinates with level-k influence at least τ^2/k. We show that Algorithm 1 computes a superset $\widetilde{\boldsymbol{C}} \supseteq C$ with high probability at line 11.

▷ Claim 26. It always holds that $|\widetilde{\boldsymbol{C}}| \leq \frac{2k^2}{\tau^2}$. Moreover, with probability at least $1 - k^{-2}$, $C \subseteq \widetilde{\boldsymbol{C}}$.

Proof. The first statement holds because $\sum_{j\in[n]} \boldsymbol{p}_j \leq k$.

For the second statement, observe that for each $j \in [n]$, $\boldsymbol{p}_j$ is the average of M independent random bits with mean $\Pr_{\boldsymbol{S}\sim\mathcal{S}_f}[j \in \boldsymbol{S} \wedge |\boldsymbol{S}| \leq k] = \mathrm{Inf}_j^{\leq k}[f]$. By Chernoff bound, with probability $1 - \exp(\log k \cdot \log \tau)$, we have $\boldsymbol{p}_j \geq \frac{\tau^2}{2k}$. The desired statement then follows from a union bound over all $j \in C$. ◁

Next, we observe that for each $R \subseteq \widetilde{\boldsymbol{C}}$ with size $|R| \leq k$, $\boldsymbol{q}_R$ is the average of M independent random bits with mean $\Pr_{\boldsymbol{S}\sim\mathcal{S}_f}[\boldsymbol{S} \subseteq R] = 1 - \mathrm{Inf}_{\overline{R}}[f]$. By Hoeffding's inequality, with probability at least $1 - \exp(100k \log k \cdot \log \tau)$ we have $|\boldsymbol{q}_R - (1 - \mathrm{Inf}_{\overline{R}}[f])| \leq \tau$. In which case, we can further apply Lemma 24 to obtain

$$\mathrm{corr}^2(f, \mathcal{J}_R) - \tau < \boldsymbol{q}_R < \mathrm{corr}(f, \mathcal{J}_R) + \tau.$$

Then by applying union bound over all $R \subseteq \widetilde{\boldsymbol{C}}$ with size $|R| = k$, with probability at least $1 - \exp(-\Omega(k \log k))$,

$$\max_{R\subseteq\widetilde{\boldsymbol{C}}:|R|\leq k} \mathrm{corr}^2(f, \mathcal{J}_R) - \tau < \widetilde{\mathbf{corr}} < \max_{R\subseteq\widetilde{\boldsymbol{C}}:|R|\leq k} \mathrm{corr}(f, \mathcal{J}_R) + \tau.$$

When $C \subseteq \widetilde{\boldsymbol{C}}$, by Lemma 8, we further have

$$\mathrm{corr}^2(f, \mathcal{J}_k) - \varepsilon = \mathrm{corr}^2(f, \mathcal{J}_k) - 2\tau - \tau < \widetilde{\mathbf{corr}} < \mathrm{corr}(f, \mathcal{J}_k) + \varepsilon.$$ ◀

References

1. Andris Ambainis, Aleksandrs Belovs, Oded Regev, and Ronald de Wolf. Efficient quantum algorithms for (gapped) group testing and junta testing. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 903–922, USA, 2016. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611974331.CH65.
2. Alp Atıcı and Rocco A Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2007. doi:10.1007/s11128-007-0061-6.
3. Zongbo Bao, Yuxuan Liu, Penghui Yao, Zekun Ye, and Jialin Zhang. Efficient non-adaptive quantum algorithms for tolerant junta testing. In *2026 SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 100–126. Society for Industrial and Applied Mathematics, 2026. doi:10.1137/1.9781611978964.9.
4. Aleksandrs Belovs. Quantum algorithm for distribution-free junta testing. In René van Bevern and Gregory Kucherov, editors, *Computer Science - Theory and Applications - 14th International Computer Science Symposium in Russia, CSR 2019, Novosibirsk, Russia, July 1-5, 2019, Proceedings*, volume 11532 of *Lecture Notes in Computer Science*, pages 50–59. Springer, 2019. doi:10.1007/978-3-030-19955-5_5.
5. Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921.
6. Eric Blais. Improved bounds for testing juntas. In Ashish Goel, Klaus Jansen, José D. P. Rolim, and Ronitt Rubinfeld, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 317–330, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-540-85363-3_26.
7. Eric Blais. Testing juntas nearly optimally. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pages 151–158, New York, NY, USA, 2009. Association for Computing Machinery. doi:10.1145/1536414.1536437.

8 Eric Blais, Clément L. Canonne, Talya Eden, Amit Levi, and Dana Ron. Tolerant junta testing and the connection to submodular optimization and function isomorphism. *ACM Trans. Comput. Theory*, 11(4):24:1–24:33, 2019. doi:10.1145/3337789.

9 Guy Blanc, Alexandre Hayderi, Caleb Koch, and Li-Yang Tan. The sample complexity of smooth boosting and the tightness of the hardcore theorem. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1431–1450, 2024. doi:10.1109/FOCS61266.2024.00092.

10 Nader H. Bshouty. Almost Optimal Distribution-Free Junta Testing. In Amir Shpilka, editor, *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:13, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2019.2.

11 Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. *Theory of Computing*, 16(10):1–71, 2020. doi:10.4086/toc.2020.v016a010.

12 Chris Calabro. *The Exponential Complexity of Satisfiability Problems*. PhD thesis, University of California at San Diego, 2009.

13 Thomas Chen, Shivam Nadimpalli, and Henry Yuen. Testing and learning quantum juntas nearly optimally. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 1163–1185. SIAM, 2023. doi:10.1137/1.9781611977554.CH43.

14 Xi Chen, Anindya De, Yuhao Li, Shivam Nadimpalli, and Rocco A. Servedio. Mildly exponential lower bounds on tolerant testers for monotonicity, unateness, and juntas. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4321–4337. Society for Industrial and Applied Mathematics, 2024. doi:10.1137/1.9781611977912.151.

15 Xi Chen and Shyamal Patel. New lower bounds for adaptive tolerant junta testing. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1778–1786, 2023. doi:10.1109/FOCS57990.2023.00108.

16 Xi Chen, Shyamal Patel, and Rocco A. Servedio. A mysterious connection between tolerant junta testing and agnostically learning conjunctions. *arXiv preprint arXiv:2504.16065*, 2025. doi:10.48550/arXiv.2504.16065.

17 Xi Chen, William Pires, Toniann Pitassi, and R. A. Servedio. Testing juntas and junta subclasses with relative error. In Nika Haghtalab and Ankur Moitra, editors, *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 5214–5245. PMLR, 30 June–04 July 2025. URL: https://proceedings.mlr.press/v291/chen25h.html.

18 Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the query complexity of non-adaptive junta testing. *J. ACM*, 65(6), 2018. doi:10.1145/3213772.

19 Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Inf. Process. Lett.*, 90(6):301–305, 2004. doi:10.1016/J.IPL.2004.01.023.

20 Anindya De, Elchanan Mossel, and Joe Neeman. Junta Correlation is Testable . In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1549–1563, Los Alamitos, CA, USA, November 2019. IEEE Computer Society. doi:10.1109/FOCS.2019.00090.

21 Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 563–574, 2006. doi:10.1109/FOCS.2006.51.

22 Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753–787, 2004. Special Issue on FOCS 2002. doi:10.1016/j.jcss.2003.11.004.

23 E. N. Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, 1952. doi:10.1002/j.1538-7305.1952.tb01393.x.

24 Uffe Haagerup. The best constants in the khintchine inequality. *Studia Mathematica*, 70(3):231–283, 1981. doi:10.4064/sm-70-3-231-283.

25 Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM J. Comput.*, 37(4):1107–1138, 2007. doi:10.1137/050645804.

26 Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi:10.2307/2282952.

27 Vishnu Iyer, Avishay Tal, and Michael Whitmeyer. Junta Distance Approximation with Sub-Exponential Queries. In Valentine Kabanets, editor, *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:38, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2021.24.

28 Vinayak M. Kumar. Most juntas saturate the hardcore lemma. In *2026 SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 142–150. Society for Industrial and Applied Mathematics, 2026. doi:10.1137/1.9781611978964.11.

29 Amit Levi and Erik Waingarten. Lower bounds for tolerant junta and unateness testing via rejection sampling of graphs. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, San Diego, California, USA, January 10-12, 2019*, volume 124 of *LIPIcs*, pages 52:1–52:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ITCS.2019.52.

30 Zhengyang Liu, Xi Chen, Rocco A. Servedio, Ying Sheng, and Jinyu Xie. Distribution-free junta testing. *ACM Trans. Algorithms*, 15(1), 2018. doi:10.1145/3264434.

31 Elchanan Mossel, Ryan O'Donnell, and Rocco P. Servedio. Learning juntas. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 206–212, New York, NY, USA, 2003. Association for Computing Machinery. doi:10.1145/780542.780574.

32 Shivam Nadimpalli and Shyamal Patel. Optimal non-adaptive tolerant junta testing via local estimators. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1039–1050, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649687.

33 Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. Approximating the distance to monotonicity of boolean functions. *Random Struct. Algorithms*, 60(2):233–260, 2022. doi:10.1002/RSA.21029.

34 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006. doi:10.1016/j.jcss.2006.03.002.

35 Mert Sağlam. Near log-convexity of measured heat in (discrete) time and consequences. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 967–978, 2018. doi:10.1109/FOCS.2018.00095.

36 Rocco A. Servedio, Li-Yang Tan, and John Wright. Adaptivity Helps for Testing Juntas. In David Zuckerman, editor, *30th Conference on Computational Complexity (CCC 2015)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 264–279, Dagstuhl, Germany, 2015. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2015.264.

37 Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *J. ACM*, 62(2), 2015. doi:10.1145/2728167.

38 Revaz R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akademii Nauk SSSR*, 117(5):739–741, 1957.

Trace Hermitian Codes Have Vanishing Bias

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

Amnon Ta-Shma
Department of Computer Science, Tel Aviv University, Israel

Kedem Yakirevitch
Department of Computer Science, Tel Aviv University, Israel

Abstract

In this work we give the first proof that Trace Hermitian codes have vanishing bias. This brings to the front the question of understanding the distance of Trace AG codes, and the fascinating possibility that in some variation they might give asymptotically good codes.

2012 ACM Subject Classification Theory of computation → Error-correcting codes; Mathematics of computing → Coding theory

Keywords and phrases algebraic geometry codes, character sums, Stepanov method, error-correcting codes

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.35

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/069/`

Funding *Swastik Kopparty*: Research supported by an NSERC Discovery Grant.
Amnon Ta-Shma: Research supported by the Israel Science Foundation (grant number 443/22).
Kedem Yakirevitch: Research supported by the Israel Science Foundation (grant number 443/22) and the Blavatnik fund.

1 Introduction

Error–correcting codes (ECC) are mathematical tools used to detect and correct errors in data transmission and storage. These codes add redundancy to the original message so that even when part of the data is corrupted (because of channel noise, hardware faults, etc.), the original message can still be recovered. The design of efficient ECC lies at the heart of information theory and has profound applications in communication systems, data storage, cryptography, and theoretical computer science.

A central trade-off in the study of ECC is the rate–distance trade-off. The *rate* of a dimension k linear code $C \subseteq \mathbb{F}_q^n$ (an $[n,k]_q$ code) is $r = k/n$, while the *relative distance* δ equals the minimum Hamming distance between distinct codewords divided by n. Determining the best possible trade-offs between r and δ is one of the most fundamental problems in coding theory.

The Gilbert–Varshamov (GV) bound [10, 33] shows that for alphabet size q and relative distance δ there exist codes with

$$r \geq 1 - H_q(\delta),$$

where H_q is the q-ary entropy function. The GV bound guarantees the existence of asymptotically good codes but is non-constructive. A major breakthrough came in the 1980s with Algebraic–Geometry (AG) codes (Tsfasman, Vlăduţ, and Zink [32], building on Goppa). Explicit codes were first given in [8, 9]. AG codes were shown to satisfy

$$r \geq 1 - \delta - \frac{1}{\sqrt{q}-1},$$



© Swastik Kopparty, Amnon Ta-Shma, and Kedem Yakirevitch;
licensed under Creative Commons License CC-BY 4.0



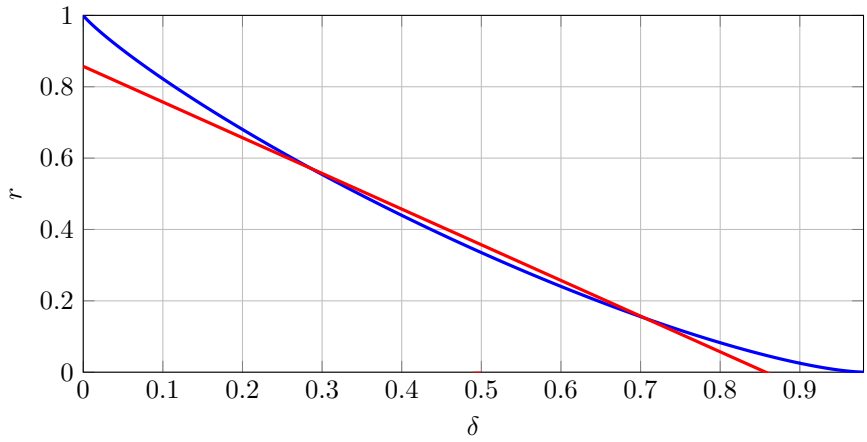


Figure 1 Rate vs distance bounds for $q = 64$. The blue curve is the Gilbert–Varshamov lower bound $r \geq 1 - H_{64}(\delta)$. The red line is the Algebraic–Geometry (AG) lower bound $r \geq 1 - \delta - \frac{1}{\sqrt{q}-1}$ and it is better than the GV bound for some interval $I = [\delta_0, \delta_1]$ of δ, with $\delta_0 \approx 0.328$ and $\delta_1 \approx 0.671$. Notice that rate zero is obtained at $1 - 1/64 = 63/64$, while the red line becomes zero at $\delta = 1 - \frac{1}{\sqrt{q}-1} = 6/7$, so the AG code bound becomes meaningless when δ approaches $1 - 1/q$.

and – remarkably – they beat the GV bound for square alphabets of size $q \geq 49$ by exploiting curves with many rational points relative to their genus. For illustration, the GV bound and the AG codes bound for $q = 64$ are shown in Figure 1.

Upper bounds such as the McEliece–Rodemich–Rumsey–Welch (MRRW) bounds [17] place limits on achievable rates; the MRRW bound is the strongest known asymptotic upper bound. The binary case ($q = 2$) is especially important. The best known lower bound is again the nonconstructive GV bound, while the best upper bound is the MRRW bound. Although AG codes beat GV for sufficiently large alphabets, it remains open whether the GV bound is tight for smaller alphabets and, in particular, for binary codes. Several explicit constructions exist for binary alphabets (for example, the Justesen code [15]), but they are generally far from the GV bound.

A particularly interesting regime is binary codes with very large relative distance,

$$\delta = \frac{1-\varepsilon}{2} \quad \text{for small } \varepsilon > 0.$$

The GV bound guarantees binary codes with such distance and rate $r = \Omega(\varepsilon^2)$, while the MRRW bound shows that any binary code with $\delta = (1-\varepsilon)/2$ must satisfy

$$r = O\Big(\varepsilon^2 \log \frac{1}{\varepsilon}\Big).$$

Many explicit binary constructions start from a good code over a larger alphabet and reduce to binary by concatenation. Two common approaches are:

- Start with an AG code over $\mathbb{F}_q$ (with $q > 2$) and reduce the alphabet by concatenation; a natural inner code is the Hadamard code. Using AG codes obtained from curves above the genus threshold yields

  $$r = \widetilde{\Omega}(\varepsilon^3),$$

  where the $\widetilde{\Omega}$ hides polylogarithmic factors in $1/\varepsilon$. AG codes below the genus lead to other trade-offs but this approach cannot reach rates near the GV bound [3].

- Alon et al. [1] proposed amplifying the distance of an asymptotically good binary code by a random walk on an expander (or any sampler), which increases the alphabet size; the alphabet is then reduced by concatenation with the Hadamard code. That construction achieves $r = \widetilde{\Omega}(\varepsilon^3)$ [1].

We also mention the early constructions of Naor and Naor [19] and Alon, Goldreich, Håstad and Peralta [2], which do not give positive rate but have good dependence on ε.

Concatenation appears to introduce an extra factor of ε in the rate in known analyses. A natural idea is to replace concatenation by a well-chosen deterministic binary function. This idea was explored in two directions. For the expander/amplification approach (not the focus of this paper), Alon and independently Rozenman and Wigderson (see [4]) studied replacing concatenation with parity; a straightforward implementation gives the weaker bound $r = \Omega(\varepsilon^4)$, but combining parity with a wide zig-zag product leads to constructions of rate $r = \Omega(\varepsilon^{2+o(1)})$ [29], see also [13, 6].

This paper focuses on the first approach: replacing concatenation with a deterministic binary-valued function applied symbolwise to an outer AG evaluation code (it is related to an earlier construction from [2]).

Assume $q = 2^\ell$. We would like to take the outer code to be the Reed-Solomon code of evaluations of all polynomials of degree at most d on all points of $\mathbb{F}_q$. To reduce the alphabet size we apply the trace map $\mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2} : \mathbb{F}_q \to \mathbb{F}_2$ coordinate-wise to each outer symbol. This unfortunately may "collapse" codewords of the Reed-Solomon code, since distinct polynomials might have identical traces (e.g., $\mathsf{Tr}(x^2) = \mathsf{Tr}(x)$). Instead we focus on a subcode of Reed-Solomon codes where codewords are guaranteed to stay distinct after taking trace. Namely, we consider the evaluation code spanned by all *odd degree* monomials x^i with $i \le d$. The key technical tool we use is the Weil bound for additive character sums:

▶ **Theorem 1.1** (Weil bound, additive characters). *Let $q = 2^\ell$, and let $f \in \mathbb{F}_q[x]$ be a polynomial of degree d that is not of the form $g(x)^2 - g(x)$. Then the additive character sum satisfies*

$$\Big| \sum_{x \in \mathbb{F}_q} (-1)^{\mathsf{Tr}(f(x))} \Big| \le (d-1)\sqrt{q}.$$

Applying this bound with $d = \deg(f)$ and

$$\varepsilon \approx \frac{d}{\sqrt{q}}$$

gives a binary $[q, k = \Omega(d),\ \delta = \frac{1-\varepsilon}{2}]_2$ code, and the block length satisfies $q = \Omega((\frac{k}{\varepsilon})^2)$. We remark that one of the constructions appearing in [2] uses the quadratic multiplicative character (quadratic residue) together with an extra linearization trick to obtain linear codes; here we use the additive character (trace) to avoid the linearization trick.

The Weil bound is known to be tight in some very special cases (see, e.g., [30][1]). A potentially stronger bound of the form $O(\sqrt{\frac{d}{q}},)$ for a large space of polynomials would immediately give $\varepsilon = O(\sqrt{\frac{d}{q}})$, i.e. $q = O(d/\varepsilon^2)$, and hence rate

$$r = \Omega\Big(\frac{d}{q}\Big) = \Omega(\varepsilon^2),$$

matching the GV bound. This raises natural questions:

[1] We thank Noan Goldgraber for the reference.

- Can this trace-based construction be improved by limiting the set of monomials? Can some adaptation of it match the GV bound? How close can it get to MRRW?
- In the regime where AG codes beat GV – does taking the trace preserve that advantage when reducing to a smaller alphabet?

In this work we give the first proof that trace AG codes have vanishing bias. For concreteness we focus on trace Hermitian codes, and later on we survey what we know about other AG codes, and following work on the problem. Also, for simplicity, the following discussion concerns a multiplicative character of order 2 (namely, the quadratic residue function χ) rather than the trace map.

The evaluation points of the Hermitian code are $H = \{(x, y) \in \mathbb{F}_q^2 \mid \mathsf{Tr}(y) = N(x)\}$ where $q = p^2$, p is a prime power, Tr and N are the trace and norm functions from $\mathbb{F}_q$ to $\mathbb{F}_p$, namely, $\mathsf{Tr}(y) = y^p + y$ and $N(x) = x^{p+1}$. The evaluation functions of the Hermitian code consist of all low-degree bi-variate polynomials. The Hermitian curve has a maximal number of rational points for its genus, and was extensively studied. The question we study is: given a non-square polynomial $p(x, y)$, how large $bias(p) = \frac{1}{|H|} \sum_{(a,b)\in H} \chi(p(a, b))$ can be?

The natural approach for attacking the problem is looking at the function field

$$\mathbb{F}_q(x)[y, z]/\langle \mathsf{Tr}(y) - N(x), z^2 - p(x, y)\rangle$$

and bounding its number of rational points using the Hasse-Weil bound. However, this approach meets an immediate obtacle: the error term in the Hasse-Weil bound is in the order of $g\sqrt{q}$ where g is the genus of the curve, and as a result the Hasse-Weil bound is vacuous when the genus is larger than $\sqrt{q}$ - as is the case with the Hermitian curve.

In [34], Vladut uses properties of zeta functions of curves and extremal properties of the Hermitian curve to prove that $bias(p)$ is smaller than some constant smaller than 1.[2] This was the first non-trivial result on the problem, and the best prior to our work.

In this work we use the Stepanov method (which we explain below) to prove the bias is vanishing (i.e., it goes down to 0 together with p). In Sections 1.1 and 1.2 we explain our approach, in Section 1.3 we present the bounds we get on the bias, and finally in Section 1.5 we conclude with some remarks, open problems and summary of following work.

1.1 The Stepanov method

In 1924 Artin conjectured the "Riemann hypothesis for finite fields". In a landmark result Weil proved in 1948 Artin's conjecture [35]. A corollary of this is the Weil bound for character sums, of which a special (representing) case is:

▶ **Theorem 1.2.** *Let $\mathbb{F}_q$ be a finite field of odd characteristic and $h \in \mathbb{F}_q[X]$ a polynomial of degree d such that h cannot be written as cg^2 with $c \in \mathbb{F}_q$ and $g \in \mathbb{F}_q[X]$. Then*

$$|\sum_{x\in\mathbb{F}_q} \chi(h(x))| \leq (d-1)\sqrt{q}.$$

Thus, the evaluations of h over $\mathbb{F}_q$ have almost the same number of quadratic residues and non-residues with a bias of at most $O(d\sqrt{q})$.

Weil's proof of the Riemann hypothesis for curves over finite fields uses algebraic-geometric arguments. Stepanov, in a series of papers around 1970, gave alternative elementary proofs for special cases of Weil's result. Stepanov's proof uses what we nowadays call the "polynomial

[2] The material in Section 1.4 is related to this approach.

method" [23, 24, 26, 25, 27]. The method was refined (independently) by Bombieri [5] and Schmidt [20, 21] who were able to use it to prove the full "Riemann hypothesis for finite fields". Further work used the method to prove bounds even in cases where algebraic geometry seems unable to provide any nontrivial ones [18, 12].

Very roughly, using the terminology of Theorem 1.2, our objective is the following. Let A denote the set of points $x \in \mathbb{F}_q$ such that $h(x)$ is a quadratic residue. Our goal is to show A is small (and then, proving a corresponding bound for non-residues the theorem follows). The polynomial method approach to this would be to show that there exists a low-degree, non-zero polynomial R vanishing on A with high multiplicity. If we are able to show such an R exists then the number of elements in A is bounded by the degree of R divided by the multiplicity. However, the naive approach of finding R by solving a linear system, where the intermediates are R's coefficients and there is one equation per each element of A, is doomed to fail.

The crux of Stepanov's argument is the observation that A is a *variety*, i.e., it is the zero locus of polynomial equations. Specifically, A is the set of points x such that $x^q = x$ (i.e., $x \in \mathbb{F}_q$) and $(h(x))^{\frac{q-1}{2}} = 1$ (i.e., $h(x)$ is a quadratic residue). Stepanov chooses R with many degrees of freedom, such that when R is restricted to the variety it is simplified to the zero polynomial. Thus, the vanishing of R on A is obtained not by a set of local constraints (one constraint per each element of A) but rather globally, by forcing R to be the zero polynomial when taken modulo the equations that define the variety.

The same approach also works to count the number of $\mathbb{F}_q$-points on more general algebraic-geometric curves (generalizing the result above, which is about the curve $y^2 = h(x)$), and, as we mentioned before, Bombieri [5] and Schmidt [20, 21] used Stepanov's method to prove the Riemann hypothesis for finite fields, which shows that the number of rational points on any small-genus curve over $\mathbb{F}_q$ is close to q: the error term is $O(g\sqrt{q})$.

These same results also automatically handle character sums of polynomials over small genus curves. Let C be an algebraic-geometric curve over $\mathbb{F}_q$ given by the equation $R(x, y) = 0$, and let g be its genus. Let $h(x, y) \in \mathbb{F}_q[x, y]$ be a low degree polynomial, viewed as an algebraic function on C. Let χ be the quadratic residue character of $\mathbb{F}_q^*$. If we want to bound:

$$|\sum_{(x,y)\in C(\mathbb{F}_q)} \chi(h(x, y))|,$$

which measures how frequently h evaluates to a quadratic residue on C, it suffices to show that the number of points on the curve C_h embedded in 3 dimensional space given by the equations:

$$R(x, y) = 0,$$

$$z^2 = h(x, y),$$

is close to q. Thus we again are trying to estimate the number of $\mathbb{F}_q$-points on a curve C_h. The Riemann hypothesis over finite fields again applies; the key fact that is needed is the Riemann-Hurwitz formula, which shows that C_h also has small genus ($\approx 2g$).

For C of high genus, the above method does not work, and the Hasse-Weil bound is usually meaningless. Indeed, Vladut [34] notes the lack of results for high-genus curves. He then studied these kinds of character sums over the high genus Hermitian curve, and obtained some results using general properties of the zeta functions of curves and the fact that the Hermitian curve has the maximum possible number of points for its genus.

Instead, we develop a version of the Stepanov method which works directly with the Hermitian curve function field, generalizing the original Stepanov method that works with the rational function field $\mathbb{F}_q(X)$. The main difficulty that we face, which stems from the fact

that the Hermitian curve has high genus, is that derivatives (that are needed for vanishing with high multiplicity) carry a penalty in high-genus curves: the derivative of a low degree function on a curve can be of significantly higher degree.

To handle this, we develop the *universal derivative-fix lemma* that we explain next.

1.2 The universal derivative-fix lemma

Our proofs use the Stepanov method, but instead of applying it over the polynomial ring, we apply it over algebraic curves of high genus. For that we need to understand high-order derivatives over algebraic curves, which we discuss first. This section requires some necessary mathematical background and notation that is given in Sections 2.1 and 2.2.

We start by considering two examples:

The polynomial ring: In the polynomial ring $\mathbb{F}_q[X]$, the derivative of a non-constant polynomial is a polynomial of a strictly smaller degree, and the more times we derive, the smaller the degree gets until we reach the zero polynomial. This gives the impression that derivatives are simpler, i.e. have less poles than the original functions.
When transitioning to rational functions, this is no longer the case. For example, when m is smaller then the characteristic of $\mathbb{F}_q$, $D_x^m(\frac{1}{x}) = \frac{c_m}{x^{m+1}}$ for some non-zero constant c_m.[3] Now, the more we derive the more poles we get, and each derivation increases the pole order by one. Similarly, if we look at $D_x^m(\frac{f}{g})$ we get some polynomial in the derivatives of f and g divided by g^{m+1}, meaning the poles at the zeroes of g increase many-fold as we derive. Thus, both in the case of $\frac{1}{x}$ and in the more general case of $\frac{f}{g}$, the poles "stay where they were", but the pole order increases.
Now consider derivatives of the form $D_g^1(f)$ where $f, g \in \mathbb{F}_q[X]$. From the chain rule $D_g^1(f) = \frac{df}{dg} = \frac{df}{dx}\frac{dx}{dg} = \frac{f'}{g'}$ we see that $D_g^1(f)$ may have poles also where g' has zeroes, and the more zeroes g' has, the more new poles we introduce when deriving. Thus, it greatly matters with respect to which function g we choose to derive.

The Hermitian function field: Now we look at $H = \mathbb{F}_{p^2}(x, y) \mod y^p + y - x^{p+1}$. The elements x and y are regular, i.e., they only have poles at a single degree one place, which we denote P_∞. It holds that $v_{P_\infty}(x) = -p$ and $v_{P_\infty}(y) = -(p+1)$. Now, $x^p = D_x(x^{p+1}) = D_x(y^p + y) = D_x(y^p) + D_x(y) = D_x(y)$, and so, $D_x(y) = x^p$ has p^2 poles at P_∞ while y has only $p+1$ poles at P_∞, an increase of $p^2 - p - 1 = 2 \cdot \text{genus}(H) - 1$.

We now give a theorem that applies generally to all function fields. We call it *the universal derivative-fix lemma*, because it fixes all new poles created by the derivation operator, and does so by multiplying by a universal element, that is independent of the function that we derive.

For a divisor D we let $(D)_0$ denote the zero-divisor of D, and $(D)_\infty$ the pole divisor of D, so that $D = (D)_0 - (D)_\infty$. We let supp be the size of the support of the pole divisor of x, i.e., the degree of the pole divisor of x when all positive coefficients are reduced to one. In Section 3 we prove:

▶ **Theorem 1.3.** *Let F/K be a function field of genus g and characteristic p. Let $x \in F$ be a separating element of F/K and P_∞ a degree one place of F. Let*

$$G = 3g - 2 + \deg(x) + \mathsf{supp},$$

[3] D_x denotes derivation by x. D_x^m is iterating D_x m times. H_x^m is the m'th Hasse derivative. For background on iterated and Hasse derivatives, and for the notation we use, see Section 2.1.

and denote

$$W = G - \max\{v_\infty(dx), 0\},$$
$$\Delta = G + \min\{v_\infty(dx), 0\}.$$

Then there exists an element

$$0 \neq \omega = \omega(x, P_\infty) \in \mathcal{L}(GP_\infty - (dx)_0) \subseteq \mathcal{L}(WP_\infty)$$

such that for every $A \geq 0$ *and every* $f \in \mathcal{L}(AP_\infty)$ *it holds that*

$$\omega \cdot H_x(f) \in \mathcal{L}((A + \Delta + 1)P_\infty).$$

We stress that ω does not depend on A or f, and depends only on x (and the place P_∞).

Theorem 1.3 can be generalized to any derivation order m. Let $H_x^m(f)$ denote the m-th Hasse derivative of f with respect to x. Then,

▶ **Theorem 1.4.** *Keeping the notation as in Theorem 1.3, there exists an element* $0 \neq \omega = \omega(x, P_\infty) \in \mathcal{L}(GP_\infty - (dx)_0) \subseteq \mathcal{L}(WP_\infty)$ *such that for every positive integer* $m < p$ *(or any positive integer* m*, if* $p = 0$*) and every* $f \in \mathcal{L}(AP_\infty)$*,*

$$\omega^{2m-1} \cdot H_x^m(f) \in \mathcal{L}(A_m P_\infty).$$

where $A_m = A - W + m \cdot (\Delta + W + 1)$.

As an example, consider the rational function field $F = K(x)$. Then $g = 0$,$(dx) = -2P_\infty$ and therefore $G = W = 0$ and $\Delta = -2$. Then $w \in \mathcal{L}(0)$ and for any $f \in \mathcal{L}(AP_\infty)$, $H_x(f) \in \mathcal{L}((A-1)P_\infty)$ which is indeed tight.

Next, we examine the Hermitian function field.

- First consider the Hermitian function field when we derive by x. $D_x(y) = x^p$. In fact, $(dx) = (2g-2)P_\infty$, i.e., it has no poles, and all its zeroes are at P_∞. Then, we can take $w = 1$. Furthermore, for every $f \in \mathcal{L}(AP_\infty)$, $-v_\infty(H_x(f)) = v_\infty(dx) - v_\infty(df) \leq 2g - 2 + A + 1$. For a general m, $A_m \leq A + m(2g-1)$.
- Next consider the Hermitian function field when we derive by y. Then, $D_y(x) = \frac{1}{x^p}$ and $(dy) = (p+2)P_\infty - p(x)_0$. Nevertheless, since all functions in $\mathcal{L}(P_\infty)$ are polynomials in x and y, we get that if we are deriving with respect to y we can choose w to be x^p to cancel out the $\frac{1}{x^p}$ which is the derivative of x with respect to y. With this choice of ω we again get that if $f \in \mathcal{L}(AP_\infty)$ then $\omega^m H^m(f) \in \mathcal{L}((A + m(2g-1))P_\infty)$.

In both cases the bounds that we get are worse, and there are several reasons for that:

- We paid an additive g to guarantee a certain Riemann-Roch space is nonempty, by forcing the degree of its divisor to be at least g. While there are divisors of degree $g-1$ which have empty Riemann-Roch spaces, there are divisors of degree 0 which have non-empty Riemann-Roch spaces. It is conceivably possible that the $3g-1$ we have is not mandatory and can be replaced with $2g-1$ as we have in the Hermitian curve. Perhaps, using the Riemann-Roch theorem with canonical divisors would do the trick.
- Additionally, the $2m$ factor is a side effect of the inductive argument which requires us to apply the induction hypothesis twice - once for $H^{m-1}(f)$ and once for $H^1\omega$. If, however, $H^1\omega$ is regular, we can apply the induction hypothesis once and so ω^m would be sufficient. Alternatively, if the poles of $D^m(f)$ which exceed those of $D^{m-1}(f)$ behave like "dividing by a function again and again", similarly to what we saw with $D_x^m(\frac{f}{g})$ in $K(x)$ or to $D_y(f)$ for regular f in the Hermitian function field, we would again get that ω^m is sufficient.

- The requirement $m < p$ is also a side effect of the induction, but when looking at the p-th Hasse derivative of y^p we get from Claim 2.4 $H_x^p(y^p) = H_x^1(y)^p = x^{p^2}$ which is of pole order $p^3 = p(p+1) + (2g-1)p$, an increase of exactly $2g-1$ times the order of the derivative.

To summarize this, we believe that the following version of Theorem 1.4 could hold:

▶ **Conjecture 1.5.** *Let F/K be a function field of genus g. Let $x \in F$ be a separating element of F/K. Let P_∞ be a degree one place of F. There exists an element $0 \neq \omega = w(x, P_\infty) \in F$ such that for every $m \in \mathbb{N}$ and every $f \in \mathcal{L}(P_\infty)$, $\omega^m \cdot H_x^m(f) \in \mathcal{L}(P_\infty)$. Furthermore, $\omega \in \mathcal{L}((2g-1+\deg(x)+\mathsf{supp})P_\infty)$ and so if $f \in \mathcal{L}(AP_\infty)$ then $\omega^m \cdot H_x^m(f) \in \mathcal{L}(A_m P_\infty)$ for $A_m = A + m(2g-1+\deg(x)+\mathsf{supp}+\min\{v_\infty(dx), 0\})$.*

1.3 Our results

We start with multiplicative characters. In Section 4.1 we give, as a warm-up, a version of Stepanov's proof of the Weil bound. In this proof we work in the function field

$$\widehat{F} = \mathbb{F}_q(x)[z] \pmod{z^2 - h(x)}, \tag{1}$$

and we use the universal derivative-fix lemma (for the function field $\widehat{F}$). We prove:

▶ **Theorem 1.6.** *Let $\mathbb{F}_q$ be a finite field of odd characteristic and $h \in \mathbb{F}_q[X]$ a square-free polynomial of odd degree d. Then*

$$\left| \sum_{x \in \mathbb{F}_q} \chi(h(x)) \right| \leq O(d\sqrt{q}).$$

This theorem is weaker then the Weil bound in two aspects. First, the error is bounded by $O(d\sqrt{q})$ instead of the tight bound of $(d-1)\sqrt{q}$ in Weil's bound. Also, we prove the claim only for square free polynomials h of odd degree, while Weil's bound holds for any h that is not a square over the algebraic closure of $\mathbb{F}_q$. On the bright side, the proof is quite generic.

After this warm-up exercise we give in Section 4.2 a corresponding proof for the Hermitian function field, i.e., p is a prime power, $q = p^2$,

$$F = \mathbb{F}_q(x) \tag{2}$$

is the rational function field, and,

$$H = \mathbb{F}_q(x, y) \mod \varphi(x, y), \tag{3}$$

where $\varphi(x, y) = y^p + y - x^{p+1}$, is the Hermitian function field. Let $\mathcal{C}$ denote all the $\mathbb{F}_q$-rational points on the curve φ. We prove

▶ **Theorem 1.7.** *Suppose p is prime and $h = \sum_{k,\ell} c_{k,\ell} x^k y^\ell \in \mathbb{F}_q[x, y]$ is a polynomial with total degree $d < \sqrt{p}$, and $(p, p+1)$-weighted degree w. The weighted degree w is obtained by a unique monomial $x^k y^\ell$. Suppose $d = k + \ell$ and $w = pk + (p+1)\ell$ and further assume d is odd. Then*

$$\left| \sum_{(x,y) \in \mathcal{C}} \chi(h(x, y)) \right| \leq O(\sqrt{d} \cdot p^{2.5}).$$

As before, we use the universal derivative-fix lemma (this time for the function field $\widehat{H} = H(z) \pmod{z^2 - h(x)}$), and as before, a drawback of the proof is that it requires additional assumptions on the polynomial h beyond assuming h is not a square polynomial. On the bright side the proof is quite generic, e.g., it also applies for the first levels of the Hermitian tower (we omit the specific claims because the parameters become more involved).

Next, we look for a character sum bound that holds for any non-square polynomial, rather than more specific polynomials. Schmidt in [22, Chapter I, section 5] showed how to use the Stepanov method to prove the Weil bound for any non-square polynomial. Schmidt works over the rational function field $\mathbb{F}_q(X)/\mathbb{F}_q$. We extend the argument to work over the Hermitian function field. The proof we give is specific for the Hermitian function field and delicate. In essence, we still follow the generic proof, which uses the Riemann-Roch theorem and the derivative-fix Lemma as black-box theorems, but we replace the black-box components with specific functions, tailored for the Hermitian function field. Consequently, in Section 5 we prove

▶ **Theorem 1.8.** *Suppose p is a prime power, $q = p^2$, $h \in \mathbb{F}_q[x, y]$, $\deg(h) = d$. If $Z^2 - h$ is absolutely irreducible then*

$$\left| \sum_{(x,y)\in\mathcal{C}} \chi(h(x,y)) \right| = O(d \cdot p^{2.5}).$$

Thus the relative bias (i.e., bias divided by the number of evaluation points) is $O(\frac{d}{\sqrt{p}})$. This is worse than the bias $O(\sqrt{\frac{d}{p}})$ in Theorem 1.7, but it holds for any non-square polynomial.

Finally, in Section 6 we give a corresponding bound for additive characters. Here, the Hasse derivatives have a very explicit structure, and we can explicitly point out the derivative fix element w guaranteed by Theorem 1.3. We prove:

▶ **Theorem 1.9.** *Suppose $h = \sum_{k,\ell} c_{k,\ell} x^k y^\ell \in \mathbb{F}_q[x, y]$ is a polynomial with total degree $d < \sqrt{p}$, and $(p, p+1)$-weighted degree w. Suppose w is odd. Then*

$$\left| \sum_{(x,y)\in\mathcal{C}} (-1)^{\mathsf{Tr}(h(x,y))} \right| \leq O(d \cdot p^{2.5}).$$

1.4 Pseudorandomness of the roots of the zeta function

Finally, we make a comment about what our results mean from the point of view of the zeta function of the relevant curves. At the very high level, it shows a further pseudorandomness property of the roots of these zeta functions.

For a curve C of genus g_C over a finite field $\mathbb{F}_q$, many interesting algebraic properties of the curve are captured by its "zeta function" $Z_C(T) \in \mathbb{C}(T)$, which is a rational function of the form $\frac{P_C(T)}{(1-T)(1-qT)}$. $P_C(T)$ is always of the form

$$P_C(T) = \prod_{i=1}^{2g_C} (1 - \omega_i T),$$

where each $\omega_i \in \mathbb{C}$ has $|\omega_i| = \sqrt{q}$. (These are called the reciprocal roots of $P_C(T)$). These ω_i have the property that the number of $\mathbb{F}_q$-points on C equals

$$q + 1 - \sum_i \omega_i.$$

This is a more refined version of the classical the Weil bound, which states that the number of $\mathbb{F}_q$-points on C is $q + 1 + O(g\sqrt{q})$.

Our result on the quadratic residue character sums on the Hermitian curve H are essentially an estimate on the number of points on some curve H'. H' is a degree 2 cover of the Hermitian curve H. We now translate our results in terms of the roots of zeta functions.

By general properties of zeta functions, we have that $P_H(T)$ divides $P_{H'}(T)$.

Let $b = g_{H'} - g_H$. Let $\omega_1, \ldots, \omega_{2g_H}$ be the reciprocal roots of $P_H(T)$, and let $\alpha_1, \ldots, \alpha_{2b}$ be the remaining reciprocal roots of $P_{H'}(T)$ (other than the ω_i). By the Riemann-Hurwitz formula, $g_{H'} \approx 2g_H$, and thus $b \approx g_H \approx q/2$.

The number of $\mathbb{F}_q$ points on H equals $q^{3/2} + 1$, and also equals

$$q + 1 - \sum_{i=1}^{2g_H} \omega_i.$$

By our results, the number of $\mathbb{F}_q$ points on H' is $q^{3/2} + 1 \pm o(q^{3/2})$, and also equals:

$$q + 1 - \sum_{i=1}^{2g_H} \omega_i - \sum_{i=1}^{2b} \alpha_i = q^{3/2} + 1 - \sum_{i=1}^{2b} \alpha_i.$$

Thus,

$$|\sum_{i=1}^{2b} \alpha_i| = o(q^{3/2}).$$

On the other hand, each α_i has magnitude $\sqrt{q}$, and b has magnitude $\Theta(q)$. This means that there is significant cancellation in the sum $\sum_{i=1}^{2b} \alpha_i$. We find this to be quite special. In general one cannot guarantee cancellation in the sum of the reciprocal roots of a zeta function. Our result thus finds a special instance where there is cancellation.

1.5 Conclusions and following work

A natural approach for getting explicit codes close to the GV bound is to start with such a code over a constant alphabet, and, somehow reduce the alphabet size. Concatenation is a natural candidate, but it seems it is not strong enough to approach the GV bound. Trace codes are also natural candidates, but the results obtained so far are poor. A natural question is whether these results can be improved:

- Is there a natural subclass of polynomials for which the code described before Theorem 1.1 is close to the GV bound?
- If we work over the Hermitian function field, and our set of evaluation functions also allows polynomials in x or y alone, then the relative bias of the Trace Hermitian code cannot be better then the relative bias of the Trace RS code. To see that suppose f is a non-square polynomial in $\mathbb{F}_q[x]$ of degree d that has bias e over $\mathbb{F}_q$. As every $x \in \mathbb{F}_q$ has p rational solutions in y that do not affect f, we get a bias of ep over the Hermitian curve, which has the same relative bias. Using Weil bound the bias we know for the Trace RS code is $O(d\sqrt{q})$. Lifting this to the Hermitian curve (by taking a polynomial that only depends on one variable) we get an absolute bias of $\Omega(dq)$. For some minor restriction on the evaluation polynomial we can bound the bias of Trace Hermitian curve by $O(\sqrt{dpq})$, which is worse (as $d < \sqrt{p}$). Is it possible to improve the bound we get for the Trace Hermitian curve to $O(dq)$?
- Finally, is it possible to improve the bound if we restrict to a subclass of monomials, e.g., monomials with odd degree both in x and y?

We believe the techniques we develop are of independent interest. Our results on Trace AG codes, in the interesting case of high genus curves, greatly improve prior work. We also believe there is much more to learn and understand about the surprising behavior of derivatives in function fields.

Our techniques (and the derivative-fix-lemma in particular) allow us to study not only the trace of the Hermitian curve, but also the trace of the codes in the first few levels of the Hermitian tower. We do not state these results because technically they get more involved. Recently, Gil Cohen, Dean Doron, Noam Goldgraber and Tomer Manket [7] generalized the techniques in our paper to general AG curves. At a very high level, the techniques of this paper and of [7] are related in the same way as Stepanov's method and Bombieri's version of Stepanov's method are related:

1. Their proof works in a more general setting of an arbitrary algebraic curve, and directly works with Riemann-Roch spaces of algebraic functions on the curve, while our proof works concretely with the equation of the Hermitian curve (and closely related curves), and represents algebraic functions on the curve more explicitly as polynomials.
2. Unlike our proof, the proof in [7] does not use multiplicities, and instead relies on Bombieri's "raising to the power $\sqrt{q}$" trick.

We are hopeful that these developments would trigger further work on this fascinating question.

2 Some mathematical background

2.1 Hasse derivatives

Let F/K be a function field. $z \in F$ is called *separating* if $F/K(z)$ is a finite separable extension. If $v_P(z) \neq 0$ for some place P then z is separating. Furthermore, if K has characteristic p, then z is separating iff z is not a p power of some element in F [28, Proposition 3.10.2].

Let $z \in F$ be separating. The m-th Hasse derivative with respect to z, denoted by H_z^m, is defined on $K[z]$ by the K-linear extension of $H_z^m(z^n) \triangleq \binom{n}{m} z^{n-m}$ to all of $K[z]$. H_z^m can be uniquely extended to all of F/K so that: (1) $H_z^0 = id_F$, (2), H_z^m vanish on K for all $m > 0$, (3) $H_z^m(fg) = \sum_{j=0}^{m} H_z^j(f) H_z^{m-j}(g)$ (Product Rule) and $H_z^m \circ H_z^n = \binom{m+n}{m} H_z^{m+n}$ (Composition Rule). These uniquely determined extensions are called the *Hasse derivatives*.

The following claim, which is an immediate corollary of [11, Corollary 2.5.14 (Taylor's Theorem)], shows Hasse derivatives capture multiplicity:

▷ Claim 2.1. Let P be a place in F/K and let t be a separating element with $v_P(t) = 1$. Let $f \in F$ and $M \in \mathbb{N} \setminus \{0\}$, then $v_P(f) \geq M \iff \forall m < M \quad (H_t^m(f))(P) = 0$.

A simple fact is

▶ **Fact 2.2** (Simple change of variable). *Let z be a separating element in F/K, let $f \in F$ and $\alpha \in K$, then for every $m > 0$, $H_z^m(f) = H_{z-\alpha}^m(f)$.*

The Hasse chain rule is:

▶ **Fact 2.3.** *Suppose $f(t) \in K((t))$. Suppose $t = t(y) \in K((y))$. Let $m \geq 0$. Then*

$$H_y^{(m)}(f(t)) = H_y^{(m)}(f \circ t(y)) = \sum_{j=1}^{m} H_t^{(j)}(f(t)) \cdot \sum_{\substack{i_1,\ldots,i_m \geq 0 \\ i_1+\ldots+i_m=j \\ i_1+2i_2+3i_3+\ldots+mi_m=m}} \binom{j}{i_1,\ldots,i_m} \cdot (H_y^{(1)}t)^{i_1} \cdot \ldots \cdot (H_y^{(m)}t)^{i_m}$$

The following lemma tells us how Hasse derivatives behave on p-th powers:

▷ Claim 2.4 ([31, Remark 2.4 and Remark 2.5], [14, Theorem 3.1]). Let $z \in F/K$ be a separating element of a function field of characteristic $p > 0$. Let $q = p^k$ be a power of p and $f \in F$. If q divides m then $H_z^m(f^q) = (H_z^{m/q}(f))^q$, and otherwise $H_z^m(f^q) = 0$.

Furthermore, the following conditions are equivalent,

1. There exists some $g \in F$ such that $f = g^q$,
2. $H_z^m(f) = 0$ for all $m = 1, ..., q-1$,
3. $H_z^1(f) = H_z^p(f) = H_z^{p^2}(f) = ... = H_z^{p^{k-1}}(f) = 0$.

▶ **Example 2.5.** We look at $y^m H_y^{(m)}(x^i)$ in the Hermitian function field.

- We start with the $i = 1$ case. On the one hand $H_y^{(\ell)} y^{p+1} = \binom{p+1}{\ell} y^{p+1-\ell}$ and on the other hand, $H_y^{(\ell)}(y^{p+1}) = H_y^{(\ell)}(x^p + x) = H_y^{(\ell)}(x)$ (as by Claim 2.4, $H_y^{(\ell)} x^p = 0$). Hence $y^\ell H_y^{(\ell)}(x) = \binom{p+1}{\ell} y^{p+1}$.
- Next we use Fact 2.3 (with $t = x$ and $f(t) = x^i$) to understand general i. For every $1 \le j \le \min\{m, i\}$, $H_x^{(j)}(x^i) \in \mathsf{Span}\{x^{i-j}\}$, $y^\ell H_y^{(\ell)} x \in \mathsf{Span}\{y^{p+1}\}$, and for every $i_1 + \ldots + i_m = j$ such that $i_1 + 2i_2 + \ldots m i_m = m$, we have $(yH_y^{(1)}x)^{i_1} \cdot (y^2 H_y^{(2)}x)^{i_2} \cdot \ldots \cdot (y^m H_y^{(m)}x)^{i_m} \in \mathsf{Span}\{y^{j(p+1)}\}$. Altogether $y^m H_y^{(m)}(x^i) \in \mathsf{Span}\{x^{\ell_1} y^{\ell_2} y^{p\ell_3} \mid \ell_1 + \ell_2 \le i, \ell_3 \le \min\{m, i\}\}$.

The following corollary will be useful to us later on:

▶ **Corollary 2.6.** *Let $z \in F/K$ be a separating element of a function field of characteristic $p > 0$. Let q be a power of p, $m < q$ and $f, g \in F$, then $H_z^m(fg^q) = H_z^m(f)g^q$.*

Proof. Since $m < q$, all of the derivatives of g^q that will appear in the expansion of $H_z^m(fg^q)$ by means of the product rule are zero (from Claim 2.4) except for the term $H_z^m(f)g^q$. ◀

2.2 Differentials

We follow the presentation at [28, Chapter 4]. Let F/K be a function field. A *derivation* is a K-linear map D from F to some F-module that upholds the product rule of derivatives, i.e. $D(fg) = fD(g) + gD(f)$. For example $H_z^1 : F \to F$ is a derivation for every separating $z \in F$. One can define an F-module Δ_F, such that all derivations of F factor through Δ_F via a canonical mapping $d : F \to \Delta_F$, i.e., if $\delta : F \to M$ is a derivation of F into some F-module M, then there exists a unique F-linear map $\mu : \Delta_F \to M$ such that $\delta = \mu \circ d$. It turns out that d is itself a derivation. For $x \in F$, $d(x)$ is called the differential associated with x and is denoted dx. The set Δ_F of differentials of F contains all elements udx where $u \in F$ and x is separating, where this set is taken modulo the equivalence relation $udx = vdy$ if and only if $\frac{u}{v} = D_x^1(y)$. With this we get a notion of division of differentials via $\frac{udy}{vdx} = \frac{u}{v} D_x^1(y)$.

We now define the notion of valuation of differentials. If $P \in \mathbb{P}_F$ is a place of F and $udx \in \Delta_F$ a differential of F, we define $v_P(udx)$ as follows. We pick a local parameter t of P (i.e., $v_P(t) = 1$) and we find $b \in F$ such that $udx = bdt$. Then $v_P(udx) = v_P(b)$. One can show that this definition is independent of the specific choice of local parameter t. As differentials have valuations, differentials also have zeroes and poles, i.e., places where the valuation is strictly positive or strictly negative, and this can be encoded in a divisor, denoted (udx) and called the divisor associated with the differential udx. It turns out any differential $udx \in \Delta_F$ has only finitely many zeroes or poles and so the associated divisor is indeed well defined. A divisor which is associated to some differential in Δ_F is called a *canonical divisor*. All canonical divisors have degree $2g - 2$ and their Riemann-Roch space has dimension g where $g = genus(F)$. The following claims will be useful:

▷ Claim 2.7. Let $u \in F$, $x, y \in F$ separating, then:

- $(udx) = (u) + (dx)$
- $(D^1_x(y)) = (\frac{dy}{dx}) = (dy) - (dx)$

The first bullet is a restatement of [28, Proposition 1.5.13] and the second one is an immediate consequence of the first bullet and the equality $1dy = D^1_x(y)dx$.

There is a close relationship between the zeroes and poles of f and the zeroes and poles of df. The following claim is stated in [16, Chapter I (6)] for the case where K is algebraically closed, but by considering a constant-field extension of F one can verify it holds exactly as stated for any perfect base field K and even when P is not a place of degree one (which is not a consideration when K is algebraically closed). We give the proof for completeness.

▷ Claim 2.8. Let $f \in F/K$, df its associated differential, then:

- For every place P, $v_P(df) \geq v_P(f) - 1$. In particular, If f has zeroes at P, df can lose at most one zero at P. Also, df can have at most one more pole at P. Also,
- If $v_P(f) \geq 0$ then $v_P(df) \geq 0$, i.e., df can have poles only at places where f has poles.

Proof of Claim 2.8. Let $f \in F/K$, $P \in \mathbb{P}_F$, $t \in F$ with $v_P(t) = 1$. We are interested in $(df)^F$ the divisor of df over F. We move to $\bar{F}/\bar{K} = \bar{K} \cdot F/K = \bar{K}F/\bar{K}$ which is the constant field extension of F/K with all of $\bar{K}$, the algebraic closure of K. Let $\bar{P} \in \mathbb{P}_{\bar{F}}$ be a place lying over P. Since $\bar{K}$ is algebraically closed we know $\deg(\bar{P}) = 1$. From [28, Theorem 3.6.3] we learn that $v_{\bar{P}}(t) = v_P(t) = 1$ and we can write:

$$f = \sum_{n=n_0 \in \mathbb{Z}}^{\infty} c_n t^n \quad (c_n \in \bar{k}, c_{n_0} \neq 0)$$

$$D^1_t(f) = \sum_{n=n_0 \in \mathbb{Z}}^{\infty} n \cdot c_n t^{n-1} \quad (c_n \in \bar{k}, c_{n_0} \neq 0)$$

From the definition of valuation for differential we know that

$$v_{\bar{P}}(1df) = v_{\bar{P}}(D^1_t(f)dt) = v_{\bar{P}}(D^1_t(f)) \geq n_0 - 1$$

From [28, Theorem 3.6.3] we know that $v_{\bar{P}}(1df) = v_P(1df)$ and $v_P(f) = v_{\bar{P}}(f)$ and so we get $v_P(1df) \geq v_P(f) - 1$. Furthermore, if f has no pole at P, then f has no pole at $\bar{P}$ and so $n_0 \geq 0$ and so $D^1_t(f)$ has no pole at $\bar{P}$ and therefore at P, meaning $v_P(1df) \geq 0$. ◁

We mentioned earlier that $\deg(df) = 2g - 2$ for canonical divisors (df), and therefore when g is large df has many more zeroes than f. We also know that df has a zero wherever f has a zero of multiplicity at least 2. Finding the other zeroes of df is a bit more complicated. It turns out that:

▷ Claim 2.9 ([28, Sections 3.4 and 3.5]). The zeroes of df are either at places that are zeroes of f, or, at places of F/K that are ramified when F is viewed as an extension of $K(f)/K$.

2.3 Kummer extensions

An algebraic function field F'/K' is a *Kummer extension* of F/K if K contains a primitive n-th root of unity,[4] and $F' = F(Z) \mod (Z^n - u)$ where $u \in F$ and $u \neq w^d$ for all $w \in F$ and $d|n$ such that $d > 1$. Kummer extensions are Galois. For $P \in \mathbb{P}_F$ we denote by

[4] When $K = \mathbb{F}_q$ this means $n|(q-1)$.

$\mathcal{L}(P) \stackrel{\text{def}}{=} \cup_{m\in\mathbb{N}}\mathcal{L}(m \cdot P)$ the K-linear, infinite dimensional vector space of all function that have poles only at P. The following claim follows from [28, Corollary 3.7.4 and Proposition 3.11.1] and the Hurwitz Genus Formula [28, Theorem 3.4.13]:

▷ Claim 2.10. Let P_∞ be a degree one place of a function field F/K of genus g, ℓ a prime number and $u \in \mathcal{L}(P_\infty)$ which is not an ℓ-th power in F. Denote $d = \deg(u) = -v_{P_\infty}(u)$ and assume d is co-prime to ℓ. Let $F' = F(Z)$ where $Z^\ell = u$ be the Kummer extension with respect to u. Then:

- F' is a degree ℓ extension of F
- P_∞ is totally ramified in F'.[5] Also K is the full constant field of F'.
- $Z \in \mathcal{L}(P'_\infty)$ and $\deg(Z) = d$
- $g' \stackrel{\text{def}}{=} \text{genus}(F')$ satisfies $\ell(g-1) \le g' - 1 \le \ell(g-1) + d$

From now on we assume that K is a finite field, $K = \mathbb{F}_q$ for some prime power q. Let ℓ be a prime number that divides $q-1$. Let P_∞ be a degree one place, and S a set of degree one places of F that does not contain P_∞. Let $u \in \mathcal{L}(P_\infty) \subset F$ such that u is not an ℓ-th power in F. We are interested in the number of $P \in S$ such that $u|_P \stackrel{\text{def}}{=} \phi_P(u) \in K$ is an ℓ-th power as an element of K (where ϕ_P is the evaluation function at P).[6] The following claim is standard and we omit its proof:

▷ Claim 2.11. Suppose $u|_P \neq 0$ for some degree one place $P \in S$. Then:

- If $u|_P$ is not an ℓ-th power in K, then there is a single place above P in F' and it is a place of degree ℓ (and ramification 1).
- If, however, $u|_P$ is a non-zero ℓ-th power in K, then the place P is totally split in F', i.e. there are ℓ distinct degree one places above P (that have all ramification 1).

▶ **Definition 2.12.** *Let F be a function field with constant field K. Let S be a set of degree one places of F/K. We say $z \in F$ is* derivative-useful *for S, or, in short, S-useful, if for every $P \in S$ there exists $\alpha \in K$ such that $v_P(z-\alpha) = 1$.*

The term "S-useful" is not standard and does not hold any deeper meaning than saying z "works" for every place of S in the sense discussed above. We have:

▷ Claim 2.13. Let $F' = F(Z) \mod (Z^\ell - u)$ be a Kummer extension with ℓ prime and $u \in \mathcal{L}(P_\infty) \subset F$ such that u is not an ℓ'th power of an element in F. Further assume K is the full constant field of F'. Let S be a set of degree one places F/K and assume $P_\infty \notin S$. Suppose $S_\ell \subset S$ is such that $u|_P$ is a non-zero ℓ-th power for all $P \in S_\ell$, and let S'_ℓ be the set of all places of F' lying over S_ℓ.

Then:

- If $Q \in S'_\ell$ lying above P then $e(Q|P) = 1$,
- If $x \in F$ is S_ℓ-useful then x when considered as an element of F' is S'_ℓ-useful.

Proof. Let $Q \in S'_\ell$ and denote $P \in S_\ell$ the place of F lying below Q. Since $u|_P$ is a non-zero ℓ-th power, P is totally split in F' (by Claim 2.11). This means there are ℓ places lying over P (Q among them), each of them with relative degree one and ramification 1 over P. Since $x \in F$ is S-useful and $P \in S_\ell \subset S$ there exists an $\alpha \in K$ such that $v_P(X-\alpha) = 1$. Since Q is lying over P and has ramification 1 we get $v_Q(X-\alpha) = e(Q|P)v_P(X-\alpha) = 1 \cdot 1 = 1$. So for any $Q \in S'_\ell$ we found $\alpha \in K$ with $v_Q(X-\alpha) = 1$ completing the proof. ◁

[5] We remind the reader that this means that P'_∞ is the only place of F' above P_∞, has degree one and its ramification index over P_∞ is ℓ.

[6] Notice that $u|_P$ is defined because u does not have a pole at any $P \in S$, and $u|_P \in K$ because any $P \in S$ is degree one.

2.4 Hyper-elliptic curve

Let $F = \mathbb{F}_q[x]$ and $\widehat{F}$ be as in Equations (1) and (2), and further assume q is odd. Let $h \in \overline{\mathbb{F}}_q[x]$ be a square-free polynomial of odd degree d and

$$\mathcal{C} = \{(x, z) \in \mathbb{F}_q^2 \mid z^2 - h(x) = 0\}.$$

By [28, Proposition 6.2.3(b)] the genus g of $\mathcal{C}$ is $\frac{d-1}{2}$. Let P_∞^F be the unique pole of x in F. By [28, Proposition 6.2.3(c)] P_∞^F totally ramifies in $\mathcal{C}/F$. Let P_∞ be the unique place of $\mathcal{C}$ that lies above P_∞^F and v_∞ its corresponding valuation function. Then $v_\infty(x) = -2$ and $2v_\infty(z) = v_\infty(z^2) = v_\infty(h(x)) = dv_\infty(x) = -2d$, i.e., $v_\infty(z) = -d$.

2.5 Extending the Hermitian curve

Suppose p is a prime power and $q = p^2$. Let $F = \overline{\mathbb{F}}_q(x, y) \bmod \phi(x, y)$ where $\phi(x, y) = y^p + y - x^{p+1}$. F is called the Hermitian function field. There is a unique place P_∞ where x and y have poles. Let v_∞ denote the valuation function at P_∞. $v_\infty(x) = -p$, $v_\infty(y) = -(p+1)$ and $\text{genus}(F) = \frac{p(p-1)}{2}$ (see [28, Lemma 6.4.4]).

Let $h \in \overline{\mathbb{F}}_q[x, y]$ be a polynomial with $v_\infty(h) = -w$ for some odd w. Let

$$\mathcal{C} = \{(x, z) \in \overline{\mathbb{F}}_q^2 \mid z^2 - h(x) = 0\}.$$

P_∞ totally ramifies in $\mathcal{C}/F$, because if $P'_\infty | P_\infty$ with valuation v'_∞, then $v'_\infty(z^2) = e(P'_\infty|P_\infty)w$, and as w is odd, $2|e(P'_\infty|P_\infty)$. Let P'_∞ denote the unique place in $\mathcal{C}$ over P_∞, and v'_∞ its associated valuation function. Then, $v'_\infty(x) = -2p$, $v'_\infty(y) = -2(p+1)$ and $v'_\infty(z) = -w$. Also, clearly, $\{1, z\}$ is a basis for $\mathcal{C}/F$ and therefore by [28, Proposition 3.11.1] $\text{genus}(\mathcal{C}) \leq 1 + 2(\text{genus}(F) - 1) + \deg(z) \leq p^2 + w$.

3 The universal derivative-fix lemma

3.1 First order derivatives

We now prove Theorem 1.3 about first order derivatives. Let us denote $f' := H_x^1(f) = D_x^1(f)$. As $f' = \frac{df}{dx}$ we have $(f') = (df) - (dx)$. It follows that the poles of f' can come either from poles of df, or, from zeroes of dx. Since $f \in \mathcal{L}(AP_\infty)$, Claim 2.8 tells us all the poles of f and df are at P_∞. Claim 2.8 also tells us that $v_\infty(df) \geq v_\infty(f) - 1 \geq -(A+1)$, and so df has at most $A+1$ poles, all of which must be at P_∞. We wish to find $\omega \in F$ s.t. $\omega \cdot f' \in \mathcal{L}(P_\infty)$ so we need to choose $\omega \in F$ that cancels the poles of f' at all places other than P_∞. These poles can arise only from zeroes of dx *outside* P_∞. Indeed, we will find $w \in \mathcal{L}(GP_\infty - (dx)_0)$ for some large enough G.

While we are interested in the zeroes of dx, we first consider the *poles* of dx. By Claim 2.8:

- The poles of dx are at the same places as the poles of x, i.e., $v_P(dx) < 0$ implies $v_P(x) < 0$, and,
- At any place P where dx and x have a pole, dx may have at most one more pole than x, i.e., $v_P(dx) \geq v_P(x) - 1$.

It therefore follows that $\deg((dx)_\infty) \leq \deg(x) + \mathsf{supp}$.

Next, we use the fact that (dx) is a canonical divisor (being a divisor associated with a differential), and therefore has degree $2g - 2$. Thus, the number of zeroes of dx is exactly $2g - 2$ more than the number of poles of dx, and in total we get

$$\deg((dx)_0) \leq \deg(x) + \mathsf{supp} + 2g - 2 = G - g,$$

and for $D = GP_\infty - (dx)_0$ we have

$$\deg(D) = G - \deg((dx)_0) \geq g.$$

By the Riemann-Roch Theorem there exists some $0 \neq \omega \in \mathcal{L}(D)$. Fix any such ω. Set $\Delta = G + \min\{v_\infty(dx), 0\}$. Then,

▷ Claim 3.1. $\omega f' = \omega \cdot \frac{df}{dx} \in \mathcal{L}((A + \Delta + 1)P_\infty)$.

Proof. For any $P \neq P_\infty$, $v_P(\omega) \geq -v_P(D) = v_P((dx)_0)$, and

$$v_P(\omega f') = v_P(\omega) + v_P(df) - v_P(dx) \geq v_P(\omega) + v_P(df) - v_P((dx)_0) \geq v_P(df) \geq 0,$$

where we have used that $v_P((dx)_0) = \max\{0, v_P(dx)\} \geq v_P(dx)$.

Next we compute the pole order of wf' at P_∞. We have $w \in \mathcal{L}(D) \subseteq \mathcal{L}((G - \max\{v_\infty(dx), 0\})P_\infty)$. Thus,

$$\begin{aligned} -v_\infty(\omega f') &= v_\infty(dx) - v_\infty(\omega) - v_\infty(df) \\ &= v_\infty(dx) + G - \max\{v_\infty(dx), 0\} - v_\infty(df) \\ &\leq A + 1 + G + v_\infty(dx) - \max\{v_\infty(dx), 0\}, \end{aligned}$$

because $v_\infty(df) \geq v_\infty(f) - 1 \geq -A - 1 = -(A + 1)$. However,

$$v_\infty(dx) - \max\{v_\infty(dx), 0\} = \min\{0, v_\infty(dx)\},$$

and the proof is complete. ◁

3.2 High order derivatives

Proof of Theorem 1.4. We use the same w as before. We prove by induction. We already saw the $m = 1$ case. Assume for m and let us prove for $m+1$. The $m+1$-th Hasse derivative is the same as the $m+1$-th iterated derivative D_x^{m+1} up to multiplication by a non-zero scalar (and using $m + 1 < p$). Now,

$$\begin{aligned} \omega^2 D_x(\omega^{2m-1} D_x^m f) &= \omega^2 \left[D_x(\omega^{2m-1}) D_x^m f + \omega^{2m-1} D_x(D_x^m f) \right] \\ &= (2m-1)\omega^{2m} \cdot D_x(\omega) \cdot D_x^m f + \omega^{2m+1} D_x^{m+1} f \end{aligned}$$

Thus,

$$\omega^{2m+1} D_x^{m+1} f = \omega^2 D_x(\omega^{2m-1} D_x^m f) - (2m-1)\omega^{2m} D_x(\omega) \cdot D_x^m f.$$

By the induction hypothesis and the $m = 1$ case:

$$\begin{aligned} \omega^{2m-1} \cdot D_x^m f &\in \mathcal{L}(A_m P_\infty), \\ \omega D_x(\omega^{2m-1} D_x^m f) &\in \mathcal{L}((A_m + (\Delta + 1))P_\infty). \end{aligned}$$

Also $\omega \in \mathcal{L}(WP_\infty)$. By the $m = 1$ case,

$$\omega \cdot D_x(\omega) \in \mathcal{L}((W + (\Delta + 1))P_\infty)$$

The term $\omega^2 D_x(\omega^{2m-1} D_x^m f)$ is in $\mathcal{L}((A_m + W + \Delta + 1)P_\infty)$. The term $\omega D_x(\omega) \cdot \omega^{2m-1} D_x^m f$ is also in $\mathcal{L}((A_m + W + \Delta + 1)P_\infty)$. Altogether, $\omega^{2m+1} D_x^{m+1} f$ is in $\mathcal{L}(A_{m+1} P_\infty) = \mathcal{L}((A_m + W + \Delta + 1)P_\infty)$. ◀

▶ Remark 3.2. Note that if $D_x(\omega)$ is in $\mathcal{L}(P_\infty)$, we can multiply by a single ω per derivative, instead of multiplying by ω^2.

4 Proving the Hermitian Curve Weil bound for the quadratic residue character

In this section we prove a Weil bound for the quadratic residue character over the Hermitian curve. We bound the bias for a restricted class of polynomials – those spanned by products of $x^i y^j$ where $i+j$ is odd. All such polynomials are non-squares, but not all non-squares are of this form. For this class of polynomials we bound the bias in the order of $O(\sqrt{\frac{d}{p}})$. We remark that in Section 5 we bound the bias for all non-square polynomials, but we get the worse bound $O(\frac{d}{\sqrt{p}})$ on the bias.

We begin in Section 4.1, as a warm-up, by first proving the Weil bound over $\mathbb{F}_q[x]$ using the same technique, and then in Section 4.2, we prove the result for the Hermitian curve.

4.1 Proof over $\mathbb{F}_q[x]$

Proof of Theorem 1.6. $F = \mathbb{F}_q(x)$ is the rational function field and $h \in \mathbb{F}_q[x]$ a square-free polynomial of odd degree. Let

$$\mathcal{C} = \{(x,z) \in \overline{\mathbb{F}}_q^2 \mid z^2 - h(x) = 0\}. \tag{4}$$

Let P_∞ be the unique pole of x and v_∞ its corresponding valuation function. We will create a nonzero algebraic function $R \in \mathcal{L}(DP_\infty)$ such that for each point $P = (\alpha, \beta)$ of $\mathcal{C}$ with $h(\alpha) \neq 0$, R vanishes on P with multiplicity at least M (i.e., $v_P(R) \geq M$). This gives an upper bound of $\frac{D}{M}$ on the number of such points. It follows that the number of elements $\alpha \in \mathbb{F}_q$ for which $h(x)$ is a square is at most $d + \frac{D}{2M}$ (because there are at most d points with $h(\alpha) = 0$ and at most $\frac{D}{2M}$ points with non-zero square evaluation).

Let $A = q-1$ and $B < Md$ an integer to be determined later. We choose R to be of the form:

$$R = \sum_{i=0}^{B-1} u_i x^{iq} + \sum_{i=0}^{B-1} v_i x^{iq} z^q,$$

where each u_i and v_i comes from $\mathcal{L}(AP_\infty)$. Observe that all the x^{iq} and $x^{iq}z^q$ all have v_∞ valuations differing by at least q, because $v_\infty(x) = -2$ is even, and $v_\infty(z) = -d$ (see Section 2.4) is by assumption odd. Since $A = q-1$ we get that R is nonzero provided some u_i or some v_i is nonzero. Now we calculate:

- **Degrees of freedom:** The total number of degrees of freedom is at least $2B(A-g+1)$, because there are $2B$ functions from $\mathcal{L}(AP_\infty)$ that we have to choose.
- **Number of constraints:** For a separating element t, we let $H_t^{(m)}$ denote the m'th order Hasse derivative with respect to t.[7] By Claim 2.1 it is enough to show that for every rational place $P = (\alpha, \beta)$ on $\mathcal{C}$ with $h(\alpha) \neq 0$, there exists a separating element t such that $v_P(t) = 1$ and for each $m \in \{0, 1, \ldots, M-1\}$, $H_t^{(m)} R(P) = 0$. In fact, if $P = (\alpha, \beta)$ is as above, then $t = x - \alpha$ is a separating element with $v_P(t) = 1$ (see Claim 2.13). Thus, by Fact 2.2 it suffices to show that for every such point P, and every $m \in \{0, 1, \ldots, M-1\}$, $H_x^{(m)} R(P) = 0$.

[7] See Section 2.1 for more details about Hasse derivatives.

For $h \in F$ and $m \in \mathbb{N}$, $m < q$, let $h^{(m)}$ denote $H_x^{(m)}h$. Then, $R^{(m)}$ vanishes on all $\mathbb{F}_q$ points of $\mathcal{C}$ with $h(\alpha) \neq 0$, follows from the condition:

$$\omega_m R^{(m)} = \sum_{i=0}^{B-1} \omega_m u_i^{(m)} x^i + \sum_{i=0}^{B-1} \omega_m v_i^{(m)} x^i z = 0,$$

where we have used Corollary 2.6 and the fact that $x^q = x$ and $z^q = z$ over all the points of $\mathcal{C}$ (as they are $\mathbb{F}_q$ rational).
By Theorem 1.4, this asks for a certain member of

$$\mathcal{L}\left((A + O(gm) + Bv_\infty(x) + v_\infty(z))\, P_\infty\right)$$

to equal 0. Thus the total number of constraints is:

$$\sum_{m=0}^{M-1} (A + 2B + O(md) + O(d)) \leq MA + 2BM + O(M^2 d).$$

We need the number of degrees of freedom to be larger than the number of constraints. Expressing $2B = M + E$, for some E, we want

$$(M + E)(A - g) > MA + 2BM + O(M^2 d),$$

i.e.,

$$E(A - g) > Mg + 2BM + O(M^2 d) = O(M^2 d)$$

because $g = O(d)$ and $B < Md$. As $A = q - 1 \geq 2d \geq 2(g - 1)$ (see Section 2.3) it is enough that $EA = \Omega(M^2 d)$. Remembering that $A = q - 1$ it is enough that $E = \Theta(\frac{M^2 d}{q})$.

The number of points $\alpha \in \mathbb{F}_q$ for which $h(\alpha)$ is a square is at most $\frac{v_\infty(R)}{2M} + d$, and

$$\left|\frac{v_\infty(R)}{2M} + d - \frac{q}{2}\right| = \left|\frac{2Bq + A + dq}{2M} + d - \frac{q}{2}\right| = \frac{Eq}{2M} + \frac{A + dq}{2M} + d = O(Md + \frac{dq}{M}).$$

Equating the two error terms we get $M^2 d = dq$ or $M = \sqrt{q}$.

It follows that the number of points $\alpha \in F_q$ for which $h(\alpha)$ is a square is at most $\frac{q}{2} + O(d\sqrt{q})$. We can get the same bound for the number of points $\alpha \in F_q$ for which $h(\alpha)$ is a non-square (e.g., by counting the number of squares ch obtains, when c is a non-square in $\mathbb{F}_q$). We therefore conclude that $|\sum_{x \in \mathbb{F}_q} \chi(h(x))|$, which is the difference between the number of squares and non-squares, is at most $O(d\sqrt{q})$, completing the proof. ◀

4.2 Proof over H

Proof of Theorem 1.7. Let $\widehat{H} = H(z) \pmod{z^2 - h(x, y)}$. Let $\mathcal{C}_H$ be the set of all $\mathbb{F}_q$ rational points of the curve,

$$\mathcal{C}_H = \{(x, y, z) \in \overline{\mathbb{F}}_q^3 \mid y^p + y = x^{p+1}, z^2 - h(x, y) = 0\}.$$

Let P_∞ be the unique pole of x in $\mathcal{C}_H$ and v_∞ its corresponding valuation function. P_∞ is also the unique pole of y and z, and $v_\infty(x) = -2p$, $v_\infty(y) = -2(p + 1)$ and $v_\infty(z) = -w$. The genus g of $\mathcal{C}_H$ is at most $g = p^2 + w \leq p^2 + d(p + 1) \leq 2p^2$ (see Section 2.5). We will create a nonzero algebraic function $R \in \mathcal{L}(DP_\infty)$ such that for each $\mathbb{F}_q$ point $P = (\alpha, \beta, \gamma)$ of $\mathcal{C}_H$ with $h(\alpha, \beta) \neq 0$, R vanishes on P with multiplicity at least M (i.e., $v_P(R) \geq M$). This will give an upper bound of $\frac{D}{M} + w$ on the number of $\mathbb{F}_q$ points of $\mathcal{C}_H$.

Let $A = pq - d$ and $B < p$ an integer to be determined later. We choose R to be of the form:

$$R = \sum_{i=0}^{B-1} u_i x^{iq} + \sum_{i=0}^{B-1} v_i x^{iq} z^q,$$

where each u_i and v_i comes from $\mathcal{L}(AP_\infty)$. Observe that $v_\infty(x^{iq})$ is an even multiple of pq. Also, $v_\infty(z) = -w = -pk - (p+1)\ell = -(k+\ell)p - \ell$ and so $v_\infty(z^q)$ is ℓq close to an odd multiple of pq. As $\ell < d$ and $A = pq - d$, all the $u_i x^{iq}$ and $v_i x^{iq} z^q$ have different v_∞ valuations, and so, R is nonzero provided some u_i or some v_i is nonzero.

Now we calculate:

- **Degrees of freedom:** There are $2B$ functions from $\mathcal{L}(A \cdot P_\infty)$ that we have to choose. So the total number of degrees of freedom is at least $2B(A - g + 1)$.
- **Number of constraints:** By Claim 2.1 it is enough to show that for every rational place $P = (\alpha, \beta, \gamma)$ on $\mathcal{C}_H$ with $h(\alpha, \beta, \gamma) \neq 0$, there exists a separating element t such that $v_P(t) = 1$ and for each $m \in \{0, 1, \ldots, M-1\}$, $H_t^{(m)} R(P) = 0$. In fact, if $P = (\alpha, \beta, \gamma)$ is as above, then $t = x - \alpha$ is a separating element with $v_P(t) = 1$ (see Claim 2.13). Thus, by Fact 2.2 it suffices to show that for every such point P, and every $m \in \{0, 1, \ldots, M-1\}$, $H_x^{(m)} R(P) = 0$.
  For $h \in F$ and $m \in \mathbb{N}$, let $h^{(m)}$ denote $H_x^{(m)} h$. Then, $R^{(m)}$ vanishes on all $\mathbb{F}_q$ points of $\mathcal{C}$ follows from the condition:

  $$\omega_m R^{(m)} = \sum_{i=0}^{B-1} \omega_m u_i^{(m)} x^i + \sum_{i=0}^{B-1} \omega_m v_i^{(m)} x^i z = 0,$$

  where we have used Corollary 2.6 and the fact that $x^q = x$ and $z^q = z$ over all the points of $\mathcal{C}$ (as they are $\mathbb{F}_q$ rational).
  By Theorem 1.4, this asks for a certain member of

  $$\mathcal{L}\left((A + O(gm) + Bv_\infty(x) + v_\infty(z))\, P_\infty\right)$$

  to equal 0. Thus the total number of constraints is:

  $$\sum_{m=0}^{M-1} (A + O(mp^2) + 2pB + w) \leq MA + 2pBM + O(M^2 p^2),$$

  because $w \leq d(p+1) = O(p^2)$.

We need the number of degrees of freedom to be larger than the number of constraints. Taking $2B = M + E$ we want

$$(M + E)(A - g) > MA + 2pBM + O(M^2 p^2),$$

i.e.,

$$E(A - g) > Mg + 2pBM + O(M^2 p^2) = O(M^2 p^2 + pBM) = O(M^2 p^2)$$

because $g = O(p^2)$ and $B < Mp$. As $A = pq - d \geq 4p^2 \geq 2g$, it is enough that $EA = \Omega(M^2 p^2)$. Remembering that $A = pq - d = \Theta(pq)$ it is enough that $E = \Theta(\frac{M^2}{p})$.

The number of points $(\alpha,\beta) \in H \cap \mathbb{F}_q^2$ for which $h(\alpha,\beta)$ is a square is at most $\frac{v_\infty(R)}{2M} + w$, and

$$|\frac{v_\infty(R)}{2M} + w - \frac{pq}{2}| = |\frac{2Bpq + A + wq}{2M} + w - \frac{pq}{2}| = \frac{Epq}{2M} + \frac{A+wq}{2M} + w = O(Mq + \frac{dpq}{M}).$$

Equating the two error terms we get $M^2 q = dpq$ or $M = \sqrt{dp}$.

It follows that the number of points $(\alpha,\beta) \in H \cap \mathbb{F}_q^2$ for which $h(\alpha,\beta)$ is a square is at most $\frac{pq}{2} + O(\sqrt{d}p^{2.5})$. We can get the same bound for the number of points $(\alpha,\beta) \in H \cap \mathbb{F}_q^2$ for which $h(\alpha,\beta)$ is a non-square (e.g., by counting the number of squares ch obtains, when c is a non-square in $\mathbb{F}_q$). We therefore conclude that $|\sum_{x\in\mathbb{F}_q} \chi(h(x))|$, which is the difference between the number of squares and non-squares, is at most $O(\sqrt{d}p^{2.5})$, completing the proof. ◀

5 Proving the Hermitian Curve Weil bound for all non-square polynomials

In this section we bound the bias of multiplicative characters over the Hermitian curve, and the bound works for all non-square polynomials. However, now we get again bias in the order of $O(\frac{d}{\sqrt{p}})$ (instead of the better $O(\sqrt{\frac{d}{p}})$ of Section 4).

We define:

▶ **Definition 5.1.** *We say $h \in H$ is* special *if it can be represented as*

$$h(x,y) = (c(x))^{q^i} \cdot \frac{g_1^2(x,y)}{g_2^2(x,y)}$$

where $i \in \mathbb{N}$, $g_1, g_2 \in \mathbb{F}_q[x,y]$ and $c \in \mathbb{F}_q[x]$ with $\deg(c) \leq \deg(h)$. We say h is non-special *otherwise.*

By the original Weil bound for $\mathbb{F}_q[x]$:

▷ Claim 5.2. If h is as in Theorem 1.8 and in addition h is special then

$$|\sum_{(x,y)\in\mathcal{C}} \chi(h(x,y))| \leq (d-1)p\sqrt{q}.$$

Proof.

$$|\sum_{(x,y)\in\mathcal{C}} \chi(h(x,y))| = |\sum_{(x,y)\in\mathcal{C}} \chi(c(x)^{q^i})| = |\sum_{(x,y)\in\mathcal{C}} \chi(c(x))|,$$

where the first equality is because χ is multiplicative and one on squares, and the second is because we only evaluate over $x \in \mathbb{F}_q$. Now if c is a square in $\overline{\mathbb{F}_q}[x]$, say, $c = z^2$, then c^{q^i} is also a square, $c^{q^i} = (z^{q^i})^2$, and therefore h is also a square. A contradiction. Hence, c is not a square, and therefore by the original Weil bound $|\sum_{(x,y)\in\mathcal{C}} \chi(c(x))| \leq (d-1)p\sqrt{q}$ (using the fact that every $x \in \mathbb{F}_q$ appears in exactly p tuples $(x,y) \in H$). ◁

It therefore follows that to prove Theorem 1.8 it is enough to prove the following theorem:

▶ **Theorem 5.3.** *Suppose p is a prime power, $q = p^2$, $h \in \mathbb{F}_q[x,y]$, $\deg(h) = d$, such that $Z^2 - h$ is absolutely reducible. Further assume h is not special. Then*

$$|\sum_{(x,y)\in\mathcal{C}} \chi(h(x,y))| = O(d \cdot p^{2.5}).$$

Proof of Theorem 5.3. Let S be the set of $\mathbb{F}_q$ rational points of H such that $h(x,y)$ is a non-zero square in $\mathbb{F}_q$. Let P_∞ be the unique pole of x in H and v_∞ its corresponding valuation function. P_∞ is also the unique pole of y, $v_\infty(x) = -p$, $v_\infty(y) = -(p+1)$. The genus g of H is $\frac{p(p-1)}{2}$ (see Section 2.5).

We create a nonzero algebraic function $R \in \mathcal{L}(DP_\infty)$ such that for each $P = (\alpha, \beta) \in S$, R vanishes on P with multiplicity at least M (i.e., $v_P(R) \geq M$). This gives an upper bound of $\frac{D}{M}$ on the number of $\mathbb{F}_q$ points of S.

Let $B_x = \lfloor \frac{p}{4} - d \rfloor$ and $B_y = \frac{p}{2} - (B_x + d) - 1 = \frac{p}{4} - 1$. We choose R to be of the form:

$$R = u + vh(x,y)^{\frac{q-1}{2}},$$

where each u and v come from $V_{B_x q, B_y}$ where $V_{c_1,c_2} = \mathsf{Span}\{x^i y^j \mid i \leq c_1, j \leq c_2\}$.

A crucial step in the proof is proving R is nonzero provided u or v is nonzero. We defer the proof to Section 5.1 and we continue assuming this claim.

Now we calculate:

- **Degrees of freedom:** There are $2qB_xB_y$ degrees of freedom,
- **Number of constraints:** By Claim 2.1 it is enough to show that for every rational place $P = (\alpha, \beta)$ from S there exists a separating element t such that $v_P(t) = 1$ and for each $m \in \{0, 1, \ldots, M-1\}$, $H_t^{(m)} R(P) = 0$. In fact, if $P = (\alpha, \beta)$ is as above, then $t = x - \alpha$ is a separating element with $v_P(t) = 1$ (see Claim 2.13). Thus, by Fact 2.2 it suffices to show that for every such point P, and every $m \in \{0, 1, \ldots, M-1\}$, $H_x^{(m)} R(P) = 0$.
  For $f \in F$ and $m \in \mathbb{N}$, let $f^{(m)}$ denote $H_x^{(m)} f$. Then, $R^{(m)}$ vanishes on all points of S follows from the condition $h^m R^{(m)} = 0$ on S.

▷ Claim 5.4. Let $I = \langle x^q - x, h^{(q-1)/2} - 1 \rangle$. Then $h^m R^{(m)} \bmod I \in V_{1, B_y + md}$.

Proof. First, we determine the effect of taking Hasse derivatives with respect to x:

- If $u \in V_{c_1,c_2}$, say, $u = x^i y^j$ with $i \leq c_1$ and $j \leq c_2$. Then for every $\ell < p$, $H_x^{(\ell)}(u) \in V_{c_1 + p\ell, c_2}$, using $H_x^{(\ell)}(y^{i_2}) = c \cdot y^{i_2 - \ell} x^{p\ell}$ for some constant c. Also,
- By Fact 2.3,

$$h^\ell H_x^{(\ell)}(h^{\frac{q-1}{2}}) = \sum_{j=1}^{\ell} \binom{\frac{q-1}{2}}{j} h^{\frac{q-1}{2} + \ell - j} \sum_{\substack{i_1, \ldots, i_\ell \geq 0 \\ i_1 + \ldots + i_\ell = j \\ i_1 + 2i_2 + \ldots + \ell i_\ell = \ell}} \binom{j}{i_1, \ldots, i_\ell} \cdot (H_x^{(1)} h)^{i_1} \cdot \ldots \cdot (H_x^{(\ell)} h)^{i_\ell}.$$

As $h \in V_{d,d}$, for $k < p$ we have

$$H_x^{(k)}(h) \in V_{d+pk,d}, \text{ and,}$$
$$(H_x^{(1)} h)^{i_1} \cdot \ldots \cdot (H_x^{(\ell)} h)^{i_\ell} \in V_{d + p(i_1 + 2i_2 + \ldots + \ell i_\ell), d(i_1 + \ldots + i_\ell)}.$$

Also, $h^{\frac{q-1}{2} + \ell - j} \bmod I = h^{\ell - j} \in V_{(\ell-j)d, (\ell-j)d}$.

Altogether,

$$h^\ell H_x^{(\ell)}(h^{\frac{q-1}{2}}) \bmod I \in V_{\max_j \{(\ell-j)d + d + \ell p\}, (\ell-j)d + dj} = V_{\ell(p+d), \ell d}.$$

By linearity, Corollary 2.6 and the above items we see that

$$h^m H_x^{(m)} R \bmod I \in V_{q, B_y + md},$$

and notice that we work now in a smaller subspace because we may replace x^q with x and $h^{\frac{q-1}{2}}$ with 1. ◁

We want to force $h^m H_x^{(m)} R \bmod I \in V_{q,B_y+md}$ to equal 0. The number of constraints this imposes is at most

$$\sum_{m=0}^{M-1} q(B_y + md) = MqB_y + O(M^2qd).$$

We need the number of degrees of freedom to be larger than the number of constraints. Taking $2B_x = M + E$ we want

$$(M+E)qB_y > MqB_y + O(M^2qd),$$

i.e., $EqB_y = \Omega(M^2qd)$, or $E = \Omega(\frac{M^2d}{B_y}) = \Omega(\frac{M^2d}{p})$.

The number of points $(\alpha, \beta) \in H \cap \mathbb{F}_q^2$ for which $h(\alpha, \beta)$ is a square is at most $\frac{\deg(R)}{M} + d(p+1)$, and

$$\begin{aligned}|\frac{v_\infty(R)}{M} + d(p+1) - \frac{pq}{2}| &= |\frac{B_x pq + B_y(p+1) + d(p+1)\frac{q-1}{2}}{M} + d(p+1) - \frac{pq}{2}| \\ &= \frac{Epq}{2M} + \frac{dpq}{M} = O(Mdq + \frac{dpq}{M}),\end{aligned}$$

using $M, B_y < p$. Equating the two error terms we get $M = \Theta(\sqrt{p})$.

It follows that the number of points $(\alpha, \beta) \in H \cap \mathbb{F}_q^2$ for which $h(\alpha, \beta)$ is a square is at most $\frac{pq}{2} + O(dp^{2.5})$. We can get the same bound for the number of points $(\alpha, \beta) \in H \cap \mathbb{F}_q^2$ for which $h(\alpha, \beta)$ is a non-square (e.g., by counting the number of squares ch obtains, when c is a non-square in $\mathbb{F}_q$). We therefore conclude that $|\sum_{x \in \mathbb{F}_q} \chi(h(x))|$, which is the difference between the number of squares and non-squares, is at most $O(dp^{2.5})$, completing the proof. ◀

5.1 Independence

▶ **Theorem 5.5.** *Suppose $h \in H$ is as in Theorem 5.3. Let $B_y < \frac{p}{2} - (B_x + d)$. Then the elements*

$$\{x^i y^j h(x,y)^{k(q-1)/2} \mid 0 \le i \le B_x q, j \le B_y, k \in \{0,1\}\}$$

are independent.

Proof. Suppose $\sum_{i \le B_x q, j \le B_y, k} c_{i,j,k} x^i y^j h(x,y)^{k(q-1)/2} = 0$. Denote

$$m_k(x,y) = \sum_{i \le B_x q, j \le B_y} c_{i,j,k} x^i y^j.$$

Then $m_0 + m_1 h^{(q-1)/2} = 0$ and $m_0^2 h = m_1^2 h^q = m_1^2 h(x^q, y^q)$.

$v_{(0,0)}(y) = p+1$, and, in fact, as $y^p + y = x^{p+1}$ we have:

$$y = \sum_{i=0}^{\infty} (-1)^i x^{p^i(p+1)}. \tag{5}$$

We observe that

$$\begin{aligned} y &= x^{p+1} - x^{p(p+1)} \ (\mathrm{mod} x^{q(p+1)}) \\ y^p &= x^{p(p+1)} \ (\mathrm{mod} x^{q(p+1)}) \\ y^q &= 0 \ (\mathrm{mod} x^{q(p+1)}). \end{aligned}$$

We have

$$m_0^2 h(x,y) = m_1^2 h(x^q, 0) \ (\mathrm{mod} x^{q(p+1)}).$$

Notice that

$$\Delta = m_1^2 h(x^q,0) - m_0^2 h(x,y) \in \mathsf{Span}\{x^i y^j \mid i < (2B_x + d)q, j \le 2B_y + d < p - (2B_x + d)\},$$

and so by Claim 5.6 that we will shortly prove we have:

$$m_0^2 h(x,y) = m_1^2 h(x^q, 0).$$

Now,

- If $m_0 \neq 0$ then

$$h(x,y) = h(x,0)^q \cdot (\frac{m_1(x,y)}{m_0(x,y)})^2,$$

  and h is special.
- If $m_0 = 0$. Then $m_1 h^{(q-1)/2} = 0$ and therefore $m_1 = 0$. Therefore, all the coefficients $c_{i,j,k}$ are zero as desired. ◀

▷ Claim 5.6. Let $c \in \mathbb{N}$. Suppose $\Delta \in \mathsf{Span}\{x^i y^j \mid i < cq, j < p - c\}$. Then $\Delta = 0 \ (\mathrm{mod} x^{q(p+1)})$ iff $\Delta = 0$.

Proof. The elements in $\mathsf{Span}\{x^i y^j \mid i \in \mathbb{N}, j < p\}$ have different v_∞ valuations and are independent in H, so there is a unique way to express $\Delta = \sum_{i<cq, j<p-c} \Delta_{i,j} x^i y^j$. We have

$$\begin{aligned}\Delta &= \sum_{i<cq, j<p-c} \Delta_{i,j} x^i (x^{p+1} - x^{p(p+1)})^j \ (\mathrm{mod} x^{q(p+1)}) \\ &= \sum_{i<cq, j<p-c} \Delta_{i,j} x^i (x^{p+1} - x^{p(p+1)})^j\end{aligned}$$

where in the first line we have used Equation (5), and in the second we notice that the degree of Δ in x is smaller than $q(p+1)$, because $i < cq$ and $j < p - c$.

Now, if $\Delta = 0$ then clearly $\Delta (\mathrm{mod} x^{q(p+1)}) = 0$. If $\Delta \neq 0$, order the tuples (i,j) first by j and then by i, and let (i_0, j_0) be the maximal tuple in that order such that $\Delta_{i_0,j_0} \neq 0$. Then, the monomial $x^{j_0 p(p+1)+i_0}$ is obtained in a unique way, and therefore has coefficient $\Delta_{i_0,j_0} \neq 0$. Hence $\Delta (\mathrm{mod} x^{q(p+1)}) \neq 0$. ◁

6 Proving the Hermitian Curve Weil bound for additive characters

▶ **Theorem 6.1.** *Assume $p = 2^n$, $q = p^2$, $H = \mathbb{F}_q(x,y) \bmod \varphi$. Let $f \in \mathbb{F}_q[x,y]$ be a degree $d \le \sqrt{p}$ polynomial such that $d' = -v_\infty(f)$ is odd. Let $\theta \in \{0,1\}$ and denote*

$$V_\theta = \{(a,b) \in \mathbb{F}_q \times \mathbb{F}_q \mid \varphi(a,b) = 0 \ , \ \mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2}(f(a,b)) = \theta\}.$$

Then, $|V_\theta| \le \frac{p^3}{2}(1 + O(\frac{d}{\sqrt{p}}))$.

Proof. We create a nonzero algebraic function $R \in \mathcal{L}(DP_\infty)$ such that for each point $P = (\alpha, \beta) \in V_\theta$ for which $\beta \neq 0$, R vanishes on P with multiplicity at least M (i.e., $v_P(R) \ge M$). This gives an upper bound of $\frac{D}{M} + 1$ on the cardinality of V_θ. Specifically,

- We look for $R \in \mathbb{F}_q[x,y]$ in the $\mathbb{F}_q$-vector space

$$\mathsf{Span}\{x^{i_1} \cdot y^{i_2+i_3p+i_4q} \cdot \mathsf{Tr}(f(x,y))^{i_5} \mid i_1+i_2 \in [A_1], i_3 \in [A_3], i_4 \in [A_4], i_5 \in \{0,1\}\}, \quad (6)$$

where A_1, A_3, A_4 will be specified later. Then, $R \in \mathcal{L}(DP_\infty)$ with $D = (p+1)(A_1+pA_3+qA_4)+d(p+1)\frac{q}{2}$.

- Define the ideal $I = \langle y^q - y, \mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2}(f(x,y)) - \theta\rangle$ of $\mathbb{F}_q[x,y]$. In Section 6.2 we show that $H_y^{(m)}R \in \mathbb{F}_q[x,y]$. We look for a non-zero polynomial R such that for every $m < M$, $y^m H_y^{(m)}R \in I$ or, equivalently, $y^m H_y^{(m)}R \bmod I = 0$. This in particular guarantees that for every $P = (\alpha,\beta) \in V_\theta$ with $\beta \neq 0$, $H_y^{(m)}R(P) = 0$ which implies that $v_P(R) \geq M$.[8]

Now we calculate:

- **Degrees of freedom:** In Section 6.1 we prove all the elements $x^{i_1} \cdot y^{i_2+i_3p+i_4q} \cdot \mathsf{Tr}(f(x,y))^{i_5}$ appearing in Equation (6) have different valuations at P_∞, and therefore are independent over $\mathbb{F}_q$. Hence, the number of degrees of freedom is $2\binom{A_1+2}{2}(A_3+1)(A_4+1)$. It is straight forward to see that all the elements $x^{i_1} \cdot y^{i_2+i_3p+i_4q}$ are independent. The crucial thing to notice is that because we may also include $\mathsf{Tr}(f(x,y))$, the dimension of the span doubles.
- **Number of constraints:** For every $0 < m < M$, we would like $y^m H_y^{(m)}R$ to be in the ideal I. In Section 6.2 we prove this can be guaranteed by imposing at most $M(A_3+M)\frac{(A_1+A_4+(M-1)d+2)^2}{2}$ linear constraints on the coefficients of R.

We now choose parameters. The number of variables is at least $2(A_4+1)\frac{A_3A_1^2}{2}$. We choose

$$A_4 + 1 = \lceil (1+4\alpha)\frac{M}{2} \rceil,$$

for some constant parameter $0 < \alpha < \frac{1}{4}$ that we fix later. Thus, the number of variables is at least $(1+4\alpha)\frac{MA_3A_1^2}{2}$. We choose parameters such that $A_3 + M \approx A_3$, $A_1 + A_4 + Md \approx A_1$, say, such that the number of constraints is at most $(1+2\alpha)\frac{MA_3A_1^2}{2}$, and so there are more variables than constraints.

Specifically, we set

$$M = \lfloor \frac{\alpha A_1}{d+1} \rfloor = \lfloor \alpha A_3 \rfloor,$$

The number of linear constraints is upper bounded by $M \cdot (1+\alpha)A_3 \cdot \frac{(1+\alpha)^2A_1^2}{2}$.[9] The claim then follows because $1+4\alpha \geq (1+\alpha)^3$ (for $\alpha \leq \frac{1}{4}$).

We therefore see that $|V_\theta| \leq \frac{D}{M}+1 = \frac{(p+1)(A_1+pA_3+qA_4)+d(p+1)\frac{q}{2}}{M}+1$. For the second term, $\frac{d(p+1)q}{2M} = \frac{p^3}{2}(\frac{d}{M}+\frac{d}{pM})$. The first term contributes $\frac{p^3}{2}(\frac{2A_4}{M}+\frac{2A_3}{pM}+\frac{2A_1}{p^2M}+\frac{2A_4}{pM}+\frac{2A_3}{p^2M}+\frac{2A_1}{p^3M})$. Notice that $\frac{2A_4}{M} \leq 1+4\alpha$. Thus, basically, $|\,|V_\theta| - \frac{p^3}{2}| \leq p^3(2\alpha+\frac{d}{2M})$ plus lower order terms, and so we set $\alpha = \frac{d}{4M}$. We also set A_1 as large as we can and we choose $A_1 = \Theta(p)$.

---

[8] To see that formally, first note that y is S-useful, for the set S of rational points of H (see Definition 2.12). Thus, by Fact 2.2, for any point $P = (\alpha,\beta) \in V_\theta$, $H_y^{(m)}(R) = H_{y-\beta}^{(m)}(R)$ and Claim 2.1 implies $v_P(R) \geq M$.

[9] As $M \leq \frac{\alpha}{d+1}A_1$, $A_4 < \frac{1+4\alpha}{2}M < M$ and $M(d+1) \leq \alpha A_1$.

This means $M^2 = \Theta(A_1) = \Theta(p)$. Thus, $M = \Theta(\sqrt{p})$, $A_3 = \Theta(\frac{M^2}{d}) = \Theta(\frac{p}{d})$, $A_1 = \Theta(p)$. Altogether,

$$d < A_4 \approx \frac{M}{2} < M < A_3 < A_1.$$

This puts the error term at $O(\frac{1}{p} + \frac{d}{M}) = O(\frac{d}{\sqrt{p}})$ as stated. ◀

6.1 Independence

▶ **Lemma 6.2.** *Assume $A_1 < \frac{p}{4}$, $A_3, A_4 < p$. The elements*

$$\{x^{i_1} \cdot y^{i_2+i_3p+i_4q} \cdot \mathsf{Tr}(f(x,y))^{i_5}\}_{i_1+i_2\in[A_1],i_3\in[A_3],i_4\in[A_4],i_5\in\{0,1\}}$$

have different valuations at P_∞ and hence are independent over $\mathbb{F}_q$.

Proof. Let $v = v_\infty$ be the valuation at P_∞. $v(x) = -(p+1)$,$v(y) = -p$. Before we start we first calculate $v(\mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2} f(x,y))$. We have: $\mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2} f(x,y) = \sum_{j=0}^{2n-1} f(x,y)^{2^j}$. Also, $v(f(x,y)^{2^j}) = 2^j v(f(x,y) = 2^j(-d')$. Hence,

$$v(\mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2} f(x,y)) = \min\{2^j \cdot (-d')\}_{j=0}^{2n-1} = -2^{2n-1}d' = -\frac{q}{2}d'.$$

Assume $x^{i_1} \cdot y^{i_2+i_3p+i_4q} \cdot \mathsf{Tr}(f(x,y))^{i_5} = x^{j_1} \cdot y^{j_2+j_3p+j_4q} \cdot \mathsf{Tr}(f(x,y))^{j_5}$, and w.l.o.g. $j_5 \geq i_5$. Set $\Delta_\ell = i_\ell - j_\ell$. Then

$$(\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5) \cdot (p+1, p, p^2, p^3, \frac{q}{2}d') = 0,$$

and $\Delta_5 \in \{0,1\}$. In particular,

$$-d'\Delta_5\frac{q}{2} \quad = \quad (p+1)\Delta_1 + p\Delta_2 + p^2\Delta_3 + p^3\Delta_4. \tag{7}$$

Assume $\Delta_5 \neq 0$. Then $\Delta_5 = 1$. Taking the equation modulo q (and remembering that d' is odd) we get:

$$\frac{q}{2} \quad = \quad p\,[\Delta_1 + \Delta_2] + \Delta_1 (\mathrm{mod}\ q),$$

Now $|\Delta_1 + \Delta_2| \leq 2A_1 < \frac{p}{2}$, and so the RHS of the equation is in the range $(-q/2, q/2)$. A contradiction. Hence $\Delta_5 = 0$. With $\Delta_5 = 0$, Equation (7) becomes $\Delta_1 + (\Delta_1 + \Delta_2)p + \Delta_3 p^2 + \Delta_4 p^3 = 0$. As $|\Delta_1| + |\Delta_2|, |\Delta_3|, |\Delta_4| < p$ this implies $\Delta_1 = \Delta_3 = \Delta_4 = 0$ and $\Delta_1 + \Delta_2 = 0$, hence $(i_1, i_2, i_3, i_4, i_5) = (j_1, j_2, j_3, j_4, j_5)$. ◀

6.2 The derivatives $H_y^{(m)}$ and the constraints

Let us denote $\mathbf{I}(A_1, A_3, A_4, A_5) = \mathsf{Span}\{x^{i_1} y^{i_2+i_3p+i_4p^2} \mathsf{Tr}(f(x,y))^{i_5} \mid i_1 + i_2 \leq A_1, i_3 \leq A_3, i_4 \leq A_4, i_5 \leq A_5\}$.

▶ **Lemma 6.3.** *For every $i \in \mathbb{N}$, $m < p$,*

1. $y^m H_y^{(m)}(y^i) \in \mathsf{Span}\{y^i\}$,
2. $y^m H_y^{(m)}(x^i) \in \mathbf{I}(i, \min\{m, i\}, 0, 0)$,
3. $y^m H_y^{(m)} f \in \mathbf{I}(d, \min\{m, d\}, 0, 0)$,
4. *If $m > 0$,* $y^m H_y^{(m)}(\mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2} f(x,y)) \in \mathbf{I}(dm, m, 0, 0)$.

Proof. Item 1 follows from $H_y^{(m)}(y^i) = \binom{i}{m} y^{i-m}$. Item 2 is Example 2.5. For the last two items:

- f has total degree d, hence it is enough to evaluate $y^m H^{(m)}(x^{j_1} y^{j_2})$ for $j_1 + j_2 \leq d$. Using the product rule (Section 2.1) and the first two items we see that it is in $\mathbf{I}(d, \min\{m, d\}, 0, 0)$,
- We have $\mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2} f(x,y) = \sum_{j=0}^{2n-1} f(x,y)^{2^j}$. The Hasse derivative is linear, and therefore we need to evaluate $y^m H_y^{(m)} f(x,y)^{2^j}$. Using Claim 2.4 it is non-zero only when $2^j | m$, i.e., $m = 2^j \ell$ for some natural number ℓ, and then $y^m H_y^{(m)}(f(x,y)^{2^j}) = (y^\ell H_y^{(\ell)} f(x,y))^{2^j}$. By the previous items, $y^\ell H_y^{(\ell)}(f(x,y)) \in \mathbf{I}(d, \min\{\ell, d\}, 0, 0)$, and $y^m H_y^{(m)}(f(x,y)^{2^j}) \in \mathbf{I}(d2^j, \min\{\ell, d\}2^j, 0, 0)$. As $\ell 2^j = m$, we see that $y^m H_y^{(m)}(\mathsf{Tr}(f(x,y))) \in \mathbf{I}(dm, m, 0, 0)$. ◀

▶ **Lemma 6.4.** *Suppose $Q \in \mathbf{I}(A_1, A_3, A_4, A_5)$ and $m > 0$. Then $y^m H_y^{(m)} Q \in \mathbf{I}(A_1 + dm, A_3 + m, A_4, A_5)$. Furthermore, each coefficient of $y^m H_y^{(m)} Q$ in the basis $\mathbf{I}(A_1 + dm, A_3 + m, A_4, A_5)$ is a linear combination (over $K = \mathbb{F}_q$) of the coefficients of Q in the basis $\mathbf{I}(A_1, A_3, A_4, A_5)$.*

Proof. It is enough to consider a single monomial $x^{i_1} y^{i_2 + i_3 p + i_4 p^2} \mathsf{Tr}(f(x,y))^{i_5}$. By the product rule it suffices to analyze $y^{m_1} H_y^{(m_1)} x^{i_1}$, $y^{m_2} H_y^{(m_2)} y^{i_2}$, $y^{m_3} H_y^{(m_3)} y^{i_3 p}$, $y^{m_4} H_y^{(m_4)} y^{i_4 p^2}$, $y^{m_5} H_y^{(m_5)} (\mathsf{Tr}(f(x,y))^{i_5}$ where $m_1 + m_2 + m_3 + m_4 + m_5 = m$. By Lemma 6.3 we see that

$$y^m H_y^{(m)} (x^{i_1} y^{i_2 + i_3 p + i_4 p^2} \mathsf{Tr}(f(x,y))^{i_5}) \in \mathbf{I}(i_1 + i_2 + dm, i_3 + m, i_4, i_5).$$

This shows that $y^m H_y^{(m)} Q \in \mathbf{I}(A_1 + dm, A_3 + m, A_4, A_5)$. It is also clear from the computation we have done that each coefficient of $y^m H_y^{(m)} Q$ is a linear combination (over $K = \mathbb{F}_q$) of the coefficients of Q. ◀

Say $R \in \mathbf{I}(A_1, A_3, A_4, A_5)$. We saw $y^m H^{(m)} R \in \mathbf{I}(A_1 + dm, A_3 + m, A_4, A_5)$, and each coefficient of $y^m H^{(m)} R$ is some linear combination of the coefficients of R. As $y^q = y \bmod I$ and $\mathsf{Tr}_{\mathbb{F}_q/\mathbb{F}_2} f(x,y) = \theta \bmod I$,

$$y^m H^{(m)} R \bmod I \quad \in \quad \mathbf{I}(A_1 + A_4 + dm, A_3 + m, 0, 0)$$

Thus, to force $y^m H^{(m)} R \bmod I = 0$ it is enough to force $\dim(\mathbf{I}(A_1 + A_4 + dm, A_3 + m, 0, 0))$ linear equations to be zero, which amounts to $\binom{A_1 + A_4 + dm + 2}{2} \cdot (A_3 + m + 1)$ linear equations. Altogether, we have $\sum_{k=0}^{M-1} \binom{A_1 + A_4 + dm + 2}{2} \cdot (A_3 + m + 1)$ linear equations which we upper bound by $M(A_3 + M) \frac{(A_1 + A_4 + (M-1)d + 2)^2}{2}$ as desired.

References

1 Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *Information Theory, IEEE Transactions on*, 38(2):509–516, 1992. doi:10.1109/18.119713.

2 Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992. doi:10.1002/RSA.3240030308.

3 Avraham Ben-Aroya and Amnon Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. *Theory OF Computing*, 9(5):253–272, 2013. doi:10.4086/TOC.2013.V009A005.

4 A. Bogdanov. A different way to improve the bias via expanders. Topics in (and out) the theory of computing, Lecture 12, 2012.

5 Enrico Bombieri. Counting points on curves over finite fields: d'après sa stepanov. In *Séminaire Bourbaki vol. 1972/73 Exposés 418–435*, pages 234–241. Springer, 1974.

6 Gil Cohen and Itay Cohen. Wide replacement products meet gray codes: Toward optimal small-bias sets. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2025.

7 Gil Cohen, Dean Doron, Noam Goldgraber, and Tomer Manket. Tracing ag codes: Toward meeting the gilbert-varshamov bound. *arXiv preprint arXiv:2511.08788*, 2025. doi:10.48550/arXiv.2511.08788.

8 Arnaldo Garcia and Henning Stichtenoth. A tower of artin-schreier extensions of function fields attaining the drinfeld-vladut bound. *Inventiones mathematicae*, 121(1):211–222, 1995.

9 Arnaldo Garcia and Henning Stichtenoth. On the asymptotic behaviour of some towers of function fields over finite fields. *Journal of Number Theory*, 61(2):248–273, 1996. doi:10.1006/jnth.1996.0147.

10 Edgar N Gilbert. A comparison of signalling alphabets. *The Bell system technical journal*, 31(3):504–522, 1952.

11 David M. Goldschmidt. *Algebraic Functions and Projective Curves*, volume 215. Springer New York, NY, 2003.

12 DR Heath-Brown. An estimate for heilbronn's exponential sum. *PROGRESS IN MATHEMATICS-BOSTON-*, 139:451–464, 1996.

13 Jun-Ting Hsieh, Sidhanth Mohanty, and Rachel Yun Zhang. Explicit almost-optimal epsilon balanced codes via free expander walks. *arXiv preprint*, 2026. arXiv:2601.12606.

14 Sangtae Jeong. Calculus in positive characteristic p. *Journal of Number Theory*, 131(6):1089–1104, 2011. doi:10.1016/j.jnt.2010.12.006.

15 Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on information theory*, 18(5):652–656, 2003.

16 R. C. Mason. *Diophantine Equations over Function Fields*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1984. doi:10.1017/CBO9780511752490.

17 Robert McEliece, Eugene Rodemich, Howard Rumsey, and Lloyd Welch. New upper bounds on the rate of a code via the delsarte-macwilliams inequalities. *IEEE transactions on Information Theory*, 23(2):157–166, 1977. doi:10.1109/TIT.1977.1055688.

18 Dmitrii Alekseevich Mit'kin. Stepanov method of the estimation of the number of roots of some equations. *Mathematical Notes*, 51(6):565–570, 1992.

19 Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput*, 22:838–856, 1993. doi:10.1137/0222053.

20 Wolfgang Schmidt. Zur methode von stepanov. *Acta Arithmetica*, 24(4):347–367, 1973.

21 Wolfgang M Schmidt. A lower bound for the number of solutions of equations over finite fields. *Journal of Number Theory*, 6(6):448–480, 1974.

22 Wolfgang M Schmidt. *Equations over finite fields: an elementary approach*, volume 536. Springer, 2006.

23 Sergei Aleksandrovich Stepanov. On the number of points of a hyperelliptic curve over a finite prime field. *Mathematics of the USSR-Izvestiya*, 3(5):1103, 1969.

24 Sergei Aleksandrovich Stepanov. An elementary method in algebraic number theory. *Acta Arith.*, 17:231–247, 1970.

25 Sergei Aleksandrovich Stepanov. Congruences in two unknowns. *Izvestiya: Mathematics*, 6(4):677–704, 1972.

26 Sergei Aleksandrovich Stepanov. An elementary proof of the hasse-weil theorem for hyperelliptic curves. *Journal of number theory*, 4(2):118–143, 1972.

27 Sergei Aleksandrovich Stepanov. On estimating rational trigonometric sums with prime denominator. *Collection of Articles Dedicated to Academician IM Vinogradov on the Eightieth Anniversary of His Birth*, 1:358, 1973.

28 Henning Stichtenoth. *Algebraic function fields and codes*, volume 254. Springer Science & Business Media, 2009.

29 Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 238–251. ACM, 2017. doi:10.1145/3055399.3055408.

30 Saeed Tafazolian. A family of maximal hyperelliptic curves. *Journal of Pure and Applied Algebra*, 216(7):1528–1532, 2012.

31 Fernando Torres. The approach of stöhr-voloch to the hasse-weil bound with applications to optimal curves and plane arcs, 2000. doi:10.48550/arXiv.MATH/0011091.

32 Michael A Tsfasman, SG Vlăduţ, and Th Zink. Modular curves, shimura curves, and goppa codes, better than varshamov-gilbert bound. *Mathematische Nachrichten*, 109(1):21–28, 1982.

33 RR Varshamov. Estimate of the number of signals in error correcting codes, dokl. acad. nauk. sssr 117. *Dokl. Acad. Nauk. SSSR 117, in Russian*, 117:739–741, 1957.

34 Serge G Vlăduţ. Two applications of weil-serre's explicit formula. *Applicable Algebra in Engineering, Communication and Computing*, 7(4):279–288, 1996. doi:10.1007/BF01195533.

35 André Weil. *Sur les courbes algébriques et les variétés qui s' en déduisent*. FeniXX, 1948.

Asymptotic Rank Speedup Theorems, Revisited

Josh Alman
Columbia University, New York, NY, USA

Baitian Li
Columbia University, New York, NY, USA

Abstract

Motivated by fast matrix multiplication and recent connections between asymptotic tensor rank and fine-grained complexity, we revisit classical tools from the matrix multiplication literature and develop a framework for obtaining improved asymptotic rank upper bounds for tensors beyond matrix multiplication.

In the 1980s, Coppersmith–Winograd and Strassen discovered a series of *speedup theorems* for asymptotic rank: in certain regimes, one can extract additional terms from a border rank upper bound on a tensor T, and then use these terms to obtain an improved asymptotic rank of T. We establish general speedup theorems that subsume these results and enable quantitative improvements. Two representative applications are:

1. The asymptotic rank of the small Coppersmith–Winograd tensor cw_q is less than its border rank. For instance, we prove $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_2) < 3.931$, improving on $\underline{\mathrm{R}}(\mathrm{cw}_2) = 4$. It is known that $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_2) = 3$ would imply $\omega = 2$.
2. A general improvement over Strassen's bound: we obtain an upper bound below $d^{2\omega/3}$ on the asymptotic rank of any $d \times d \times d$ tensor.

To make full use of speedups, we analyze degenerations in which both sides are nontrivial direct sums, a setting where the optimal quantitative bound one can achieve was previously unclear. We do so via an approach we call *Strassen calculus*: a systematic method for converting such degeneration data into explicit asymptotic rank bounds using Strassen's theory of the asymptotic spectrum.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases tensor rank, matrix multiplication, Coppersmith–Winograd tensor, asymptotic spectrum

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.36

Funding *Josh Alman*: Supported in part by NSF Grant CCF-2238221 and a Packard Foundation Fellowship.
Baitian Li: Supported in part by NSF Grant CCF-2238221, a Packard Foundation Fellowship, and a Columbia SEAS Presidential Fellowship.

1 Introduction

The exponent of matrix multiplication, ω, is defined as the smallest real number such that two $n \times n$ matrices can be multiplied using at most $n^{\omega+o(1)}$ field operations. Because of the fundamental and widespread applications of matrix multiplication algorithms, determining the value of ω has become one of the most important open problems in computer science. Strassen's famous algorithm [43] showed the first nontrivial bound $\omega \leqslant \log_2(7) < 2.81$. Since then, a long line of follow-up work has culminated in the current best bound $\omega < 2.371339$ [2]. The only known lower bound is the straightforward $\omega \geqslant 2$, since the input and output sizes are n^2. In fact, it is frequently conjectured that $\omega = 2$.

It has been known since Strassen's work [44] that determining ω is equivalent to understanding the *asymptotic rank* of the matrix multiplication tensor. In fact, there are three different rank notions for a (3-mode/3-dimensional) tensor T which have played a prominent

© Josh Alman and Baitian Li;
licensed under Creative Commons License CC-BY 4.0

role in the study of matrix multiplication algorithms: the rank $\mathrm{R}(T)$, the *border rank* $\underline{\mathrm{R}}(T)$ (roughly, the smallest r such that T is the limit of a sequence of tensors of rank r), and the asymptotic rank $\underset{\sim}{\mathrm{R}}(T)$, where one takes the ranks of large tensor powers of T:

$$\underset{\sim}{\mathrm{R}}(T) := \lim_{n\to\infty} (\mathrm{R}(T^{\otimes n})^{1/n}).$$

For instance, Strassen's algorithm [43] showed that the rank of the tensor for multiplying 2×2 matrices is at most 7. More generally, because of the recursive structure of matrix multiplication under tensor products, to upper bound ω it suffices to bound the asymptotic rank of matrix multiplication tensors. It also suffices to bound the border rank because of a connection due to Bini [8] which showed that, for any tensor T, we have

$$\underset{\sim}{\mathrm{R}}(T) \leqslant \underline{\mathrm{R}}(T) \leqslant \mathrm{R}(T).$$

More recently, the asymptotic rank of tensors has found other algorithmic applications beyond matrix multiplication. Björklund and Kaski [10] and Pratt [34] showed that improved asymptotic rank bounds for certain tensors would give a faster algorithm for the *Set Cover* problem, which is conjectured in fine-grained complexity theory to be impossible. Follow-up work showed similar connections for other algorithmic problems including computing the chromatic number of a graph [9] and computing the permanent of a matrix [11]. Many exponential and parameterized algorithms rely on a (generalized) convolution problem, the complexity of which is determined by the asymptotic rank of the corresponding tensor [14].

Amazingly, Strassen conjectured that *every* tensor has the minimum possible asymptotic rank given its dimensions (as long as it is "tight", a technical condition satisfied by most tensors of interest):

▶ **Conjecture 1** (Asymptotic Rank Conjecture [45]). *Every $d \times d \times d$ tensor T which is concise and tight has $\underset{\sim}{\mathrm{R}}(T) = d$.*

This would imply $\omega = 2$ and lead to breakthrough algorithms for all the aforementioned problems.

Unfortunately, despite the importance of asymptotic rank, we don't know many techniques for bounding the asymptotic rank of a tensor T beyond bounding the border rank of T or a small tensor power of T. Even this approach, which often gives suboptimal bounds, is difficult to use, as bounding the rank or border rank of fairly small tensors has proven to be quite challenging. For example, it remains an open problem to determine the rank or border rank of $\langle 3, 3, 3\rangle$, the tensor for multiplying 3×3 matrices (which is a $9 \times 9 \times 9$ tensor).

Beyond this, the two most successful techniques for bounding asymptotic rank for the purposes of bounding ω have been the laser method (a combinatorial technique for restricting tensor powers of structured tensors into direct sums of matrix multiplication tensors), and the *asymptotic sum inequality* (a.k.a. *Schönhage's τ theorem*) for taking advantage of such direct sums:

▶ **Theorem 2** (Asymptotic Sum Inequality [38]). *Suppose a direct sum of matrix multiplication tensors has a border rank bound*

$$\underline{\mathrm{R}}\left(\bigoplus_{i=1}^{\ell} \langle n_i, m_i, p_i\rangle\right) \leqslant r,$$

and that this is nontrivial (i.e., $n_i m_i p_i > 1$ for at least one i). Let τ be the unique solution of the transcendental equation

$$\sum_{i=1}^{\ell} (n_i m_i p_i)^{\tau} = r.$$

Then, $\omega \leqslant 3\tau$.

It was then shown in follow-up work of Coppersmith and Winograd [22] that the asymptotic sum inequality is *never tight*: One always has the strict inequality $\omega < 3\tau$! Their proof is constructive, in the sense that they found a systematic way to convert a given direct sum construction $\underline{R}(T) \leqslant r$ into another one $\underline{R}(T') \leqslant r'$, which yields a *more effective* bound on the matrix multiplication exponent ω.

This type of result is reminiscent of Blum's classical *speedup* theorem for Turing machines [13], and so we refer to such results as speedup theorems in this paper. That said, Blum's speedup theorem concerns non-constructible time functions, whereas the tensor speedup theorems we discuss here concern explicit problems of great interest like matrix multiplication.

Our approach. In this paper, we revisit the approach of bounding the asymptotic ranks of tensors via speedup theorems. We establish a generalization of prior speedup theorems, and analyze them using a method we call *Strassen calculus* which makes use of Strassen's theory of the asymptotic spectrum, in order to give new quantitative asymptotic rank bounds. We will discuss this in much more detail in Section 2 below, but we first present some applications of our approach.

Small Coppersmith–Winograd tensor. As a main application, we give the first nontrivial upper bound on the asymptotic rank of the *small Coppersmith–Winograd tensor*, cw_q. This is, for any integer $q \geqslant 2$, the $(q+1) \times (q+1) \times (q+1)$ tensor given by

$$\mathrm{cw}_q = \sum_{i=1}^{q} x_0 y_i z_i + x_i y_0 z_i + x_i y_i z_0.$$

The cw_q tensor was introduced in the celebrated work of Coppersmith and Winograd [23]. They further developed Strassen's laser method, and applied it first to cw_q as their easy (warmup) construction [23, §6], then later to the *big Coppersmith–Winograd tensor*, CW_q, the $(q+2) \times (q+2) \times (q+2)$ tensor which adds three "corner terms" to cw_q,

$$\mathrm{CW}_q = x_0 y_0 z_{q+1} + x_0 y_{q+1} z_0 + x_{q+1} y_0 z_0 + \mathrm{cw}_q.$$

Coppersmith and Winograd showed that both cw_q and CW_q have border rank $q+2$ (and thus, asymptotic rank at most $q+2$). By virtue of the extra corner terms, they found CW_q to be more valuable in their laser method analysis, giving their bound $\omega < 2.376$. The big tensor CW_q has since been used for all record-holding bounds on ω, including the current bound $\omega < 2.371339$ [2].

However, the tensor CW_q is now known to be subject to a number of *barrier* results [6, 4, 18, 1] which show that it is *not* valuable enough to prove $\omega = 2$. (The exact numerical barrier depends on which techniques one allows to analyze CW_q, but the most general shows that even with the theoretically optimal asymptotic restrictions/degenerations, one cannot achieve a better bound on ω than 2.16 using CW_q.)

There is no known barrier like this to using cw_q. The key difference is in the dimensions of cw_q compared to CW_q. Since CW_q is a concise (roughly meaning "nondegenerate") tensor of dimensions $(q+2) \times (q+2) \times (q+2)$, it follows that $\underset{\sim}{R}(\mathrm{CW}_q) \geqslant q+2$, and so the bound $\underset{\sim}{R}(\mathrm{CW}_q) \leqslant q+2$ from [23] is tight. By contrast, for cw_q there is a gap between the lower bound $\underset{\sim}{R}(\mathrm{cw}_q) \geqslant q+1$ from its dimensions, and the upper bound $\underset{\sim}{R}(\mathrm{CW}_q) \leqslant q+2$ from [23]. Using their laser method analysis, Coppersmith and Winograd [23, §11] observed that closing this gap could have major implications:

▶ **Theorem 3** ([23]). *If* $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_2) = 3$ *then* $\omega = 2$.

Prior to our work, no bound better than Coppersmith and Winograd's border rank bound $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_2) \leqslant \underline{\mathrm{R}}(\mathrm{cw}_2) = 4$ was known. But because of Theorem 3, many prior works have studied and asked about improved bounds on $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_q)$. See, for instance, [15, Remark 15.44 and Exercise 15.25], [12, Problem 9.8], and [21, Question 3].

Most recently, Conner, Landsberg, Gesmundo, and Ventura [21] studied whether the border ranks of small Kronecker powers $\mathrm{cw}_q^{\otimes k}$ for $k = 2, 3$ could be used to improve the upper bound on $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_q)$, but they achieved a negative result: $\underline{\mathrm{R}}(\mathrm{cw}_q^{\otimes 2}) = (q+2)^2$ (for all $q > 2$) and $\underline{\mathrm{R}}(\mathrm{cw}_q^{\otimes 3}) = (q+2)^3$ (for all $q > 4$). This suggests that one may need to take large Kronecker powers of cw_q before any border rank submultiplicativity is observed.

As an application of our approach for bounding the asymptotic ranks of tensors, we achieve the first upper bound improvement:

▶ **Theorem 4.** $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_2) < 3.931$.

More generally, for any $q \geqslant 2$, we show there is a $\delta_q > 0$ such that $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_q) < q + 2 - \delta_q$; a table of these bounds for $2 \leqslant q \leqslant 10$ is given in Table 1 in Section 7.1 below.

We remark that, in hindsight, the existence of such a $\delta_q > 0$ actually follows from Strassen's original work on the asymptotic spectrum [41, Theorem 3.14]. That said, we believe that a quantitative bound on how large δ_q is requires the new analysis we introduce here. We will discuss this more in the technique overview in the next section.

Improved general upper bound on asymptotic rank. Beyond Theorem 4, we prove additional upper bounds on the asymptotic rank of tensors, as well as new restrictions involving direct sums of tensors of interest. As one example, we obtain a new upper bound on the maximum possible asymptotic rank of a $d \times d \times d$ tensor.

For small d, the generic rank bounds [39, 32] already yield nontrivial estimates, and recent geometric methods can slightly improve them [30, 31]; however, these bounds remain $\Omega(d^2)$. For larger d, the best known general upper bound instead comes from a generic reduction of an arbitrary tensor to matrix multiplication, which implies that every $d \times d \times d$ tensor T satisfies $\underset{\sim}{\mathrm{R}}(T) \leqslant d^{\frac{2}{3}\omega}$ [41].

It is natural to wonder whether this can be improved, and recently Kaski [29, Open Problem 1] asked this explicitly in the context of further understanding sequences of tensors related to Strassen's asymptotic rank conjecture. We improve it as follows:

▶ **Theorem 5.** *For any* $d \geqslant 3$*, any* $d \times d \times d$ *tensor* T *satisfies*

$$\underset{\sim}{\mathrm{R}}(T) \leqslant \sqrt{d^{\frac{4}{3}\omega} - \left\lfloor \frac{d}{3} \right\rfloor \cdot (2^{\frac{\omega}{3}} - 1) d^{\frac{2}{3}\omega}} < d^{\frac{2}{3}\omega}.$$

2 Technique overview

2.1 Strassen calculus: Bounds via asymptotic spectrum

Traditionally, techniques for bounding the matrix multiplication exponent ω proceed by constructing a chain of border-rank upper bounds. This requires carefully tracking the orders of magnitude of intermediate tensors, and typically culminates in a degeneration of the form $\langle n, n, n\rangle \trianglelefteq \langle r\rangle$, which bounds ω via the basic inequality $\omega \leqslant \log_n \underline{\mathrm{R}}(\langle n, n, n\rangle)$. A major drawback is that these arguments often rely on *ad hoc* derivation chains. A prominent example is the standard proof of the asymptotic sum inequality (cf. [15, §15.5], [12, §7]), which hinges on a specific and rather involved degeneration.

Furthermore, it is often unclear whether such a derivation yields the optimal bound theoretically obtainable from a given set of inequalities. For instance, in 1981, Schönhage [38] obtained the (record-holding at the time) bound $\omega < 2.521801$ by proving and analyzing the direct sum degeneration (specifically, setting $k = 5$ and $n = 10$):

$$\langle 1, k, 2n\rangle \oplus \langle n, 2, k\rangle \oplus \langle 2k, n, 1\rangle \trianglelefteq \langle 2(n+1)(k+1)\rangle \oplus k \odot \langle 1, 1, 2\rangle.$$

Schönhage's analysis, however, was not optimal. Romani [36] later improved this estimate to $\omega < 2.516649$ by exploiting the same identity more effectively. Even this is not tight: the best bound obtainable from the identity is $\omega \leqslant 3\tau < 2.516094$, where τ is the solution of the following equation (which looks like a natural generalization of asymptotic sum inequality):

$$3 \cdot (2kn)^{\tau} \leqslant 2(n+1)(k+1) + k \cdot 2^{\tau}, \quad k = 5, n = 10.$$

This was simultaneously noted by Pan [33, Appendix A] and Schönhage (and as claimed in Pan's paper, a full proof is included in a unpublished manuscript of Schönhage). As far as we are aware, there is still no accessible proof of this analysis, likely because the community's attention shifted to other methods to design matrix multiplication algorithms. We provide a proof in Appendix A below that $\omega \leqslant 2.516094$ can be achieved by analyzing this direct sum identity, and is the *optimal* bound one can achieve from it.

In Section 4, we introduce *Strassen calculus* – a systematic framework for deriving bounds from a fixed collection of degenerations between direct sums of tensors – via Strassen's theory of the *asymptotic spectrum* [41, 42, 46].

Informally, the asymptotic spectrum provides a family of numerical "measures" for tensors. Concretely, an element of the asymptotic spectrum $\mathcal{X}$ is a function

$$\phi : \{\mathbb{F}\text{-tensors}\} \to \mathbb{R}_{\geqslant 0}$$

that behaves well with respect to the tensor operations. We refer to such a function ϕ as a *spectral point*; see Section 4 for the precise definitions.

The key structural statement underlying the theory of the asymptotic spectrum is Strassen duality (Theorem 6). Roughly speaking, it says that asymptotic degeneration relations are exactly the relations that hold after applying *all* spectral points: if T asymptotically degenerates to S, then $\phi(T) \leqslant \phi(S)$ for every ϕ, and conversely, if this inequality holds for all ϕ, then this implies the existence of the asymptotic degeneration. In particular, this characterizes asymptotic rank via a max formula,

$$\underset{\sim}{R}(T) = \max_{\phi \in \mathcal{X}} \phi(T).$$

Conceptually, this turns questions about asymptotic rank into optimization problems over the asymptotic spectrum.

Of course, the space $\mathcal{X}$ of all spectral points is defined *nonconstructively* and may be *infinite-dimensional a priori*, so constructing explicit spectral points is generally difficult, although there has been significant progress in the past decade [19, 17, 37]. Our approach is therefore not to "parametrize" $\mathcal{X}$ directly. Instead, we start from a fixed collection of tensor degenerations, translate them into numerical inequalities that must be satisfied by every spectral point, and then use Strassen duality to turn the resulting constraints into explicit upper bounds on asymptotic rank. In favorable regimes one can solve the corresponding optimization problems analytically; more generally, one can obtain sharp bounds via computer-aided evaluation. Moreover, Strassen calculus is *complete*: for a fixed

set of degenerations, optimizing over these constraints yields the best bound obtainable from those relations. In Appendix A, we provide a rigorous treatment of the notion of "optimal usage", and show the completeness under this formalism.

We emphasize that although the theory of the asymptotic spectrum was originally motivated by the study of tensor rank, its subsequent development has followed a curious trajectory. While the framework has flourished in other areas such as (quantum) information theory, its application to the *asymptotic rank* of tensors has remained largely conceptual. Prior to this work, the theory occupied a somewhat dormant role in its own founding domain, as to the best of our knowledge, *no* practical instances were identified to yield concrete, improved upper bounds.

The following examples illustrate the utility of the framework in other areas, and highlight the gap between its success elsewhere and its (so far) limited impact on tensor rank:

- *Strassen's Foundation.* In his seminal paper [41], Strassen introduced the asymptotic spectrum to provide a new proof of the asymptotic sum inequality, setting the stage for a theory that, while foundational, remained difficult to apply to specific new bounds.
- *Barriers to Fast Matrix Multiplication.* Improvements to the exponent ω over the past thirty years have largely relied on analyzing the big Coppersmith–Winograd tensor CW_q. However, Christandl, Vrana, and Zuiddam [18] utilized spectral points to show that the laser method – even in a highly generalized form – applied on big CW tensors, cannot yield an upper bound better than 2.16. Subsequent work [16] has established analogous barriers for the rectangular case.
- *Graph Theory.* Zuiddam [49] applied the general theory of asymptotic spectrum (not confined to tensors) to study the Shannon capacity $\Theta(G)$ of graphs. Wigderson and Zuiddam [47, Theorem 4.1] utilized the asymptotic spectrum to show a striking equivalence: for graphs G and H, where $G \boxtimes H$ and $G \sqcup H$ denote the strong product and disjoint union respectively, $\Theta(G \boxtimes H) > \Theta(G)\Theta(H)$ if and only if $\Theta(G \sqcup H) > \Theta(G) + \Theta(H)$.[1]
- *Circuit Complexity.* Robere and Zuiddam [35] developed an analogous theory to study and generalize some results in the complexity of catalytic branching programs. Alman and Li [3] applied asymptotic spectrum theory to derive a duality theorem for depth-2 circuits, resolving the problem of optimally combining fixed constructions in that context.

These developments demonstrate that Strassen duality is a mature and productive framework in other contexts. Our work aims to *re-apply* this machinery to its original setting by identifying concrete tensor degenerations for which it provably yields new, improved upper bounds.

In Section 4 we provide two self-contained examples of Strassen calculus. They focus on how one can turn degenerations of the form

$$T \oplus \langle 1, t, 1\rangle \trianglelefteq \langle r\rangle \oplus \langle 1, s, 1\rangle$$

into explicit upper bounds on asymptotic rank. On the one hand, from such identities – with rectangular matrix multiplication tensors on the right-hand side – it was not previously understood what the optimal resulting asymptotic rank bound is. On the other hand, relations of this type (and more complicated variants) arise naturally in our generalized speedup theorems. For this reason, it is important to use the Strassen calculus in conjunction with our speedups.

[1] To appreciate this result, note that examples of graphs G, H satisfying the first strict inequality were found in [26], yet it took decades before examples satisfying the second were discovered [5].

In particular, in Section 4.2, our first example is a *toy example* illustrating the basic principles of the framework, together with a detailed derivation. Despite its simplicity, it already yields a novel and interesting consequence: every $3 \times 3 \times 3$ tensor T satisfies $\underset{\sim}{\mathrm{R}}(T) \leqslant 3 + 2^{\omega/3}$ (Corollary 12), giving a first glimpse of the calculus in action. (Previously, to the best of our knowledge, the upper bound on $\underset{\sim}{\mathrm{R}}(T)$ for a generic $3 \times 3 \times 3$ tensor T was 5, which is the generic rank. This result improves it to $3 + 2^{\omega/3} < 4.7297$.)

2.2 Speedup theorems

In Section 5, we prove a general speedup theorem that subsumes the results of Coppersmith–Winograd [22] and Strassen [41, Theorem 3.14]. The Coppersmith–Winograd theorem fits the following paradigm: starting from a border rank bound $T \trianglelefteq \langle r \rangle$, one can (in certain regimes) "speed up" the degeneration by taking a direct sum with a one-slice matrix multiplication tensor on both sides, obtaining

$$T \oplus \langle 1, t, 1 \rangle \trianglelefteq \langle r \rangle \oplus \langle 1, s, 1 \rangle.$$

Strassen later generalized this by showing that the right-hand side does not necessarily need to be a diagonal tensor (i.e., a border rank bound). Specifically, a degeneration $T \trianglelefteq S$ can also be accelerated to $T \oplus \langle 1, t, 1 \rangle \trianglelefteq S \oplus \langle 1, s, 1 \rangle$. Our main theorem further generalizes this idea: under suitable hypotheses, it allows one to add *arbitrary* tensors on both sides of a degeneration, producing relations of the form $T \oplus T' \trianglelefteq S \oplus S'$.

In Section 6, we present three concrete instantiations of the general speedup theorem. Each has different strengths and ranges of applicability; together, they form the main toolkit we use to derive explicit upper bounds on asymptotic rank.

- The first result relates a certain variant of the min-rank problem to one-slice speedups of the form $T \oplus \langle 1, t, 1 \rangle \trianglelefteq S \oplus \langle 1, s, 1 \rangle$, thereby recovering Strassen's speedup theorem.
- The second result shows that, in a suitable regime, a degeneration of the form $T \oplus \langle 1, t, 1 \rangle \trianglelefteq S'$ can be further accelerated to $T \oplus \langle 1, t', 1 \rangle \trianglelefteq S'$ for some $t' > t$, while keeping the right-hand side fixed. In particular, if one takes an identity obtained from the first result and takes a tensor power of both sides, this acceleration always applies; thus the first method can be *iterated*. This generalizes a result of Coppersmith and Winograd [22, Theorem B].
- The third result shows that, in a certain regime, one can add multiple one-slice tensors on both sides, obtaining degenerations of the form $T \oplus \bigoplus_i \langle 1, t_i, 1 \rangle \trianglelefteq S \oplus \bigoplus_j \langle 1, s_j, 1 \rangle$.

As a motivating illustration, we use these methods to show that any non-minimal border rank bound $\underline{\mathrm{R}}(T) \leqslant r$ can be strictly improved (a *qualitative* fact first established by Strassen [41, Theorem 3.14]). The three instantiations above are formalized as Theorem 26, Theorem 28, and Theorem 29, and we give a *quantitative* comparison of the improvement achieved by each method using Strassen calculus.

In Section 7, we apply these methods to obtain explicit bounds for concrete tensors of interest.

- In Section 7.1, we prove Theorem 4, improving the asymptotic rank of the small Coppersmith–Winograd tensors cw_q; for instance, we improve the best known bound for $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_2)$ from 4 (coming from border rank) to 3.931 (see Table 1).
- In Section 7.2, we prove Theorem 5, improving general upper bounds on asymptotic rank.
- In Section 7.3, we use the speedup theorem to construct new degenerations between direct sums of matrix multiplication tensors in which the right-hand side is itself a direct sum of nontrivial matrix multiplication tensors. While this does not (at present) improve the matrix multiplication exponent, the resulting identities do not appear to follow easily from previously known ones, and may provide insights for future developments.

Finally, it is natural to wonder whether this method yields faster algorithms for problems such as matrix multiplication or set cover. We find that our techniques are still not effective enough, although we are optimistic that further improvements along these lines may succeed. See Remark 35 and Remark 40 for further discussion.

3 Preliminaries

For a positive integer n, let $[n] := \{1, \dots, n\}$. We use Iverson bracket notation: for a proposition P, let $[\![P]\!] = 1$ if P is true and $[\![P]\!] = 0$ otherwise.

3.1 Tensors

Our notation largely follows [24, 48, 2].

Fix a base field $\mathbb{F}$. Let U, V, W be finite-dimensional vector spaces with bases $\{x_1, \dots, x_n\}$ of U, $\{y_1, \dots, y_m\}$ of V and $\{z_1, \dots, z_p\}$ of W. A tensor is an element $T \in U \otimes V \otimes W$. With respect to the chosen bases, we write T via its coefficients $a_{ijk} \in \mathbb{F}$ as

$$T = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} a_{ijk} x_i y_j z_k.$$

We say U, V, W are the *modes* of the tensor.

For two tensors $S \in U \otimes V \otimes W$ and $T \in U' \otimes V' \otimes W'$, we write their direct sum $S \oplus T$ as the image of their sum inside the common space $(U \oplus U') \otimes (V \oplus V') \otimes (W \oplus W')$, and their tensor product as the natural image inside the tensor product space $(U \otimes U') \otimes (V \otimes V') \otimes (W \otimes W')$.

A *restriction* from $S \in U \otimes V \otimes W$ to $T \in U' \otimes V' \otimes W'$ is specified by linear maps $A \colon U \to U'$, $B \colon V \to V'$ and $C \colon W \to W'$ such that $T = (A \otimes B \otimes C)S$. Concretely,

$$T = \sum_{i,j,k} a_{ijk} A(x_i) B(y_j) C(z_k).$$

In this case we say S *restricts to* T, and write $T \leqslant S$.

A *degeneration* is a one-parameter family of linear maps $A_\lambda, B_\lambda, C_\lambda$ (with entries rational in λ) such that $(A_\lambda \otimes B_\lambda \otimes C_\lambda)S = T + O(\lambda^d)$ for some d. In other words, write each entry of $A_\lambda, B_\lambda, C_\lambda$ as a rational function in λ and then expand the resulting expression as a Laurent series in λ:

$$A_\lambda = \sum_{i \geqslant i_0} A_i \lambda^i,$$

and similarly for B_λ, C_λ, we have the λ^ℓ-coefficient of $(A_\lambda \otimes B_\lambda \otimes C_\lambda)S$ is

$$\sum_{i+j+k=\ell} (A_i \otimes B_j \otimes C_k)S.$$

We require that the λ^0-coefficient is exactly T, and that the λ^ℓ-coefficient is zero for all $\ell < 0$. In this case we say T is a degeneration of S, write S *degenerates to* T, and denote this by $T \trianglelefteq S$.

Two central families of tensors are diagonal tensors and matrix multiplication tensors. For any $n \geqslant 1$, the diagonal tensor is

$$\langle n \rangle = \sum_{i=1}^{n} x_i y_i z_i,$$

and for any $n, m, p \geqslant 1$, the matrix multiplication tensor is

$$\langle n, m, p \rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} x_{ij} y_{jk} z_{ki}.$$

The rank $\mathrm{R}(T)$ of a tensor T is the minimum r such that $T \leqslant \langle r \rangle$. Similarly, the subrank $\mathrm{Q}(T)$ is the maximum q such that $\langle q \rangle \leqslant T$ (equivalently, $T \geqslant \langle q \rangle$). The *border rank* $\underline{\mathrm{R}}(T)$ and *border subrank* $\underline{\mathrm{Q}}(T)$ are defined analogously by replacing $\leqslant$ with $\trianglelefteqslant$.

We write the n-fold direct sum of T as $n \odot T$; note that $n \odot T$ is equivalent to $\langle n \rangle \otimes T$.

The *asymptotic rank* and *subrank* are defined as

$$\underset{\sim}{\mathrm{R}}(T) = \lim_{n \to \infty} \mathrm{R}(T^{\otimes n})^{1/n},$$
$$\underset{\sim}{\mathrm{Q}}(T) = \lim_{n \to \infty} \mathrm{Q}(T^{\otimes n})^{1/n}.$$

By Fekete's lemma, for $\underset{\sim}{\mathrm{R}}$ ($\underset{\sim}{\mathrm{Q}}$ resp.), the $\lim_{n\to\infty}$ can also be replaced by $\inf_{n \geqslant 1}$ ($\sup_{n \geqslant 1}$ resp.). Therefore, the matrix multiplication exponent ω is characterized by $\underset{\sim}{\mathrm{R}}(\langle n, n, n \rangle) = n^{\omega}$.

4 Strassen calculus

We first briefly recall the notion of the asymptotic spectrum. The asymptotic spectrum of $\mathbb{F}$-tensors is the set $\mathcal{X} = \mathcal{X}_{\mathbb{F}}$ of functions ϕ from tensors to nonnegative real numbers $\mathbb{R}_{\geqslant 0}$ satisfying:

- Additive under direct sum: $\phi(T \oplus S) = \phi(T) + \phi(S)$.
- Multiplicative under tensor product: $\phi(T \otimes S) = \phi(T)\phi(S)$.
- Normalized on diagonal tensors: $\phi(\langle n \rangle) = n$.
- Monotone under restriction (degeneration[2]): $\phi(T) \leqslant \phi(S)$ when $T \leqslant S$ ($T \trianglelefteqslant S$).

From this definition, it is immediate that $\mathcal{X}$ only "remembers" asymptotic relations: if $T^{\otimes n} \leqslant S^{\otimes n + o(n)}$, then $\phi(T) \leqslant \phi(S)$ for all $\phi \in \mathcal{X}$. Surprisingly, the converse holds as well; this is the *Strassen duality theorem* (cf. [41, Theorem 3.2], [47, §3]).

▶ **Theorem 6** (Strassen duality). *For two tensors T, S, the following are equivalent:*

1. $T^{\otimes n} \leqslant S^{\otimes n + o(n)}$,
2. $T^{\otimes n} \trianglelefteqslant S^{\otimes n + o(n)}$,
3. *For any $\phi \in \mathcal{X}$, have $\phi(T) \leqslant \phi(S)$.*

Moreover,

$$\underset{\sim}{\mathrm{R}}(T) = \lim_{n \to \infty} \mathrm{R}(T^{\otimes n})^{1/n} = \lim_{n \to \infty} \underline{\mathrm{R}}(T^{\otimes n})^{1/n} = \max_{\phi \in \mathcal{X}} \phi(T),$$

and

$$\underset{\sim}{\mathrm{Q}}(T) = \lim_{n \to \infty} \mathrm{Q}(T^{\otimes n})^{1/n} = \lim_{n \to \infty} \underline{\mathrm{Q}}(T^{\otimes n})^{1/n} = \min_{\phi \in \mathcal{X}} \phi(T).$$

For our purposes, it is enough to know Strassen duality; no further background on the asymptotic spectrum is required. For a complete treatment – including further properties and applications – we refer to Strassen's founding paper [41] and the survey of Wigderson and Zuiddam [47].

[2] It turns out that *restriction* and *degeneration* give equivalent definitions, i.e., they yield the same collection of functions; cf. [41, Theorem 3.1].

4.1 Matrix multiplication

Before proceeding to the toy example, we first introduce a fundamental concept to facilitate our use of Strassen calculus.

▶ **Proposition 7** ([41, Proposition 4.3]). *For any $\phi \in \mathcal{X}$, there exist unique real numbers $\theta_1, \theta_2, \theta_3$ such that*

$$\phi(\langle n, m, p\rangle) = n^{\theta_1} m^{\theta_2} p^{\theta_3}$$

for any n, m, p.

Proof. Let $\theta_1 = \log_2 \phi(\langle 2, 1, 1\rangle)$. We will show that $\phi(\langle n, 1, 1\rangle) = n^{\theta_1}$ for all n. For any k, let $t = \lfloor \log_2(n^k) \rfloor$, thus $2^t \leqslant n^k < 2^{t+1}$ and we have the restrictions $\langle 2^t, 1, 1\rangle \leqslant \langle n^k, 1, 1\rangle \leqslant \langle 2^{t+1}, 1, 1\rangle$. Since ϕ is monotone, we have

$$\phi(\langle 2^t, 1, 1\rangle) \leqslant \phi(\langle n^k, 1, 1\rangle) \leqslant \phi(\langle 2^{t+1}, 1, 1\rangle).$$

Since $\langle n^k, 1, 1\rangle = \langle n, 1, 1\rangle^{\otimes k}$ and ϕ is multiplicative, we can pull out the exponent:

$$\phi(\langle 2, 1, 1\rangle)^t \leqslant \phi(\langle n, 1, 1\rangle)^k \leqslant \phi(\langle 2, 1, 1\rangle)^{t+1}.$$

Thus

$$2^{-\theta_1} n^{k\theta_1} \leqslant 2^{\theta_1 t} \leqslant \phi(\langle n, 1, 1\rangle)^k \leqslant 2^{\theta_1(t+1)} \leqslant 2^{\theta_1} n^{k\theta_1}.$$

Taking kth roots and letting $k \to \infty$, we conclude $\phi(\langle n, 1, 1\rangle) = n^{\theta_1}$. Similarly, we can obtain θ_2, θ_3 such that $\phi(\langle 1, m, 1\rangle) = m^{\theta_2}$ and $\phi(\langle 1, 1, p\rangle) = p^{\theta_3}$.

Finally, since $\langle n, m, p\rangle = \langle n, 1, 1\rangle \otimes \langle 1, m, 1\rangle \otimes \langle 1, 1, p\rangle$ and ϕ is multiplicative, we have

$$\begin{aligned}\phi(\langle n, m, p\rangle) &= \phi(\langle n, 1, 1\rangle \otimes \langle 1, m, 1\rangle \otimes \langle 1, 1, p\rangle) \\ &= \phi(\langle n, 1, 1\rangle) \cdot \phi(\langle 1, m, 1\rangle) \cdot \phi(\langle 1, 1, p\rangle) \\ &= n^{\theta_1} m^{\theta_2} p^{\theta_3}.\end{aligned}$$

◀

Therefore, the behavior of the asymptotic spectrum on matrix multiplication tensors can be parametrized just by θ_1, θ_2 and θ_3. This leads to the notion of the *spectrum of matrix multiplication*.

▶ **Definition 8** ([41, §4]). *The (logarithmic) spectrum $\Delta_{\mathrm{MM}} \subset \mathbb{R}^3$ of matrix multiplication are the points $(\theta_1, \theta_2, \theta_3)$ that satisfy Proposition 7.*

All the basic facts of matrix multiplication tensor transfer to statements of Δ_{MM}.

▶ **Proposition 9.** *Every θ_i lies in $[0, 1]$. The value $\theta_1 + \theta_2 + \theta_3$ over Δ_{MM} attains minimum 2 and maximum ω.*

Proof. For any positive integer n, since $\underset{\sim}{\mathrm{Q}}(\langle n, 1, 1\rangle) = 1$ and $\underset{\sim}{\mathrm{R}}(\langle n, 1, 1\rangle) = n$, we have $\min \theta_1 = 0$ and $\max \theta_1 = 1$, and similarly for θ_2, θ_3. The value $\theta_1 + \theta_2 + \theta_3$, by definition, equals $\log_n \phi(\langle n, n, n\rangle)$. Thus by Theorem 6, we have

$$\max_{(\theta_1,\theta_2,\theta_3)\in\Delta_{\mathrm{MM}}} (\theta_1 + \theta_2 + \theta_3) = \max_{\phi\in\mathcal{X}} \log_n \phi(\langle n, n, n\rangle) = \log_n \underset{\sim}{\mathrm{R}}(\langle n, n, n\rangle),$$

and it follows from the definition of ω that $\underset{\sim}{\mathrm{R}}(\langle n, n, n\rangle) = n^\omega$. Similarly, we have

$$\min_{(\theta_1,\theta_2,\theta_3)\in\Delta_{\mathrm{MM}}} (\theta_1 + \theta_2 + \theta_3) = \log_n \underset{\sim}{\mathrm{Q}}(\langle n, n, n\rangle),$$

and matrix multiplication has optimal asymptotic subrank $\underset{\sim}{\mathrm{Q}}(\langle n, n, n\rangle) = n^2$ [40, Theorem 6.6]. ◀

4.2 Toy example

▶ **Proposition 10.** *Suppose $\mathbb{F}$ is an algebraically closed field. For any $3 \times 3 \times 3$ tensor T, we have $T \trianglelefteq \langle 3 \rangle \oplus \langle 1, 2, 1 \rangle$.*

Proof. It is known that degeneration can be characterized topologically (cf. [40, Theorem 5.8], [15, Theorem 20.24]). Let $C \subseteq \mathbb{F}^3 \otimes \mathbb{F}^3 \otimes \mathbb{F}^3$ be the set of tensors that can be restricted from $\langle 3 \rangle \oplus \langle 1, 2, 1 \rangle$; then $T \trianglelefteq \langle 3 \rangle \oplus \langle 1, 2, 1 \rangle$ if and only if $T \in \overline{C}$, where $\overline{C}$ denotes the Zariski closure of C. Thus, it suffices to prove $T \leqslant \langle 3 \rangle \oplus \langle 1, 2, 1 \rangle$ for a *generic* T, and the statement follows.

If T is generic, we may first assume that the z_1-slice of T is a nonsingular matrix. By a change of basis, we assume that the z_1-slice is of the form $x_1y_1 + x_2y_2 + x_3y_3$. Furthermore, we can assume the z_2-slice is a diagonalizable matrix; we then apply the corresponding similarity transformation which preserves the form of the first slice and transforms the second slice into $\alpha_1 x_1 y_1 + \alpha_2 x_2 y_2 + \alpha_3 x_3 y_3$. Now we can write the tensor T as

$$\sum_{i=1}^{3} x_i y_i (z_1 + \alpha_i z_2) + (xMy) z_3$$

for some matrix M. Letting λ be one of the eigenvalues of M, then we have

$$T = \sum_{i=1}^{3} x_i y_i (z_1 + \alpha_i z_2 + \lambda z_3) + x(M - \lambda I) y z_3$$

where the first summation can be restricted from $\langle 3 \rangle$ and the second term is a matrix of rank at most 2, which is a restriction of $\langle 1, 2, 1 \rangle$. ◀

▶ **Proposition 11.** *If tensor T satisfies the degeneration relations on three directions,*

$$T \trianglelefteq \langle r \rangle \oplus \langle s, 1, 1 \rangle, \tag{4.1}$$
$$T \trianglelefteq \langle r \rangle \oplus \langle 1, s, 1 \rangle, \tag{4.2}$$
$$T \trianglelefteq \langle r \rangle \oplus \langle 1, 1, s \rangle, \tag{4.3}$$

then its asymptotic rank satisfies $\underset{\sim}{\mathrm{R}}(T) \leqslant r + s^{\omega/3}$.

Proof. In this proof, we demonstrate the general principle of *Strassen calculus*.

Step 1: Pass to asymptotic spectrum. Let $\phi \in \mathcal{X}$ be a point of the asymptotic spectrum. Recall that ϕ is *monotone* – it preserves degenerations – so we can apply it to (4.1), resulting in

$$\phi(T) \leqslant \phi(\langle r \rangle \oplus \langle s, 1, 1 \rangle).$$

Using the *additivity* of ϕ, we can pull out the direct sum:

$$\phi(T) \leqslant \phi(\langle r \rangle) + \phi(\langle s, 1, 1 \rangle).$$

Moreover, ϕ is *normalized*, so $\phi(\langle r \rangle) = r$. As shown in Proposition 7, the value of a matrix multiplication tensor can be parametrized by $(\theta_1, \theta_2, \theta_3)$ uniquely determined by ϕ, such that $\phi(\langle s, 1, 1 \rangle) = s^{\theta_1}$. We can apply the same argument to (4.2) and (4.3), obtaining

$$\phi(T) \leqslant r + s^{\theta_1}, \quad \phi(T) \leqslant r + s^{\theta_2}, \quad \phi(T) \leqslant r + s^{\theta_3}.$$

Step 2: Use known inequalities to estimate spectral values. Each of the three degenerations gives an upper bound of $\phi(T)$. To obtain the tightest bound, we take the minimum:

$$\phi(T) \leqslant r + \min\{s^{\theta_1}, s^{\theta_2}, s^{\theta_3}\}.$$

To translate this into an absolute value independent of the spectral point ϕ, we utilize Proposition 9: $\theta_1 + \theta_2 + \theta_3 \leqslant \omega$. We remind the reader that this inequality essentially stems from the fact that $\underset{\sim}{\mathrm{R}}(\langle n, n, n\rangle) = n^\omega$. Since the θ_i are all non-negative, at least one must be less than or equal to the average $\omega/3$, thus

$$\phi(T) \leqslant r + s^{(\theta_1+\theta_2+\theta_3)/3} \leqslant r + s^{\omega/3}.$$

Final step: Return to primal estimate. Now we have an absolute upper bound of $\phi(T)$ for any ϕ. By Strassen duality theorem (Theorem 6), the asymptotic rank $\underset{\sim}{\mathrm{R}}(T)$ of T is the maximum of $\phi(T)$ over $\mathcal{X}$, so we get $\underset{\sim}{\mathrm{R}}(T) \leqslant r + s^{\omega/3}$. ◀

▶ **Corollary 12.** *Any* $3 \times 3 \times 3$ *tensor* T *has* $\underset{\sim}{\mathrm{R}}(T) \leqslant 3 + 2^{\omega/3}$.

Proof. Since asymptotic rank is invariant under field extension [15, Proposition 15.17], we may assume $\mathbb{F}$ is algebraically closed. Then by Proposition 10, we have $T \trianglelefteq \langle 3\rangle \oplus \langle 1, 2, 1\rangle$. Applying Proposition 10 on the tensor with modes relabeled yields the other two degenerations required by Proposition 11. Thus we have $\underset{\sim}{\mathrm{R}}(T) \leqslant 3 + 2^{\omega/3}$. ◀

Previously, to the best of our knowledge, the upper bound on $\underset{\sim}{\mathrm{R}}(T)$ for a generic $3 \times 3 \times 3$ tensor T was 5, which is the generic rank. We see that $3 + 2^{\omega/3} < 3 + 2 = 5$ represents an improvement. Numerically, using the current upper bound for ω, we have $3 + 2^{\omega/3} < 4.7297$.

The generic rank (Strassen–Lickteig [39, 32]) establishes that for any $d \times d \times d$ tensor T, $T \trianglelefteq \langle r_d\rangle$, where

$$r_d = \begin{cases} 5, & \text{if } d = 3, \\ \left\lceil \frac{d^3}{3d-2} \right\rceil, & \text{otherwise.} \end{cases}$$

This bound works well for small d, while for large d the current best asymptotic rank bound comes from $T \leqslant \langle d, 1, d\rangle$ (though we will refine this in Section 7.2). This suggests an alternative approach to improving the asymptotic rank of general tensors:

▶ **Problem 13.** Find an intermediate tensor S_d that admits degenerations for all $d \times d \times d$ tensors (or significant subclasses) to obtain better asymptotic rank estimates.

We leave this question for future research.

▶ Remark 14. For this example, it is not difficult to derive the same upper bound via *traditional derivations*. We encourage the reader to consider such an alternative proof, which may help in understanding the interplay between classical deductions and the operations in Strassen calculus. However, as the given degenerations become more complicated, the complexity of classical deduction grows rapidly and becomes unintuitive. In contrast, the *Strassen calculus* remains systematic, typically adding only a few lines of derivation. In certain regimes, such as those analyzed in Appendix A, the optimum of the optimization problem may be difficult to describe by an easy criterion, making it reasonable to rely on computer-aided evaluation. In such cases, a traditional derivation would need to incorporate different regimes and sensitively detect shifts in behavior as parameters change, which is a prohibitively complicated task.

4.3 Complicated example

In this subsection we present another example of Strassen calculus, based on a slightly more complicated set of identities that will also be used in later applications.

▶ **Proposition 15.** *If tensor T satisfies the degeneration relations on three directions*

$$T \oplus \langle t,1,1\rangle \trianglelefteq \langle r\rangle \oplus \langle s,1,1\rangle, \tag{4.4}$$
$$T \oplus \langle 1,t,1\rangle \trianglelefteq \langle r\rangle \oplus \langle 1,s,1\rangle, \tag{4.5}$$
$$T \oplus \langle 1,1,t\rangle \trianglelefteq \langle r\rangle \oplus \langle 1,1,s\rangle, \tag{4.6}$$

for some $s,t \geqslant 1$, then its asymptotic rank satisfies

$$\underset{\sim}{R}(T) \leqslant \begin{cases} r + s^{2/3} - t^{2/3}, & \text{if } t > s, \\ r + s^{\omega/3} - t^{\omega/3}, & \text{if } t \leqslant s. \end{cases}$$

Proof. Fix an arbitrary spectral point $\phi \in \mathcal{X}$. Applying ϕ to the three degeneration relations yields:

$$\begin{aligned} \phi(T) &\leqslant r + s^{\theta_1} - t^{\theta_1} && \text{(apply } \phi \text{ to (4.4))} \\ \phi(T) &\leqslant r + s^{\theta_2} - t^{\theta_2} && \text{(apply } \phi \text{ to (4.5))} \\ \phi(T) &\leqslant r + s^{\theta_3} - t^{\theta_3} && \text{(apply } \phi \text{ to (4.6))} \end{aligned}$$

where $\phi(\langle r\rangle) = r$ by normalization, and $\phi(\langle t,1,1\rangle) = t^{\theta_1}$, $\phi(\langle 1,t,1\rangle) = t^{\theta_2}$, $\phi(\langle 1,1,t\rangle) = t^{\theta_3}$ by Proposition 7.

Taking the strongest bound among the three directions yields

$$\begin{aligned} \phi(T) &\leqslant r + \min_{i\in\{1,2,3\}} \left(s^{\theta_i} - t^{\theta_i}\right) && \text{(take the minimum over directions)} \\ &= r + \min_{i\in\{1,2,3\}} \tau(\theta_i) && \text{(set } \tau(x) = s^x - t^x\text{).} \end{aligned}$$

Now we split into two cases.

Case 1: $t > s$. Then τ is decreasing on $\mathbb{R}_{\geqslant 0}$, so $\min_i \tau(\theta_i) = \tau(\max_i \theta_i)$. Since $\theta_1+\theta_2+\theta_3 \geqslant 2$ (Proposition 9), we have $\max_i \theta_i \geqslant 2/3$, and thus

$$\phi(T) \leqslant r + \tau(\max_i \theta_i) \leqslant r + \tau(2/3) = r + s^{2/3} - t^{2/3}.$$

Case 2: $t \leqslant s$. Then τ is increasing on $\mathbb{R}_{\geqslant 0}$, so $\min_i \tau(\theta_i) = \tau(\min_i \theta_i)$. Since $\theta_1+\theta_2+\theta_3 \leqslant \omega$ (Proposition 9), we have $\min_i \theta_i \leqslant (\theta_1+\theta_2+\theta_3)/3 \leqslant \omega/3$, and therefore

$$\phi(T) \leqslant r + \tau(\min_i \theta_i) \leqslant r + \tau(\omega/3) = r + s^{\omega/3} - t^{\omega/3}.$$

Since the above inequality holds for every $\phi \in \mathcal{X}$, Strassen duality gives

$$\underset{\sim}{R}(T) = \max_{\phi\in\mathcal{X}} \phi(T) \leqslant \begin{cases} r + s^{2/3} - t^{2/3}, & \text{if } t > s, \\ r + s^{\omega/3} - t^{\omega/3}, & \text{if } t \leqslant s. \end{cases}$$ ◀

▶ Remark 16. The proof illustrates a typical feature of Strassen calculus: it can *turn lower bounds into upper bounds*. In the case $t > s$ we used the nontrivial inequality $\theta_1+\theta_2+\theta_3 \geqslant 2$, which ultimately comes from the optimal asymptotic subrank bound $\underset{\sim}{Q}(\langle n,n,n\rangle) = n^2$.

5 Speedup theorems

5.1 Notation and blockwise restrictions

To clarify our statements, we first introduce notation for handling tensor modes and restrictions. Let $T \in U' \otimes V' \otimes W'$ and $S \in U \otimes V \otimes W$ be tensors. If a tensor mode is identified as a direct sum, such as $U = U_1 \oplus U_2$ and $U' = U'_1 \oplus U'_2$, the linear map A in a restriction $T \leqslant S$ can be written in blockwise form:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

where A_{ij} is a linear map from U_j to U'_i. Correspondingly, we may decompose the tensor S as $S = S_1 + S_2$ where $S_j \in U_j \otimes V \otimes W$. In this context, the restriction $T = (A \otimes B \otimes C)S$ can be expanded as:

$$T = \sum_{i,j} (A_{ij} \otimes B \otimes C)(S_j).$$

We also need to clarify the relationship between subspaces of functional spaces and the linear maps used in restrictions. Let $U^\vee$ denote the dual space of U, consisting of all linear functions $f : U \to \mathbb{F}$. Suppose $A \subseteq U^\vee$ is a subspace of dimension d. We can identify the entire subspace A with a surjective linear map $U \to \mathbb{F}^d$: choosing a basis $\{f_1, \ldots, f_d\}$ for A defines the coordinates of the image in $\mathbb{F}^d$. While this specific map depends on the choice of basis, any two such choices yield the same resulting tensor up to isomorphism. Thus, we may treat a subspace of $U^\vee$ as an "aggregate" linear map in a restriction $(A \otimes B \otimes C)S$ without ambiguity.

5.2 Free-lunch speedup theorem

We begin by describing a condition that allows one to obtain a direct sum on the left-hand side *for free*, without the need to add any additional terms to the right-hand side. For readers unfamiliar with these works, it is worth noting that the arguments of Coppersmith–Winograd [22, Theorem A] and Strassen [41, Lemma 3.11] likewise proceed in two distinct stages. Our approach follows this established logic, beginning here with the formulation of the first step.

▶ **Theorem 17** (Free-lunch speedup). *Let $T \in U' \otimes V' \otimes W'$ and $S \in U \otimes V \otimes W$ be two tensors, with a restriction $T \leqslant S$ given by the maps (A, B, C). Let $C^\perp \subseteq W^\vee$ be the subspace of linear functions $f : W \to \mathbb{F}$ such that $(A \otimes B \otimes f)S = 0$. For any chosen subspace $C' \subseteq C^\perp$, define the subspaces $A' \subseteq U^\vee$ and $B' \subseteq V^\vee$ as follows:*

$$A' = \{u \in U^\vee : (u \otimes B \otimes C')S = 0\},$$
$$B' = \{v \in V^\vee : (A \otimes v \otimes C')S = 0\}.$$

Then we have the direct sum degeneration:

$$T \oplus T' \trianglelefteq S,$$

where $T' = (A' \otimes B' \otimes C')S$.

Proof. The construction of the degeneration proceeds in two steps.

Step 1. We first construct a restriction by augmenting the linear maps. We define the new maps as (A, A'), (B, B'), and (C, C'), where the block structure corresponds to the direct sum of the target spaces. Consider the resulting tensor:

$$R = \left[\begin{pmatrix} A \\ A' \end{pmatrix} \otimes \begin{pmatrix} B \\ B' \end{pmatrix} \otimes \begin{pmatrix} C \\ C' \end{pmatrix} \right] S.$$

We examine the blocks of R arising from the terms involving C'. By the construction of our subspaces:

- $(A \otimes B \otimes C')S = 0$ by the definition of C'.
- $(A' \otimes B \otimes C')S = 0$ by the definition of A'.
- $(A \otimes B' \otimes C')S = 0$ by the definition of B'.

Consequently, the only potentially non-zero block involving C' is $(A' \otimes B' \otimes C')S = T'$. In the overall block structure of R, the diagonal blocks are $R_{111} = T$ and $R_{222} = T'$, while the blocks R_{112}, R_{122}, and R_{212} are guaranteed to be zero.

Step 2. We now eliminate the remaining junk terms (namely R_{121}, R_{211}, and R_{221}) via a *monomial degeneration*. We apply the following diagonal transformation to the tensor R:

$$\begin{pmatrix} 1 & \\ & \lambda \end{pmatrix} \otimes \begin{pmatrix} 1 & \\ & \lambda \end{pmatrix} \otimes \begin{pmatrix} \lambda^2 & \\ & 1 \end{pmatrix} R = \lambda^2 (T + T') + O(\lambda^3).$$

By scaling by λ^{-2} and taking the limit as $\lambda \to 0$, the junk terms are eliminated, yielding the direct sum $T \oplus T'$. In conclusion, we have obtained $T \oplus T' \trianglelefteq R \leqslant S$, which implies $T \oplus T' \trianglelefteq S$. ◀

One might be concerned about why Theorem 17 only speaks about starting with a *restriction* rather than *degeneration*. This is because the degeneration version is obtained by bootstrapping the result: It is not using any specific property of degeneration, but just regarding it as a restriction over the field extension $\mathbb{F}(\lambda)$.

▶ **Corollary 18** (Free-lunch speedup, degeneration version)**.** *Let $T \trianglelefteq S$ be a degeneration of $\mathbb{F}$-tensors. More specifically, let T_λ be a $\mathbb{F}(\lambda)$-tensor before taking $\lambda \to 0$, i.e.,*

$$T + O(\lambda) = T_\lambda = (A_\lambda \otimes B_\lambda \otimes C_\lambda)S$$

is the restriction of $\mathbb{F}(\lambda)$-tensors. Plug in $T \leqslant S$ with data A_λ, B_λ and C_λ into Theorem 17, for any choice of C' thereby, let the resulting $\mathbb{F}(\lambda)$ tensor be T'_λ, T' be the lowest order term of T'_λ (0 if T'_λ is completely zero). Then

$$T \oplus T' \trianglelefteq S.$$

Proof. Firstly, by Theorem 17, we obtain a degeneration of $\mathbb{F}(\lambda)$-tensors

$$T_\lambda \oplus T'_\lambda \trianglelefteq S,$$

say this degeneration is parametrized by another symbol ε. We may scale by λ and ε, to assume that the degeneration is given by $A_{\lambda,\varepsilon}, B_{\lambda,\varepsilon}$ and $C_{\lambda,\varepsilon}$, and $T'_\lambda = T' + O(\lambda)$, such that

$$(T \oplus T') + O(\lambda) + O_\lambda(\varepsilon) = (A_{\lambda,\varepsilon} \otimes B_{\lambda,\varepsilon} \otimes C_{\lambda,\varepsilon})S.$$

The notation means that $O(\lambda)$ term is purely a λ-high order term tensor over $\mathbb{F}$, but $O_\lambda(\varepsilon)$ term is a ε-high order term tensor that can also contain λ. One can again reduce to the

regular definition of degeneration by substituting ε by a large enough power $\varepsilon = \lambda^k$, then the whole restriction again becomes

$$(T \oplus T') + O(\lambda) = (A_{\lambda,\lambda^k} \otimes B_{\lambda,\lambda^k} \otimes C_{\lambda,\lambda^k})S.$$

This concludes the degeneration. ◀

5.2.1 Isolated summand

We now introduce the notion of an *isolated* summand, a property defined by Coppersmith and Winograd [22, Definition 3.1] that is crucial for iterating speedup theorems.

Let $T \oplus T' \trianglelefteq S$ be a degeneration defined by the polynomial maps A_λ, B_λ, and C_λ. Let $R_\lambda = (A_\lambda \otimes B_\lambda \otimes C_\lambda)S$ denote the $\mathbb{F}(\lambda)$-tensor prior to taking the limit $\lambda \to 0$. By definition, we must have $(R_\lambda)_{111} = T + O(\lambda)$ and $(R_\lambda)_{222} = T' + O(\lambda)$, while all other blocks $(R_\lambda)_{ijk}$ (where indices are not all 1 or all 2) must satisfy $(R_\lambda)_{ijk} = O(\lambda)$. We say that the direct summand T' is an *isolated summand* in the third mode with respect to the degeneration $(A_\lambda, B_\lambda, C_\lambda)$ if $(R_\lambda)_{ij2}$ is identically zero for all $(i,j) \neq (2,2)$.

One can verify that this definition essentially coincides with Coppersmith and Winograd's original formulation when $T' = \langle 1, t, 1\rangle$. Moreover, the following facts are readily established.

▶ **Proposition 19.** *The direct summand T' of $T \oplus T' \trianglelefteq S$ produced by Theorem 17 or Corollary 18 is isolated in the third mode.*

The following proposition generalizes [22, Proposition 4.1]:

▶ **Proposition 20.** *Let T_1', T_2' be isolated direct summands of degenerations*

$$T_1 \oplus T_1' \trianglelefteq S_1, \quad T_2 \oplus T_2' \trianglelefteq S_2.$$

Then, in the tensor product of these degenerations

$$(T_1 \otimes T_2) \oplus (T_1 \otimes T_2') \oplus (T_1' \otimes T_2) \oplus (T_1' \otimes T_2') \trianglelefteq S_1 \otimes S_2,$$

the term $T_1' \otimes T_2'$ remains an isolated summand in the third mode.

5.3 Speedup by one slice

In this subsection, we recover the classical speedup theorem of Coppersmith and Winograd, as well as Strassen's generalization, in the case where the tensors appended to both sides are one-slice matrix multiplication tensors. We also refine the notion of *fullness* originally introduced by Coppersmith and Winograd [22, Definition 3.1].

Usually, one cannot hope to directly use the free-lunch speedup theorem, because the first step requires choosing a subspace $C' \subseteq C^\perp$. However, for a reasonable degeneration $T \trianglelefteq S$, the corresponding space $C^\perp$ is typically zero. Indeed, if $C^\perp$ were non-zero, then viewing S as an "upper bound" for T, the tensor S would contain redundant W-slices; removing these slices would essentially yield a more effective upper bound for T.

Therefore, the standard speedup strategy proceeds as follows: First, artificially add a tensor S' to the right-hand side. This new summand S' contributes nothing to the degeneration $T \trianglelefteq S \oplus S'$ because its component in the C-map is set to zero. However, its components in the A and B maps can be carefully chosen to satisfy the conditions of Theorem 17 (or Corollary 18), enabling the extraction of a non-trivial summand T'.

We first examine the implications of the free-lunch speedup theorem when C' is a one-dimensional subspace.

▶ **Proposition 21.** *Assume the setting of Theorem 17. Let $C' \subseteq C^\perp$ be a one-dimensional subspace spanned by a linear function $f : W \to \mathbb{F}$. Let $r = \mathrm{R}((\mathrm{id}_U \otimes \mathrm{id}_V \otimes f)S)$ be the rank of the resulting matrix in $U \otimes V$. Then we have $T' \cong \langle 1, t, 1\rangle$, where*

$$t \geqslant r - n - m,$$

with $n = \dim U'$ and $m = \dim V'$.

Proof. The restriction $T' = (A' \otimes B' \otimes f)S$ can be viewed as first contracting the W-mode: $M = (\mathrm{id}_U \otimes \mathrm{id}_V \otimes f)S$ yields a matrix of rank r. The subspace A' is defined as the kernel of the map $u \mapsto (u \otimes B \otimes f)S$. Since this map maps to V', its kernel A' has codimension at most $\dim V' = m$. Thus, restricting M to A' reduces the rank by at most m. Similarly, restricting to B' reduces the rank by at most n. Consequently, $T' = (A' \otimes B')M$ has rank at least $t \geqslant r - n - m$. A rank-t matrix (viewed as a tensor) is isomorphic to $\langle 1, t, 1\rangle$. ◀

We now explain how to append a one-slice tensor to both sides of the degeneration.

▶ **Proposition 22** (One-slice speedup [41, Lemma 3.12])**.** *Let $T \leqslant S$ be a restriction defined by maps A, B, C. For a linear function $f : W \to \mathbb{F}$, let $q = \mathrm{R}((\mathrm{id}_U \otimes \mathrm{id}_V \otimes f)S)$ be the rank of the corresponding matrix in $U \otimes V$, and let $s = \mathrm{R}((A \otimes B \otimes f)S)$ be the rank of the matrix in $U' \otimes V'$. Then there exists a restriction $T \leqslant S \oplus \langle 1, s, 1\rangle$ extending A, B, C, such that there exists a linear function $f' : (W \oplus \mathbb{F}) \to \mathbb{F}$ (where $W \oplus \mathbb{F}$ corresponds to the third mode of $S \oplus \langle 1, s, 1\rangle$) satisfying the conditions of Proposition 21 with $r = q + s$.*

Proof. Since $M = (A \otimes B \otimes f)S$ has rank s, there exist linear maps $A' : \mathbb{F}^s \to U'$ and $B' : \mathbb{F}^s \to V'$ such that $M = (A' \otimes B')\langle 1, s, 1\rangle$. It is then straightforward to verify that we can define the new restriction as

$$T = \left[\begin{pmatrix} A & A' \end{pmatrix} \otimes \begin{pmatrix} B & B' \end{pmatrix} \otimes \begin{pmatrix} C & 0 \end{pmatrix}\right](S \oplus \langle 1, s, 1\rangle),$$

with the new linear function $f' = \begin{pmatrix} f & -1 \end{pmatrix}$. This function lies in the appropriate kernel, since

$$\left[\begin{pmatrix} A & A' \end{pmatrix} \otimes \begin{pmatrix} B & B' \end{pmatrix} \otimes \begin{pmatrix} f & -1 \end{pmatrix}\right](S \oplus \langle 1, s, 1\rangle) = M - (A' \otimes B')\langle 1, s, 1\rangle = 0,$$

and $(\mathrm{id} \otimes \mathrm{id} \otimes f')(S \oplus \langle 1, s, 1\rangle)$ is the direct sum of the matrices $(\mathrm{id}_U \otimes \mathrm{id}_V \otimes f)S$ and $\langle 1, s, 1\rangle$, which has rank $r = q + s$. ◀

Finally, we generalize Coppersmith and Winograd's notion of *fullness*. Given a degeneration $T \oplus \langle 1, t, 1\rangle \trianglelefteq S$, let $f : W \to \mathbb{F}(\lambda)$ be the component of the C-map projecting to the slice $\langle 1, t, 1\rangle$. We define the *fullness index* of $\langle 1, t, 1\rangle$ (or f) as $q = \mathrm{R}((\mathrm{id}_U \otimes \mathrm{id}_V \otimes f)S)$. If $q = \dim U = \dim V$, we say the slice is *full*.

Like the isolated property, the fullness index behaves well under speedup theorems and tensor products.

▶ **Proposition 23.** *The slice $\langle 1, t, 1\rangle$ constructed in Proposition 21 has fullness index r.*

▶ **Proposition 24.** *Consider two degenerations*

$$T_1 \oplus \langle 1, t_1, 1\rangle \trianglelefteq S_1, \quad T_2 \oplus \langle 1, t_2, 1\rangle \trianglelefteq S_2,$$

wherein the summands $\langle 1, t_1, 1\rangle$ and $\langle 1, t_2, 1\rangle$ have fullness indices q_1 and q_2, respectively. Then the direct summand $\langle 1, t_1 t_2, 1\rangle$ in the tensor product of these two degenerations has fullness index $q_1 q_2$. In particular, if f_1 and f_2 are full, then $f_1 \otimes f_2$ is full.

5.4 Direct sum and compression of matrix multiplication

In the proof of Proposition 21, we used the following basic fact: for a one-slice matrix multiplication tensor $\langle 1, r, 1\rangle$, restricting its first mode to a codimension-p subspace yields a tensor isomorphic to $\langle 1, r-p, 1\rangle$. Although this observation looks elementary, it can be viewed as the simplest instance of a much more general phenomenon.

Strassen proved the following result (see [41, Proposition 6.4])[3]: if T is a direct sum of matrix multiplication tensors and $A \subseteq U^\vee$ is a subspace of small codimension, then the partially restricted tensor $(A \otimes \mathrm{id}_V \otimes \mathrm{id}_W)T$ still contains, as a restriction, a "valuable" tensor T' that is again a direct sum of matrix multiplication tensors. Informally, matrix multiplication is robust under losing a small amount of information in one mode. This is often referred to as the *compression* property of matrix multiplication tensors; see [47, Part III] for an exposition and generalizations. Algorithmic versions have also been developed recently [27, 28].

For our purposes, we only need a compression statement in the special case where all summands are one-slice tensors. This case is simple enough that we include a short proof for the reader's convenience. We also note that the concrete applications in this paper do not require the full generality of this theorem: the tensors and linear maps we use enjoy additional structure that simplifies the analysis. Nonetheless, we expect that the compression statement in this general form may be useful for more complicated degenerations in the future, for instance when the relevant linear maps are found via computer search.

▶ **Proposition 25.** *Let $T = \bigoplus_{i=1}^{k} \langle 1, n_i, 1\rangle$ be a direct sum of one-slice matrix multiplication tensors, and let $A \subseteq U^\vee$ be a subspace of codimension p. Then there exists a tensor*

$$T' \leqslant (A \otimes \mathrm{id}_V \otimes \mathrm{id}_W)T$$

that is still a direct sum of one-slice matrix multiplication tensors:

$$T' \cong \bigoplus_{i=1}^{k} \langle 1, n_i - p_i, 1\rangle,$$

where the integers p_i satisfy $0 \leqslant p_i \leqslant n_i$ and $\sum_{i=1}^{k} p_i = p$.

Proof. Let $(x_j^{(i)})_{1\leqslant j\leqslant n_i}$ and $(y_j^{(i)})_{1\leqslant j\leqslant n_i}$ denote the bases of the U- and V-modes of T corresponding to the decomposition $T = \bigoplus_{i=1}^{k} \langle 1, n_i, 1\rangle$. Identify A with a surjective linear map $A : U \to U'$ where $\dim U' = \dim U - p$.

Consider the set of vectors $\{A(x_j^{(i)})\} \subseteq U'$. Choose a subset $D \subseteq \{x_j^{(i)}\}$ of size $\dim U'$ such that $\{A(x) : x \in D\}$ is linearly independent (hence a basis of U'). Now restrict T by zeroing out every basis element $x_j^{(i)}$ (and the corresponding $y_j^{(i)}$) with $x_j^{(i)} \notin D$. This reduces the ith summand $\langle 1, n_i, 1\rangle$ to $\langle 1, n_i - p_i, 1\rangle$, where p_i is the number of removed basis elements from the ith block. By construction, $\sum_i p_i = p$.

Finally, the restriction of A to $\operatorname{span} D$ is an isomorphism onto U', so after the above zeroing-out operation the resulting tensor is indeed a restriction of $(A \otimes \mathrm{id}_V \otimes \mathrm{id}_W)T$ and has the claimed direct-sum form. ◀

[3] Strassen's original formulation and subsequent expositions (e.g., [41, Proposition 6.4] and [47, Part III]) often present a closely related variant in terms of an *additive* perturbation in the same tensor space: if $S = T + U$ where U has small flattening rank in the U-mode, then one can find a "valuable" restriction $T' \leqslant S$ that remains a direct sum of matrix multiplication tensors. The formulation we use here (starting from a direct sum and restricting one mode by a small-codimension subspace) is equivalent to that variant via the substitution method; we include this footnote to reconcile the two viewpoints.

6 Instantiations of speedup theorems

The first method is the most naive version of the speedup theorem, which generalizes [22, Corollary 3.1].

▶ **Theorem 26** (One-slice speedup for nonminimal border rank). *Let T be an $n \times n \times n$ tensor over $\mathbb{F}$, and suppose that $T \trianglelefteq \langle r \rangle$. Let $\{a_i\}, \{b_i\}, \{c_i\}$ be the $\mathbb{F}(\lambda)$-vectors representing the border rank decomposition, i.e.,*

$$\sum_{i=1}^{r} a_i b_i c_i = T + O(\lambda).$$

If there exist nonzero scalars $c_i' \in \mathbb{F}(\lambda) \setminus \{0\}$ *such that the matrix*

$$M = \sum_{i=1}^{r} a_i b_i c_i' \in U' \otimes V'$$

has rank at most s, then there exists a degeneration

$$T \oplus \langle 1, r+s-2n, 1 \rangle \trianglelefteq \langle r \rangle \oplus \langle 1, s, 1 \rangle.$$

When $r > 2n$, this yields the improved asymptotic rank bound

$$\underset{\sim}{\mathrm{R}}(T) \leqslant r + s^{2/3} - (r+s-2n)^{2/3} < r.$$

In particular, if $r \geqslant n$, we have

$$T \oplus \langle 1, r-n, 1 \rangle \trianglelefteq \langle r \rangle \oplus \langle 1, n, 1 \rangle,$$

and for $r > 2n$, this provides the strict improvement

$$\underset{\sim}{\mathrm{R}}(T) \leqslant r + n^{2/3} - (r-n)^{2/3} < r.$$

Proof. The hypothesis exactly matches the condition of Proposition 22. Applying that result, all the degeneration parameters align:

$$T \oplus \langle 1, r+s-2n, 1 \rangle \trianglelefteq \langle r \rangle \oplus \langle 1, s, 1 \rangle.$$

We refer to the setting of the three-direction speedup inequalities Equation (4.4)–Equation (4.6) with parameters $t = r + s - 2n$ and the same s. Since $r > 2n$, we have $t > s$, and thus Equation (4.4)–Equation (4.6) imply the asymptotic rank bound

$$\underset{\sim}{\mathrm{R}}(T) \leqslant r + s^{2/3} - t^{2/3} = r + s^{2/3} - (r+s-2n)^{2/3}.$$ ◀

We can assess the efficacy of this method by the order of magnitude of the improvement. For $r \geqslant 3n$, the bound simplifies to

$$\underset{\sim}{\mathrm{R}}(T) \leqslant r - \Omega(r^{2/3}).$$

Although this method appears to yield an improvement only when $r > 2n$, we can bootstrap this result to show that any non-minimal border rank upper bound for T is not tight for its asymptotic rank.

▶ **Corollary 27.** *Let T be an $n \times n \times n$ tensor with $\underline{\mathrm{R}}(T) \leqslant r$. If $r > n$, then $\underset{\sim}{\mathrm{R}}(T) < r$.*

Proof. Since $T \trianglelefteq \langle r \rangle$, taking tensor powers yields $T^{\otimes k} \trianglelefteq \langle r^k \rangle$. By choosing k sufficiently large such that $r^k > 2n^k$, Theorem 26 implies $\underset{\sim}{\mathrm{R}}(T)^k = \underset{\sim}{\mathrm{R}}(T^{\otimes k}) < r^k$. ◀

The second method demonstrates that the identity obtained via the first method can essentially always be *iterated* to yield a superior identity. This generalizes [22, Theorem B].

▶ **Theorem 28.** *Under the setting of Theorem 26, let $t = r + s - 2n$. There exists a degeneration*

$$T^{\otimes 2} \oplus 2 \odot T \otimes \langle 1, t, 1 \rangle \oplus \langle 1, t^2 + 2n^2, 1 \rangle \trianglelefteq \langle r^2 \rangle \oplus 2r \odot \langle 1, s, 1 \rangle \oplus \langle 1, s^2, 1 \rangle.$$

This effectively replaces the summand $\langle 1, t^2, 1 \rangle$ in the tensor square of the result from Theorem 26 with $\langle 1, t^2 + 2n^2, 1 \rangle$.

Proof. Since the summand $\langle 1, t, 1 \rangle$ appearing in the degeneration

$$T \oplus \langle 1, t, 1 \rangle \trianglelefteq \langle r \rangle \oplus \langle 1, s, 1 \rangle$$

is produced via the one-slice speedup theorem (Theorem 26), Proposition 19 and Proposition 23 guarantee that this summand is full and isolated w.r.t. the degeneration. Upon taking the tensor square, Proposition 20 and Proposition 24 guarantee that the summand $\langle 1, t^2, 1 \rangle$ in the degeneration

$$T^{\otimes 2} \oplus 2 \odot T \otimes \langle 1, t, 1 \rangle \oplus \langle 1, t^2, 1 \rangle \trianglelefteq \langle r^2 \rangle \oplus 2r \odot \langle 1, s, 1 \rangle \oplus \langle 1, s^2, 1 \rangle$$

remains full and isolated. Let this summand be projected by a linear functional f in the third mode. First, we zero out the variables of this summand in all three directions, so that for the remaining degeneration

$$T^{\otimes 2} \oplus 2 \odot T \otimes \langle 1, t, 1 \rangle \trianglelefteq \langle r^2 \rangle \oplus 2r \odot \langle 1, s, 1 \rangle \oplus \langle 1, s^2, 1 \rangle,$$

f satisfies the condition of Proposition 21, as f is isolated. The parameter r corresponds to the fullness of f, which in our case (since f is full) is the dimension of the right-hand side, i.e., $(r + s)^2$. Thus the replaced direct summand

$$T^{\otimes 2} \oplus 2 \odot T \otimes \langle 1, t, 1 \rangle \oplus \langle 1, t', 1 \rangle \trianglelefteq \langle r^2 \rangle \oplus 2r \odot \langle 1, s, 1 \rangle \oplus \langle 1, s^2, 1 \rangle$$

has

$$t' = (r + s)^2 - 2(n^2 + 2nt) = t^2 + 2n^2.$$ ◀

Naturally, this iterative process can be continued indefinitely, but its improvement on the asymptotic rank would be marginal. Also, after some analysis, it turns out that Theorem 28 still gives an asymptotic rank bound of type $r - \Omega(r^{2/3})$ when $r \geqslant 3n$.

The third method involves appending not merely a single one-slice tensor to both sides, but a direct sum of such tensors.

▶ **Theorem 29.** *Let $T \trianglelefteq \langle r \rangle$, and let $\{a_i\}, \{b_i\}, \{c_i\}$ be $\mathbb{F}(\lambda)$-vectors representing the degeneration. Suppose there exist nonzero scalars c'_i and a partition of $[r]$ into p groups $[r] = I_1 \sqcup \cdots \sqcup I_p$. For each $1 \leqslant \alpha \leqslant p$, let $r_\alpha = |I_\alpha|$, let $M_\alpha = \sum_{i \in I_\alpha} a_i b_i c'_i$, and let $s_\alpha \geqslant \mathrm{R}(M_\alpha)$. If $r_\alpha + s_\alpha \geqslant 2n$ for all α, then*

$$T \oplus \bigoplus_{\alpha=1}^{p} \langle 1, r_\alpha + s_\alpha - 2n, 1 \rangle \trianglelefteq \langle r \rangle \oplus \bigoplus_{\alpha=1}^{p} \langle 1, s_\alpha, 1 \rangle.$$

We provide two proofs of this result. They are essentially equivalent; the first is simpler, while the second employs the concept of the compression theorem and generalizes to cases where the added terms are not direct sums of one-slice tensors (Theorem 42) or where the starting upper bound S is not a diagonal tensor (Section 7.2).

First proof. In the spirit of Corollary 18, it suffices to prove the restriction case, i.e., working over $\mathbb{F}$ instead of $\mathbb{F}(\lambda)$. In that setting, we can write T as a sum of tensors $T = \sum_{\alpha=1}^{p} T_\alpha$, where $T_\alpha = \sum_{i \in I_\alpha} a_i b_i c_i$. Then $\mathrm{R}(T_\alpha) \leqslant r_\alpha$. We can apply Theorem 26 to each T_α individually and sum the results:

$$\bigoplus_{\alpha=1}^{p} (T_\alpha \oplus \langle 1, r_\alpha + s_\alpha - 2n, 1\rangle) \trianglelefteq \langle r\rangle \oplus \bigoplus_{\alpha=1}^{p} \langle 1, s_\alpha, 1\rangle.$$

The conclusion follows from $T \leqslant \bigoplus_\alpha T_\alpha$. ◀

Second proof. As in the first proof, we focus on the restriction case. We begin with the redundant restriction

$$T \leqslant \langle r\rangle \oplus \bigoplus_{\alpha=1}^{p} \langle 1, s_\alpha, 1\rangle.$$

Write

$$S \coloneqq \langle r\rangle \oplus \bigoplus_{\alpha=1}^{p} \langle 1, s_\alpha, 1\rangle = \sum_{i=1}^{r} u_i v_i w_i - \sum_{\alpha=1}^{p} \sum_{j=1}^{s_\alpha} u'_{\alpha j} v'_{\alpha j} w'_\alpha,$$

where the minus sign is introduced to simplify subsequent cancellations. Since M_α has rank at most s_α, there exist linear maps A_α, B_α such that $(A_\alpha \otimes B_\alpha)\langle 1, s_\alpha, 1\rangle = M_\alpha$. The restriction is defined by the linear maps

$$\begin{aligned}
A: &\quad a_i \mapsto u_i, \quad A_\alpha(u'_{\alpha j}) \mapsto u'_{\alpha j},\\
B: &\quad b_i \mapsto v_i, \quad B_\alpha(v'_{\alpha j}) \mapsto v'_{\alpha j},\\
C: &\quad c_i \mapsto w_i, \qquad\quad 0 \mapsto w'_\alpha.
\end{aligned}$$

To apply the free-lunch speedup theorem (Theorem 17), define a subspace $C' \subseteq W^\vee$ via the linear map defined as follows, where $\alpha(i)$ is the index such that $i \in I_\alpha$:

$$C': \quad c'_i w'_{\alpha(i)} \mapsto w_i, \quad w'_\alpha \mapsto w'_\alpha.$$

A direct computation shows that $(A \otimes B \otimes C')S = 0$.

Applying Theorem 17 yields a direct summand $T' = (A' \otimes B' \otimes C')S$ on the left-hand side. Let us first unwind the form of $(\mathrm{id} \otimes \mathrm{id} \otimes C')S$:

$$\begin{aligned}
(\mathrm{id} \otimes \mathrm{id} \otimes C')S &= \sum_{i=1}^{r} u_i v_i \cdot c'_i w'_{\alpha(i)} - \sum_{\alpha=1}^{p} \sum_{j=1}^{s_\alpha} u'_{\alpha j} v'_{\alpha j} w'_\alpha\\
&= \sum_{\alpha=1}^{p} \left(\sum_{i \in I_\alpha} c'_i u_i v_i - \sum_{j=1}^{s_\alpha} u'_{\alpha j} v'_{\alpha j} \right) w'_\alpha,
\end{aligned}$$

which is evidently $\bigoplus_{\alpha=1}^{p} \langle 1, r_\alpha + s_\alpha, 1\rangle$. Recall that $A' \subseteq U^\vee$, the space of linear functionals, is given by those $f \in U^\vee$ such that the following expression vanishes:

$$(f \otimes B \otimes C')S = \sum_{\alpha=1}^{p} \left(\sum_{i \in I_\alpha} c'_i f(u_i) b_i - \sum_{j=1}^{s_\alpha} f(u'_{\alpha j}) B_\alpha(v'_{\alpha j}) \right) w'_\alpha$$

The image, viewed as a vector in an np-dimensional space, defines np linear equations on f, meaning A' is a subspace of codimension $\leqslant np$. Similarly, B' restricts to a subspace of codimension $\leqslant np$. Proposition 25 applies, which allows us to reduce the direct sum of matrix multiplication tensors by $2np$ terms. Notice that each equation associated with a component in w'_α only concerns the values of f on $\{u_i\}_{i\in I_\alpha}$ and $u'_{\alpha j}$. Thus, we do not need the full generality of Proposition 25, but can instead perform compression on each direct summand separately. Consequently, T' can be compressed to $T' \geqslant \bigoplus_\alpha \langle 1, r_\alpha + s_\alpha - 2n, 1\rangle$, proving the claim. ◀

We now discuss the quantitative advantage of this method compared with previous ones. Using the naive asymptotic assumption $s \leqslant n$, one can take $p = \Omega(r/n)$ and let every $r_\alpha \geqslant 3n$. This gives the degeneration

$$T \oplus p \odot \langle 1, 2n, 1\rangle \trianglelefteq \langle r\rangle \oplus p \odot \langle 1, n, 1\rangle,$$

and similar statements hold for the other two directions. Strassen calculus then yields an upper bound $\underset{\sim}{\mathrm{R}}(T) \leqslant r - (2^{2/3} - 1)pn^{2/3} = r - \Omega(r/n^{1/3})$, which is a superior bound when $r = \omega(n)$.

▶ Remark 30. An intriguing question is how far we can improve the order of magnitude in the speedup theorem. For example, can we prove that any border rank upper bound $\underline{\mathrm{R}}(T) \leqslant r$ with $r > n^{1.0001}$ implies $\underset{\sim}{\mathrm{R}}(T) \leqslant 0.9999r$? Any mild strengthening of this question would have surprising consequences in fine-grained complexity theory: the *balanced partition tensor* T_n proposed by Pratt [34] is a family of tensors for which the currently known border rank upper bound is $r_n \approx N^{1.08}$ [25, Proposition 6.18], where N is the dimension of T_n. If one could improve the asymptotic rank of T_n to $o(r_n/\log r_n)$, then Pratt showed that this would refute the set cover conjecture.

7 Applications

7.1 Small CW tensors

Recall that the small Coppersmith–Winograd (CW) tensor cw_q has $q+1$ variables in each mode. It is commonly written as

$$\mathrm{cw}_q = \sum_{i=1}^{q} x_0 y_i z_i + x_i y_0 z_i + x_i y_i z_0.$$

The following identity, originally due to Coppersmith–Winograd, gives a border-rank upper bound of $q+2$.

$$\begin{aligned}
&\sum_{i=1}^{q} \lambda^{-2}(x_0 + \lambda x_i)(y_0 + \lambda y_i)(z_0 + \lambda z_i) \\
&- \lambda^{-3}\left(x_0 + \lambda^2 \sum_{i=1}^{q} x_i\right)\left(y_0 + \lambda^2 \sum_{i=1}^{q} y_i\right)\left(z_0 + \lambda^2 \sum_{i=1}^{q} z_i\right) \\
&+ (\lambda^{-3} - q\lambda^{-2}) x_0 y_0 z_0 \\
&= \mathrm{cw}_q + O(\lambda).
\end{aligned}$$

It is known that this border-rank bound is tight, i.e. $\underline{\mathrm{R}}(\mathrm{cw}_q) = q+2$. Moreover, Conner, Gesmundo, Landsberg, and Ventura [20, Theorem 1.4] proved that the border rank does not drop for small powers: when $q > 2$ and $n = 2$, one has $\underline{\mathrm{R}}(\mathrm{cw}_q^{\otimes n}) = (q+2)^n$, and likewise when $q > 4$ and $n = 3$. We show that the asymptotic rank of cw_q is strictly smaller than $q+2$, and hence this phenomenon must eventually fail for larger n.

We begin by finding low-rank matrices in order to apply Theorem 26.

▶ **Lemma 31.** *Let a_i, b_i, c_i be the data of the border-rank upper bound $\mathrm{cw}_q \trianglelefteq \langle q+2\rangle$ given above. Then there exist nonzero scalars c'_i such that the matrix $M = \sum_{i=1}^{q+2} c'_i \cdot a_i b_i \in U' \otimes V'$ has rank q.*

Proof. We first write the linear combination as

$$M = \sum_{i=1}^{q} c'_i (x_0 + \lambda x_i)(y_0 + \lambda y_i) + c'_{q+1}\left(x_0 + \lambda^2 \sum_{i=1}^{q} x_i\right)\left(y_0 + \lambda^2 \sum_{i=1}^{q} y_i\right) + c'_0 x_0 y_0.$$

For notational convenience, we apply the change of basis $x'_i \coloneqq x_0 + \lambda x_i$ for $i > 0$ (and similarly for y). Then

$$M = c'_0 x_0 y_0 + \sum_{i=1}^{q} c'_i x'_i y'_i + c'_{q+1}\left((1-q\lambda)x_0 + \lambda \sum_{i=1}^{q} x'_i\right)\left((1-q\lambda)y_0 + \lambda \sum_{i=1}^{q} y'_i\right).$$

Assume $c'_i = 1$ for all $i \leqslant q$. We choose c'_{q+1} so that M becomes singular. Writing M in the form $M = I + c'_{q+1} a b^{\mathsf{T}}$, we have

$$\det(M) = \det(I + c'_{q+1} a b^{\mathsf{T}}) = 1 + c'_{q+1} b^{\mathsf{T}} a,$$

and a direct calculation gives $b^{\mathsf{T}} a = (1-q\lambda)^2 + q\lambda^2 \neq 0$. Taking $c'_{q+1} = -1/(b^{\mathsf{T}} a)$ yields $\mathrm{R}(M) < q+1$. On the other hand,

$$\mathrm{R}(M) = \mathrm{R}(I + c'_{q+1} a b^{\mathsf{T}}) \geqslant \mathrm{R}(I) - \mathrm{R}(c'_{q+1} a b^{\mathsf{T}}) = q.$$

Therefore, this choice produces a rank-q matrix. ◀

▶ **Proposition 32.** *For any $q \geqslant 2$ and $n \geqslant 1$, we have the tensor degeneration*

$$\mathrm{cw}_q^{\otimes n} \oplus \langle 1, (q+2)^n - 2(q+1)^n + q^n, 1\rangle \trianglelefteq \langle (q+2)^n\rangle \oplus \langle 1, q^n, 1\rangle.$$

Proof. Let f be the linear functional on the W-mode given by the previous lemma. Then $M = (A \otimes B \otimes f)\langle q+2\rangle$ satisfies $\mathrm{R}(M) = q$. Tensoring these maps gives a degeneration $\mathrm{cw}_q^{\otimes n} \trianglelefteq \langle (q+2)^n\rangle$, and $f^{\otimes n}$ is a linear functional such that

$$M^{\otimes n} = (A^{\otimes n} \otimes B^{\otimes n} \otimes f^{\otimes n})\langle (q+2)^n\rangle$$

has rank q^n. Applying Theorem 26 yields the desired degeneration. ◀

Combining the above degeneration with Proposition 15 immediately yields the following asymptotic-rank bound.

▶ **Corollary 33.** *For any $q \geqslant 2$ and $n \geqslant 1$, define*

$$\gamma_{q,n} \coloneqq \left((q+2)^n + q^{2n/3} - ((q+2)^n - 2(q+1)^n + q^n)^{2/3}\right)^{1/n}$$

and

$$\gamma_q^{(33)} \coloneqq \min_{n \geqslant 1} \gamma_{q,n}.$$

Then the small CW tensor satisfies

$$\underset{\sim}{\mathrm{R}}(\mathrm{cw}_q) \leqslant \gamma_q^{(33)} < q+2.$$

We let $n^{(33)}$ denote the n that reaches the minimum in Corollary 33.

We can further apply the second method to iteratively speed up the degeneration. By Theorem 28, we obtain

$$\begin{aligned}
&\mathrm{cw}_q^{\otimes 2n} \ \oplus\ 2 \odot \mathrm{cw}_q^{\otimes n} \otimes \langle 1, (q+2)^n - 2(q+1)^n + q^n, 1\rangle \\
&\oplus \langle 1, [(q+2)^n - 2(q+1)^n + q^n]^2 + 2(q+1)^{2n}, 1\rangle \\
&\trianglelefteqslant (\langle (q+2)^n \rangle \oplus \langle 1, q^n, 1\rangle)^{\otimes 2}.
\end{aligned}$$

Strassen calculus then gives the upper bound

$$\underset{\sim}{\mathrm{R}}(\mathrm{cw}_q) \leqslant \max_{\theta \in [2/3,1]} F(\theta)$$

where

$$F(\theta) \coloneqq \sqrt{(r+s^\theta)^2 - (t^2 + 2(q+1)^{2n})^\theta + t^{2\theta}} - t^\theta$$

with $r = (q+2)^n$, $s = q^n$, and $t = r + s - 2(q+1)^n$. We denote the value obtained by taking $n = n^{(33)}$ by $\gamma_q'^{(33)}$.

The values of $\gamma_q^{(33)}, n^{(33)}$ and $\gamma_q'^{(33)}$ for the first few values of q are shown in Table 1 (computed via computer-aided evaluation; in fact, the maximizing θ in the definition of $\gamma_q'^{(33)}$ equals $\theta = 2/3$).

Table 1 Asymptotic rank upper bound for small CW tensors.

q	$\underline{\mathrm{R}}(\mathrm{cw}_q)$	$\gamma_q^{(33)}$	$n^{(33)}$	$\gamma_q'^{(33)}$
2	4	3.933484	4	3.930872
3	5	4.970152	4	4.967092
4	6	5.986817	5	5.986293
5	7	6.994597	6	6.994516
6	8	7.997665	6	7.997596
7	9	8.998978	7	8.998969
8	10	9.999595	8	9.999593
9	11	10.999850	9	10.999850
10	12	11.999929	9	11.999929

Note that this degeneration can be iterated indefinitely. We do not pursue further iterations here, as the resulting numerical improvements are marginal.

We also remark that the following variant of the cw_2 tensor, which first appears in [23, §11], is equally useful from the perspective of the laser method:

$$\mathrm{cw}_2' \coloneqq x_1y_2z_3 + x_1y_3z_2 + x_2y_1z_3 + x_2y_3z_1 + x_3y_1z_2 + x_3y_2z_1,$$

In particular, if one could prove $\underset{\sim}{\mathrm{R}}(\mathrm{cw}'_2) = 3$, then it would follow that $\omega = 2$. Currently, the best known upper bound is $\mathrm{R}(\mathrm{cw}'_2) = 4$ (over fields of characteristic $\neq 2$), via

$$\begin{aligned}\mathrm{cw}'_2 = \frac{1}{4}[&(x_1+x_2+x_3)(y_1+y_2+y_3)(z_1+z_2+z_3)\\ &+(-x_1+x_2-x_3)(-y_1+y_2-y_3)(-z_1+z_2-z_3)\\ &+(x_1-x_2-x_3)(y_1-y_2-y_3)(z_1-z_2-z_3)\\ &+(-x_1-x_2+x_3)(-y_1-y_2+y_3)(-z_1-z_2+z_3)].\end{aligned}$$

Let the four rank-one summands in the decomposition above be $a_ib_ic_i$. Then

$$a_1b_1 + a_2b_2 - a_3b_3 - a_4b_4 = 4(x_1y_3 + x_3y_1)$$

has rank 2, matching the same parameter regime as in Lemma 31. Therefore, the preceding analysis applies to cw'_2 as well.

▶ **Corollary 34.** *If $\mathbb{F}$ has characteristic $\neq 2$, then we also have $\underset{\sim}{\mathrm{R}}(\mathrm{cw}'_2) \leqslant {\gamma'_2}^{(33)}$.*

▶ Remark 35. If the asymptotic rank of the small CW tensor were optimal, i.e., $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_q) = q+1$, then applying the laser method directly to cw_q would in fact yield a better upper bound on ω than the standard analysis based on the big CW tensors CW_q. Since we improve the best known upper bounds on $\underset{\sim}{\mathrm{R}}(\mathrm{cw}_q)$, it is natural to ask whether these improvements already translate into a better bound on ω.

Starting from the degeneration above, the most straightforward way to bound ω combines the laser method with our new asymptotic rank bounds. Interestingly, we remark that this incurs a "composition loss": the asymptotic rank bound transfers some matrix multiplication structure into diagonal tensors, which the laser method cannot exploit. A more faithful approach would start from the degeneration used in our proof and combine Strassen calculus with the laser method at that level, and this indeed achieves a slightly better bound than the straightforward approach.

Nonetheless, our experiments indicate that, with the identities currently produced by our speedup theorems, this hybrid approach still performs worse than Coppersmith–Winograd's analysis of the laser method applied to the first power $\mathrm{CW}_q^{\otimes 1}$.

7.2 General tensors

A nontrivial general upper bound on the asymptotic rank of tensors, due to Strassen (implicit in [41, Proposition 3.6]), states that every $d \times d \times d$ tensor satisfies $\underset{\sim}{\mathrm{R}}(T) \leqslant d^{\frac{2}{3}\omega}$. In this subsection, we show how to slightly improve this bound.

To explain our improvement, we first recall Strassen's $\frac{2}{3}\omega$ bound. The following straightforward rewriting shows the restriction $T \leqslant \langle d, 1, d\rangle$:

$$T = \sum_{i,j,k=1}^{d} a_{ijk}x_iy_jz_k = \sum_{i,j=1}^{d} x_iy_j\left(\sum_{k=1}^{d} a_{ijk}z_k\right) \leqslant \sum_{i,j,k=1}^{d} x_iy_jw_{ij} = \langle d, 1, d\rangle.$$

Applying the same argument to the other modes yields $T \leqslant \langle d, d, 1\rangle$ and $T \leqslant \langle 1, d, d\rangle$. Tensoring these three restrictions gives $T^{\otimes 3} \leqslant \langle d^2, d^2, d^2\rangle$, and hence $\underset{\sim}{\mathrm{R}}(T)^3 \leqslant d^{2\omega}$.

We first present a one-slice speedup of the above argument.

▶ **Theorem 36.** *When $d \geqslant 3$, any $d \times d \times d$ tensor T satisfies*

$$\underset{\sim}{\mathrm{R}}(T) \leqslant \sqrt{d^{\frac{4}{3}\omega} + d^{\frac{2}{3}\omega} - (d^3 - d^2)^{\frac{\omega}{3}}} < d^{\frac{2}{3}\omega}.$$

Proof. We start with the restriction $T^{\otimes 2} \leqslant \langle d, d^2, d\rangle$.

We wish to apply Proposition 22 to append a slice to both sides of this restriction. To do so, we first need a linear functional $f\colon W \to \mathbb{F}$ for $S = \langle d, d^2, d\rangle$ such that $(\mathrm{id} \otimes \mathrm{id} \otimes f)S$ has large rank. Write

$$S = \sum_{i=1}^{d^2} \sum_{j=1}^{d} \sum_{k=1}^{d^2} x_{ij} y_{jk} z_{ki}.$$

Let f be defined by $f(z_{ki}) = [\![i = k]\!]$. Then

$$(\mathrm{id} \otimes \mathrm{id} \otimes f)S = \sum_{i=1}^{d^2} \sum_{j=1}^{d} x_{ij} y_{ji}$$

is diagonal, hence has rank $s = d^3$. Applying Proposition 22 and Proposition 21, we obtain the degeneration

$$T^{\otimes 2} \oplus \langle 1, d^3 - d^2, 1\rangle \trianglelefteq \langle d, d^2, d\rangle \oplus \langle 1, d^2, 1\rangle.$$

Passing to the asymptotic spectrum, this means

$$\phi(T)^2 \leqslant d^{\theta_1 + 2\theta_2 + \theta_3} + d^{2\theta_2} - (d^3 - d^2)^{\theta_2}.$$

Writing $\varpi \coloneqq \theta_1 + \theta_2 + \theta_3$, by permuting modes we obtain analogous inequalities, we have

$$\phi(T)^2 \leqslant d^{\varpi + \theta_i} + d^{2\theta_i} - (d^3 - d^2)^{\theta_i}.$$

Since $\theta_1 + \theta_2 + \theta_3 = \varpi$ and $\theta_1, \theta_2, \theta_3 \geqslant 0$, there exists i such that $\theta_i \leqslant \varpi/3$. One can show that the function

$$F(\theta) = d^{\varpi + \theta} + d^{2\theta} - (d^3 - d^2)^{\theta}, \quad \theta \in [0, \varpi/3]$$

is maximized at $\theta = \varpi/3$ (the proof is deferred to Lemma 48). Furthermore, by Lemma 49, the quantity $F(\varpi/3)$ is increasing in ϖ on $[2, \omega]$. Since $\varpi \leqslant \omega$ for all $(\theta_1, \theta_2, \theta_3) \in \Delta_{\mathrm{MM}}$, we obtain

$$\phi(T)^2 \leqslant d^{\frac{4}{3}\omega} + d^{\frac{2}{3}\omega} - (d^3 - d^2)^{\omega/3}.$$

The claim then follows from Strassen duality. ◀

By appending direct sums rather than a single slice, we obtain a stronger bound.

▶ **Lemma 37.** *Let $\{d_\alpha\}_{1 \leqslant \alpha \leqslant p}$ be positive integers such that $\sum_\alpha d_\alpha \leqslant d$. Then we have degeneration*

$$T^{\otimes 2} \oplus \bigoplus_{\alpha=1}^{p} \langle 1, (d_\alpha - 1)d^2, 1\rangle \trianglelefteq \langle d, d^2, d\rangle \oplus p \odot \langle 1, d^2, 1\rangle.$$

Proof. We start from the redundant restriction $T^{\otimes 2} \leqslant \langle d, d^2, d\rangle \oplus p \odot \langle 1, d^2, 1\rangle$. Write

$$S \coloneqq \langle d, d^2, d\rangle \oplus p \odot \langle 1, d^2, 1\rangle = \sum_{i=1}^{d}\sum_{j=1}^{d^2}\sum_{k=1}^{d} u_{ij}v_{jk}w_{ki} - \sum_{\alpha=1}^{p}\sum_{\ell=1}^{d^2} u'_{\alpha\ell}v'_{\alpha\ell}w'_{\alpha},$$

where the minus sign is introduced to simplify later cancellations. Fix a partition $[\sum_\alpha d_\alpha] = I_1 \sqcup \cdots \sqcup I_p$ where $|I_\alpha| = d_\alpha$. Define the linear map C by augmenting the corresponding linear map of $T^{\otimes 2} \leqslant \langle d, d^2, d\rangle$ by zero on new components, and C' by

$$C' : \quad w_{ki} \mapsto \begin{cases} w'_\alpha & i = k \in I_\alpha, \\ 0 & \text{otherwise}, \end{cases} \qquad w'_\alpha \mapsto w'_\alpha,$$

and choose linear maps A, B so that

$$\begin{aligned} 0 &= (A \otimes B \otimes C')S \\ &= \sum_{\alpha=1}^{p}\left(\sum_{i\in I_\alpha}\sum_{j=1}^{d^2} A(u_{ij})B(v_{ji}) - \sum_{\ell=1}^{d^2} A(u'_{\alpha\ell})B(v'_{\alpha\ell})\right)w'_\alpha, \end{aligned}$$

where the action on the u- and v-variables is already determined (since it stems from $T^{\otimes 2} \leqslant \langle d, d^2, d\rangle$). Since $T^{\otimes 2}$ has dimension d^2 in each mode, we can choose $A(u'_{\alpha\ell})$ and $B(v'_{\alpha\ell})$ so that the terms cancel.

Applying Theorem 17 yields a direct summand $T' = (A' \otimes B' \otimes C')S$ on the left-hand side. We first expand the form of $(\mathrm{id} \otimes \mathrm{id} \otimes C')S$:

$$(\mathrm{id} \otimes \mathrm{id} \otimes C')S = \sum_{\alpha=1}^{p} \underbrace{\left(\sum_{i\in I_\alpha}\sum_{j=1}^{d^2} u_{ij}v_{ji} - \sum_{\ell=1}^{d^2} u'_{\alpha\ell}v'_{\alpha\ell}\right)}_{\langle 1,(d_\alpha+1)d^2,1\rangle} w'_\alpha .$$

Then, as in the second proof of Theorem 29, after applying the contraction $A' \otimes B'$, each summand $\langle 1, (d_\alpha + 1)d^2, 1\rangle$ restricts to $\langle 1, (d_\alpha - 1)d^2, 1\rangle$. This completes the proof. ◀

Compared with Theorem 36, this yields an improved bound in which the subtractive term is asymptotically stronger.

▶ **Theorem 38.** *When $d \geqslant 3$, any $d \times d \times d$ tensor T satisfies*

$$\underset{\sim}{R}(T) \leqslant \sqrt{d^{\frac{4}{3}\omega} - \left\lfloor \frac{d}{3} \right\rfloor \cdot (2^{\frac{\omega}{3}} - 1)d^{\frac{2}{3}\omega}}.$$

Proof. Apply the previous lemma with $p = \lfloor d/3 \rfloor$ and $d_\alpha = 3$ for all α. It gives the degeneration

$$T^{\otimes 2} \oplus p \odot \langle 1, 2d^2, 1\rangle \trianglelefteq \langle d, d^2, d\rangle \oplus p \odot \langle 1, d^2, 1\rangle.$$

Fix an arbitrary spectral point $\phi \in \mathcal{X}$, and let $(\theta_1, \theta_2, \theta_3) \in \Delta_{\mathrm{MM}}$ be its parameters. Passing to the asymptotic spectrum yields

$$\phi(T)^2 \leqslant d^{\theta_1+2\theta_2+\theta_3} - (2^{\theta_2} - 1)p\, d^{2\theta_2}.$$

By permuting tensor modes, we obtain the analogous bounds for $i \in \{1, 2, 3\}$:

$$\phi(T)^2 \leqslant d^{\varpi+\theta_i} - (2^{\theta_i} - 1)p\, d^{2\theta_i}, \qquad \text{where } \varpi \coloneqq \theta_1 + \theta_2 + \theta_3.$$

Let $\theta \coloneqq \min\{\theta_1, \theta_2, \theta_3\}$, so $\theta \leqslant \varpi/3$. Define

$$h_\varpi(x) \coloneqq d^{\varpi+x} - (2^x - 1)p\, d^{2x}.$$

Then $\phi(T)^2 \leqslant h_\varpi(\theta)$. By Lemma 50, h_ϖ is increasing on $[0, \varpi/3]$, hence

$$\phi(T)^2 \leqslant h_\varpi(\varpi/3) = d^{4\varpi/3} - (2^{\varpi/3} - 1)p\, d^{2\varpi/3}.$$

By Lemma 51, the right-hand side is increasing in ϖ on $[2, \omega]$; since $\varpi \leqslant \omega$ for every $(\theta_1, \theta_2, \theta_3) \in \Delta_{\mathrm{MM}}$, we obtain

$$\phi(T)^2 \leqslant d^{4\omega/3} - (2^{\omega/3} - 1)p\, d^{2\omega/3}.$$

Finally, taking the maximum over $\phi \in \mathcal{X}$ and using Strassen duality yields

$$\underset{\sim}{\mathrm{R}}(T)^2 \leqslant d^{4\omega/3} - (2^{\omega/3} - 1)p\, d^{2\omega/3},$$

i.e.,

$$\underset{\sim}{\mathrm{R}}(T) \leqslant \sqrt{d^{\frac{4}{3}\omega} - \left\lfloor \frac{d}{3} \right\rfloor \cdot (2^{\frac{\omega}{3}} - 1) d^{\frac{2}{3}\omega}}.$$ ◀

Let $\sigma(d)$ denote the exponent of $d \times d \times d$ tensors [30], i.e., the smallest σ such that every $d \times d \times d$ tensor T satisfies $\underset{\sim}{\mathrm{R}}(T) \leqslant d^\sigma$. Then the above results imply the following.

▶ **Corollary 39.** *For any $d > 0$, $\sigma(d) < \frac{2}{3}\omega$.*

▶ Remark 40. This answers an open problem posed by Kaski [29, Open Problem 1], which asks whether Strassen's bound $\sigma(d) \leqslant \frac{2}{3}\omega$ can be improved. It is worth emphasizing, however, that our improvement has the form $\sigma(d) \leqslant \frac{2}{3}\omega - \varepsilon_d$, where $\varepsilon_d \to 0$ rapidly as $d \to \infty$.

In particular, to make progress on fine-grained questions such as the set cover conjecture [10, 34], one would need to achieve $\sigma(d) \leqslant 1.08$. While in the most ideal case $\omega = 2$, currently we only have an exponent bound $\sigma(d) \leqslant 4/3 - o(1)$. On the other hand, this current approach also seems ill-suited for improving ω: the strategy is to reduce a generic tensor T to matrix multiplication, but when T is itself a matrix multiplication tensor, this reduction becomes too indirect to be effective.

To make progress toward the (extended; see [30, Conjecture 2]) asymptotic rank conjecture, we propose the following question, which seems to require ideas beyond a naive application of the techniques developed in this paper.

▶ **Problem 41.** Prove that there exists $\delta > 0$ such that $\sigma(d) \leqslant \frac{2}{3}\omega - \delta$ for all d.

7.3 A direct sum identity

In this subsection, we use the speedup theorems developed above to derive an intriguing direct sum identity that generalizes Schönhage's classical construction [38, Lemma 6.1]:

$$\langle n, 1, m\rangle \oplus \langle 1, (n-1)(m-1), 1\rangle \trianglelefteq \langle nm + 1\rangle.$$

▶ **Theorem 42.** *Let p, q be positive integers, and let $\{n_\alpha\}_{1\leqslant\alpha\leqslant p}$ and $\{m_\beta\}_{1\leqslant\beta\leqslant q}$ be sequences of positive integers. Set*

$$n = \sum_{\alpha=1}^{p} n_\alpha, \qquad m = \sum_{\beta=1}^{q} m_\beta.$$

Then there exists a degeneration between direct sums of matrix multiplication tensors:

$$\langle n,1,m\rangle \oplus \bigoplus_{\substack{1\leqslant\alpha\leqslant p\\ 1\leqslant\beta\leqslant q}} \langle 1,(n_\alpha-1)(m_\beta-1),1\rangle \trianglelefteq \langle nm\rangle \oplus \langle p,1,q\rangle.$$

Before proving Theorem 42, we briefly compare it with Schönhage's original direct sum identity. Taking the direct sum over all (α,β) gives

$$\bigoplus_{\alpha,\beta}\langle n_\alpha,1,m_\beta\rangle \oplus \bigoplus_{\alpha,\beta}\langle 1,(n_\alpha-1)(m_\beta-1),1\rangle \trianglelefteq \langle nm\rangle\oplus\langle pq\rangle.$$

Compared with Theorem 42, the direct-sum tensor $\bigoplus_{\alpha,\beta}\langle n_\alpha,1,m_\beta\rangle$ on the left-hand side is more valuable than $\langle n,1,m\rangle$. However, this also comes with a bigger cost on the right-hand side: the slack term $\langle pq\rangle$ replaces the one-slice tensor $\langle p,1,q\rangle$.

In the special case $n_\alpha = n$ and $m_\beta = m$ for all α,β, our identity becomes

$$\langle pn,1,qm\rangle \oplus pq\odot\langle 1,(n-1)(m-1),1\rangle \trianglelefteq \langle pqnm\rangle\oplus\langle p,1,q\rangle.$$

If instead one tensors Schönhage's identity for (n,m) with $\langle p,1,q\rangle$, one obtains

$$\langle pn,1,qm\rangle \oplus \langle p,(n-1)(m-1),q\rangle \trianglelefteq nm\odot\langle p,1,q\rangle\oplus\langle p,1,q\rangle.$$

Here the direct summand $nm\odot\langle p,1,q\rangle$ on the right-hand side is cheaper than $\langle pqnm\rangle$, but the tensor $\langle p,(n-1)(m-1),q\rangle$ on the left-hand side is also less valuable than $pq\odot\langle 1,(n-1)(m-1),1\rangle$.

In sum, our new identity does not seem to be easily derived from Schönhage's identity alone.

Proof of Theorem 42. We start from a redundant restriction

$$\langle n,1,m\rangle \leqslant \langle nm\rangle\oplus\langle p,1,q\rangle.$$

Write

$$\langle n,1,m\rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} x_i y_j z_{ij}, \qquad S \coloneqq \langle nm\rangle\oplus\langle p,1,q\rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} u_{ij}v_{ij}w_{ij} - \sum_{\alpha=1}^{p}\sum_{\beta=1}^{q} u'_\alpha v'_\beta w'_{\alpha\beta},$$

where the minus sign is introduced to simplify later cancellations.

Fix partitions $[n] = I_1\sqcup\cdots\sqcup I_p$ and $[m] = J_1\sqcup\cdots\sqcup J_q$ with $|I_\alpha| = n_\alpha$ and $|J_\beta| = m_\beta$. Define linear maps A,B,C by

$$\begin{aligned} A:&\quad x_i \mapsto u_{ij}, && \sum_{i\in I_\alpha} x_i \mapsto u'_\alpha,\\ B:&\quad y_j \mapsto v_{ij}, && \sum_{j\in J_\beta} y_j \mapsto v'_\beta,\\ C:&\quad z_{ij} \mapsto w_{ij}, && 0 \mapsto w'_{\alpha\beta}. \end{aligned}$$

These maps define a restriction $\langle n,1,m\rangle \leqslant S$.

To apply the free-lunch speedup theorem (Theorem 17), define a subspace $C'\subseteq W^\vee$ via the linear map

$$C':\quad w'_{\alpha(i)\beta(j)} \mapsto w_{ij}, \qquad w'_{\alpha\beta}\mapsto w'_{\alpha\beta},$$

where $\alpha(i)$ (resp. $\beta(j)$) is the unique index such that $i \in I_{\alpha(i)}$ (resp. $j \in J_{\beta(j)}$). A direct computation shows that $(A \otimes B \otimes C')S = 0$.

Applying Theorem 17 yields a direct summand $T' = (A' \otimes B' \otimes C')S$ on the left-hand side.

Let us make the resulting subspaces $A' \subseteq U^\vee$ and $B' \subseteq V^\vee$ explicit. By definition,

$$A' = \{u \in U^\vee : (u \otimes B \otimes C')S = 0\}, \qquad B' = \{v \in V^\vee : (A \otimes v \otimes C')S = 0\}.$$

Unwinding the maps A, B and the definition of C', one checks that A' is the annihilator of the subspace spanned by the vectors

$$u'_\alpha - \sum_{i \in I_\alpha} u_{ij} \quad (\text{for all } \alpha \in [p],\ j \in [m]),$$

and similarly B' is the annihilator of the subspace spanned by

$$v'_\beta - \sum_{j \in J_\beta} v_{ij} \quad (\text{for all } \beta \in [q],\ i \in [n]).$$

In particular, in T' the relations

$$u'_\alpha = \sum_{i \in I_\alpha} u_{ij} \quad (\text{for all } \alpha \in [p],\ j \in [m]) \qquad \text{and} \qquad v'_\beta = \sum_{j \in J_\beta} v_{ij} \quad (\text{for all } \beta \in [q],\ i \in [n])$$

hold simultaneously. For each α choose a distinguished index $i^*_\alpha \in I_\alpha$ and for each β choose a distinguished index $j^*_\beta \in J_\beta$. These relations allow us to eliminate the variables $u_{i^*_\alpha j}$ and $v_{ij^*_\beta}$ via

$$u_{i^*_\alpha j} = u'_\alpha - \sum_{i \in I_\alpha \setminus \{i^*_\alpha\}} u_{ij}, \qquad v_{ij^*_\beta} = v'_\beta - \sum_{j \in J_\beta \setminus \{j^*_\beta\}} v_{ij}.$$

We now impose a further restriction by applying suitable projections on the first two modes. Define linear maps π_U and π_V by

$$\pi_U(u'_\alpha) = 0, \qquad \pi_U(u_{i\, j^*_\beta}) = 0\ (\forall i, \beta), \qquad \pi_U(u_{ij}) = u_{ij}\ (j \neq j^*_{\beta(j)}),$$

and

$$\pi_V(v'_\beta) = 0, \qquad \pi_V(v_{i^*_\alpha\, j}) = 0\ (\forall \alpha, j), \qquad \pi_V(v_{ij}) = v_{ij}\ (i \neq i^*_{\alpha(i)}).$$

Applying π_U and π_V to the first two modes of T' yields a further restriction

$$T^* \coloneqq (\pi_U \otimes \pi_V \otimes \mathrm{id})T' \leqslant T'.$$

By construction, π_U kills all u'_α and all variables in the distinguished columns $\{u_{i\, j^*_\beta}\}$, and π_V kills all v'_β and all variables in the distinguished rows $\{v_{i^*_\alpha\, j}\}$. After eliminating $u_{i^*_\alpha j}$ and $v_{ij^*_\beta}$ using the identities above, every term that contains an eliminated variable involves either u'_α (or a distinguished column) or v'_β (or a distinguished row), and hence is annihilated by $\pi_U \otimes \pi_V$. Therefore, the only surviving terms are those with $i \in I_\alpha \setminus \{i^*_\alpha\}$ and $j \in J_\beta \setminus \{j^*_\beta\}$.

The resulting tensor is

$$T^* = \sum_{\alpha=1}^{p} \sum_{\beta=1}^{q} \left(\sum_{i \in I_\alpha \setminus \{i^*_\alpha\}} \sum_{j \in J_\beta \setminus \{j^*_\beta\}} u_{ij} v_{ij} \right) w_{\alpha\beta} \ \cong \ \bigoplus_{\alpha,\beta} \langle 1, (n_\alpha - 1)(m_\beta - 1), 1 \rangle.$$

Composing the restrictions and degenerations yields the claimed identity. ◀

References

1 Josh Alman. Limits on the universal method for matrix multiplication. *Theory Comput.*, 17:Paper No. 1, 30, 2021. doi:10.4086/toc.2021.v017a001.

2 Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2005–2039. SIAM, 2025. doi:10.1137/1.9781611978322.

3 Josh Alman and Baitian Li. Kronecker powers, orthogonal vectors, and the asymptotic spectrum. In *2025 IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1411–1441. IEEE, 2025. doi:10.1109/FOCS63196.2025.00074.

4 Josh Alman and Virginia Vassilevska Williams. Limits on all known (and some unknown) approaches to matrix multiplication. *SIAM J. Comput.*, 52(6):FOCS18–285–FOCS18–315, 2023. doi:10.1137/19M124695X.

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

6 Andris Ambainis, Yuval Filmus, and François Le Gall. Fast matrix multiplication: limitations of the coppersmith-winograd method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*, pages 585–593, 2015. doi:10.1145/2746539.2746554.

7 Theodore Baker, John Gill, and Robert Solovay. Relativizations of the $\mathcal{P} =? \mathcal{NP}$ question. *SIAM J. Comput.*, 4(4):431–442, 1975. doi:10.1137/0204037.

8 D. Bini. Relations between exact and approximate bilinear algorithms. Applications. *Calcolo*, 17(1):87–97, 1980. doi:10.1007/BF02575865.

9 Andreas Björklund, Radu Curticapean, Thore Husfeldt, Petteri Kaski, and Kevin Pratt. Fast deterministic chromatic number under the asymptotic rank conjecture. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2804–2818. SIAM, 2025. doi:10.1137/1.9781611978322.91.

10 Andreas Björklund and Petteri Kaski. The asymptotic rank conjecture and the set cover conjecture are not both true. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 859–870, 2024. doi:10.1145/3618260.3649656.

11 Andreas Björklund, Petteri Kaski, Tomohiro Koana, and Jesper Nederlof. Kronecker scaling of tensors with applications to arithmetic circuits and algorithms. In *53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 36:1–36:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi:10.4230/LIPIcs.ICALP.2026.36.

12 Markus Bläser. *Fast Matrix Multiplication.* Number 5 in Graduate Surveys. Theory of Computing Library, 2013. doi:10.4086/toc.gs.2013.005.

13 Manuel Blum. A machine-independent theory of the complexity of recursive functions. *J. Assoc. Comput. Mach.*, 14:322–336, 1967. doi:10.1145/321386.321395.

14 Cornelius Brand, Radu Curticapean, Baitian Li, and Kevin Pratt. Faster convolutions: Yates and strassen revisited. In *2026 SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 328–339. SIAM, 2026. doi:10.1137/1.9781611978964.25.

15 Peter Bürgisser, Michael Clausen, and Mohammad A Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013. doi:10.1007/978-3-662-03338-8.

16 Matthias Christandl, François Le Gall, Vladimir Lysikov, and Jeroen Zuiddam. Barriers for rectangular matrix multiplication. *Comput. Complexity*, 34(1):Paper No. 4, 38, 2025. doi:10.1007/s00037-025-00264-9.

17 Matthias Christandl, Vladimir Lysikov, and Jeroen Zuiddam. Weighted slice rank and a minimax correspondence to Strassen's spectra. *J. Math. Pures Appl. (9)*, 172:299–329, 2023. doi:10.1016/j.matpur.2023.02.006.

18 Matthias Christandl, Péter Vrana, and Jeroen Zuiddam. Barriers for fast matrix multiplication from irreversibility. *Theory Comput.*, 17:Paper No. 2, 32, 2021. doi:10.4086/toc.2021.v017a002.

19 Matthias Christandl, Péter Vrana, and Jeroen Zuiddam. Universal points in the asymptotic spectrum of tensors. *J. Amer. Math. Soc.*, 36(1):31–79, 2023. doi:10.1090/jams/996.

20 Austin Conner, Fulvio Gesmundo, Joseph M. Landsberg, and Emanuele Ventura. Rank and border rank of Kronecker powers of tensors and Strassen's laser method. *Comput. Complexity*, 31(1):Paper No. 1, 40, 2022. doi:10.1007/s00037-021-00217-y.

21 Austin Conner, Joseph M. Landsberg, Fulvio Gesmundo, and Emanuele Ventura. Kronecker powers of tensors and strassen's laser method. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151, pages 10:1–10:28, 2020. doi:10.4230/LIPIcs.ITCS.2020.10.

22 D. Coppersmith and S. Winograd. On the asymptotic complexity of matrix multiplication. *SIAM J. Comput.*, 11(3):472–492, 1982. doi:10.1137/0211038.

23 Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.*, 9(3):251–280, 1990. doi:10.1016/S0747-7171(08)80013-2.

24 Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *2023 IEEE 64th annual symposium on Foundations of Computer Science (FOCS)*, pages 2129–2138. IEEE, 2023. doi:10.1109/FOCS57990.2023.00130.

25 Cosimo Flavi, Joachim Jelisiejew, and Mateusz Michałek. Symmetric powers: structure, smoothability, and applications. *Int. Math. Res. Not. IMRN*, 2025(18):Paper No. rnaf277, 40, 2025. doi:10.1093/imrn/rnaf277.

26 Willem Haemers. On some problems of Lovász concerning the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(2):231–232, 1979. doi:10.1109/TIT.1979.1056027.

27 Shuichi Hirahara and Nobutaka Shimizu. Error-correction of matrix multiplication algorithms. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 785–794, 2025. doi:10.1145/3717823.3718244.

28 Shuichi Hirahara and Nobutaka Shimizu. An optimal error-correcting reduction for matrix multiplication. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, pages 97–1, 2025. doi:10.4230/LIPIcs.ICALP.2025.97.

29 Petteri Kaski. Tensor exponents and tensor sequences beyond matrix multiplication. Talk at the Simons Institute for the Theory of Computing, October 2025. URL: https://simons.berkeley.edu/talks/petteri-kaski-aalto-university-2025-10-31.

30 Petteri Kaski and Mateusz Michałek. A universal sequence of tensors for the asymptotic rank conjecture. In *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, pages 64:1–64:24, 2025. doi:10.4230/LIPIcs.ITCS.2025.64.

31 Kisun Lee. Asymptotic rank bounds: a numerical census, 2026. arXiv:2601.08119.

32 Thomas Lickteig. Typical tensorial rank. *Linear Algebra Appl.*, 69:95–120, 1985. doi:10.1016/0024-3795(85)90070-9.

33 V. Ya. Pan. New combinations of methods for the acceleration of matrix multiplication. *Comput. Math. Appl.*, 7(1):73–125, 1981. doi:10.1016/0898-1221(81)90009-2.

34 Kevin Pratt. A stronger connection between the asymptotic rank conjecture and the set cover conjecture. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 871–874, 2024. doi:10.1145/3618260.3649620.

35 Robert Robere and Jeroen Zuiddam. Amortized circuit complexity, formal complexity measures, and catalytic algorithms. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 759–769. IEEE, 2021. doi:10.1109/FOCS52979.2021.00079.

36 Francesco Romani. Some properties of disjoint sums of tensors related to matrix multiplication. *SIAM J. Comput.*, 11(2):263–267, 1982. doi:10.1137/0211020.

37 Keiya Sakabe, M. Levent Doğan, and Michael Walter. Strassen's support functionals coincide with the quantum functionals, 2026. arXiv:2601.21553.

38 A. Schönhage. Partial and total matrix multiplication. *SIAM J. Comput.*, 10(3):434–455, 1981. doi:10.1137/0210032.

39 V. Strassen. Rank and optimal computation of generic tensors. *Linear Algebra Appl.*, 52/53:645–685, 1983. doi:10.1016/0024-3795(83)80041-X.

40 V. Strassen. Relative bilinear complexity and matrix multiplication. *J. Reine Angew. Math.*, 375/376:406–443, 1987. doi:10.1515/crll.1987.375-376.406.

41 V. Strassen. The asymptotic spectrum of tensors. *J. Reine Angew. Math.*, 384:102–152, 1988. doi:10.1515/crll.1988.384.102.

42 V. Strassen. Degeneration and complexity of bilinear maps: some asymptotic spectra. *J. Reine Angew. Math.*, 413:127–180, 1991. doi:10.1515/crll.1991.413.127.

43 Volker Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354–356, 1969. doi:10.1007/BF02165411.

44 Volker Strassen. Vermeidung von Divisionen. *J. Reine Angew. Math.*, 264:184–202, 1973. doi:10.1515/crll.1973.264.184.

45 Volker Strassen. Algebra and complexity. In *First European Congress of Mathematics Paris, July 6–10, 1992: Vol. II: Invited Lectures (Part 2)*, pages 429–446. Springer, 1994.

46 Volker Strassen. Komplexität und Geometrie bilinearer Abbildungen. *Jahresber. Deutsch. Math.-Verein.*, 107(1):3–31, 2005.

47 Avi Wigderson and Jeroen Zuiddam. Asymptotic spectra: Theory, applications and extensions. *Bull. Amer. Math. Soc.*, 63(2):199–331, 2026. doi:10.1090/bull/1880.

48 Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835. SIAM, 2024. doi:10.1137/1.9781611977912.

49 Jeroen Zuiddam. The asymptotic spectrum of graphs and the Shannon capacity. *Combinatorica*, 39(5):1173–1184, 2019. doi:10.1007/s00493-019-3992-5.

A Degenerations between direct sums of matrix multiplication

The goal of this appendix is to clarify how the direct-sum degeneration used by Schönhage [38, §8],

$$\langle 1,5,20\rangle \oplus \langle 10,2,5\rangle \oplus \langle 10,10,1\rangle \trianglelefteq \langle 132\rangle \oplus 5 \odot \langle 1,1,2\rangle, \tag{A.1}$$

can be analyzed (via Strassen calculus) to obtain the inequality

$$3 \cdot 100^{\omega/3} \leqslant 132 + 5 \cdot 2^{\omega/3}. \tag{A.2}$$

Moreover, this analysis is *optimal* under a suitable formalization.

More generally, given a single degeneration between direct sums of matrix multiplication tensors, one may ask for the best upper bound on ω that can be extracted from it:

$$\bigoplus_{i=1}^{t} \langle n_i^{(1)}, n_i^{(2)}, n_i^{(3)}\rangle \trianglelefteq \bigoplus_{j=1}^{s} \langle m_j^{(1)}, m_j^{(2)}, m_j^{(3)}\rangle. \tag{A.3}$$

We begin with a Strassen-calculus proof of the classical asymptotic sum inequality, which yields an upper bound in the special case where the right-hand side is diagonal, i.e., $m_j^{(1)} = m_j^{(2)} = m_j^{(3)} = 1$. This proof is due to Strassen [41, §4].

▶ **Theorem 43** (Asymptotic sum inequality [38, Theorem 7.1]). *If there exists a degeneration,*

$$\bigoplus_{i=1}^{t} \langle n_i^{(1)}, n_i^{(2)}, n_i^{(3)}\rangle \trianglelefteq \langle s\rangle,$$

let τ be the (unique) solution to the equation

$$\sum_{i=1}^{t} (n_i^{(1)} n_i^{(2)} n_i^{(3)})^{\tau} = s,$$

then $\omega \leqslant 3\tau$.

Proof.

Step 1: Pass to the asymptotic spectrum. Fix $\phi \in \mathcal{X}$ with parameters $(\theta_1, \theta_2, \theta_3) \in \Delta_{\mathrm{MM}}$. Applying ϕ to the degeneration yields

$$F(\theta_1, \theta_2, \theta_3) \coloneqq \sum_{i=1}^{t} (n_i^{(1)})^{\theta_1} (n_i^{(2)})^{\theta_2} (n_i^{(3)})^{\theta_3} \leqslant s.$$

Step 2: Symmetry of tensors. We can permute the modes and obtain degenerations that permute the parameters of matrix multiplication. For any $\sigma \in \mathfrak{S}_3$, we also have a degeneration

$$\bigoplus_{i=1}^{t} \langle n_i^{(\sigma^{-1}(1))}, n_i^{(\sigma^{-1}(2))}, n_i^{(\sigma^{-1}(3))} \rangle \trianglelefteqslant \langle s \rangle,$$

hence $F(\theta_{\sigma(1)}, \theta_{\sigma(2)}, \theta_{\sigma(3)}) \leqslant s$.

Step 3: Average by convexity. The function F is convex (it is a sum of exponentials), so by Jensen's inequality,

$$F\Big(\mathop{\mathbb{E}}_{\sigma}[(\theta_{\sigma(1)}, \theta_{\sigma(2)}, \theta_{\sigma(3)})]\Big) \leqslant \mathop{\mathbb{E}}_{\sigma}[F(\theta_{\sigma(1)}, \theta_{\sigma(2)}, \theta_{\sigma(3)})] \leqslant s.$$

Let $\varpi \coloneqq \theta_1 + \theta_2 + \theta_3$. Averaging over $\sigma \in \mathfrak{S}_3$ gives

$$\mathop{\mathbb{E}}_{\sigma}[(\theta_{\sigma(1)}, \theta_{\sigma(2)}, \theta_{\sigma(3)})] = (\varpi/3, \varpi/3, \varpi/3).$$

Since $\varpi \leqslant \omega$ for all $(\theta_1, \theta_2, \theta_3) \in \Delta_{\mathrm{MM}}$ and F is coordinatewise nondecreasing, we obtain $F(\omega/3, \omega/3, \omega/3) \leqslant s$, which implies the claim. ◀

We emphasize that the first two steps in the proof above amount to a simple geometric relaxation, which extends verbatim to the case where the right-hand side is also a direct sum of matrix multiplication tensors:

Step 1: Pass to the asymptotic spectrum. Every point $(\theta_1, \theta_2, \theta_3) \in \Delta_{\mathrm{MM}}$ must lie in the feasible region

$$\Delta_{\mathrm{MM}} \subseteq \Delta_{(\mathrm{A.3})} \coloneqq \left\{ (\theta_1, \theta_2, \theta_3) : \sum_{i=1}^{t} (n_i^{(1)})^{\theta_1} (n_i^{(2)})^{\theta_2} (n_i^{(3)})^{\theta_3} \leqslant \sum_{j=1}^{s} (m_j^{(1)})^{\theta_1} (m_j^{(2)})^{\theta_2} (m_j^{(3)})^{\theta_3} \right\}.$$

Step 2: Symmetry. Permuting tensor modes yields the containment

$$\Delta_{\mathrm{MM}} \subseteq \Delta_{(\mathrm{A.3})}^{\sigma} = \{(\theta_1, \theta_2, \theta_3) : (\theta_{\sigma(1)}, \theta_{\sigma(2)}, \theta_{\sigma(3)}) \in \Delta_{(\mathrm{A.3})}\}$$

for all $\sigma \in \mathfrak{S}_3$.

Step 3: Optimize over the relaxation. Intersecting all six permuted regions gives the strongest symmetric relaxation: $\Delta_{\mathrm{MM}} \subseteq \Delta_{(\mathrm{A.3})}^{\mathfrak{S}} \coloneqq \bigcap_{\sigma \in \mathfrak{S}_3} \Delta_{(\mathrm{A.3})}^{\sigma}$. By Strassen duality, we then obtain an upper bound on ω:

$$\omega = \max_{\Delta_{\mathrm{MM}}} (\theta_1 + \theta_2 + \theta_3) \leqslant \max_{\Delta_{(\mathrm{A.3})}^{\mathfrak{S}}} (\theta_1 + \theta_2 + \theta_3).$$

From this viewpoint, the classical asymptotic sum inequality says that, in the diagonal case, the region $\Delta_{(A.3)}$ is convex, so the maximum of $\theta_1 + \theta_2 + \theta_3$ over $\Delta^{\mathfrak{S}}_{(A.3)}$ is attained on the principal diagonal $\theta_1 = \theta_2 = \theta_3$, which simplifies the analysis.

Therefore, the analysis of the direct-sum identity (A.1) can be carried out via computer-aided optimization. The inequality induced by the degeneration is

$$\Delta_{(A.1)} \coloneqq \left\{(\theta_1, \theta_2, \theta_3) : 5^{\theta_2} \cdot 20^{\theta_3} + 10^{\theta_1} \cdot 2^{\theta_2} \cdot 5^{\theta_3} + 10^{\theta_1} \cdot 10^{\theta_2} \leqslant 132 + 5 \cdot 2^{\theta_3}\right\}.$$

It turns out (and can be verified in standard mathematical software packages) that the maximum of $\theta_1 + \theta_2 + \theta_3$ over $\Delta^{\mathfrak{S}}_{(A.1)}$ is attained at $\theta_1 = \theta_2 = \theta_3$. This justifies (A.2) as a valid upper bound on ω.

One may then ask whether an asymptotic-sum-inequality statement holds in *full generality*. Namely, can one conclude

$$(A.3) \implies \sum_{i=1}^{t} (n_i^{(1)} n_i^{(2)} n_i^{(3)})^{\omega/3} \leqslant \sum_{j=1}^{s} (m_j^{(1)} m_j^{(2)} m_j^{(3)})^{\omega/3}. \tag{A.4}$$

The answer is nuanced: in a certain relativized sense the answer is *no*, but it is also plausible that the answer is *yes* for tensors.

We first explain the *no* side. A natural first attempt is to let τ be the largest real satisfying

$$\sum_{i=1}^{t} (n_i^{(1)} n_i^{(2)} n_i^{(3)})^{\tau} \leqslant \sum_{j=1}^{s} (m_j^{(1)} m_j^{(2)} m_j^{(3)})^{\tau},$$

Then $(\tau, \tau, \tau) \in \Delta^{\mathfrak{S}}_{(A.3)}$, so 3τ provides a lower bound on the best upper bound on ω obtainable from the identity. However, the converse need not hold: the maximum of $\theta_1 + \theta_2 + \theta_3$ over $\Delta^{\mathfrak{S}}_{(A.3)}$ may be attained away from the principal diagonal $\theta_1 = \theta_2 = \theta_3$. In such cases, the optimal bound on ω extracted from the degeneration can be strictly larger than 3τ.

A cautious reader may notice that we have been deliberately vague about what it means to obtain "the best bound on ω from a single identity". In particular, one should clarify what is meant by "using only the identity" as opposed to using additional tensor-specific facts.

The underlying intuition is that the classical asymptotic sum inequality – and the bound (A.2) derived from (A.1) – is, in a sense, *blind* to the arithmetic structure of tensors. Once one abstracts the basic axioms governing rank and degeneration, the same argument goes through without using any additional tensor-specific structure. From the perspective of complexity theory, this can be phrased as a form of *relativization* [7]: in any mathematical structure that satisfies the same axioms and "knows" only the single identity (A.3), an asymptotic-sum-inequality type bound still holds. In particular, one can show that there exists a relativized world in which ω cannot be smaller than $\max_{\Delta^{\mathfrak{S}}_{(A.3)}} (\theta_1 + \theta_2 + \theta_3)$. One way to formalize this is to work in a *universal commutative semiring* that encodes the axioms together with the given identity. We give a rigorous treatment in the subsequent subsection.

We briefly indicate why the answer *might* nevertheless be yes. The discussion above only shows that (A.4) is too strong to hold in all relativized settings; additional structure specific to tensors might still imply it. A natural candidate is *Strassen's convexity conjecture*, which asserts that Δ_{MM} is a convex subset of $\mathbb{R}^3$. If Δ_{MM} were convex, then (A.4) would follow almost immediately. Moreover, the conjecture is not entirely speculative: Strassen proved partial results showing that Δ_{MM} is star-convex with respect to the points $(0, 1, 1)$, $(1, 0, 1)$, and $(1, 1, 0)$; see [41, §5]. For further discussion, see [41, §4, 5] or the more detailed exposition in [47, Part II, III].

A.1 Strassen duality and asymptotic sum inequality in general rings

To make the preceding discussion precise, we briefly recall the general notion of rank and Strassen duality in the abstract setting of commutative semirings. This framework is explained in detail by Wigderson and Zuiddam [47, Part I].

General framework of Strassen duality. Let $\mathcal{R}$ be a commutative semiring with additive identity 0 and multiplicative identity 1. We think of elements of $\mathcal{R}$ as the "objects" of interest. For example, in the $\mathbb{F}$-tensor world, $\mathcal{R} = \mathcal{R}_{\mathbb{F}}$ can be taken to be the set of all $\mathbb{F}$-tensors (up to isomorphism), with addition given by direct sum $\oplus$ and multiplication given by tensor product $\otimes$; the zero tensor plays the role of 0, and the unit diagonal tensor $\langle 1 \rangle$ plays the role of 1.

We also equip $\mathcal{R}$ with a preorder P (written $\leqslant_P$), which encodes the basic comparison relation between objects (e.g., restriction or degeneration). In this setting, the rank of an element $a \in \mathcal{R}$ is defined as

$$\mathrm{R}_P(a) \coloneqq \min\{n \in \mathbb{N} : a \leqslant_P n\},$$

where $n \in \mathcal{R}$ denotes the sum of n copies of the unit 1. To ensure that this notion behaves as expected, one assumes the following axioms; a preorder satisfying them will be called a *Strassen preorder*:

(Preorder) Reflexivity ($a \leqslant_P a$) and transitivity ($a \leqslant_P b, b \leqslant_P c \implies a \leqslant_P c$).

(Compatibility) $0 \leqslant_P a$, and the preorder respects addition and multiplication: $a \leqslant_P b \implies a + c \leqslant_P b + c$ and $ac \leqslant_P bc$.

(Faithfulness on natural numbers) For $n, m \in \mathbb{N}$, one has $n \leqslant_P m$ if and only if $n \leqslant m$ as natural numbers.

(Strong Archimedean property) For any $a \neq 0$, there exists $n \in \mathbb{N}$ such that $1 \leqslant_P a \leqslant_P n$.

From now on, we omit P when unambiguous. Under these axioms, the rank function is subadditive and submultiplicative, and one can define the asymptotic rank $\underset{\sim}{\mathrm{R}}(a)$ in the usual way. For example, in the tensor setting, both restriction $\leqslant$ and degeneration $\trianglelefteqslant$ satisfy these axioms.

One can then define the asymptotic spectrum $\mathcal{X}$ of $(\mathcal{R}, \leqslant)$: it consists of all functions $\phi \colon \mathcal{R} \to \mathbb{R}_{\geqslant 0}$ that

(Semiring homomorphism) $\phi(a + b) = \phi(a) + \phi(b)$ and $\phi(ab) = \phi(a)\phi(b)$.

(Normalization) $\phi(n) = n$ for all $n \in \mathbb{Z}_{\geqslant 0}$.

(Monotonicity) $a \leqslant b \implies \phi(a) \leqslant \phi(b)$.

Strassen duality (as discussed in Section 4) remains valid in this abstract setting. For our purposes, we mainly use the identity

$$\underset{\sim}{\mathrm{R}}(a) = \max_{\phi \in \mathcal{X}} \phi(a).$$

Relativization of matrix multiplication. We now formulate an abstract semiring *representing* matrix multiplication, in order to study the asymptotic sum inequality. We define $\mathcal{R}_{\mathrm{MM}}$ to be the commutative semiring of finite formal $\mathbb{N}$-linear combinations of symbols $\langle n, m, p \rangle$:

$$\mathcal{R}_{\mathrm{MM}} = \left\{ \sum_{n,m,p} a_{nmp} \langle n, m, p \rangle : a_{nmp} \in \mathbb{N}, \sum_{n,m,p} a_{nmp} < \infty \right\}.$$

Addition in $\mathcal{R}_{\mathrm{MM}}$ is defined by adding coefficients coordinatewise, and multiplication is defined on generators by $\langle n,m,p\rangle \cdot \langle n',m',p'\rangle = \langle nn',mm',pp'\rangle$ and extended bilinearly:

$$\left(\sum_{n,m,p} a_{nmp}\langle n,m,p\rangle\right)\left(\sum_{n',m',p'} b_{n'm'p'}\langle n',m',p'\rangle\right) = \sum_{\substack{n,m,p\\ n',m',p'}} a_{nmp}b_{n'm'p'}\langle nn',mm',pp'\rangle.$$

Note that $\langle 1,1,1\rangle$ is the multiplicative identity, so we will identify $\langle 1,1,1\rangle$ with 1.

One may view elements of $\mathcal{R}_{\mathrm{MM}}$ as formal analogues of direct sums of matrix multiplication tensors, e.g., $\sum_{n,m,p} a_{nmp}\langle n,m,p\rangle$ corresponds heuristically to $\bigoplus_{n,m,p} a_{nmp} \odot \langle n,m,p\rangle$. However, elements of $\mathcal{R}_{\mathrm{MM}}$ are purely formal: they do not "know" anything about tensors beyond the axioms we impose below.

We next define a preorder by specifying a set of axioms. Let Σ be a collection of inequalities $a \leqslant b$ with $a,b \in \mathcal{R}_{\mathrm{MM}}$, encoding a minimal set of properties of matrix multiplication. Concretely, Σ contains the following families of inequalities (for all $n \in \mathbb{Z}_+$):

(Monotonicity) For any $n \in \mathbb{Z}_+$, we have $\langle n,1,1\rangle \leqslant \langle n+1,1,1\rangle$, $\langle 1,n,1\rangle \leqslant \langle 1,n+1,1\rangle$ and $\langle 1,1,n\rangle \leqslant \langle 1,1,n+1\rangle$.

(Trivial upper bound) For any $n \in \mathbb{Z}_+$, we have $\langle n,1,1\rangle \leqslant n$, $\langle 1,n,1\rangle \leqslant n$ and $\langle 1,1,n\rangle \leqslant n$.

Given a set of axioms Σ, we write $\Sigma \vdash a \leqslant b$ if the inequality $a \leqslant b$ can be *derived* from Σ in finitely many steps using only (Preorder) and (Compatibility).

We define a preorder $P(\Sigma)$ on $\mathcal{R}_{\mathrm{MM}}$ by declaring $a \leqslant_{P(\Sigma)} b$ if and only if $\Sigma \vdash a \leqslant b$.

▶ **Proposition 44.** *The preorder $P(\Sigma)$ is a Strassen preorder.*

Proof. By construction, $P(\Sigma)$ is a preorder and is compatible with addition and multiplication. We verify the remaining axioms.

For (Strong Archimedean property), let $a = \sum_{n,m,p} a_{nmp}\langle n,m,p\rangle \in \mathcal{R}_{\mathrm{MM}}$. From Σ one can derive

$$\sum_{n,m,p} a_{nmp} \leqslant \sum_{n,m,p} a_{nmp}\langle n,m,p\rangle \leqslant \sum_{n,m,p} a_{nmp} \cdot nmp.$$

If $a \neq 0$, then $\sum_{n,m,p} a_{nmp} \geqslant 1$, so in particular $1 \leqslant a \leqslant N$ for some $N \in \mathbb{N}$.

Finally, for (Faithfulness on natural numbers), observe that each inequality in Σ holds when interpreting elements of $\mathcal{R}_{\mathrm{MM}}$ in the tensor semiring $\mathcal{R}_{\mathbb{F}}$ (with restriction preorder). Hence, if $\Sigma \vdash n \leqslant m$, then $\langle n\rangle \leqslant \langle m\rangle$ in the tensor world, which forces $n \leqslant m$ as natural numbers. ◀

Since $P(\Sigma)$ is a Strassen preorder, Strassen duality applies, and it becomes meaningful to describe its asymptotic spectrum. In particular, the parametrization in Proposition 7 still holds: re-running the proof shows that it relies only on (Monotonicity), which is already contained in Σ. Moreover, in this setting the asymptotic spectrum is naturally parametrized by triples $(\theta_1,\theta_2,\theta_3)$, since every element of $\mathcal{R}_{\mathrm{MM}}$ is built from generators $\langle n,m,p\rangle$.

Now fix a collection Γ of additional identities of interest (i.e., inequalities $a \leqslant b$ in $\mathcal{R}_{\mathrm{MM}}$). In principle, Γ may be infinite – for instance, it could encode an abstracted family of degenerations arising from the laser method – but the reader may keep in mind the case where Γ consists of a single identity, such as Schönhage's.

We analogously define the preorder $P(\Sigma \cup \Gamma)$ by closing $\Sigma \cup \Gamma$ under derivations. It is immediate that $P(\Sigma \cup \Gamma)$ satisfies the axioms of a Strassen preorder except, potentially, (Faithfulness on natural numbers): for an arbitrary choice of Γ, one could force a contradiction (e.g., by including $1 \leqslant 0$). In the applications we have in mind, Γ comes from genuine tensor identities, and faithfulness holds.

From now on, we assume $P(\Sigma \cup \Gamma)$ is indeed a Strassen preorder, and now try to describe the structure of the corresponding Δ_{MM} in this case, denoted as $\Delta_{\mathrm{MM}}(\Gamma)$.

For an element in $\mathcal{R}_{\mathrm{MM}}$:

$$a = \sum_{n,m,p} a_{nmp}\langle n, m, p\rangle,$$

and for $(\theta_1, \theta_2, \theta_3)$ we write

$$a(\theta_1, \theta_2, \theta_3) \coloneqq \sum_{n,m,p} a_{nmp} n^{\theta_1} m^{\theta_2} p^{\theta_3}.$$

▶ **Proposition 45.** *Let Δ_Γ be the feasible region cut out by the inequalities in Γ:*

$$\Delta_\Gamma \coloneqq \Big\{(\theta_1, \theta_2, \theta_3) : a(\theta_1, \theta_2, \theta_3) \leqslant b(\theta_1, \theta_2, \theta_3) \text{ for all } (a \leqslant b) \in \Gamma\Big\}.$$

If $P(\Sigma \cup \Gamma)$ is a Strassen preorder, then $\Delta_{\mathrm{MM}}(\Gamma) = [0, 1]^3 \cap \Delta_\Gamma$.

Proof. The spectrum $\Delta_{\mathrm{MM}}(\Gamma)$, by definition, should be the set of $(\theta_1, \theta_2, \theta_3)$ that the function defined by

$$\phi(a) = a(\theta_1, \theta_2, \theta_3)$$

satisfies all the axioms of an asymptotic-spectrum point. It is straightforward to verify that ϕ is nonnegative and satisfies (Semiring homomorphism) and (Normalization).

The only nontrivial condition is (Monotonicity). If $\Sigma \cup \Gamma \vdash a \leqslant b$, then we must have $a(\theta_1, \theta_2, \theta_3) \leqslant b(\theta_1, \theta_2, \theta_3)$. Since the derivation rules are compatible with passing to nonnegative real numbers, it suffices to enforce these inequalities for the generating set $\Sigma \cup \Gamma$.

The inequalities in Γ impose the region Δ_Γ, while the axioms in Σ impose:

- (Monotonicity) implies $\theta_i \geqslant 0$.
- (Trivial upper bound) implies $\theta_i \leqslant 1$.

Thus Σ forces $(\theta_1, \theta_2, \theta_3) \in [0, 1]^3$, and the claim follows. ◀

Define

$$\omega_\Gamma \coloneqq \log_2\big(\underset{\sim}{\mathrm{R}}_{P(\Sigma\cup\Gamma)}(\langle 2, 2, 2\rangle)\big).$$

Since Proposition 9 can also be checked to relativize[4], we obtain the characterization

$$\omega_\Gamma = \max_{\Delta_{\mathrm{MM}}(\Gamma)} (\theta_1 + \theta_2 + \theta_3).$$

On the one hand, ω_Γ is the optimum of $\theta_1 + \theta_2 + \theta_3$ over the feasible region $[0, 1]^3 \cap \Delta_\Gamma$. On the other hand, unwinding the definition yields the following "derivation-only" interpretation:

- Start from the axioms Σ together with the auxiliary identities Γ.
- Derive (in finitely many steps, using only (Preorder) and (Compatibility)) an inequality of the form $\langle n, n, n\rangle \leqslant r$.
- Conclude the bound $\omega \leqslant \log_n r$.
- Take the infimum over all such bounds (using $\omega = \inf_{n\geqslant 1} \log_n \mathrm{R}(\langle n, n, n\rangle)$ in this abstract setting).

This provides a precise meaning of "using only the identities in Γ". Under this notion, Strassen calculus is *complete*: the optimum over $[0, 1]^3 \cap \Delta_\Gamma$ is not merely an upper bound on ω_Γ; it equals ω_Γ.

[4] Strictly speaking, something breaks down for asymptotic subrank $\underset{\sim}{\mathrm{Q}}$, but we will not use it.

Asymptotic sum inequality. We are now ready to formulate a relativized version of the asymptotic sum inequality. There is, however, one remaining ingredient that we have implicitly used but have not yet formalized in $\mathcal{R}_{\mathrm{MM}}$: the symmetry under permuting tensor modes.

Let a be an element

$$a = \sum_{n^{(1)},n^{(2)},n^{(3)}} a_{n^{(1)}n^{(2)}n^{(3)}} \langle n^{(1)}, n^{(2)}, n^{(3)} \rangle,$$

and let $\sigma \in \mathfrak{S}_3$ be a permutation. We write

$$a^\sigma \coloneqq \sum_{n^{(1)},n^{(2)},n^{(3)}} a_{n^{(1)}n^{(2)}n^{(3)}} \langle n^{(\sigma^{-1}(1))}, n^{(\sigma^{-1}(2))}, n^{(\sigma^{-1}(3))} \rangle.$$

(Symmetry) For any inequality $a \leqslant b$ and any $\sigma \in \mathfrak{S}_3$, the permuted inequality $a^\sigma \leqslant b^\sigma$ also holds.

At first glance, (Symmetry) appears to be an additional derivation rule, which would force us to rebuild the preceding formalism. This is unnecessary: we can instead close the axiom set under permutations once and for all.

▶ **Lemma 46.** *Let* $\Gamma^\sigma \coloneqq \{a^\sigma \leqslant b^\sigma : (a \leqslant b) \in \Gamma\}$*, and let* $\Gamma^{\mathfrak{S}} \coloneqq \bigcup_{\sigma\in\mathfrak{S}_3} \Gamma^\sigma$*. Then* $P(\Sigma \cup \Gamma^{\mathfrak{S}})$ *coincides with the preorder obtained from* $\Sigma \cup \Gamma$ *by allowing derivations using (Symmetry) in addition to (Preorder) and (Compatibility).*

Proof. Let P' denote the preorder obtained by closing $\Sigma \cup \Gamma$ under derivations using (Preorder), (Compatibility), and (Symmetry). Since P' is closed under (Symmetry) by definition, it contains every permuted axiom in $\Gamma^{\mathfrak{S}}$, and hence $\Sigma \cup \Gamma^{\mathfrak{S}} \subseteq P'$. Therefore $P(\Sigma \cup \Gamma^{\mathfrak{S}}) \subseteq P'$.

Conversely, we claim that $P(\Sigma \cup \Gamma^{\mathfrak{S}})$ is closed under (Symmetry). Indeed, if $\Sigma \cup \Gamma^{\mathfrak{S}} \vdash a \leqslant b$, then there is a derivation of $a \leqslant b$ from axioms in $\Sigma \cup \Gamma^{\mathfrak{S}}$ using only (Preorder) and (Compatibility). Applying a fixed permutation $\sigma \in \mathfrak{S}_3$ to each line of the derivation yields a valid derivation of $a^\sigma \leqslant b^\sigma$, because $\Sigma \cup \Gamma^{\mathfrak{S}}$ is stable under permutation and the preorder/compatibility rules commute with permuting modes. Thus $P(\Sigma \cup \Gamma^{\mathfrak{S}})$ is closed under (Symmetry), so $P' \subseteq P(\Sigma \cup \Gamma^{\mathfrak{S}})$.

Combining the inclusions gives $P' = P(\Sigma \cup \Gamma^{\mathfrak{S}})$. ◀

Putting the preceding formalization together, we obtain a clean "complete" statement for the generalized asymptotic sum inequality in this relativized setting.

▶ **Proposition 47** (Generalized asymptotic sum inequality)**.** *Let* Γ *be a collection of inequalities in* $\mathcal{R}_{\mathrm{MM}}$*. Allowing derivations using (Preorder), (Compatibility), and (Symmetry), the optimal bound on* ω *obtainable from* Γ *equals* $\omega_{\Gamma^{\mathfrak{S}}}$*. Equivalently, it is the maximum of* $\theta_1 + \theta_2 + \theta_3$ *over the feasible region* $[0,1]^3 \cap \Delta_{\Gamma^{\mathfrak{S}}}$*.*

B Analysis facts

▶ **Lemma 48.** *Let* $d \geqslant 3$ *and* $\varpi \geqslant 2$*. Define*

$$f(\theta) = d^{\varpi+\theta} + d^{2\theta} - (d^3 - d^2)^\theta, \qquad \theta \in [0, \varpi/3].$$

Then f *is strictly increasing on* $[0, \varpi/3]$*, hence*

$$\max_{0\leqslant\theta\leqslant\varpi/3} f(\theta) = f(\varpi/3).$$

Proof. Differentiate:

$$f'(\theta) = (\ln d)\, d^{\varpi+\theta} + 2(\ln d)\, d^{2\theta} - \ln(d^3 - d^2)\,(d^3 - d^2)^{\theta}.$$

Write $d^3 - d^2 = d^3(1 - 1/d)$ and set

$$\delta_d \coloneqq -\frac{\ln(1 - 1/d)}{\ln d} > 0, \quad \text{so} \quad \ln(d^3 - d^2) = (3 - \delta_d)\ln d.$$

Also $(d^3 - d^2)^{\theta} = d^{3\theta}(1 - 1/d)^{\theta} \leqslant d^{3\theta}$. Hence

$$\frac{f'(\theta)}{\ln d} \geqslant d^{\varpi+\theta} + 2d^{2\theta} - (3 - \delta_d)d^{3\theta} = d^{3\theta}\Big(d^{\varpi-2\theta} + 2d^{-\theta} - (3 - \delta_d)\Big).$$

For $\theta \in [0, \varpi/3]$,

$$d^{\varpi-2\theta} + 2d^{-\theta} \geqslant d^{\varpi/3} + 2d^{-\varpi/3} \geqslant d^{2/3} + 2d^{-2/3},$$

where the last inequality uses that $x \mapsto d^x + 2d^{-x}$ is increasing for $x \geqslant \frac{\ln\sqrt{2}}{\ln d}$ and $\frac{\ln\sqrt{2}}{\ln d} < \frac{2}{3} \leqslant \varpi/3$. Therefore,

$$\frac{f'(\theta)}{\ln d} \geqslant d^{3\theta}\Big(d^{2/3} + 2d^{-2/3} - 3\Big).$$

Let $g(d) = d^{2/3} + 2d^{-2/3}$. Since

$$g'(d) = \frac{2}{3}d^{-5/3}\big(d^{4/3} - 2\big) > 0 \quad (d \geqslant 3)$$

and $g(3) = 3^{2/3} + 2 \cdot 3^{-2/3} > 3$, we have $d^{2/3} + 2d^{-2/3} - 3 > 0$ for all $d \geqslant 3$. Thus $f'(\theta) > 0$ on $[0, \varpi/3]$, so f is strictly increasing there and the maximum is attained at $\theta = \varpi/3$. ◀

▶ **Lemma 49.** *Let $d \geqslant 3$. For $\varpi \geqslant 2$, define*

$$F(\varpi) \coloneqq d^{4\varpi/3} + d^{2\varpi/3} - (d^3 - d^2)^{\varpi/3}.$$

Then F is strictly increasing on $[2, 3]$.

Proof. Differentiate:

$$\frac{\mathrm{d}}{\mathrm{d}\varpi}F(\varpi) = \frac{4}{3}(\ln d)d^{4\varpi/3} + \frac{2}{3}(\ln d)d^{2\varpi/3} - \frac{1}{3}\ln(d^3 - d^2)(d^3 - d^2)^{\varpi/3}.$$

Write $d^3 - d^2 = d^3(1 - 1/d)$ and set

$$\delta_d \coloneqq -\frac{\ln(1 - 1/d)}{\ln d} > 0, \qquad \text{so that} \qquad \ln(d^3 - d^2) = (3 - \delta_d)\ln d.$$

Moreover, $(d^3 - d^2)^{\varpi/3} = d^{\varpi}(1 - 1/d)^{\varpi/3} \leqslant d^{\varpi}$. Hence

$$\frac{3}{\ln d}\,\frac{\mathrm{d}}{\mathrm{d}\varpi}F(\varpi) \geqslant 4d^{4\varpi/3} + 2d^{2\varpi/3} - (3 - \delta_d)d^{\varpi}.$$

Factor out $d^{2\varpi/3}$ to obtain

$$\frac{3}{\ln d}\,\frac{\mathrm{d}}{\mathrm{d}\varpi}F(\varpi) \geqslant d^{2\varpi/3}\Big(4d^{2\varpi/3} + 2 - (3 - \delta_d)d^{\varpi/3}\Big).$$

For $\varpi \in [2, 3]$, we have $d^{\varpi/3} \in [d^{2/3}, d]$, hence

$$4d^{2\varpi/3} + 2 - (3 - \delta_d)d^{\varpi/3} \geqslant 4d^{4/3} + 2 - (3 - \delta_d)d.$$

Using $\delta_d > 0$ and $d \geqslant 3$, the right-hand side satisfies

$$4d^{4/3} + 2 - (3 - \delta_d)d > 4d^{4/3} + 2 - 3d = d\big(4d^{1/3} - 3\big) + 2 > 0.$$

Therefore $\frac{\mathrm{d}}{\mathrm{d}\varpi}F(\varpi) > 0$ for all $\varpi \in [2, 3]$, and F is strictly increasing on this interval. ◀

▶ **Lemma 50.** *Let $d \geqslant 3$, $p \in \mathbb{R}_{\geqslant 0}$, and $\varpi \geqslant 2$. Define*

$$h_\varpi(\theta) \coloneqq d^{\varpi+\theta} - (2^\theta - 1)p\, d^{2\theta}, \qquad \theta \in [0, \varpi/3].$$

If $p \leqslant \lfloor d/3 \rfloor$, then h_ϖ is strictly increasing on $[0, \varpi/3]$.

Proof. Differentiate:

$$h'_\varpi(\theta) = (\ln d)\, d^{\varpi+\theta} - p\,\frac{\mathrm{d}}{\mathrm{d}\theta}\big((2^\theta - 1)d^{2\theta}\big).$$

Using

$$\frac{\mathrm{d}}{\mathrm{d}\theta}\big((2^\theta - 1)d^{2\theta}\big) = d^{2\theta}\Big((\ln 2)2^\theta + 2(\ln d)(2^\theta - 1)\Big),$$

we obtain

$$h'_\varpi(\theta) = d^{2\theta}\Big((\ln d)\, d^{\varpi-\theta} - p\big((\ln 2)2^\theta + 2(\ln d)(2^\theta - 1)\big)\Big).$$

For $\theta \in [0, \varpi/3]$ we have $d^{\varpi-\theta} \geqslant d^{2\varpi/3}$ and, since $\varpi \leqslant \omega < 3$ in our applications, also $\theta \leqslant 1$ and hence $2^\theta \leqslant 2$ and $2^\theta - 1 \leqslant 1$. Therefore,

$$h'_\varpi(\theta) \geqslant d^{2\theta}\Big((\ln d)\, d^{2\varpi/3} - p(2\ln 2 + 2\ln d)\Big).$$

Using $p \leqslant \lfloor d/3 \rfloor \leqslant d/3$ and $\varpi \geqslant 2$ (so $d^{2\varpi/3} \geqslant d^{4/3}$), the bracket is at least

$$(\ln d)d^{4/3} - \frac{d}{3}(2\ln 2 + 2\ln d) = d\Big((\ln d)d^{1/3} - \tfrac{2}{3}(\ln 2 + \ln d)\Big).$$

For $d \geqslant 3$, we have $d^{1/3} > 1$ and thus

$$(\ln d)d^{1/3} - \tfrac{2}{3}(\ln 2 + \ln d) \geqslant (\ln 3)3^{1/3} - \tfrac{2}{3}(\ln 2 + \ln 3) > 0.$$

Hence $h'_\varpi(\theta) > 0$ on $[0, \varpi/3]$, so h_ϖ is strictly increasing there. ◀

▶ **Lemma 51.** *Let $d \geqslant 3$, $p \in \mathbb{R}_{\geqslant 0}$, and define for $\varpi \geqslant 2$,*

$$g(\varpi) \coloneqq d^{4\varpi/3} - (2^{\varpi/3} - 1)p\, d^{2\varpi/3}.$$

If $p \leqslant \lfloor d/3 \rfloor$, then g is strictly increasing on $[2, 3]$.

Proof. Differentiate:

$$g'(\varpi) = \frac{4}{3}(\ln d)\, d^{4\varpi/3} - \frac{p}{3}\,\frac{\mathrm{d}}{\mathrm{d}\varpi}\Big((2^{\varpi/3} - 1)d^{2\varpi/3}\Big).$$

Since

$$\frac{\mathrm{d}}{\mathrm{d}\varpi}\Big((2^{\varpi/3} - 1)d^{2\varpi/3}\Big) = d^{2\varpi/3}\Big(\tfrac{\ln 2}{3}2^{\varpi/3} + \tfrac{2\ln d}{3}(2^{\varpi/3} - 1)\Big),$$

we can write

$$g'(\varpi) = \frac{1}{3}d^{2\varpi/3}\Big(4(\ln d)\, d^{2\varpi/3} - p\big((\ln 2)2^{\varpi/3} + 2(\ln d)(2^{\varpi/3} - 1)\big)\Big).$$

For $\varpi \in [2, 3]$ we have $2^{\varpi/3} \leqslant 2$ and $2^{\varpi/3} - 1 \leqslant 1$, hence

$$g'(\varpi) \geqslant \frac{1}{3}d^{2\varpi/3}\Big(4(\ln d)\, d^{2\varpi/3} - p(2\ln 2 + 2\ln d)\Big).$$

Using $p \leqslant d/3$ and $\varpi \geqslant 2$ (so $d^{2\varpi/3} \geqslant d^{4/3}$), it suffices to check

$$4(\ln d)d^{4/3} - \frac{d}{3}(2\ln 2 + 2\ln d) > 0.$$

This is equivalent to

$$d\Big(4(\ln d)d^{1/3} - \tfrac{2}{3}(\ln 2 + \ln d)\Big) > 0,$$

which holds for all $d \geqslant 3$ (e.g., already at $d = 3$ the bracket is positive, and it increases with d). Therefore $g'(\varpi) > 0$ on $[2, 3]$, and g is strictly increasing there. ◀

Separations Above TFNP from Sherali-Adams Lower Bounds

Noah Fleming
Lund University, Sweden
Columbia University, New York, NY, USA

Anna Gál
University of Texas at Austin, TX, USA

Deniz Imrek
University of Texas at Austin, TX, USA

Christophe Marciot
Lund University, Sweden

Abstract

Unlike in TFNP, for which there is an abundance of problems capturing natural existence principles which are incomparable (in the black-box setting), Kleinberg et al. [15] observed that many of the natural problems considered so far in the second level of the total function polynomial hierarchy ($\mathsf{TF}\Sigma_2$) reduce to the Strong Avoid problem. In this work, we prove that the Linear Ordering Principle does not reduce to Strong Avoid in the black-box setting, exhibiting the first $\mathsf{TF}\Sigma_2$ problem that lies outside of the class of problems reducible to Strong Avoid.

The proof of our separation exploits a connection between total search problems in the polynomial hierarchy and proof complexity, recently developed by Fleming, Imrek, and Marciot [9]. In particular, this implies that to show our separation, it suffices to show that there is no small proof of the Linear Ordering Principle in a Σ_2-variant of the Sherali-Adams proof system. To do so, we extend the classical pseudo-expectation method to the Σ_2 setting, showing that the existence of a Σ_2 pseudo-expectation precludes a Σ_2 Sherali-Adams proof. The main technical challenge is in proving the existence of such a pseudo-expectation, we manage to do so by solving a combinatorial covering problem about permutations. We also show that the extended pseudo-expectation bound implies that the Linear Ordering Principle cannot be reduced to any problem admitting a low-degree Sherali-Adams refutation.

2012 ACM Subject Classification Theory of computation → Proof complexity; Theory of computation → Problems, reductions and completeness

Keywords and phrases TFNP, Range Avoidance, Linear Ordering Principle, Separation, Sherali-Adams, Pseudo-Expectation

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.37

Funding *Noah Fleming*: Supported by the Swedish Research Council under grant number 2025-06762.

Acknowledgements We thank the anonymous referees for helpful comments.

1 Introduction

In recent years total search problems in the second level of the polynomial hierarchy ($\mathsf{TF}\Sigma_2$) have received considerable attention. A substantial reason for this is that this class contains a variety of important explicit construction problems, such as finding truth tables of functions with high circuit complexity, pseudo-random generators, rigid matrices, time-bounded Kolmogorov random strings, and extractors. The totality of these problems is witnessed by union-bound type arguments, which can be formalized as reductions to the $\mathsf{TF}\Sigma_2$ problem Avoid [15, 17].


© Noah Fleming, Anna Gál, Deniz Imrek, and Christophe Marciot;
licensed under Creative Commons License CC-BY 4.0
41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 37; pp. 37:1–37:23




Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Avoid

Given $C : [2^n] \to [2^{n+1}]$ output $y \in [2^{n+1}]$ such that for every $x \in [2^n]$, $C(x) \neq y$.

By developing new algorithms for this problem, as well as the harder Linear Ordering Principle, researchers have obtained state-of-the-art circuit lower bounds [4, 18, 20] and data structure lower bounds [14, 19].

Linear Ordering Principle (LOP)

Given $\prec$: $[2^n] \times [2^n] \to \{0, 1\}$, output either

- x such that for every $y \neq x$, $x \prec y$, or *(Minimal element)*
- $x \neq y \neq z$ such that either (i) $x \prec x$, or (ii) $x \not\prec y$ and $y \not\prec x$, or (iii) $x \prec y, y \prec z$ but $x \not\prec z$.

(Linear ordering violation)

Kleinberg et al. [15] initiated the study of $\mathsf{TF}\Sigma_2$ as a class, introducing many of the aforementioned explicit construction problems along with a number of other natural search problems, including Ramsey-Erdős Completion which captures the proof that the n-th Ramsey number is at least $2^{n/2}$. In doing so they observed that, while TFNP has a variety of incomparable subclasses capturing various existence principles, every $\mathsf{TF}\Sigma_2$ problem that they considered admitted a reduction to the following strong variant of the Avoid problem.

Strong Avoid

Given $C : [2^n] \to [2^n + 1]$ output $y \in [2^n + 1]$ such that for every $x \in [2^n]$, $C(x) \neq y$.

[15] raised the question of whether there are any natural problems in $\mathsf{TF}\Sigma_2$ which do not reduce to the problem StrongAvoid. The main contribution of our work is to give the first $\mathsf{TF}\Sigma_2$ problem which is not contained within $\mathsf{StrongAvoid}^{dt}$, the class of problems which admit efficient black-box reductions to StrongAvoid. This answers the question in the black-box setting. Note that a separation in the Turing Machine setting would separate P from Σ_2^P.

▶ **Theorem 1.** LOP $\notin \mathsf{StrongAvoid}^{dt}$.

This complements the work of Korten and Pitassi [18], who show a separation in the other direction. Using our technique, we are also able to show that LOP does not reduce to the $\mathsf{TF}\Sigma_2$ problem LeastNumber, introduced by Thapen [23].

Least Number

Given $f : \{0, 1\}^n \to \{0, 1\}$, output either $\perp$ if $f(x) = 0$ for all x, or x such that $f(x) = 1$ and for all $y < x$, $f(y) = 0$.

▶ **Theorem 2.** LOP $\notin \mathsf{LeastNumber}^{dt}$.

Our separations, along with those known already in the literature, are represented pictorially in Figure 1.

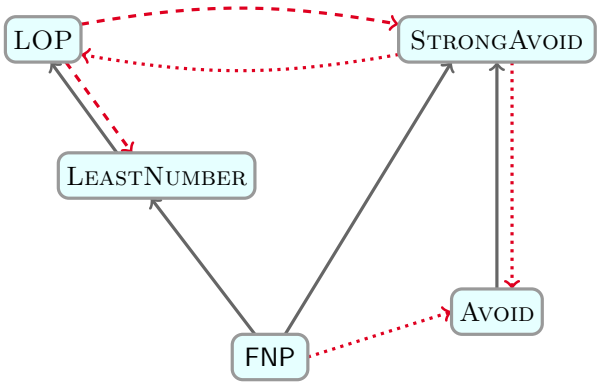


Figure 1 Relationships of some $\mathsf{TF}\Sigma_2^{dt}$ classes. A black arrow from a class A to a class B means that $A \subseteq B$ ([15]). A dashed or dotted arrow from a class A to a class B means that $A \not\subseteq B$. The dashed separations are proved in this paper. [18] proves that StrongAvoid does not reduce to Avoid and LOP. [7] proves that Avoid^{dt} does not contain all of TFNP^{dt}. This implies that none of the other problems in the diagram reduces to Avoid.

1.1 Technical Highlights

To prove Theorem 1 we make use of a connection between $\mathsf{TF}\Sigma_2^{dt}$ (where the superscript dt denotes black-box reductions) and proof complexity, developed in [9]. In doing so we provide the first proof of a separation between black-box $\mathsf{TF}\Sigma_2$ classes that uses proof complexity, confirming that this connection can indeed be useful. In particular, [9] shows that black-box reductions to StrongAvoid are equivalent to efficient proofs in a Σ_2-variant of the Sherali-Adams proof system (see Definition 8) which equips it with a Σ_2-weakening step, extending the weakening rule for resolution to depth-2 formulas, as follows.

▶ **Definition 3.** *Let D be a DNF formula, a* Σ_2-weakening *of D is a collection of DNFs $\{D_i\}_{i\in[m]}$ such that $D \implies D_i$ for every $i \in [m]$. A Σ_2-weakening of a collection of DNFs is a union of Σ_2-weakening of those DNFs.*

This connection with proof complexity reduces proving Theorem 1 to showing that the unsatisfiable formula which encodes the totality of LOP does not have efficient Σ_2-Sherali-Adams proofs (Definition 8).

Pseudo-Expectations

To prove our Sherali-Adams lower bound, we extend the well-developed pseudo-expectation technique for Sherali-Adams proofs to the Σ_2 setting. A pseudo-expectation is an object which appears to be a probability distribution over satisfying assignments to our (unsatisfiable) formula when examining low-degree marginals. The existence of a pseudo-expectation precludes a low-degree Sherali-Adams proof. A Σ_2-pseudo-expectation for LOP is a collection of pseudo-expectations, one for each set of weakenings of the constraints of LOP. See Definition 10 for details. The main technical challenge is to construct such a Σ_2-pseudo-expectation. While Dantchev et al. [5] showed that LOP itself has a high-degree pseudo-expectation, it is not clear how to extend their argument to all Σ_2-weakenings.

A Covering Problem of Permutations

We observe that constructing a Σ_2-pseudo-expectation requires us to solve the following covering problem. Let Ord be the set of all total orders on the set of numbers $\{1, \ldots, n\}$, and let $\text{Ord}^{*1} \subseteq \text{Ord}$ be the set of total orders for which at least one number comes before the element 1. For a fixed $d \leq n$, a set $S \subseteq [n]$ of size d and an order σ on S, let $\mathcal{C}_{S,\sigma} = \{\pi \in \text{Ord} \mid \pi \text{ induces the order } \sigma \text{ on } S\}$ be the set of all total orders on $[n]$ that are consistent with σ on S. Is there a set of strictly less than $d!$ pairs (S_i, σ_i), with each S_i of size d such that Ord^{*1} is contained in the union $\bigcup_i \mathcal{C}_{S_i,\sigma_i}$?

Note that one can cover the set of all total orders with exactly $d!$ such collections $\mathcal{C}_{S,\sigma}$. The question is thus "can we do better if we do not need to cover orders starting with 1?". One can extract from the lower bound of Dantchev et al. [5] that for $d = 2$, such coverings are impossible. Our proof of Theorem 1 implies that the answer to this question is no for any $d \leq n/100$. In our argument, we show that weakenings of some axioms of LOP can be modified to a normalized form without increasing their degree by much. Proving that our pseudo-expectation function is non-negative on these normalized weakenings is equivalent to a lower bound in the above covering problem.

Our linear lower bound on d is tight up to constant factors, as one can construct a covering with less than $d!$ collections for $d = n/2$.

Further Separations from Sherali-Adams Upper Bounds

Leveraging our Σ_2-pseudo-expectation, we show that if a $\mathsf{TF}\Sigma_2$ problem admits a Sherali-Adams upper bound then this problem is separated from the Linear Ordering Principle. Together with a simple Sherali-Adams upper bound for the LeastNumber we obtain Theorem 2. To prove this, we show that reductions can be split in two parts: first, a potentially hard to verify Σ_2-weakening step, followed by a *counter-example reduction*, a type of reduction introduced in [16, 23] that actually happens in TFNP^{dt}. This reveals how the two types of black-box reductions that have been studied in the literature for total functions in the polynomial hierarchy relate.

1.2 Related Works

Ghentiyala et al. [11] showed that LOP does not reduce to Avoid, even in the non-blackbox setting, assuming that $\mathsf{P}^{\mathsf{prMA}} \not\subset \mathsf{AM} \cap \mathsf{coAM}$. The only other known separation in $\mathsf{TF}\Sigma_2^{dt}$ was proven by Korten and Pitassi [18] who showed that the problem StrongAvoid $\notin \mathsf{LOP}^{dt}$. Their separation was obtained by proving a lower bound for depth-3 circuits via a switching lemma. This contrasts with our separation which uses proof complexity techniques.

Proof complexity techniques have been the main method for obtaining separations between classes in TFNP in the black-box setting. This has led to a complete understanding of the relationships between the major classes [1, 2, 8, 12, 13, 21]. Proof complexity has been useful in TFNP because of the fact that membership in (sufficiently uniform) TFNP^{dt} subclasses is equivalent to efficient provability in some associated proof system [3]. This has been used to show that, for example, the class PPADS^{dt} is equivalent to the unary Sherali-Adams proof system [12]. This connection was recently extended to subclasses of the total search problems in the polynomial hierarchy in [9], which is the basis for our work.

Recently, Thapen in [23] exhibited a new type of reductions, namely *counter-example reductions*. They, along with the notion of *herbrandization*, are used to find a TFNP^{dt} translation of what happens at the second level $\mathsf{TF}\Sigma_2^{dt}$ and they can also be used to define

TFNP^{dt} subclasses from $\mathsf{TF}\Sigma_2^{dt}$ problems. In Section 5, we add a bridge between this notion of counter-example reduction and the notion of Σ_2-weakening defined in [9] and show that they are sufficient ingredients to capture reductions.

2 Preliminaries on Black-Box $\mathsf{TF}\Sigma_2$

A *query search problem* is a sequence of relations $R_n \subseteq \{0,1\}^n \times \mathcal{O}_n$, one for each $n \in \mathbb{N}$. It is *total* if for every $x \in \{0,1\}^n$ there is an $o \in \mathcal{O}_n$ such that $(x,o) \in R_n$. We think of $x \in \{0,1\}^n$ as a bit string which can be accessed by querying individual bits, and we will measure the complexity of solving R_n as the number of bits that must be queried. Hence, an efficient algorithm for R_n will be one which finds a suitable o while making at most $\mathsf{polylog}(n)$-many queries to the input. We will not charge the algorithm for other computational steps, and therefore an efficient algorithm corresponds to a shallow decision tree. Total query search problems which can be computed by decision trees of depth $\mathsf{polylog}(n)$ belong to the class FP^{dt}, where dt indicates that it is a black-box class. While search problems are formally defined as sequences $R = (R_n)$, we will often want to speak about individual elements in the sequence. For readability, we will abuse notation and refer to elements R_n in the sequence as total search problems. In addition, we will often drop the subscript n, and rely on context to differentiate.

In this paper we will be considering total query search problems in the second level of the polynomial hierarchy $\mathsf{TF}\Sigma_2^{dt}$.

▶ **Definition 4.** *A total query search problem $R = (R_n)$, where $R_n \subseteq \{0,1\}^n \times \mathcal{O}_n$, belongs to $\mathsf{TF}\Sigma_2^{dt}$ if for every n, $o \in \mathcal{O}_n$ and $x \in \{0,1\}^n$*

$$R_n(x,o) \iff \forall z \in \{0,1\}^\ell, V_{o,z}(x) = 1,$$

where $\ell = \mathsf{polylog}(n)$ and for every $z \in \{0,1\}^\ell$, $V_{o,z}$ is a $\mathsf{polylog}(n)$-depth decision tree.

We can compare the complexity of total search problems by taking reductions between them. The following defines *decision tree reductions* (known as *formulations*) between total query search problems, the query analogue of polynomial-time reductions.

▶ **Definition 5.** *For total query search problems $R \subseteq \{0,1\}^n \times \mathcal{O}_n, S \subseteq \{0,1\}^m \times \mathcal{O}'_m$, there is an S-*formulation of R *if, for every $i \in [m]$ and $o \in \mathcal{O}'_m$, there are functions $f_i : \{0,1\}^n \to \{0,1\}$ and $g_o : \{0,1\}^n \to \mathcal{O}_n$ such that*

$$S(f(x),o) \implies R(x, g_o(x)),$$

where $f(x) = (f_1(x) \dots f_m(x))$. The depth *of the formulation is*

$$d := \max\left(\{\mathsf{depth}(f_i) : i \in [m]\} \cup \{\mathsf{depth}(g_o) : o \in \mathcal{O}'_m\}\right),$$

where $\mathsf{depth}(f)$ denotes the minimum depth of any decision tree which computes f. The size *of the formulation is m, the number of input bits to S. The* complexity *of the formulation is $\log m + d$. The* complexity of reducing R to S *is the minimum complexity of all possible S-formulations of R.*

We extend this definition to sequences in the natural way. If $S = (S_m)$ is a sequence and R_n is a single search problem, then the complexity of reducing R_n to S is the minimum over m of the complexity of reducing R_n to S_m. For two sequences of search problems $S = (S_m)$ and $R = (R_n)$, the complexity of reducing R to S is the complexity of reducing R_n to S, as a function of n. A reduction from R to S is efficient if its complexity is $\mathsf{polylog}(n)$; we denote this by $R \leq_{dt} S$.

Black-box $\mathsf{TF}\Sigma_2$ can be viewed as the study of the following family of total search problems. This will allow us to leverage a close connection between $\mathsf{TF}\Sigma_2^{dt}$ and proof complexity [9] to prove our main result.

▶ **Definition 6.** *Let $F = D_1 \wedge \ldots \wedge D_t$ be a formula in which each D_i is a DNF. The* false formula *search problem* $\mathrm{FF}_F \subseteq \{0,1\}^n \times [t]$ *is defined as*

$$\mathrm{FF}_F(x, o) \iff D_o(x) = 0.$$

▶ **Lemma 7** ([9])**.** *For any $R \in \mathsf{TF}\Sigma_2^{dt}$ there is an unsatisfiable formula $F_R = D_1 \wedge \ldots \wedge D_t$ where each D_i is a DNF of* $\mathsf{polylog}(n)$*-width, such that $R =_{dt} \mathrm{FF}_{F_R}$.*

This extends the well-known connection between TFNP^{dt} and the false clause search problem. The proof of this lemma proceeds by writing down the totality of R as a Σ_3-formula and then taking its negation to obtain F_R. Throughout this paper, we will abuse notation and use R to refer to F_R, using context to differentiate. Hence, $\mathrm{FF}_R = \mathrm{FF}_{F_R}$.

3 Proof Complexity and a Sufficient Condition for Separations

To prove Theorem 1 we make use of a connection between $\mathsf{TF}\Sigma_2^{dt}$ and proof complexity. Fleming, Imrek, and Marciot [9] showed that decision tree reductions between $\mathsf{TF}\Sigma_2^{dt}$ problems are equivalent to efficient proofs in certain proof systems equipped with the Σ_2-weakening rule from Definition 3. Using this connection they showed that proofs in the unary Σ_2-Sherali-Adams proof system are equivalent to reductions to StrongAvoid, which we describe next.

For any boolean formula F, we will assume without loss of generality that all negations occur at the leaves and let $\mathsf{Vars}^+(F)$ be the positive literals in F and $\mathsf{Vars}^-(F)$ be the negative literals. For any conjunct $t = \bigwedge_{x_i \in \mathsf{Vars}^+(t)} x_i \wedge \bigwedge_{x_j \in \mathsf{Vars}^-(t)} \neg x_j$, we associate the following polynomial $t(x) = \prod_{x_i \in \mathsf{Vars}^+(t)} x_i \prod_{x_j \in \mathsf{Vars}^-(t)} (1 - x_j)$. We refer to the polynomials $t(x)$ also as conjuncts and also denote them by t. We say that a *conical junta* is a sum of conjuncts $\mathcal{J} := \sum_i t_i$.

Let $D = \bigvee_{t \in D} t$ be any DNF. We can express D as a polynomial

$$D(x) = \sum_{t \in D} t(x) - 1. \tag{1}$$

The degree of this polynomial is $\deg(D) := \max_{t \in D} \deg(t)$. We refer to $\deg(D)$ as the *degree of the DNF D*. Observe that for any $x \in \{0,1\}^n$, the DNF $D(x)$ is true iff $\sum_{t \in D} t(x) - 1 \geq 0$. We will abuse notation and denote by D both the DNF and the associated polynomial, using context to differentiate.

Throughout we will work with *multi-linear arithmetic*, associating $x_i^2 = x_i$ for every variable x. This has the effect of restricting the underlying linear program to $\{0,1\}$-points.

▶ **Definition 8.** *Let $F = \{D_i\}_{i \in [m]}$ be an unsatisfiable collection of DNFs (a collection with no shared satisfying assignment). A* Σ_2-unary-Sherali-Adams *(denoted* uSA*) proof Π of F is a Σ_2-weakening $F' = \{D'_i\}_{i \in [m']}$ of F together with a list of conical juntas $\mathcal{J}_i, \mathcal{J}$, such that*

$$\sum_{i \in [m']} D'_i \mathcal{J}_i + \mathcal{J} = -1.$$

The degree $\deg(\Pi)$ *is the maximum degree among $D_i, D'_i\mathcal{J}_i$, and $\mathcal{J}$, and the* size $\mathsf{size}(\Pi)$ *is the number of monomials in $D_i, D'_i\mathcal{J}_i, \mathcal{J}$ counted with multiplicity (i.e., each monomial is counted again each time it appears). The* complexity *of the proof is given by* $\mathsf{uSA}(\Pi) := \deg(\Pi) + \log \mathsf{size}(\Pi)$*, and the complexity of proving F is* $\mathsf{uSA}(F) := \min_\Pi \mathsf{uSA}(\Pi)$*, where the minimum is taken over all* uSA *proofs Π of F.*

The only difference between the standard Sherali-Adams proof system and its unary variant is the measure of size; in particular, we cannot use large coefficients. When measuring degree, these systems are identical.

To prove Theorem 1 it suffices to show that there is no efficient Σ_2-uSA proof of LOP.

▶ **Theorem 9** ([9]). *For any $R \in \mathsf{TF}\Sigma_2^{dt}$ there exists a $\mathsf{polylog}(n)$-complexity Σ_2-uSA proof of R iff $R \in \mathsf{StrongAvoid}^{dt}$.*

To prove our lower bound for LOP we generalize the method of pseudo-expectations, which has been developed for Sherali-Adams (see [10] for a survey), to Σ_2-Sherali-Adams degree lower bounds.

▶ **Definition 10.** *For any collection of DNFs $F = \{D_i\}_{i\in[m]}$ a* degree-d pseudo-expectation *for F is a linear function $\tilde{\mathbb{E}} : \mathbb{R}[x] \to \mathbb{R}$ satisfying*
1. $\tilde{\mathbb{E}}[1] = 1$,
2. $\tilde{\mathbb{E}}[\mathcal{J}] \geq 0$ *for every conical junta $\mathcal{J}$ of degree at most d,*
3. $\tilde{\mathbb{E}}[D_i\mathcal{J}] \geq 0$ *for every $D_i \in F$ and conical junta J such that* $\deg(D_i\mathcal{J}) \leq d$.

*A degree-d Σ_2-*pseudo-expectation *for F is a family of degree-d pseudo-expectations $\{\tilde{\mathbb{E}}_G\}$, one for every Σ_2-weakening G of F.*

Note that a Σ_2-pseudo expectation for F is equivalent to a standard pseudo-expectation for the collection of DNFs F' which includes every Σ_2-weakening of the DNFs in F.

▶ **Lemma 11.** *There is a degree-d Σ_2-pseudo-expectation for F iff there is no degree-d Σ_2-uSA proof of F. Moreover, if F does not have a degree-d Σ_2-uSA, then there is a degree-d pseudo-expectation that works for all weakenings of F.*

Proof. The proof follows from the fact that for every unsatisfiable collection of DNFs (and indeed any collection of polynomials with no $\{0,1\}$-solutions), a degree-d pseudo-expectation exists iff a Sherali-Adams proof does not (see e.g. [10]). Applying this to each weakening completes the proof. However, since we will use the forward direction of this theorem we include a proof.

Let $\{\tilde{\mathbb{E}}_G\}$ be a degree-d Σ_2-pseudo-expectation and suppose for contradiction that there is also a degree-d Σ_2-Sherali-Adams proof (G, Π) of F, where $G = \{D'_i\}_{i\in[m']}$ is a Σ_2-weakening of F and Π is a list of conical juntas $\{\mathcal{J}_i\}, \mathcal{J}$ constituting a Sherali-Adams proof of G. Let $\tilde{\mathbb{E}}_G$ be the pseudo-expectation which corresponds to G, then

$$-1 = \tilde{\mathbb{E}}_G[-1] = \tilde{\mathbb{E}}\Big[\sum_{i\in[m']} D'_i\mathcal{J}_i + \mathcal{J}\Big] = \sum_{i\in[m']} \tilde{\mathbb{E}}[D'_i\mathcal{J}_i] + \tilde{\mathbb{E}}[\mathcal{J}] \geq 0.$$

Consider $\mathcal{G} = \bigcup_G G$, where the union is taken over every Σ_2-weakenings of F. Then $\tilde{\mathbb{E}}_{\mathcal{G}}$ must be a d-pseudo-expectation for all weakenings of F, as they are all subsets of $\mathcal{G}$. ◀

4 Separating LOP from Strong Avoid

In this section we prove the following.

▶ **Theorem 12.** *There exists Σ_2-pseudo-expectation for* LOP *of degree* $(\frac{n}{400} - 1)$.

From this, the separation LOP $\notin \mathsf{StrongAvoid}^{dt}$ follows immediately. Note that Theorem 12 is much stronger than we need in order to rule out a reduction to StrongAvoid, as works irrespective of the size of the uSA proofs, or size of the reduction.

Proof of Theorem 1. By Theorem 9 LOP $\in$ StrongAvoid^{dt} iff there exists a polylog(n) complexity Σ_2-uSA proof of the propositional encoding of LOP given by Lemma 7. In particular, such a proof must have polylog(n) degree. However, by Lemma 11, the existence of a degree $\Omega(n)$ Σ_2-pseudo-expectation precludes the existence of such a Σ_2-uSA proof. ◀

In the remainder of this section we prove Theorem 12. Applying Lemma 7 to LOP, we obtain an unsatisfiable formula which consists of the following constraints (axioms) over variables $x_{i,j}$, where $x_{i,j} = 1$ means "$i \prec j$":

- M_i: $\bigvee_{j \in [n] \setminus i} x_{j,i}$, for $i \in n$, *(Non-minimality)*
- R_i: $\neg x_{i,i}$, for $i \in [n]$, *(Irreflexivity)*
- $A_{i,j}$: $\neg x_{i,j} \vee \neg x_{j,i}$, for distinct $i, j \in [n]$, *(Asymmetry)*
- $T_{i,j,k}$: $x_{i,k} \vee \neg x_{i,j} \vee \neg x_{j,k}$, for distinct $i, j, k \in [n]$, *(Transitivity)*
- $O_{i,j}$: $x_{i,j} \vee x_{j,i}$, for distinct $i, j \in [n]$. *(Totality)*

Note that even though each of these axioms could be expressed as a clause, we consider them as DNFs of degree 1. We will use the same pseudo-expectation for each weakening of LOP, which will be the standard choice [5]: a uniform distribution over total orders. Let $\mathrm{Ord} = \{x \in [n]^n \mid \forall i \neq j, x_i \neq x_j\}$ be the set of total orders on $[n]$. We associate total orders on $[n]$ with both strings in $[n]^n$ (simply listing the elements in the given order) and bijective maps $[n] \to [n]$. For $z \in \mathrm{Ord}$, and $T \subseteq [n]$, we define $z{\restriction}T$ as the permutation on T, $\pi : |T| \to T$, that orders those elements in the same order as z does. We denote $z{\restriction}([n] \setminus T)$ by $z \setminus T$ for convenience. For $x \in \mathrm{Ord}$ and a monomial t we denote by $t(x)$ the monomial t evaluated over the variables $x_{i,j}$, where $x_{i,j} = 1$ means "$i \prec j$" (as above) in the order x.

▶ **Definition 13.** *We define our pseudo-expectation* $\tilde{\mathbb{E}}$ *on monomials* t *as*

$$\tilde{\mathbb{E}}\,[t] = \frac{|\{x \in \mathrm{Ord} \mid t(x) = 1\}|}{|\,\mathrm{Ord}\,|}.$$

The definition is extended to arbitrary polynomials by linearity.

By our encoding of DNFs as polynomials (Equation 1), for a DNF $D = \bigvee_i t_i$, we use the notation $\tilde{\mathbb{E}}\,[D]$ to mean $\tilde{\mathbb{E}}\,[\sum_i t_i - 1]$. Note that $\tilde{\mathbb{E}}\,[t + t']$ corresponds to the number of orders either t or t' accepts, plus the number of orders they both accept (all of this divided by $|\,\mathrm{Ord}\,|$). As an example,

$$\tilde{\mathbb{E}}\,[M_i] = \tilde{\mathbb{E}}\Big[\sum_{j \in [n] \setminus i} x_{j,i} - 1\Big] = \sum_{j \in [n] \setminus i} \tilde{\mathbb{E}}\,[x_{j,i}] - \tilde{\mathbb{E}}\,[1] = \sum_{j \in [n] \setminus i} \frac{1}{2} - 1 = \frac{n-1}{2} - 1 = \frac{n-3}{2},$$

as exactly half of the total orders are such that $j \prec i$.

When taking the pseudo-expectation of a sum of terms (or when considering a DNF), we will refer to total orders accepted by more than one term of the sum (or the DNF) as being *over-counted*. If a given order is accepted by $c > 1$ terms, we say that this order contributes $c - 1$ *extra occurrences*. In particular, in the above example, a total order which has i as the 10th element of its order will be accepted by 9 terms, and is thus over-counted. Moreover, 8 of those 9 terms provide extra occurrences of the order.

We will show that our function $\tilde{\mathbb{E}}$ gives a degree-d Σ_2-pseudo-expectation for the set of axioms of LOP for any $d \leq \frac{n}{400} - 1$. We proceed by showing that it satisfies the conditions of Definition 10.

Condition 1. Note that the constant 1 is treated as a monomial such that $1(x) = 1$ for every x, and thus $\tilde{\mathbb{E}}\,[1] = 1$.

Condition 2. Notice that conjuncts containing $x_{i,i}$ are never satisfied by any total order. We can thus always delete those conjuncts without changing the pseudo-expectation. On the other hand, the negated variables $x_{i,i}$ are always true in any total order by the axioms R_i, thus we can simply remove (or replace by constant 1) any negated $x_{i,i}$ without changing the pseudo-expectation.

Additionally we can rewrite any conjunct t over the LOP variables $x_{i,j}$ for $i \neq j$ to an equivalent conjunct t' without using negations, simply replacing any negated variable $x_{i,j}$ by the variable $x_{j,i}$. To see this, notice that for any total order on LOP variables, if $i \neq j$, $x_{j,i}$ holds iff $x_{i,j}$ does not hold, and so the total orders which satisfy t are exactly those that satisfy t'. Thus, $\tilde{\mathbb{E}}[t] = \tilde{\mathbb{E}}[t']$. Observe that the polynomial corresponding to t' is simply a monomial, and $\tilde{\mathbb{E}}[t] = \tilde{\mathbb{E}}[t'] \geq 0$ by definition. Since any conical junta is a sum of conjuncts, this implies that $\tilde{\mathbb{E}}[\mathcal{J}] \geq 0$ for any conical junta $\mathcal{J}$.

Condition 3. It remains to show that condition 3 holds for our function $\tilde{\mathbb{E}}$. Our goal is thus to show that for all conical juntas $\mathcal{J}$, and any weakening D' of any axiom of LOP, $\tilde{\mathbb{E}}[D'\mathcal{J}] \geq 0$, if $\deg(D'\mathcal{J})$ is small enough. First we observe that this easily follows for the axioms other than type M_i. We will handle axioms of type M_i in Section 4.1.

The axioms of LOP that are not of type M_i must hold for every total order. For any such axiom $D = \bigvee_{t \in D} t$ we have $\cup_{t \in D}\{x \in \mathrm{Ord} \,|\, t(x)\} = \mathrm{Ord}$. Furthermore, this holds for any weakening W of D, as W must be satisfied by at least the same assignments as D. Hence, for any conical junta $\mathcal{J}$, we can deduce that

$$\tilde{\mathbb{E}}[W\mathcal{J}] = \left(\sum_{j \in J}\sum_{t \in W} \lambda_j \frac{|\{x \in \mathrm{Ord} \mid t(x)j(x) = 1\}|}{|\,\mathrm{Ord}\,|}\right) - \sum_{j \in J} \lambda_j \frac{|\{x \in \mathrm{Ord} \mid j(x) = 1\}|}{|\,\mathrm{Ord}\,|} \geq 0,$$

where j sums over every conjunct in $\mathcal{J}$, and λ_j is the number of times j appears in $\mathcal{J}$. As every order x must satisfy at least one term t in W, the last inequality holds. This proves that the pseudo-expectation is non-negative for any conical junta and any weakening of an axiom that is not of type M_i, regardless of the degree.

4.1 The No-Minimal-Element Axioms

By symmetry, it suffices to consider weakenings of M_1, weakenings of M_i can be handled analogously. It thus suffices, without loss of generality, to prove that for every low-degree conical junta $\mathcal{J}$, and every low-degree weakening W of M_1, $\tilde{\mathbb{E}}[W\mathcal{J}] \geq 0$. The main difficulty is to handle the diversity of the weakenings of M_1.

We begin by fixing some notation. It will be convenient to go back and forth between total orders and canonical terms representing them. For a set $S \subseteq [n]$ with $|S| \geq 2$, consider an ordering $\pi : [|S|] \to S$ of its elements. We will denote by $[\![\pi(1)\dots\pi(|S|)]\!]$, or $[\![S]\!]_\pi$, the term $x_{\pi(1),\pi(2)}x_{\pi(2),\pi(3)}\cdots x_{\pi(|S|-1),\pi(|S|)}$.

We say that a term *mentions* i if it contains one of the variables $x_{i,j}$ or $x_{j,i}$ for some $j \neq i$. For example, $x_{1,2}x_{2,3}$ mentions $\{1,2,3\}$. We refer to the set of elements that a term mentions as its *support* and the number of such elements as the *support size* of the term.

To simplify our argument, we will first show that we can assume that each DNF has several convenient properties.

▶ **Definition 14.** *We say that a DNF D is* normalized *if*

1. *Every term t of D is of the form $t = [\![S]\!]_\pi$ for some set S which includes 1, and*
2. *All terms have the same support size, which we refer to as the* support size of D.

▶ **Lemma 15.** *Let W be a DNF of degree d. There exists a normalized DNF N of support size $\leq 2d+1$ such that W accepts exactly the same total orders as N does, and $\tilde{\mathbb{E}}[W] = \tilde{\mathbb{E}}[N]$.*

Proof. Let $t \not\equiv 0$ be one of the terms of W. We first transform t into a term t' with no negated literals as follows: we replace each negated literal $\neg x_{i,j}$ in t with $x_{j,i}$, noting that for any total order, $x_{j,i}$ holds iff $x_{i,j}$ does not, and so the total orders satisfying t are exactly those satisfying t'. Hence, $\tilde{\mathbb{E}}[t] = \tilde{\mathbb{E}}[t']$.

Next, we replace t by a collection of terms of the form $[\![S]\!]_\pi$ for some $S \subseteq [n]$ containing 1 (defined below), and a collection of orderings π on S. Let T be the support of t and note that $|T| \leq 2d$ as each variable in t can mention at most two elements. We modify T to define S as follows: Include the element 1 if it was not included yet in T, and pad T with additional elements as necessary, so that the resulting set S has size $|S| = 2d+1$. We use the resulting set S as the support for the collection of terms replacing t.

Let Π be the set of total orders on S that are consistent with t. We claim that for every total order z, $t(z) \Leftrightarrow \bigvee_{\pi\in\Pi}[\![S]\!]_\pi(z)$, and $\tilde{\mathbb{E}}[t] = \tilde{\mathbb{E}}\left[\sum_{\pi\in\Pi}[\![S]\!]_\pi\right]$. In other words, $\bigvee_{\pi\in\Pi}[\![S]\!]_\pi$ accepts exactly the same total orders t does, and each total order accepted by t is consistent with exactly one $\pi \in \Pi$ (thus total orders accepted by t are accepted by a unique term in $\bigvee_{\pi\in\Pi}[\![S]\!]_\pi$). We set N to be the resulting DNF $\bigvee_{t\in W}\bigvee_{\pi\in\Pi_t}[\![S_t]\!]_\pi$. ◀

Note that the transformation in the proof of Lemma 15 can highly increase the number of terms in W, but it does not change its pseudo-expectation, and it at most doubles the degree: Notice that for a normalized DNF with support size k its degree is exactly $k-1$. The support size of the resulting DNF N is at most $2d+1$, thus the degree of N is at most $2d$.

Suppose that M_1 has a weakening W of degree d such that $\tilde{\mathbb{E}}[W] < 0$. This lemma implies that there is also a normalized weakening N of M_1 with $\tilde{\mathbb{E}}[N] < 0$ and of not much larger degree.

▶ **Lemma 16.** *For any normalized DNF N of support size k, every term t of N has $\tilde{\mathbb{E}}[t] = 1/k!$.*

Proof. Let $t = [\![T]\!]_\pi$ be one of the terms of N. As a reminder, $\tilde{\mathbb{E}}[t] = |\{x \in \mathrm{Ord} \mid t(x) = 1\}|/|\mathrm{Ord}|$ where Ord is the set of total orders. Since $t = [\![T]\!]_\pi$, $\tilde{\mathbb{E}}[t] = |\{x \in \mathrm{Ord} \mid x{\restriction}T = \pi\}|/|\mathrm{Ord}|$. To count the number of orders consistent with t, one can simply select the positions of the elements in T, set them in the right order, and then fill in the rest any possible way, thus,

$$|\{x \in \mathrm{Ord} \mid x{\restriction}T = \pi\}| = \binom{n}{k}(n-k)! = \frac{n!}{k!}.$$ ◀

▶ **Lemma 17.** *Let N be normalized DNF with support size k that is a weakening of M_1, and let R be the set of total orders rejected by N. Then the following holds:*

1. *$\tilde{\mathbb{E}}[N] \geq 0$ if and only if N has at least $k!$ terms,*
2. *Let S_i be the set of total orders accepted by exactly i terms in N, then $\tilde{\mathbb{E}}[N] \geq 0$ if and only if $|R| \leq \sum_{i=2}^{\infty}(i-1)|S_i|$.*

We remark that the equivalence with the covering problem presented in the introduction comes from the first point in this lemma. Finding a covering of Ord^{*1} with less than $d!$ sets (that correspond here to normalized terms) is equivalent to finding a normalized weakening N with support size d and negative pseudo-expectation.

Proof. For the first point, using Lemma 16,

$$\tilde{\mathbb{E}}[N] = \tilde{\mathbb{E}}\left[\sum_{[\![T]\!]_\pi\in N}[\![T]\!]_\pi - 1\right] = \sum_{[\![T]\!]_\pi\in N}\tilde{\mathbb{E}}[[\![T]\!]_\pi] - 1 = \frac{\#\text{of terms in } N}{k!} - 1.$$

This is non-negative if and only if there are at least $k!$ terms in N. For the second point, recall that for terms of the form $t = [\![T]\!]_\pi$, we have $\tilde{\mathbb{E}}[t] = |\{x \in \mathrm{Ord} \mid x{\restriction}T = \pi\}|/|\mathrm{Ord}|$. Thus

$$\begin{aligned}
\tilde{\mathbb{E}}[N] &= \frac{1}{|\mathrm{Ord}|}\left(\sum_{[\![T]\!]_\pi \in N} |\{x \in \mathrm{Ord} \mid x{\restriction}T = \pi\}|\right) - 1 \\
&= \frac{1}{|\mathrm{Ord}|}\left(\sum_{x \in \mathrm{Ord}} (|\{[\![T]\!]_\pi \in N \mid x{\restriction}T = \pi\}|\right) - \frac{|\mathrm{Ord}|}{|\mathrm{Ord}|} \\
&= \frac{1}{|\mathrm{Ord}|}\left(\sum_{i=1}^{\infty} i|S_i|\right) - \frac{1}{|\mathrm{Ord}|}\left(|R| + \sum_{i=1}^{\infty}|S_i|\right) \\
&= \frac{1}{|\mathrm{Ord}|}\left(\sum_{i=2}^{\infty}(i-1)|S_i| - |R|\right).
\end{aligned}$$

◀

The proof of Theorem 1 uses the second point of this lemma: we show that if the degree of a normalized weakening is low enough, then a large number of orders must be accepted by several terms of $W\mathcal{J}$.

4.1.1 A Warmup

We begin by considering the case when $\mathcal{J} = 1$. As a warmup, we show a simplified (but not sufficient) argument, where we only focus on orders accepted by at least two terms. Later we present the stronger argument considering orders accepted by at least 3 terms which is sufficient to prove our theorem. We do this for the general case of arbitrary conical juntas in Section 4.1.3.

We use the notation $1z$ to denote a total order on $[n]$ where the element 1 is first and z is a total order of the remaining $n-1$ elements. The notation $i1z$ is similarly defined: it denotes orders where i is first, 1 is second, and z is a total order of the remaining $n-2$ elements.

▶ **Lemma 18.** *For a normalized DNF,*

1. *Total orders starting with 1 can only be accepted by terms $[\![T]\!]_\pi$ with $\pi(1) = 1$.*
2. *Total orders with first element i and second element 1 can only be accepted by terms $[\![T]\!]_\pi$ with either $\pi(1) = 1$ and $i \notin T$, or $\pi(1) = i$ and $\pi(2) = 1$.*

Proof. For the first point, since 1 is in the support T for each term in a normalized DNF, if $\pi(1) \neq 1$ then π is not equal to $1z{\restriction}T$ for any $1z \in \mathrm{Ord}$. For the second, consider the following two cases: If $\pi(1) = 1$ and $i \in T$, then $i1z{\restriction}T \neq \pi$ for any $i1z \in \mathrm{Ord}$. If $\pi(2) = 1$ and $\pi(1) \neq i$, then $i1z{\restriction}T \neq \pi$. Otherwise, if $\pi^{-1}(1) > 2$, then $i1z{\restriction}T \neq \pi$. ◀

The following definitions help to describe our combinatorial argument. To simplify notation, we denote $z\backslash\{i\}$ by $z\backslash i$, and $[n]\backslash\{1\}$ by $[n]\backslash 1$.

▶ **Definition 19.** *Let z be an order on $[n]\backslash 1$. We say that a term $[\![T]\!]_\pi$ is z-*good *for i if $i \in T$, $1 \in T$, $\pi(1) = i, \pi(2) = 1$, and $i1(z\backslash i){\restriction}T = \pi$. That is, the order π on T places i first, 1 second, and is consistent with the total order $i1(z\backslash i)$.*

Notice that if a term $[\![T]\!]_\pi$ is z-good for i then it accepts the order $i1(z\backslash i)$. Moreover, it also accepts the orders $ij1(z\backslash\{i,j\})$ and $ji1(z\backslash\{i,j\})$ for any j that is not in its support.

▶ **Definition 20.** *Let z be an order on $[n]\backslash 1$. We call an unordered pair of distinct elements $(i,j) \in ([n]\backslash 1)^2$ a* hitting pair *for z if the following conditions hold:*

- *N has a term $[\![T_1]\!]_{\pi_1}$ such that $\pi_1(1) = i, \pi_1(2) = 1, j \notin T_1$ and $i1(z\backslash i)\restriction T_1 = \pi_1$.*
- *N has a term $[\![T_2]\!]_{\pi_2}$ such that $\pi_2(1) = j, \pi_2(2) = 1, i \notin T_2$ and $j1(z\backslash j)\restriction T_2 = \pi_2$.*

In other words, a pair (i,j) is a hitting pair for z, if there is some term that is z-good for i but its support does not contain j, and there is some term that is z-good for j but its support does not contain i.

Notice that if (i,j) is a hitting pair for z, then *both* $ij1(z\backslash\{i,j\})$ and $ji1(z\backslash\{i,j\})$ are accepted by *both* $[\![T_1]\!]_{\pi_1}$ and $[\![T_2]\!]_{\pi_2}$. Therefore, we can get a lower bound on the number of orders accepted by several terms of N by getting a lower bound on the number of hitting pairs.

▶ **Lemma 21.** *Let N be a normalized DNF with support size k that is a weakening of M_1. Let $R \subseteq$ Ord be the set of orders rejected by N. If $k \leq n/10$, there are at least $4|R|/5$ total orders accepted by more than one term of N.*

Proof. If $R = \emptyset$, we are done, so suppose otherwise. Recall that we assume for normalized DNFs that each term contains the element 1, so we do not explicitly state this condition.

Given an order z on $[n]\backslash 1$, we define a matrix H_z as follows. Let H_z be an $(n-1)\times(n-1)$ matrix with rows and columns indexed by $i,j \in [n]\setminus 1$, respectively, where $H_z(i,j) = 1$ iff $i \neq j$, and N has a term $[\![T]\!]_\pi$ such that $\pi(1) = i$, $\pi(2) = 1$, $j \notin T$, and $i1(z\backslash i)\restriction T = \pi$. That is, the (i,j)th entry is 1 iff there is a term that is z-good for i but its support does not contain j. Thus, (i,j) is a hitting pair iff both $H_z(i,j) = 1$ and $H_z(j,i) = 1$.

Let $1z \in R$ be an order rejected by N. Note that since N is a weakening of M_1, every order it rejects must start with 1. Moreover, for every $i \in [n]\backslash 1$, N must accept the total order $i1(z\backslash i)$, as it is accepted by M_1.

If N had any terms $[\![T]\!]_\pi$ with $\pi(1) = 1$ and $1z\restriction T = \pi$, $1z$ would not be rejected. Thus, by point 2 of Lemma 18, for each $i \in [n]\backslash 1$, the order $i1(z\backslash i)$ must be accepted by a term $[\![T]\!]_\pi$ of N with $\pi(1) = i, \pi(2) = 1$.

That is, for each $i \in [n]\backslash 1$, and any z such that $1z$ is rejected by N, there is a term of N that is z-good for i. Since N is normalized, the support of each term has size k, and so row i of H_z must have at least $n-k$ entries with value 1 (there are $n-k$ elements j not included in the support of any particular term). The matrix thus contains at least $(n-1)(n-k)$ entries which are 1.

Next, observe that the largest number of 1-entries in a matrix without a hitting pair is at most $\binom{n-1}{2}$: by definition, any such matrix has a 1 in at most one of the entries $H_z(i,j)$ or $H_z(j,i)$, for each unordered pair (i,j). Moreover, the following holds.

▶ **Observation 22.** *Any $(n-1)\times(n-1)$ $\{0,1\}$-matrix with $q > \binom{n-1}{2}$ 1-entries outside of the diagonal must contain at least $q - \binom{n-1}{2}$ hitting pairs.*

Proof. Let H be an arbitrary $(n-1)\times(n-1)$ $\{0,1\}$-matrix with $q > \binom{n-1}{2}$ 1-entries, none of which are on the diagonal. Since the number of 1-entries is more than $\binom{n-1}{2}$, H must contain at least one hitting pair (i,j). Replace one of the entries corresponding to this pair (either the entry $H(i,j)$ or the entry $H(j,i)$) by 0. We can repeat this step at least $q - \binom{n-1}{2}$ times, accounting for at least $q - \binom{n-1}{2}$ hitting pairs. ◀

Thus, in the matrix H_z there are at least $(n-1)(n-k) - \binom{n-1}{2}$ hitting pairs for z. This holds for each z such that $1z \in R$.

Each of the hitting pairs in H_z causes (at least) two orders to be accepted twice, in particular, as discussed above, if (i,j) is a hitting pair for z, then *both* $ij1(z\backslash\{i,j\})$ and $ji1(z\backslash\{i,j\})$ are accepted twice. Notice that the total orders we count twice this way due to (i,j) are the same for z and z' iff $z\backslash\{i,j\} = z'\backslash\{i,j\}$. For any given z, the number of z' such that $z\backslash\{i,j\} = z'\backslash\{i,j\}$ is at most $(n-1)(n-2)$.

Thus, we can conclude that there must be at least

$$\frac{2\left((n-1)(n-k) - \binom{n-1}{2}\right)}{(n-1)(n-2)}|R|$$

orders accepted by at least two terms of N. This value is always smaller than $|R|$. However, if $k \leq n/10$, it is at least $4|R|/5$. ◀

4.1.2 Normalization with Respect to Juntas

The above argument shows that focusing on hitting pairs is almost enough to conclude that $\tilde{\mathbb{E}}[W] \geq 0$ for weakenings W, even if they have a fairly high degree. Our final proof for the general case is similar to this argument, but focuses on hitting triples. Before stating and proving this result, we need to deal with conical juntas. We extend the normalization done in Lemma 15 to handle the additional inclusion of a conical junta. This is captured by the following lemmas.

▶ **Lemma 23.** *Let the DNF W be a weakening of M_1 and $\mathcal{J}$ be a conical junta. If $\tilde{\mathbb{E}}[W\mathcal{J}] < 0$ there exists N and t such that*

1. *N is the normalized form of W given by Lemma 15,*
2. *$t = [\![T]\!]_\pi$ such that $1 \in T$, $\pi(1) = 1$, and the support size of t is $\leq 2\deg(\mathcal{J}) + 1$*
3. *$\deg(Nt) \leq 2\deg(W) + 2\deg(\mathcal{J})$,*
4. *$\tilde{\mathbb{E}}[Nt] < 0$.*

Proof. Let $\mathcal{J} = \sum_{t\in\mathcal{J}} t$ and suppose that $0 > \tilde{\mathbb{E}}[W\mathcal{J}] = \sum_{t\in\mathcal{J}} \tilde{\mathbb{E}}[Wt]$, where the equality follows by linearity. As the sum is negative, there must be at least one term t^* such that $\tilde{\mathbb{E}}[Wt^*] < 0$. Let T be the support of t^*. Note that without loss of generality we can assume that 1 belongs to T: If $1 \notin T$ then let Π be the set of permutations on $T \cup \{1\}$ consistent with t^*. As shown in the proof of Lemma 15, t^* is equivalent to $\bigvee_{\pi\in\Pi}[\![T\cup\{1\}]\!]_\pi$, and $\tilde{\mathbb{E}}[t^*] = \tilde{\mathbb{E}}\left[\sum_{\pi\in\Pi}[\![T\cup\{1\}]\!]_\pi\right]$. In particular, every order accepted by t^* is accepted by $[\![T\cup\{1\}]\!]_\pi$ for exactly one $\pi \in \Pi$. Thus, $\tilde{\mathbb{E}}[Wt^*] = \sum_{\pi\in\Pi}\tilde{\mathbb{E}}[W[\![T\cup\{1\}]\!]_\pi]$, and if $\tilde{\mathbb{E}}[Wt^*] < 0$ then there must exist $\pi \in \Pi$ such that $\tilde{\mathbb{E}}[W[\![T\cup\{1\}]\!]_\pi] < 0$.

Suppose that $\pi(1) \neq 1$, then $W \wedge [\![T]\!]_\pi$ does not accept any order that starts with 1, since $[\![T]\!]_\pi$ does not. As W is a weakening of M_1, it must accept every order that does not start with 1. Thus, $W \wedge [\![T]\!]_\pi$ is equivalent to $[\![T]\!]_\pi$ and $\sum_{w\in W}\tilde{\mathbb{E}}[w[\![T]\!]_\pi] \geq \tilde{\mathbb{E}}[[\![T]\!]_\pi]$, as every order accepted by $[\![T]\!]_\pi$ must be accepted by at least one term w of W. This implies that

$$\tilde{\mathbb{E}}\big[W[\![T]\!]_\pi\big] = \tilde{\mathbb{E}}\left[\Big(\sum_{w\in W} w - 1\Big)[\![T]\!]_\pi\right] = \sum_{w\in W}\tilde{\mathbb{E}}[w[\![T]\!]_\pi] - \tilde{\mathbb{E}}[[\![T]\!]_\pi] \geq \tilde{\mathbb{E}}[[\![T]\!]_\pi] - \tilde{\mathbb{E}}[[\![T]\!]_\pi] = 0.$$

Combining this with the previous argument, we get that if there is some conical junta such that $\tilde{\mathbb{E}}[W\mathcal{J}] < 0$ then there is also a term $t = [\![T]\!]_\pi$, with $1 \in T$ and $\pi(1) = 1$, such that $\tilde{\mathbb{E}}[Wt] < 0$. Note that the support size of $t = [\![T]\!]_\pi$ is at most $2\deg(\mathcal{J}) + 1$, and its degree is at most $2\deg(\mathcal{J})$.

Next, we apply Lemma 15 to the DNF W to obtain a normalized DNF N. Recall that the degree of N will be at most $2\deg(W)$. Since both the degree of N and t might have at most doubled, and $\deg(Nt) \leq \deg(N) + \deg(t)$, we get that $\deg(Nt) \leq 2\deg(W) + 2\deg(\mathcal{J})$, concluding (3).

We now argue that $\tilde{\mathbb{E}}[Wt] < 0$ implies that $\tilde{\mathbb{E}}[Nt] < 0$. Expanding Nt, we have

$$\tilde{\mathbb{E}}[Nt] = \tilde{\mathbb{E}}\Big[\Big(\sum_{v\in N} v - 1\Big)t\Big] = \sum_{v\in N} \tilde{\mathbb{E}}[vt] - \tilde{\mathbb{E}}[t].$$

Since N is the normalization of W, the terms of N can be partitioned into groups, one for each term of W, such that the term of W accepts a given order iff a unique term in its corresponding group accepts that order, and the term of W rejects a given order iff all terms in the corresponding group reject it. Thus, $\sum_{v\in N} \tilde{\mathbb{E}}[vt] = \sum_{w\in W} \tilde{\mathbb{E}}[wt]$, and so $\tilde{\mathbb{E}}[Nt] = \tilde{\mathbb{E}}[Wt] < 0$. ◀

Note that in the above lemma, both N and t are normalized, but the DNF corresponding to $N \wedge t$ (obtained by taking the conjunction of each term of N with t) is not. We address this issue next.

▶ **Lemma 24.** *Let $t = [\![T]\!]_{\pi^*}$ be a term of support size ℓ such that $1 \in T$ and $\pi^*(1) = 1$, and let N be a normalized DNF of support size h that is a weakening of M_1. Then there is a DNF N' such that*

1. *For every total order x, $N'(x) \Leftrightarrow N(x) \wedge t(x)$. Hence, N' is a weakening of $M_1 \wedge t$,*
2. *N' is normalized with support size at most $\ell + h - 1$, and degree at most $\ell + h - 2 \leq \deg(N) + \deg(t)$,*
3. *$\tilde{\mathbb{E}}[Nt] = \tilde{\mathbb{E}}[N' + 1 - t]$.*

Coming back to our analogy with the covering problem presented in the introduction, proving that $\tilde{\mathbb{E}}[N' + 1 - t]$ is non-negative for all such N' of small degree (for a fixed t), is equivalent to saying that, even if we restrict the sets Ord and Ord^{*1} to their orders consistent with t, it is impossible to get a covering of $(\mathrm{Ord} \wedge t)^{*1}$ with fewer sets than the number needed to cover $\mathrm{Ord} \wedge t$, if the support size of the sets is small.

Proof of Lemma 24. We construct N' by "merging" terms of N and t. Formally, let $[\![S]\!]_\sigma$ be a term of N, and let $\Pi_{S,\sigma}$ be the set of permutations on $S \cup T$ consistent with both σ and π^*; note that this set might be empty. Define $N' := \bigvee_{[\![S]\!]_\sigma \in N} \bigvee_{\pi\in\Pi_{S,\sigma}} [\![S \cup T]\!]_\pi$ and observe that this DNF accepts the same total orders Nt does. Indeed, an order is consistent with a term $[\![S]\!]_\sigma$ of N and t iff there is an ordering in $\Pi_{S,\sigma}$ consistent with it. Every term in N' is normalized, but since for different terms $[\![S]\!]_\sigma, [\![S']\!]_\sigma$ of N, the sizes of $S \cup T$ and $S' \cup T$ might be different, terms in N' might be of different sizes. As in the previous arguments, one can simply add elements to the support of smaller terms to get them to a fixed size. The maximal support size of a term in N' is the maximal size of $S \cup T$, where S is the support of some term in N. Since N and t are both normalized, and since they both contain 1, this value is at most $\ell + h - 1$. This also implies that the degree $\deg(N') \leq \ell + h - 2 \leq \deg(N) + \deg(t)$.

It remains to prove that $\tilde{\mathbb{E}}[Nt] = \tilde{\mathbb{E}}[N' + 1 - t]$. For a single term $[\![S]\!]_\sigma$ of N it holds that $\tilde{\mathbb{E}}[[\![S]\!]_\sigma t] = \tilde{\mathbb{E}}\left[\sum_{\pi\in\Pi_{S,\sigma}} [\![S \cup T]\!]_\pi\right]$, as every total order accepted by $[\![S]\!]_\sigma t$ is consistent with a unique $\pi \in \Pi_{S,\sigma}$. Thus,

$$\begin{aligned}
\tilde{\mathbb{E}}[Nt] &= \tilde{\mathbb{E}}\Big[\sum_{[\![S]\!]_\sigma\in N} [\![S]\!]_\sigma t - t\Big] = \sum_{[\![S]\!]_\sigma\in N} \tilde{\mathbb{E}}\big[[\![S]\!]_\sigma t\big] - \tilde{\mathbb{E}}[t] \\
&= \sum_{[\![S]\!]_\sigma\in N} \tilde{\mathbb{E}}\Big[\sum_{\pi\in\Pi_{S,\sigma}} [\![S \cup T]\!]_\pi\Big] - \tilde{\mathbb{E}}[t] = \tilde{\mathbb{E}}\Big[\sum_{[\![S]\!]_\sigma\in N} \sum_{\pi\in\Pi_{S,\sigma}} [\![S \cup T]\!]_\pi\Big] - \tilde{\mathbb{E}}[t] \\
&= \tilde{\mathbb{E}}[N' + 1] - \tilde{\mathbb{E}}[t] = \tilde{\mathbb{E}}[N' + 1 - t].
\end{aligned}$$

◀

4.1.3 The General Case

We are now ready to state and prove our main theorem about weakenings of M_1.

▶ **Theorem 25.** *Let $t = [\![T^*]\!]_{\pi^*}$ be a term of support size ℓ such that $\pi^*(1) = 1$, and let N be a normalized DNF of support size h that is a weakening of M_1. If $\ell + h \leq n/100$, then $\tilde{\mathbb{E}}[Nt] \geq 0$.*

In Section 4.1.1 we presented a warmup argument for the case when the conical junta, $\mathcal{J}$, was assumed to be the trivial junta $\mathcal{J} = 1$. This simpler argument (considering only "pairs") is not sufficient to get a good bound, even for the case when $\mathcal{J} = 1$, but it illustrates our approach well. To get a good enough bound, we extend this argument to triples. To prove Theorem 25, we allow $\mathcal{J}$ to be an arbitrary conical junta. This also includes the case when $\mathcal{J}$ could be the trivial junta $\mathcal{J} = 1$, in which case we define its support to be $T^* = \{1\}$.

Proof of Theorem 25. Let N' be the weakening of $M_1 \wedge t$ given by Lemma 24; that is $\tilde{\mathbb{E}}[Nt] = \tilde{\mathbb{E}}[N' + 1 - t]$. We will prove that $\tilde{\mathbb{E}}[N' + 1 - t] \geq 0$. Let $\ell + g$ be the support size of N', and recall that this is at most $\ell + h - 1 \leq n/100 - 1$. Let R be the set of total orders that are consistent with t and are rejected by N', and hence also by $N \wedge t$. Recall also that by Lemma 24 N' only accepts orders consistent with t. We will prove that the over-counting in the pseudo-expectation that is due to orders that are accepted by several terms of N' is sufficiently large compared to the number of orders rejected by N' that are consistent with t. A sufficiently large "over-count" will be enough to show that $\tilde{\mathbb{E}}[N' + 1 - t] \geq 0$, since $\tilde{\mathbb{E}}[N' + 1 - t]$ is exactly the sum of the number of orders that each term of N' accepts minus the number of orders consistent with t, divided by $n!$. $\tilde{\mathbb{E}}[N' + 1 - t]$ is thus non-negative, if and only if the number of extra occurrences of orders accepted by more than one term of N' is at least $|R|$ (Recall that $|R|$ is the number of orders consistent with t that are not accepted by any term of N').

Given an order z on $[n] \backslash 1$, we call an unordered triple of distinct elements $(i, j, k) \in ([n] \backslash T^*)^3$ a *hitting triple for z* if the following hold:

1. N' has a term $[\![T_1]\!]_{\pi_1}$ such that $\pi_1(1) = i, \pi_1(2) = 1, j, k \notin T_1$ and $i1(z \backslash i) \restriction T_1 = \pi_1$.
2. N' has a term $[\![T_2]\!]_{\pi_2}$ such that $\pi_2(1) = j, \pi_2(2) = 1, i, k \notin T_2$ and $j1(z \backslash j) \restriction T_2 = \pi_2$.
3. N' has a term $[\![T_3]\!]_{\pi_3}$ such that $\pi_3(1) = k, \pi_3(2) = 1, i, j \notin T_3$ and $k1(z \backslash k) \restriction T_3 = \pi_3$.

In other words, (i, j, k) is a hitting triple for z, if there is some term that is z-good for i but j and k are missing from its support, there is some term that is z-good for j but i and k are missing from its support, and there is some term that is z-good for k but i and j are missing from its support. Analogous to hitting pairs, the presence of hitting triples implies that there are orders accepted by several terms. Indeed, if (i, j, k) is a hitting triple for z, then for all six permutations σ of $\{i, j, k\}$, the order $\sigma(1)\sigma(2)\sigma(3)1(z \setminus \{i, j, k\})$ is accepted by at least three terms.

Given an order z on $[n] \backslash 1$, define an array $H_z : ([n] \backslash T^*)^3 \to \{0, 1\}$, such that $H_z(i, j, k) = 1$ iff i, j, k are distinct and not in T^*, and N' has a term that is z-good for i but j and k are missing from its support (that is N' has a term $[\![S]\!]_\pi$ with $\pi(1) = i, \pi(2) = 1$, $j, k \notin S$ and $i1(z \backslash i) \restriction S = \pi$). Note that by this definition, $H_z(i, j, k) = H_z(i, k, j)$. Furthermore, notice that (i, j, k) is a hitting triple for z iff for all six permutations π of $\{i, j, k\}$, $H_z(\pi(1), \pi(2), \pi(3)) = 1$. In other words, 1-entries now come in pairs in the array, and one needs six 1-entries to get a hitting-triple. Thus, there can be at most $4\binom{n-\ell}{3}$ 1-entries in an array H_z without any hitting triples. The $-\ell$ comes from the fact that we do not consider elements of T^* in the triples. Similarly to Observation 22, we get the following.

▶ **Observation 26.** *Let H be an $(n-\ell)\times(n-\ell)\times(n-\ell)$ {0,1}-array with $q > 4\binom{n-\ell}{3}$ 1-entries, where all 1 entries are in positions with distinct coordinates i,j,k, such that $H(i,j,k) = H(i,k,j)$ for all i,j,k. Then H must contain at least $\frac{1}{2}\left(q - 4\binom{n-\ell}{3}\right)$ hitting triples.*

Proof. Let H be an $(n-\ell)\times(n-\ell)\times(n-\ell)$ {0,1}-array with $q > 4\binom{n-\ell}{3}$ 1-entries, where all 1-entries are in positions with distinct coordinates i,j,k, such that $H(i,j,k) = H(i,k,j)$ for all i,j,k. Since the 1-entries come in pairs, we have $q \geq 4\binom{n-\ell}{3} + 2$ entries that are 1, which implies that for at least one triple with distinct i,j,k, all 6 corresponding entries must be 1. Thus H contains at least one hitting triple. Replace one pair of entries corresponding to this triple, say $H(i,j,k)$ and $H(i,k,j)$, by 0. We can repeat this step at least $\frac{1}{2}\left(q - 4\binom{n-\ell}{3}\right)$ times, accounting for at least $\frac{1}{2}\left(q - 4\binom{n-\ell}{3}\right)$ hitting triples. ◀

Let $1z \in R$ be a rejected order that is consistent with t. By a similar argument to the proof of Lemma 21, for every $i \in [n]\backslash T^*$, there is a term of N' that is z-good for i (and thus this term accepts the order $i1(z\backslash i)$). This only holds for i that are not in T^* (the support of t), since N' only accepts total orders that are accepted by t, that is only total orders that are consistent with π^* on T^*, and moving any element in T^* before 1 would contradict π^*.

Since for all $i \notin T^*$, the order $i1(z\backslash i)$ is accepted by a term of support size $\ell + g$, with $\pi(1) = i, \pi(2) = 1$, there must be at least $(n-\ell-g)(n-\ell-g-1)$ 1-entries in H_z at positions starting with i. Hence, there are at least $\frac{1}{2}\left((n-\ell)(n-\ell-g)(n-\ell-g-1) - 4\binom{n-\ell}{3}\right)$ hitting triples for z. Each hitting triple gives (at least) six orders accepted by (at least) three terms: specifically, as noted earlier, for all six permutations σ of $\{i,j,k\}$, the order $\sigma(1)\sigma(2)\sigma(3)1(z\backslash\{i,j,k\})$ is accepted by at least three terms. Thus each hitting triple gives (at least) 12 total extra occurences of over-counted orders. By a similar reasoning to the proof of Lemma 21, each of these specific orders could come from hitting triples of at most $(n-1)(n-2)(n-3)$ different z-s, since for any given order z on $[n]\backslash 1$ consistent with t, the number of orders z' on $[n]\backslash 1$ consistent with t such that $z\backslash\{i,j,k\} = z'\backslash\{i,j,k\}$ is at most $(n-1)(n-2)(n-3)$.

Thus, we get that there are at least

$$\frac{6\left((n-\ell)(n-\ell-g)(n-\ell-g-1) - 4\binom{n-\ell}{3}\right)}{(n-1)(n-2)(n-3)}|R|$$

total extra occurrences of over-counted orders.

Plugging in $\ell + g + 1 \leq n/100$, we get

$$\begin{aligned}
&\frac{6\left((n-\ell)(n-\ell-g)(n-\ell-g-1) - 4\binom{n-\ell}{3}\right)}{(n-1)(n-2)(n-3)}|R| \\
&= \left(\frac{6(n-\ell)(n-\ell-g)(n-\ell-g-1)}{(n-1)(n-2)(n-3)} - \frac{24\binom{n-\ell}{3}}{(n-1)(n-2)(n-3)}\right)|R| \\
&\geq \left(\frac{6\left((99n/100)^3\right)}{n^3} - \frac{4(n-\ell)(n-\ell-1)(n-\ell-2)}{(n-1)(n-2)(n-3)}\right)|R| \\
&\geq \left(\frac{6\left((99n/100)^3\right)}{n^3} - 4\right)|R| = \left(6\frac{99^3}{100^3} - 4\right)|R| \geq |R|
\end{aligned}$$ ◀

We can now conclude the proof of the main theorem of this section.

Proof of Theorem 12. As we argued above, it suffices to consider weakenings of M_1. By Lemma 23, if $\tilde{\mathbb{E}}[W\mathcal{J}] < 0$ for a weakening W of M_1 and conical junta $\mathcal{J}$, both of degree at most d, then there is a normalized weakening N of support size $h \leq 2d+1$ and a normalized term t of support size $\ell \leq 2d+1$ such that $\tilde{\mathbb{E}}[Nt] < 0$. Notice that if $d \leq (\frac{n}{400} - 1)$, then $\ell + h \leq 4d + 2 \leq n/100$. By Theorem 25, this cannot happen. Thus, there is a Σ_2-pseudo-expectation for LOP of degree $(\frac{n}{400} - 1)$. ◀

Note that the $n/100$ bound was picked for convenience, one could get a better bound by counting more than the hitting triples, and by being more careful in the computation.

5 A Criterion for Non-Reducibility of LOP

In this section, we show that our pseudo-expectation for LOP (Lemma 11) implies a general criterion for the non-reducibility of LOP to other problems in $\mathsf{TF}\Sigma_2^{dt}$.

▶ **Corollary 27.** *Suppose that $R \in \mathsf{TF}\Sigma_2^{dt}$ and R has a $\mathsf{polylog}(n)$-degree uSA-refutation. Then* LOP $\not\leq_{dt} R$.

Using this, we show that LOP $\notin \mathsf{LeastNumber}^{dt}$.

Proof of Theorem 2. The axioms of the formula of $\textsc{LeastNumber}_n$ are:

- $\bigvee_{i\in[n]} x_i$, giving the inequality $\sum_{i\in[n]} x_i - 1 \geq 0$; *(Not all 0)*
- $\overline{x_i} \vee \bigvee_{j<i} x_j$, giving the inequality $(1-x_i) + \sum_{j<i} x_j - 1 = -x_i + \sum_{j<i} x_j \geq 0$ for each $i \in [n]$. *(i is the least index taking value 1)*

This admits the following simple uSA-refutation,

$$\sum_{i=1}^{n} x_i - 1 + \sum_{i=0}^{n-1} 2^i \Big(- x_{n-i} + \sum_{j<n-i} x_j \Big) = -1 + \sum_{i=1}^{n} \Big(1 + \sum_{j=0}^{n-i-1} 2^j - 2^{n-i} \Big) x_i = -1.$$

Although, the refutation is of exponential size, it has constant degree. By Corollary 27, LOP $\notin \mathsf{LeastNumber}^{dt}$. ◀

We now prove Corollary 27. To do so, in Lemma 32 we show that we can factorize any reduction between $\mathsf{TF}\Sigma_2^{dt}$ problems into two parts (i) a weakening step, akin to Σ_2-weakening, and (ii) a *counter-example reduction* [16,23] which takes place in TFNP^{dt}. This elucidates how the two types of reductions between $\mathsf{TF}\Sigma_2^{dt}$ problems that have been considered (formulations and counter-example reductions) relate. Using this, our proof proceeds as follows:

1. Consider an R-formulation of LOP where R admits a low-degree uSA-refutation.
2. Factorize the R-formulation into two parts using Lemma 32: a weakening from LOP to a problem P, and a counter-example reduction from P to R.
3. Argue that we can derive an uSA-refutation of P from an uSA-refutation of R while roughly maintaining the degree (Proposition 34).

5.1 Factoring Reductions

To improve readability, we will use a slightly different notation for $\mathsf{TF}\Sigma_2^{dt}$ problems in this section.

▶ **Notation 28.** *A $\mathsf{TF}\Sigma_2^{dt}$ search problem R is a sequence of search problems $R_m \subseteq \{0,1\}^m \times \mathcal{O}_m^R \times \mathcal{W}_m^R$ for each $m \in \mathbb{N}$. $\mathcal{O}_m^R$ is the set of possible R_m-outputs and $\mathcal{W}_m^R$ is the set of R_m-witnesses. We think of $b \in \mathcal{O}_m^R$ as being a solution to $a \in \{0,1\}^m$ for R_m if $R_m(a,b;c)$ holds for all $c \in \mathcal{W}_m^R$.*

A weaker notion of reducibility between $\mathsf{TF}\Sigma_2$ problems has been considered, known as *counter-example reducibility* [23]. Such a reduction from a search problem Q to a search problem R can be interpreted as the following game: Suppose Alice wants to solve Q on input x. She heard that Bob is actually able to solve R using some solver S, and thus, using the reduction, she sends $f(x)$ to Bob. In turns, Bob uses his solver $b := S\big(f(x)\big)$ and Alice transforms b into the output $y := g(x, b)$. Now, if y is indeed an output for x, Alice got what she wanted. But let us say that Alice does not trust the solution and, after some searching, is able to find z such that $Q(x, y; z)$ does not hold (and hence y is not a correct output for x). She confronts Bob about the issue, claiming that Bob's solver is not correct. Bob, who is very proud of his solver, dismisses her by asking for a proof. Then by computing $c := h(x, b, z)$, Alice is capable of finding a counter-example to b being a correct output for $f(x)$, and hence proving that S is incorrect.

▶ **Definition 29.** *Let Q and R be two $\mathsf{TF}\Sigma_2^{dt}$ problems. A* counter-example reduction from Q to R *or a* counter-example R-formulation of Q *is an R-formulation (f, g) along with, for each $b \in \mathcal{O}^R_{s(n)}$ and $z \in \mathcal{W}^Q_n$, a function $h_{b,z} : \{0,1\}^n \to \mathcal{W}^R_{s(n)}$ such that*

$$R_{s(n)}\big(f(x), b; h_{b,z}(x)\big) \implies Q_n\big(x, g_b(x); z\big)$$

where $s(n)$ is the size of the reduction. The depth, and complexity of the formulation are defined as in Definition 5.

By abuse of notation, an R-formulation of a problem Q will also be said to be a counter-example reduction if there exists a function h such that the triplet (f, g, h) is a counter-example reduction. We also define another type of reduction that is the translation of the Σ_2-weakenings.

▶ **Definition 30.** *Let Q and R be two $\mathsf{TF}\Sigma_2^{dt}$ search problems and let (f, g) be an R-formulation of Q. We say that (f, g) is a* weakening *if*

- *f is the identity,*
- *for each b, g_b is a constant function.*

In particular, weakenings are formulations that are always of size n and depth 1.

▶ **Proposition 31.** *Let Q and R be two $\mathsf{TF}\Sigma_2^{dt}$ search problems. Then the following are equivalent:*

- *There exists a weakening from Q to R,*
- *F_{R_n} is a Σ_2-weakening of F_{Q_n} for every n.*

For a decision tree $T(x)$ whose leaves are labeled with 0 and 1, we will abuse notation and also denote by $T(x)$ the DNF obtained from taking the OR over the terms corresponding to its accepting paths. We also write $\overline{T}(x)$ for the function $1 - T(x)$. Its decision tree is the same as T but with inverted labels and its DNF is obtained from taking the OR over the terms corresponding to the rejecting paths of T.

Proof of Proposition 31. Writing $s := s(n)$, say we have a weakening R-formulation (f, g) of Q. Then for each $b \in \mathcal{O}^R_s$ the axiom $\bigvee_c \overline{R}_n(x, b; c)$ is a Σ_2-weakening of $\bigvee_z \overline{Q}_n(x, g_b; z)$ by the correctness of the reduction, where c ranges over $\mathcal{W}^R_s$ and z over $\mathcal{W}^Q_n$. If F_{R_n} is a Σ_2-weakening of F_{Q_n}, then set $g_b = y_b$ such that $\bigvee_c \overline{R}_n(x, b; c)$ is a weakening of $\bigvee_z \overline{Q}_n(x, y_b, z)$. ◀

Considering only weakenings and counter-example reductions seems very restrictive, as the correctness of counter-example reductions is efficiently verifiable (since they actually occur at the TFNP level as highlighted in [23]) and weakenings are somewhat trivial. Surprisingly enough, they seem to capture reductions in their entirety in the sense of the following lemma.

▶ **Lemma 32.** *Let Q and R be two $\mathsf{TF}\Sigma_2^{dt}$ problems and let (f, g) be an R-formulation of Q. Then there exists $P \in \mathsf{TF}\Sigma_2^{dt}$ such that (f, g) decomposes into two reductions $Q \to P \to R$ and:*

- *the reduction $Q \to P$ is a weakening,*
- *the reduction $P \to R$ is a counter-example reduction.*

This decomposition lemma mirrors the characterization given by Theorem 9: One part corresponds to taking a weakening of a formula while the other has a validity that can be efficiently verified (on one hand uSA-refutations and on the other counter-example reductions – See more about this in Section 6). In order to prove this lemma, we first need to introduce the *reduced problem* associated with a reduction, as it will be our choice for P.

For a search problem $R_m(a, b; c)$, we write $R_{m,b,c}(a)$ for the decision tree computing $R_m(a, b; c)$ to put an emphasis on the fact that its only variable is a.

▶ **Definition 33.** *Let Q and R be two $\mathsf{TF}\Sigma_2^{dt}$ search problems. For (f, g) an R-formulation of Q of size $s := s(n)$, the* reduced problem *$R(f, g)$ is the sequence of ternary relations $R(f, g)_n \subseteq \{0,1\}^n \times (\mathcal{O}_n^Q \times \mathcal{O}_s^R) \times \mathcal{W}_s^R$ given by*

$$R(f,g)_n\big(x,(y,b);c\big) \iff g_b(x) = y \text{ and } R_s\big(f(x),b;c\big).$$

The formula associated to the reduced problem $F_{R(f,g)_n}$ has axioms

$$\overline{G}_{b,y}(x) \vee \bigvee_c \overline{R}_{s,b,c} \circ f(x)$$

and $G_{b,y}(x)$ is the indicator function for the event $g_b(x) = y$. We see $G_{b,y}(x)$ as the decision tree (and hence can also be thought of as a DNF) obtained from the decision tree computing g_b and replacing the labels of its leaf with 1 if the leaf is labeled with y and 0 otherwise.

The formula associated to the reduced problem $R(f, g)$ states "the reduction given by (f, g) is not correct", and for the tuple (b, y), the meaning of the corresponding axiom is "either $g_b(x) \neq y$, or there is a counter-example c to b being an R-output for $f(x)$". The formula encodes the following procedure: Say we are given a Q-input x and a purported output y. To verify that y is a Q-output for x, instead of computing $Q(x, y; z)$ for each z, one can do the following: compute $f(x)$ and, given b an R-output for $f(x)$, compute $R(f(x), b; c)$ for each c and verify that $g_b(x) = y$.

Proof of Lemma 32. Let Q, R, and (f, g) be as in the statement of the lemma, set P to be $R(f, g)$ and denote by $s := s(n)$ the size of the reduction. The formula $F_{R(f,g)_n}$ has an axiom $\overline{G}_{b,y}(x) \vee \bigvee_c \overline{R}_{s,b,c} \circ f(x)$ for every pair y, b, where the variable c ranges over $\mathcal{W}_s^R$, and F_{Q_n} has an axiom $\bigvee_z \overline{Q}_{n,y,z}(x)$ for each y where the variable z ranges over $\mathcal{W}_n^Q$. By the correctness of the reduction (if $y = g_b(x)$ and b is an R-output for $f(x)$, then y is a Q-output for x), we get that the formula

$$\bigvee_z \overline{Q}_{n,y,z}(x) \implies \overline{G}_{b,y}(x) \vee \bigvee_c \overline{R}_{s,b,c} \circ f(x)$$

is a tautology and $F_{R(f,g)_n}$ is indeed a Σ_2-weakening of F_{Q_n}.

Now, setting $f'(x) = f(x)$ for all x, $g'_b(x) = \big(g_b(x), b\big)$ for all b, and $h'_{b,c}(x) = c$ for each pair b, c, the triplet (f', g', h') forms a counter-example R-formulation of $R(f, g)$ since for every x, y, b, and c we get

$$R_s\big(f'(x), b; h'_{b,c}(x)\big) = R_{s,b,c} \circ f(x) = R(f,g)_n\big(x,(b,g_b(x));c\big) = R(f,g)_n\big(x,g'_b(x);c\big). \quad \blacktriangleleft$$

5.2 Proving Corollary 27

▶ **Proposition 34.** *Let Q and R be two $\mathsf{TF}\Sigma_2^{dt}$ search problems and let (f, g) be an R-formulation of Q of size $s(n)$ and depth $d(n)$. If F_{R_m} admits an uSA-refutation of degree $d'(m)$, then $F_{R(f,g)_n}$ admits one of degree $d(n)(d'(s(n)) + 2)$.*

In the following proof, for a polynomial $p(a) = p(a_1, \dots, a_m)$ over m variables and a collection of m decision trees $f(x) = (f_i(x))_{i\in[m]}$, the polynomial $p \circ f(x)$ is $p\big(f_1(x), \dots, f_m(x)\big)$ where each f_i is seen as the sum of the monomials representing its accepting paths. Note that when p is a junta, the resulting polynomial can also be represented by a junta. Indeed, for a term $t(a)$, one can compute $t \circ f(x)$ by querying f_i if the variable a_i appears in t. This gives a decision tree that can in turn be transformed into a conical junta. The degree of $p \circ f(x)$ is equal to $\deg(p) \cdot \mathsf{depth}(f)$. We also define $T \circ f(x)$ a similar way when T is a decision tree and its depth is equal to $\mathsf{depth}(T) \cdot \mathsf{depth}(f)$.

Proof. Writing $s := s(n)$, let us first recall the axioms (in their polynomial inequality form) of the two formulas.

- Axioms of F_{R_s}: $\sum_c \overline{R}_{s,b,c}(a) - 1 \geq 0$ for each $b \in \mathcal{O}_s^R$ with c ranging over $\mathcal{W}_s^R$;
- Axioms of $F_{R(f,g)_n}$: $\overline{G}_{b,y}(x) + \sum_c \overline{R}_{s,b,c} \circ f(x) - 1 \geq 0$ for each pair $b \in \mathcal{O}_s^R$ and $y \in \mathcal{O}_n^Q$ with c ranging over $\mathcal{W}_s^R$. Recall that each axiom encodes the fact that either $g_b(x) \neq y$ or c witnesses that b is not an R-output of $f(x)$ and their conjunction contradicts the correctness of the reduction.

As the axioms of $F_{R(f,g)_n}$ closely resemble the ones of F_{R_s}, our goal is to transform the uSA-refutation of the later formula to one of the former. The idea is pretty simple: compose every polynomial appearing in the refutation of R_s with the function f and the function computed by the sum should still be -1. Let $\sum_b J_b(a)(\sum_c \overline{R}_{s,b,c}(a) - 1) + J(a) = -1$ be an uSA-refutation of R_s. We explain how to deduce an uSA-refutation step by step.

1. For each axiom $\overline{G}_{b,y}(x) + \sum_c \overline{R}_{s,b,c} \circ f(x) - 1$, multiply by $G_{b,y}(x)$. Since $T(x) \cdot \overline{T}(x) = 0$ for any decision tree, we get $G_{b,y}(x) \cdot (\sum_c \overline{R}_{s,b,c} \circ f(x) - 1)$ for each tuple (b, y).
2. Using the fact that $\sum_y G_{b,y} = 1$ (as we sum over the indicator variables of all paths of the decision tree g_b, and every assignment must be consistent with at least one of those paths), summing over y and multiplying by $J_b \circ f$ gives us
$$J_b \circ f(x)\Big(\sum_c \overline{R}_{s,b,c} \circ f(x) - 1\Big).$$
3. Finally, summing over all b and summing $J \circ f(x)$, the result becomes
$$\sum_b J_b \circ f(x)\Big(\sum_c \overline{R}_{s,b,c} \circ f(x) - 1\Big) + J \circ f(x)$$

This actually gives a valid uSA-refutation since we (1) Multiply our axioms by conical juntas, (2) Sum all the results, and (3) Add an extra junta to the sum. Looking at this polynomial as a function over the boolean cube, we get the following: let χ be an assignment of x and α be the resulting assignment $f(\chi)$ of a. Then

$$\sum_b J_b \circ f(\chi)\Big(\sum_c \overline{R}_{s,b,c} \circ f(\chi) - 1\Big) + J \circ f(\chi) = \sum_b J_b(\alpha)\Big(\sum_c \overline{R}_{s,b,c}(\alpha) - 1\Big) + J(\alpha) = -1.$$

Since functional equality is the same as polynomial equality in multi-linear arithmetic we get the equality $\sum_b J_b \circ f(x)\Big(\sum_c \overline{R}_{s,b,c} \circ f(x) - 1\Big) + J \circ f(x) = -1$. Hence, setting $J'_{b,y}(x) = J_b \circ f(x) \cdot G_{b,y}(x)$ and $J'(x) = J \circ f(x)$, we get the uSA-refutation

$$\sum_{b,y} J'_{b,y}(x)\Big(\overline{G}_{b,y}(x) + \sum_c \overline{R}_{s,b,c} \circ f(x) - 1\Big) + J'(x) = -1$$

of $F_{R(f,g)_n}$. The degree of the refutation is $d(n)(d'(s(n)) + 2)$. ◀

Using this last result, we are able to prove Corollary 27.

Proof of Corollary 27. Let R be a $\mathsf{TF}\Sigma_2^{dt}$ search problem such that R_m admits a uSA-refutation of poly-logarithmic degree $d'(m)$ and let (f, g) be an R-formulation of LOP of size $s(n)$ and degree $d(n)$. Then by Lemma 32, we have an R-formulation of similar size and depth of $R(f, g)$ and by the same lemma as well as Proposition 34 and the proof of Theorem 1, we have

- $R(f, g)_n$ is a weakening of LOP_n for each n and hence admits a super-poly-logarithmic lower bound on the degree of its uSA-refutations;
- $R(f, g)_n$ admits an uSA-refutation of degree $d(n)(d'(s(n)) + 2)$.

We then conclude that either $s(n)$ is super quasi-polynomial or either $d(n)$ is super poly-logarithmic since the sum $d(n)(d'(s(n)) + 2)$ is super poly-logarithmic and d' is poly-logarithmic and hence the R-formulation is not efficient. ◀

6 Open Problems

Separating LeastNumber from StrongAvoid

Can we show that LeastNumber $\not\leq_{dt}$ StrongAvoid? While LeastNumber has a low-degree uSA proof, the size of this proof is exponential, and therefore this does not imply a reduction to StrongAvoid. The existence of a low-degree uSA proof precludes us from using the techniques in this paper to obtain this separation. Furthermore, to our knowledge, all of the size lower bounds that have been proven for Sherali-Adams proceed by first proving a degree lower bound via pseudo-expectations, and then applying the size-degree tradeoff [22]. The only lower bounds which do not proceed via this strategy apply strictly to uSA [6, 12], they do not apply to the general Sherali-Adams proof system. Can these techniques be used in order to separate LeastNumber from StrongAvoid?

Counter-Example Reductions and Herbrandization

Thapen [23] gave a notion of *herbrandization* for $\mathsf{TF}\Sigma_2^{dt}$ problems, which associates a $\mathsf{TF}\Sigma_2$ problem R with a TFNP problem Checkable(R) satisfying the following property.

▶ **Lemma 35** ([23]). *Let $Q, R \in \mathsf{TF}\Sigma_2^{dt}$. If Q admits low-complexity counter-example R-formulation, then* Checkable(Q) *admits a low-complexity* Checkable(R)*-formulation.*

This previous lemma, coupled with our decomposition (Lemma 32), implies the following.

▶ **Lemma 36.** *If for every weakening P of $Q \in \mathsf{TF}\Sigma_2^{dt}$,* Checkable($P$) $\not\leq_{dt}$ Checkable(R)*, then $Q \not\leq_{dt} R$.*

In light of this, if P is a weakening of Q, then it is natural to ask whether there is anything that we can infer about the relationship between Checkable(Q) and Checkable(P)? Heuristically, to obtain intuition as to whether Q reduces to R it has been useful to look at the relationship between Checkable(Q) and Checkable(R) in TFNP^{dt}. This question asks to what degree this heuristic can be formalized.

Decision-tree reductions between problems in TFNP and proofs in certain polynomial-time verifiable proof systems are tightly connected. Counter-example reductions are a polynomial-time verifiable way to relate search problems in higher levels of the polynomial hierarchy. We ask whether one can also obtain characterizations of counter-example reductions to $\mathsf{TF}\Sigma_2^{dt}$ problems by proof systems. Furthermore, if this is the case, how do these proof systems compare to the ones obtained by first herbrandizing the $\mathsf{TF}\Sigma_2$ problem R to a TFNP problem Checkable(R) and then taking its corresponding proof system via [3].

References

1 Paul Beame, Stephen A. Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *J. Comput. Syst. Sci.*, 57(1):3–19, 1998. doi:10.1006/jcss.1998.1575.

2 Josh Buresh-Oppenheim and Tsuyoshi Morioka. Relativized NP search problems and propositional proof systems. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 54–67. IEEE Computer Society, 2004. doi:10.1109/CCC.2004.1313795.

3 Sam Buss, Noah Fleming, and Russell Impagliazzo. TFNP characterizations of proof systems and monotone circuits. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 30:1–30:40. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.30.

4 Lijie Chen, Shuichi Hirahara, and Hanlin Ren. Symmetric exponential time requires near-maximum circuit size. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1990–1999. ACM, 2024. doi:10.1145/3618260.3649624.

5 Stefan S. Dantchev, Barnaby Martin, and Mark Nicholas Charles Rhodes. Tight rank lower bounds for the sherali-adams proof system. *Theor. Comput. Sci.*, 410(21-23):2054–2063, 2009. doi:10.1016/J.TCS.2009.01.002.

6 Susanna F. de Rezende, Aaron Potechin, and Kilian Risse. Clique is hard on average for unary sherali-adams. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 12–25. IEEE, 2023. doi:10.1109/FOCS57990.2023.00008.

7 Noah Fleming, Stefan Grosser, Siddhartha Jain, Jiawei Li, Hanlin Ren, Morgan Shirley, and Weiqiang Yuan. Total search problems in ZPP. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference, ITCS 2026, Bocconi University, Milan, Italy, January 27-30, 2026*, volume 362 of *LIPIcs*, pages 60:1–60:26. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi:10.4230/LIPIcs.ITCS.2026.60.

8 Noah Fleming, Stefan Grosser, Toniann Pitassi, and Robert Robere. Black-box PPP is not turing-closed. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1405–1414. ACM, 2024. doi:10.1145/3618260.3649769.

9 Noah Fleming, Deniz Imrek, and Christophe Marciot. Provably Total Functions in the Polynomial Hierarchy. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference (CCC 2025)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:40, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2025.28.

10 Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Found. Trends Theor. Comput. Sci.*, 14(1-2):1–221, 2019. doi:10.1561/0400000086.

11 S. Ghenntiyala, L. Zeyong, and N. Stephens-Davidowitz. Range avoidance, arthur-merlin, and tfnp. *In Submission*, 20225.

12 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in proof complexity and TFNP. *CoRR*, abs/2205.02168, 2022. doi:10.48550/arXiv.2205.02168.

13 Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, volume 124 of *LIPIcs*, pages 38:1–38:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ITCS.2019.38.

14 Venkatesan Guruswami, Xin Lyu, and Xiuhan Wang. Range avoidance for low-depth circuits and connections to pseudorandomness. *ACM Trans. Comput. Theory*, 17(2):14:1–14:23, 2025. doi:10.1145/3718745.

15 Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos H. Papadimitriou. Total functions in the polynomial hierarchy. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 44:1–44:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.44.

16 Leszek Aleksander Kolodziejczyk and Neil Thapen. Approximate counting and NP search problems. *J. Math. Log.*, 22(3):2250012:1–2250012:31, 2022. doi:10.1142/S021906132250012X.

17 Oliver Korten. The hardest explicit construction. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 433–444. IEEE, 2021. doi:10.1109/FOCS52979.2021.00051.

18 Oliver Korten and Toniann Pitassi. Strong vs. weak range avoidance and the linear ordering principle. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 1388–1407. IEEE, 2024. doi:10.1109/FOCS61266.2024.00089.

19 Oliver Korten, Toniann Pitassi, and Russell Impagliazzo. Stronger cell probe lower bounds via local prgs. *Electron. Colloquium Comput. Complex.*, TR25-030, 2025. URL: https://eccc.weizmann.ac.il/report/2025/030.

20 Zeyong Li. Symmetric exponential time requires near-maximum circuit size: Simplified, truly uniform. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 2000–2007. ACM, 2024. doi:10.1145/3618260.3649615.

21 Tsuyoshi Morioka. Classification of search problems and their definability in bounded arithmetic. *Electron. Colloquium Comput. Complex.*, TR01-082, 2001. URL: https://eccc.weizmann.ac.il/eccc-reports/2001/TR01-082/index.html.

22 Toniann Pitassi and Nathan Segerlind. Exponential lower bounds and integrality gaps for tree-like lovász-schrijver procedures. *SIAM J. Comput.*, 41(1):128–159, 2012. doi:10.1137/100816833.

23 Neil Thapen. How to fit large complexity classes into TFNP. *arXiv:2412.09984 [cs.CC]*, 2024. doi:10.48550/arXiv.2412.09984.

Non-Clifford Gates Are Required for Long-Term Memory

Jon Nelson[1]
Joint Center for Quantum Information & Computer Science, University of Maryland, College Park, MD, USA
NIST, Gaithersburg, MD, USA

Joel Rajakumar[1]
Joint Center for Quantum Information & Computer Science, University of Maryland, College Park, MD, USA
NIST, Gaithersburg, MD, USA
IBM T. J. Watson Research Center, Yorktown Heights, NY, USA

Michael J. Gullans
Joint Center for Quantum Information & Computer Science, University of Maryland, College Park, MD, USA
National Institute of Standards and Technology, Gaithersburg, MD, USA

Abstract

We show that all Clifford circuits under interspersed depolarizing noise lose memory of their input exponentially quickly in the depth of the circuit, even when given access to a supply of fresh qubits initialized in arbitrary states. This result applies only in the absence of intermediate measurements, i.e. without intervention by a noiseless external observer such as a classical computer. Nonetheless, this result is surprising given that non-Clifford circuits with access to a supply of fresh qubits can preserve quantum states for arbitrary lengths of time, and even perform fault tolerant quantum computation on them, without requiring any intermediate measurements (Aharonov et al., STOC 1997). Our result shows that such fault tolerance protocols are impossible using only Clifford gates, demonstrating that non-Clifford gates are fundamentally required to store quantum or classical information for long periods of time.

2012 ACM Subject Classification Theory of computation → Quantum information theory; Theory of computation → Quantum complexity theory

Keywords and phrases Fault Tolerance, Quantum Error Correction

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.38

Funding This material is based upon work supported by the U.S. Department of Energy, Office of Science, Accelerated Research in Quantum Computing, Fundamental Algorithmic Research toward Quantum Utility (FAR-Qu). Additional support is acknowledged from IBM, where JR perfomed a portion of this research as an intern. This material is based upon work supported in part by the NSF QLCI award OMA2120757. This work was performed in part at the Kavli Institute for Theoretical Physics (KITP), which is supported by grant NSF PHY-2309135. JN is supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 2236417.

Acknowledgements We thank Abhinav Deshpande, Kunal sharma, Oles Shtanko, Dominik Hangleiter, Zhi-Yuan Wei, Daniel Malz and Alexey Gorshkov for helpful discussions.

[1] These authors contributed equally to this work.

© Jon Nelson, Joel Rajakumar, and Michael J. Gullans;
licensed under Creative Commons License CC-BY 4.0
41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 38; pp. 38:1–38:13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

It has been known for some time that noisy quantum circuits can fault tolerantly perform arbitrary quantum computations when given access to a constant supply of fresh qubits – or equivalently, the ability to reset a qubit at any time to a fixed state (the *reset gate*) [3]. A weaker task (than computation) that can be made fault tolerant is the implementation of a long-term memory, i.e. storing quantum or classical information in the form of encoded states for long periods of time. Note that the protocol of [3] trivially achieves this task by fault tolerantly implementing the logical identity operation. On the flip side, it has been shown that for noisy quantum circuits without non-unital operations like the reset gate, output states converge towards the maximally mixed state over time [36, 2][2], after which the input state is information-theoretically irrecoverable thus ruling out long-term memory. Here, we show that non-Cliffordness holds a similar power to the reset gate in terms of its necessity for fault-tolerance. In particular, removing the non-Clifford gates from a noisy quantum (or classical) circuit that possesses reset gates also causes it to lose memory of its input exponentially quickly.

We note crucially that we disallow intermediate measurements in our circuit models, which is an identical setup to the threshold theorem proven in [3][3]. This setting is practically relevant as it captures the limitations of near-term quantum devices, where intermediate measurements remain challenging [12]. These restrictions also address the question of whether noisy quantum circuits can protect quantum information *on their own* without the intervention of a noiseless external observer – this has motivated the field of self-correcting quantum memory [18, 5] and is a primary motivation in several fault-tolerance results [2, 3, 7]. Moreover, noiseless classical resources trivially allow the long-term memory of *classical states*, and further, Clifford circuits can store *stabilizer states* when provided with intermediate measurements and noiseless classical processing of measurement outcomes (e.g. syndrome decoding)[23]. In our work, we explicitly disallow noiseless classical resources, which allows us to achieve the strong result that all Clifford protocols lose memory of *any* input state in the presence of noise.

Several recent results have demonstrated a similar loss of memory in noisy *random* quantum circuits that contain resets and other non-unital operations [35, 6, 34, 38]. However, these results differ from ours in a primary aspect, even apart from the assumption of randomness: they show that the input state has negligible effect on the expected value of any *fixed observable* after high enough depth, whereas we show that the input state has negligible effect on all possible observables at once, i.e. that there is no observable that can distinguish two different input states. This is because we prove a convergence in *trace distance*, which, to our knowledge, has never been shown before in a setting that includes non-unital operations and noise of any constant rate. Thus, we anticipate that our analysis techniques may be of broad interest. We state our main theorem below,

▶ **Theorem 1.** *Let Φ denote the channel produced by a circuit of d layers of Clifford gates and reset gates on n qubits, with single-qubit depolarizing noise of strength γ on each qubit after each layer. Let ρ and σ be two arbitrary input states. When $d > d^*$ where $d^* = O(\gamma^{-1} \log(n) \log(n/\epsilon))$,*

$$\|\Phi(\rho) - \Phi(\sigma)\|_1 \leq \epsilon \tag{1}$$

[2] Here, we assume depolarizing noise of any constant rate, on each qubit, after each time step

[3] Note that intermediate measurements can be coherently simulated using the deferred measurement principle, but then any classical processing of the measurement outcomes must be noisy, reversible, and, in our setting, Clifford.

Operationally, our result implies that any two input states become information-theoretically indistinguishable after polylogarithmically many layers of the circuit in the sense that any distinguisher correctly guesses which state it possesses with probability at most $1/2 + o(1/\operatorname{poly}(n))$. This rules out long-term memory in such circuits, since they are incapable of storing quantum or classical information in a manner that can be efficiently extracted (in terms of number of copies) after large time scales. Again, we contrast this with the result of [3], which enables two encoded states to remain efficiently distinguishable for time that scales exponentially with system size[4].

Besides the assumption that the circuit contains only Clifford gates and qubit resets, our result makes no additional assumptions on the circuit. In other words, the statement holds for *any* Clifford circuit using mid-circuit resets in *any* desired locations. In fact, the gates are not required to be geometrically local or even $O(1)$-local. The only requirement is that after each layer of resets and gates a round of constant-strength single-qubit depolarizing channels is applied to each qubit. Finally, the convergence in trace distance occurs for *any* two quantum states. Despite the generality of the result, our proof is relatively short, relying primarily on a counting argument.

1.1 Implications

Our result is somewhat surprising since many error-correction gadgets can be constructed using Clifford gates. For example, one of the most popular types of error correcting codes is a stabilizer code which uses Clifford circuits for both its fault-tolerant encoding and fault-tolerant syndrome extraction protocols [24]. Clifford circuits are also desirable since they propagate Pauli errors nicely making it easier to track errors throughout the circuit, a property which has led to several fault-tolerant implementations of noisy logical Clifford circuits [17, 15, 8, 11]. Although these properties seem useful for correcting errors, our results point out that they are not so powerful on their own, and one must introduce a non-Clifford gate at some point of the computation in order to achieve long-term memory. For example, in order to make intermediate syndrome measurements robust to errors it is often required to perform a majority vote, which cannot be done using Clifford gates. In this way, our result highlights the necessary ingredients for effective fault-tolerant protocols.

Another reason our result is surprising is that magic is often discussed as a computational resource rather than as an error-correction resource. This is because Clifford circuits can be classically simulated efficiently [22] even when augmented with a small number of non-Clifford gates [1, 14]. Thus, a high degree of non-Cliffordness is essential for achieving quantum advantage. In this work, we do not consider the problem of computational hardness but instead focus on the ability to preserve quantum states in memory. This is quite a different task and on the surface does not seem to require departure from Cliffordness as a necessary ingredient. However, our result shows that non-Clifford gates are crucial for error correction in addition to their importance in quantum advantage.

A final feature of our result is that the intermediate reset gates are allowed to produce fresh magic states. One might think that this would allow for the implementation of non-Clifford gates through magic state injection. However, this technique requires a classically controlled Clifford operation as the final step, which is itself a non-Clifford gate. Thus, it is not clear how to use fresh magic states to overcome the limitations of the Clifford circuit. In fact, our no-go result shows that Clifford gates alone cannot exploit the resource

[4] This can be seen by noting that the space and depth overheads of this protocol are polylogarithmic.

of magic states for fault-tolerance, highlighting a distinction between the utility of magic states and non-Clifford gates. This contrasts with the case of computational hardness, where the inclusion of magic states is sufficient to promote Clifford circuits to a model that is hard to sample from classically [13, 45].

1.2 Proof Strategy

Many related noisy circuit problems can be formulated in the Pauli path framework. Broadly speaking, a Pauli path is a trajectory in the Feynman path integral of the circuit with respect to the Pauli basis. In our case, to prove the loss of memory it is sufficient to show that the total contribution of each Pauli path connecting the input to the output state is negligible. To do this, it will be crucial to leverage the property that depolarizing noise causes the contribution of a Pauli path to decay exponentially in the number of non-identities in the path, also called the "weight". This property is the key to many prior works that have also studied noise dynamics in quantum circuits [4, 16, 38, 21, 27].

However, reset gates present a challenge for directly adapting these techniques as they introduce new Pauli paths throughout the circuit. This directly thwarts the Pauli path decay approach, which attempts to suppress the contribution of these paths. To apply Pauli path techniques to circuits with non-unital channels, we use a similar technique to [35, 38, 6, 34], which considers the adjoint map of the noisy circuit. The reason for this is that the adjoint map has the nice property that it is once again unital, and thus the Pauli path decay techniques can be employed on the adjoint map. Unitalness is important for these decay techniques since by definition it means that the map has identity as a fixed point and the presence of depolarizing noise causes these paths to decay until the fixed point is reached.

Even after reformulating our problem as a Pauli path decay problem, the difficulty of the reset gate still presents itself in other subtle ways that require careful analysis. In particular, the adjoint map of the $|0\rangle$-reset gate maps Z to $\mathbb{I}$, which can lower the Pauli weight. This makes the Pauli path less susceptible to noise decay. To address these points, we first treat the depolarization noise as stochastically applying a complete depolarizing error or leaving the qubit unchanged, where a depolarization error has the effect of mapping non-identity Pauli operators to zero. We then reduce our problem to bounding the probability that any non-identity Pauli operator remains at the end of the adjoint circuit. This stochasticization technique was also used for Clifford circuits in [37], but here, we apply it to the adjoint map of the circuit instead. Note that the action of the adjoint reset can now actually help decay the total number of remaining non-identity Pauli operators since it can map two Pauli operators to one (e.g. it maps both $\mathbb{I}$ and Z to $\mathbb{I}$). This property that the number of remaining Pauli operators can only decrease throughout the adjoint circuit is crucial to our analysis.

In our proof, we consider the adjoint circuit in batches of d' layers at a time. Our proof proceeds inductively by alternating two steps. First, we show that all Pauli paths above a given weight w_0 are mapped to zero by a batch of d' layers with high probability. Then we bound the number of remaining Pauli operators by using the property that each input Pauli operator to the batch of layers is mapped to at most one Pauli operator at the output. We then iterate these steps setting $w_1 = w_0/4$ and so on. It turns out that in order to achieve a sufficient decay, it is required that $d' = O(\log n)$ assuming a constant noise strength. Since we must repeat for $O(\log n)$ rounds to decay all possible Pauli weights, we get that no Pauli operators remain at the end of the adjoint circuit after $d = O(\log^2 n)$ total layers.

1.3 Open Questions

Prior work has shown that when circuits have no mechanism for expelling entropy, this entropy accumulates and eventually consumes the circuit [2, 36]. In our work, we analyze the effects of noise in the setting where the circuit is able to pump entropy out of the system by utilizing reset gates. It would be interesting to consider other mechanisms in which the circuit could reduce its entropy. The closest next step would be to generalize our results to show that all Clifford circuits with amplitude damping noise cannot have long-term memory. Similar to the case considered in this paper, there are known fault-tolerant protocols using non-Clifford gates to preserve quantum information under amplitude damping noise [39, 10]. One reason why this could be a tractable problem to solve is that amplitude damping noise acts in a similar manner to a "random reset" error, albeit with some subtle differences. On the other hand, amplitude damping noise also has a unique feature in that it can act as a coherent, but noisy, rotation on stabilizer states [41]; however, it is unclear whether this mechanism can be used as a mid-circuit non-Clifford gate.

Another promising direction is to show that noisy monitored random circuits also have short-term memory. It is known that in the noiseless case, these circuits exhibit a measurement-induced phase transition [29, 40, 28] where above a critical measurement rate, the output of the circuit converges to a single pure state regardless of the input state. On the other hand, below this measurement rate, the circuit preserves an extensive logical codespace for exponentially long times [25]. When noise is introduced, it has been argued using a heuristic mapping of the circuit to a classical Ising model that this phase transition disappears and the circuit loses memory for any constant noise strength and measurement rate [9, 19, 26, 30, 31, 30, 33, 32]. It would be interesting to determine whether our techniques can be used to give a rigorous proof of this statement for the case of random Clifford circuits. Notice that the assumption of randomness is now required since there are known fault-tolerant quantum memory protocols using Clifford gates and intermediate measurements.

Additionally, it would be desirable to lift any of these statements to the case of Haar random circuits. In particular, showing that Haar random circuits under amplitude damping noise have short-term memory would strengthen the result of [35] to the case of trace distance rather than observable estimation which would be a significant improvement. We also point out concurrent work [42], which gives further evidence for this conjecture. Our hope is that the equivalence of second-moment quantities between Clifford circuits and Haar-random circuits may provide one avenue to make progress.

Finally, it is an interesting future direction to consider whether our no-go results on fault-tolerant quantum memory can be lifted to no-go results on computational hardness in certain noise regimes. For example, can we use these techniques to classically simulate Clifford-magic circuits with non-unital noise, similar to [37]? If we can also extend our results to Haar-random circuits as suggested earlier, we may be able to make progress towards classically simulating noisy Haar-random circuits with non-unital noise, which currently presents a significant gap in the literature [20].

2 Background and Notation

2.1 Pauli and Clifford Group

We use $\mathbb{I}$ to denote the identity operator, where the dimension can be inferred by context, and Id to denote the identity channel. The Pauli group is defined as:

$$\mathsf{P}_n := \{1, i, -1, -i\} \times \{\mathbb{I}, X, Y, Z\}^{\otimes n}$$

We define the phaseless Pauli group $\hat{\mathsf{P}}_n \leqslant \mathsf{P}_n$ as:

$$\hat{\mathsf{P}}_n := \{\mathbb{I}, X, Y, Z\}^{\otimes n}$$

We let $|O|$ denote the weight of the Pauli operator $p(O)$, i.e. the number of non-identity qubits. For two sets A and B, we will use $A\backslash B$ to denote $A - A \cap B$.

The Clifford group is defined as:

$$\mathsf{C}_n := \{U \in \mathsf{U}(2^n) : UPU^\dagger \in \mathsf{P}_n \ \forall P \in \mathsf{P}_n\}$$

where $\mathsf{U}(2^n)$ is the group of $2^n \times 2^n$ unitary matrices.

2.2 Stochastic Noise

We define the single-qubit depolarizing noise channel with noise strength γ as:

$$\mathcal{N}_\gamma(\rho) = (1-\gamma)\rho + \gamma\frac{\mathbb{I}}{2}\operatorname{Tr}\rho \tag{2}$$

This channel can be viewed as stochastically applying a complete depolarization error with probability γ, which traces out the given qubit and replaces it with the maximally mixed state. If no error occurs then the channel does nothing. Denoting this depolarization error as $\mathcal{D}$, we have that $\mathcal{N}_\gamma := (1-\gamma)\mathrm{Id} + \gamma\mathcal{D}$. For a given noisy circuit Φ with mid-circuit depolarizing channels, let b be a bitstring that contains a bit for each depolarizing channel representing whether or not $\mathcal{D}$ is applied. We then let Φ_b represent the circuit where each noise channel is replaced with either Id or $\mathcal{D}$ as specified by b. Sampling b according to a binomial distribution parameterized by γ, i.e. $b \sim Bin(nd, \gamma)$, we have that by definition

$$\Phi = \mathbf{E}_b \Phi_b \tag{3}$$

2.3 Reset Channel

We define the qubit reset channel to an arbitrary state ρ as follows:

$$\mathcal{R}_\rho(\sigma) = \rho \operatorname{Tr}\sigma \tag{4}$$

2.4 Adjoint Maps

In our analysis, it will be convenient to use the adjoint of the noisy circuit. For a given channel $\mathcal{N}$, the adjoint map $\mathcal{N}^\dagger$ is defined as follows [43, 44].

$$\langle X, \mathcal{N}(Y)\rangle = \langle \mathcal{N}^\dagger(X), Y\rangle \tag{5}$$

where $\langle X, Y\rangle := \operatorname{Tr} X^\dagger Y$ and X and Y are linear operators on the channel's Hilbert space. From this definition, it can be shown that the adjoint of a given channel $\mathcal{N}(\rho) = \sum_i K_i \rho K_i^\dagger$ is represented by $\mathcal{N}^\dagger(\rho) = \sum_i K_i^\dagger \rho K_i$. Furthermore, given the Pauli transfer matrix of a channel, $T(\mathcal{N})$, which is a matrix with entries $T_{i,j}(\mathcal{N}) = 2^{-n}\langle P_i, \mathcal{N}(P_j)\rangle$ for $P_i, P_j \in \hat{\mathsf{P}}_n$, it is also true that $T(\mathcal{N}^\dagger) = T(\mathcal{N})^\top$.

It is next helpful to explicitly characterize the adjoint map for completely depolarizing errors and qubit resets. First, a depolarizing error $\mathcal{D}$ maps the Pauli matrices as follows

$$\begin{aligned}\mathcal{D}(\mathbb{I}) &= \mathbb{I}\\ \mathcal{D}(X) &= 0\\ \mathcal{D}(Y) &= 0\\ \mathcal{D}(Z) &= 0\end{aligned} \tag{6}$$

Taking the transpose of the associated Pauli transfer matrix we have that $\mathcal{D}^\dagger = \mathcal{D}$.

Next the qubit reset channel maps the Pauli matrices as:

$$\begin{aligned}\mathcal{R}_\rho(\mathbb{I}) &= 2\rho \\ \mathcal{R}_\rho(X) &= 0 \\ \mathcal{R}_\rho(Y) &= 0 \\ \mathcal{R}_\rho(Z) &= 0 \end{aligned} \tag{7}$$

For the adjoint of the reset channel, we can similarly take the transpose of the Pauli matrix associated with Equation (7):

$$\mathcal{R}_\rho^\dagger(\mathbb{I}) = \mathbb{I} \tag{8}$$
$$\mathcal{R}_\rho^\dagger(X) = \mathrm{Tr}(\rho X)\mathbb{I} \tag{9}$$
$$\mathcal{R}_\rho^\dagger(Y) = \mathrm{Tr}(\rho Y)\mathbb{I} \tag{10}$$
$$\mathcal{R}_\rho^\dagger(Z) = \mathrm{Tr}(\rho Z)\mathbb{I} \tag{11}$$

Finally, note that the adjoint of each Clifford gate is also a Clifford gate.

3 Analysis

In general, we will use $A \propto B$ for matrices A, B to refer to the condition that $\exists c \in \mathbb{R} : A = cB$ (note that this condition is satisfied even if $A = 0$). The primary property that we will use in our analysis is that $\Phi_b^\dagger$ is *many-to-one* on the set of Pauli operators since each input Pauli is mapped to at most one output Pauli, as stated below,

▶ **Observation 2.** *For Φ as in Theorem 1 and bitstring b as defined in Section 2.2, for all $s_0 \in \hat{\mathsf{P}}_n$, there exists some $s_1 \in \hat{\mathsf{P}}_n$ such that $\Phi_b^\dagger(s_0) \propto s_1$*

Note that this condition that fails when considering non-Clifford gates, whose adjoint maps can take one Pauli operator to many. Our techniques apply generally to all noisy circuits that can be represented as a mixture over channels whose adjoint maps have this many-to-one property.

3.1 Reduction to Pauli Path Survival Probability

We begin by analyzing our desired quantity in the Pauli basis, which allows us to reduce our problem to bounding the probability that a non-identity Pauli operator is output by the adjoint of the circuit. This is captured in the following lemma,

▶ **Lemma 3.** *For $\Phi, \rho, \sigma, \gamma$ as in Theorem 1,*

$$\|\Phi(\rho) - \Phi(\sigma)\|_1 \leq 2\Pr(\exists s \in \hat{\mathsf{P}}_n : \Phi_b^\dagger(s) \not\propto \mathbb{I}) \tag{12}$$

where the probability $\Pr(\cdot)$ is taken over bitstrings $b \sim Bin(nd, \gamma)$, which specify the locations of depolarization errors in Φ_b.

Proof. For any error configuration b, note the following fact.

$$\Phi_b(\rho) = \Phi_b(\sigma) \tag{13}$$
$$\iff \forall s \in \hat{\mathsf{P}}_n : \mathrm{Tr}(\Phi_b(\rho)s) = \mathrm{Tr}(\Phi_b(\sigma)s) \tag{14}$$
$$\iff \forall s \in \hat{\mathsf{P}}_n : \mathrm{Tr}(\rho\Phi_b^\dagger(s)) = \mathrm{Tr}(\sigma\Phi_b^\dagger(s)) \tag{15}$$

The above property follows from the fact that all density matrices have a unique decomposition in the Pauli basis. Now,

$$\begin{aligned}
\|\Phi(\rho) - \Phi(\sigma)\|_1 &= \|\mathbf{E}_b \Phi_b(\rho) - \Phi_b(\sigma)\|_1 && (16)\\
&\leq \mathbf{E}_b \|\Phi_b(\rho) - \Phi_b(\sigma)\|_1 && (17)\\
&\leq \Pr(\Phi_b(\rho) = \Phi_b(\sigma)) \cdot 0 + \Pr(\Phi_b(\rho) \neq \Phi_b(\sigma)) \cdot 2 && (18)\\
&= 2 \Pr(\Phi_b(\rho) \neq \Phi_b(\sigma)) && (19)\\
&= 2 \Pr(\exists s \in \hat{\mathsf{P}}_n : \mathrm{Tr}(\rho \Phi_b^\dagger(s)) \neq \mathrm{Tr}(\sigma \Phi_b^\dagger(s))) && (20)\\
&\leq 2 \Pr(\exists s \in \hat{\mathsf{P}}_n : \Phi_b^\dagger(s) \propto \mathbb{I}) && (21)
\end{aligned}$$

In the last step, notice that if $\Phi_b^\dagger(s) = c\mathbb{I}$ for some $c \in \mathbb{R}$ then $\mathrm{Tr}(\rho c\mathbb{I}) = \mathrm{Tr}(\sigma c\mathbb{I}) = c$. Therefore, the only way for $\mathrm{Tr}(\rho\Phi_b^\dagger(s)) \neq \mathrm{Tr}(\sigma\Phi_b^\dagger(s))$ is if $\Phi_b^\dagger(s)$ is not proportional to the identity. A crucial step of this proof is Equation (18), where we have used the fact that for any choice of b, the trace distance $\|\Phi_b(\rho) - \Phi_b(\sigma)\|_1$ is always bounded by two. This is true because Φ_b is a quantum channel and the trace distance between two density matrices is at most two. ◀

It now remains to upper bound $\Pr(\exists s \in \hat{\mathsf{P}}_n : \Phi_b^\dagger(s) \propto \mathbb{I})$.

3.2 Grouping Pauli Operators Based on Weight

Our argument will rely on a careful accounting of the non-identity Pauli operators that remain at various points in the adjoint of the circuit. The probability that a Pauli operator is mapped to zero is closely related to its weight since the depolarizing channel maps non-identity Pauli operators to zero with constant probability. For a circuit Φ, it is useful to divide a given set of Pauli operators G into subsets based on their minimum weight throughout the circuit. These subsets are formally defined as follows

▶ **Definition 4.** *For Φ as in Theorem 1, let $S_w(\Phi, G) \subseteq \hat{\mathsf{P}}_n$ be the set:*

$$S_w(\Phi, G) := \{s \in G : \min_i |\Phi_{i\leftarrow}(s)| = w\}. \tag{22}$$

where $\Phi_{i\leftarrow}$ denotes the first i layers of the circuit. Here, each layer constitutes a round of gates, a round of depolarizing channels (or sampled depolarizing errors) and a round of arbitrarily placed qubit resets. When G is not defined we take it to be $\hat{\mathsf{P}}_n$. We define $S_{\geq w}(\Phi, G)$ and $S_{<w}(\Phi, G)$ in a similar manner.

A key fact we will use is the following,

▶ **Observation 5.** *For Φ, d, γ as in Theorem 1, for all $s \in S_w(\Phi, G)$, $\Pr(\Phi_b(s) \neq 0) \leq (1-\gamma)^{wd}$*

This is simply due to the fact that each layer of depolarizing channels will encounter at least w non-identity qubits since this is the minimum weight throughout each layer of the circuit. The Pauli operator survives each depolarizing channel acting on a non-identity with probability $1 - \gamma$. Because there are d layers, this must occur at least wd times, and so the probability of surviving all depolarizing errors is $(1-\gamma)^{wd}$.

3.3 Batched Counting Argument

Our strategy will be to consider $d' = O(\gamma^{-1} \log n)$ layers at a time and show that with high probability the number of remaining Pauli operators decreases substantially after each batch of layers.

▶ **Lemma 6.** *For Φ, d, γ as in Theorem 1, if $d > d^*$ where $d^* = O(\gamma^{-1} \log(n) \log(n/\epsilon))$, then*

$$\Pr(\exists s \in \hat{\mathsf{P}}_n : \Phi_b^\dagger(s) \not\propto \mathbb{I}) \leq \epsilon \tag{23}$$

where the probability $\Pr(\cdot)$ is taken over bitstrings $b \sim Bin(nd, \gamma)$, which specify the locations of depolarization errors in Φ_b.

Note that Theorem 1 is implied by the combination of Lemma 3 and Lemma 6.

Proof. First, notice that we do not need to differentiate between two Pauli operators with different real-valued coefficients at any point in the adjoint map of the circuit. This is because, the coefficient will not affect whether or not this operator gets mapped to zero or identity by the adjoint map of the circuit. To handle this, we introduce the following notation: for any operator $O \propto P$ where $P \in \hat{\mathsf{P}}_n$, we let $p(O) := P$.

Our proof proceeds by first breaking up the adjoint of the circuit into batches of d' layers. Let $\Phi_{b,j}^\dagger$ be the jth batch of d' layers of the adjoint of the circuit. For each j, we will associate a weight limit w_j, where $w_j = \lceil w_{j-1}/4 \rceil$ and $w_0 = n$. Define the sequences of sets $F_0, \dots F_{j-1}, F_j$ and $G_0, \dots G_{j-1}, G_j$ inductively as follows,

$$F_0 = G_0 = \{\hat{\mathsf{P}}_n\} \tag{24}$$

$$F_j = \{p(\Phi_{b,j}^\dagger(s)) \ \forall s \in F_{j-1} \text{ s.t. } \Phi_{b,j}^\dagger(s) \neq 0\} \tag{25}$$

$$G_j = \{p(\Phi_{b,j}^\dagger(s)) \ \forall s \in S_{<w_j}(\Phi_{b,j}^\dagger, G_{j-1}) \text{ s.t. } \Phi_{b,j}^\dagger(s) \neq 0\} \tag{26}$$

In other words, F_j and G_j are defined as the sets of all unscaled output Pauli operators of $\Phi_{b,j}^\dagger$ starting from the input Pauli operators in F_{j-1} and G_{j-1} respectively, where G_j has the added restriction that the minimum weight of the operator as it passes through $\Phi_{b,j}^\dagger$ is less than w_j. With this notation, our new goal is to prove that $\Pr(F_j \neq \{\mathbb{I}\}) \leq \epsilon$ for some large enough j.

We will iterate through the batches while computing probabilities over $b \sim Bin(nd, \gamma)$ as follows: when we are considering the jth batch of the circuit, we will fix all locations of depolarizing errors in batches $1 \dots j-1$, and use $\Pr_j(\cdot)$ to to denote the probability over locations of depolarizing errors within $\Phi_{b,j}^\dagger$. Our inductive hypothesis is that $\Pr_{j-1}(F_{j-1} \neq G_{j-1}) \leq (j-1)\epsilon'$ and that $|G_{j-1}| \leq d'(9n)^{w_{j-1}}$. This is clearly satisfied in the base case of $j - 1 = 0$. Our inductive step will be to show that $\Pr_j(F_j \neq G_j) \leq j'\epsilon$, and that $|G_j| \leq d'(9n)^{w_j}$. We will iterate this argument until $w_j = 1$, which occurs at $j = O(\log n)$, at which point $G_j = \{\mathbb{I}\}$ since all operators in G_j must have weight less than w_j[5]. At this point, by induction, $\Pr(F_j \neq \{\mathbb{I}\}) \leq O(\epsilon' \log n)$, and so we choose $\epsilon' = O(\epsilon/\log n)$ to complete the proof.

[5] Note that the bounds $|G_j| \leq d'(9n)^{w_j}$ never reaches 1, and we have established $|G_j| = 1$ via a different argument

First, we upper bound $\Pr(F_j \neq G_j)$,

$$\begin{aligned}
\Pr_j(F_j \neq G_j) &= \Pr_j(F_j \neq G_j | F_{j-1} = G_{j-1}) \Pr_{j-1}(F_{j-1} = G_{j-1}) && (27)\\
&+ \Pr_j(F_j \neq G_j | F_{j-1} \neq G_{j-1}) \Pr_{j-1}(F_{j-1} \neq G_{j-1}) && (28)\\
&\leq \Pr_j(F_j \neq G_j | F_{j-1} = G_{j-1}) + (j-1)\epsilon' && \text{(inductive hypothesis)}\\
&\leq \Pr_j(\exists s \in S_{\geq w_j}(\Phi^\dagger_{b,j}, G_{j-1}) : \Phi^\dagger_{b,j}(s) \neq 0) + (j-1)\epsilon' && (29)\\
&\leq \sum_{s \in S_{\geq w_j}(\Phi^\dagger_{b,j}, G_{j-1})} \Pr_j(\Phi^\dagger_{b,j}(s) \neq 0) + (j-1)\epsilon' && \text{(union bound)}\\
&\leq |G_{j-1}|(1-\gamma)^{w_j d'} + (j-1)\epsilon' && (30)\\
&\leq d'(9n)^{w_{j-1}}(1-\gamma)^{w_{j-1}d'/4} + (j-1)\epsilon' && \text{(inductive hypothesis)}\\
&\leq \epsilon' + (j-1)\epsilon' && (\text{setting } d' = O(\gamma^{-1}\log(n/\epsilon')))\\
&= j\epsilon' && (31)
\end{aligned}$$

Next, we upper bound $|G_j|$. Consider every possible *intermediate* Pauli operator at a given layer of $\Phi^\dagger_{b,j}$, whose weight is less than w_j. We can exploit the property that each such Pauli operator, whichever layer it is in, gets mapped by $\Phi^\dagger_{b,j}$ to at most one output Pauli operator, which may or may not be in G_j. Therefore, we can upper bound $|G_j|$ using the size of this set. There are d' layers, w_j possible values of the minimum weight, at most $\binom{n}{w_j}$ ways to choose the locations of its non-identities [6], and at most 3^{w_j} ways to assign non-identities to X, Y, and Z. So we have,

$$\begin{aligned}
|G_j| &\leq w_j d' \binom{n}{w_j} 3^{w_j} && (32)\\
&\leq w_j d' (\frac{en}{w_j})^{w_j} 3^{w_j} && (33)\\
&\leq d'(9n)^{w_j} && (34)
\end{aligned}$$

◀

References

1 Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70:052328, November 2004. doi:10.1103/PhysRevA.70.052328.

2 D. Aharonov, M. Ben-Or, R. Impagliazzo, and N. Nisan. Limitations of noisy reversible computation, 1996. arXiv:quant-ph/9611028.

3 Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 38(4):1207–1282, 2008. doi:10.1137/S0097539799359385.

4 Dorit Aharonov, Xun Gao, Zeph Landau, Yunchao Liu, and Umesh Vazirani. A polynomial-time classical algorithm for noisy random circuit sampling. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 945–957, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585234.

5 R. Alicki, M. Horodecki, P. Horodecki, and R. Horodecki. On thermal stability of topological qubit in kitaev's 4d model. *Open Systems & Information Dynamics*, 17(01):1–20, 2010. doi:10.1142/S1230161210000023.

[6] assuming $w_j \leq n/2$, which is valid since only $w_0 > n/2$ and this is the base case

6 Armando Angrisani, Antonio A. Mele, Manuel S. Rudolph, M. Cerezo, and Zoe Holmes. Simulating quantum circuits with arbitrary local noise using pauli propagation, 2025. arXiv:2501.13101.

7 A. Anshu, Y. Liu, Q. Nguyen, and C. Pattison, 2025. Simons talk available at: https://simons.berkeley.edu/talks/quynh-nguyen-harvard-university-2025-05-28.

8 Dave Bacon, Steven T. Flammia, Aram W. Harrow, and Jonathan Shi. Sparse quantum codes from quantum circuits. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 327–334, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2746539.2746608.

9 Yimu Bao, Soonwon Choi, and Ehud Altman. Symmetry enriched phases of quantum circuits. *Annals of Physics*, 435:168618, 2021. doi:10.1016/j.aop.2021.168618.

10 Michael Ben-Or, Daniel Gottesman, and Avinatan Hassidim. Quantum refrigerator, 2013. arXiv:1301.1995.

11 Thiago Bergamaschi and Yunchao Liu. On Fault Tolerant Single-Shot Logical State Preparation and Robust Long-Range Entanglement. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:9, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2025.16.

12 Dolev Bluvstein, Simon J. Evered, Alexandra A. Geim, Sophie H. Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, J. Pablo Bonilla Ataides, Nishad Maskara, Iris Cong, Xun Gao, Pedro Sales Rodriguez, Thomas Karolyshyn, Giulia Semeghini, Michael J. Gullans, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997):58–65, 2024.

13 Adam Bouland, Joseph F. Fitzsimons, and Dax Enshan Koh. Complexity Classification of Conjugated Clifford Circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference (CCC 2018)*, volume 102 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:25, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2018.21.

14 Sergey Bravyi and David Gosset. Improved classical simulation of quantum circuits dominated by clifford gates. *Phys. Rev. Lett.*, 116:250501, June 2016. doi:10.1103/PhysRevLett.116.250501.

15 Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, July 2020. doi:10.1038/s41567-020-0948-z.

16 Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. Achieving quantum supremacy with sparse and noisy commuting quantum computations. *Quantum*, 1:8, April 2017. doi:10.22331/q-2017-04-25-8.

17 Nicolas Delfosse and Adam Paetznick. Spacetime codes of clifford circuits, 2023. doi:10.48550/arXiv.2304.05943.

18 Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, September 2002. doi:10.1063/1.1499754.

19 Beatriz C. Dias, Domagoj Perkovic, Masudul Haque, Pedro Ribeiro, and Paul A. McClarty. Quantum Noise as a Symmetry-Breaking Field. *arXiv preprint*, 2022. arXiv:2208.13861.

20 Bill Fefferman, Soumik Ghosh, Michael Gullans, Kohdai Kuroiwa, and Kunal Sharma. Effect of nonunital noise on random-circuit sampling. *PRX Quantum*, 5:030317, July 2024. doi:10.1103/PRXQuantum.5.030317.

21 Xun Gao and Luming Duan. Efficient classical simulation of noisy quantum computation, 2018. arXiv:1810.03176.

22 Daniel Gottesman. The heisenberg representation of quantum computers, 1998. arXiv:quant-ph/9807006.

23 Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation, 2009. arXiv:0904.2557.

24 Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Info. Comput.*, 14(15–16):1338–1372, November 2014. doi:10.26421/QIC14.15-16-5.

25 Michael J. Gullans and David A. Huse. Dynamical purification phase transition induced by quantum measurements. *Physical Review X*, 10(4), October 2020. doi:10.1103/physrevx.10.041020.

26 Shao-Kai Jian, Chunxiao Liu, Xiao Chen, Brian Swingle, and Pengfei Zhang. Quantum error as an emergent magnetic field. *arXiv preprint*, 2021. arXiv:2106.09635.

27 Julia Kempe, Oded Regev, Falk Unger, and Ronald de Wolf. Upper bounds on the noise threshold for fault-tolerant quantum computing. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, pages 845–856, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-540-70575-8_69.

28 Yaodong Li, Xiao Chen, and Matthew P. A. Fisher. Quantum zeno effect and the many-body entanglement transition. *Phys. Rev. B*, 98:205136, November 2018. doi:10.1103/PhysRevB.98.205136.

29 Yaodong Li, Xiao Chen, and Matthew P. A. Fisher. Measurement-driven entanglement transition in hybrid quantum circuits. *Phys. Rev. B*, 100:134306, October 2019. doi:10.1103/PhysRevB.100.134306.

30 Yue Li and Martin Claassen. Statistical mechanics of monitored dissipative random circuits. *Phys. Rev. B*, 108:104310, September 2023. doi:10.1103/PhysRevB.108.104310.

31 Zhi Li, Shengqi Sang, and Timothy H. Hsieh. Entanglement dynamics of noisy random circuits. *Phys. Rev. B*, 107:014307, January 2023. doi:10.1103/PhysRevB.107.014307.

32 Shuo Liu, Ming-Rui Li, Shi-Xin Zhang, and Shao-Kai Jian. Entanglement structure and information protection in noisy hybrid quantum circuits. *Phys. Rev. Lett.*, 132:240402, June 2024. doi:10.1103/PhysRevLett.132.240402.

33 Shuo Liu, Ming-Rui Li, Shi-Xin Zhang, Shao-Kai Jian, and Hong Yao. Universal kardar-parisi-zhang scaling in noisy hybrid quantum circuits. *Phys. Rev. B*, 107:L201113, May 2023. doi:10.1103/PhysRevB.107.L201113.

34 Victor Martinez, Armando Angrisani, Ekaterina Pankovets, Omar Fawzi, and Daniel Stilck França. Efficient simulation of parametrized quantum circuits under nonunital noise through pauli backpropagation. *Phys. Rev. Lett.*, 134:250602, June 2025. doi:10.1103/j1gg-s6zb.

35 Antonio Anna Mele, Armando Angrisani, Soumik Ghosh, Sumeet Khatri, Jens Eisert, Daniel Stilck França, and Yihui Quek. Noise-induced shallow circuits and absence of barren plateaus, 2024. arXiv:2403.13927.

36 Alexander Müller-Hermes, Daniel Stilck França, and Michael M. Wolf. Relative entropy convergence for depolarizing channels. *Journal of Mathematical Physics*, 57(2), January 2016. doi:10.1063/1.4939560.

37 Jon Nelson, Joel Rajakumar, Dominik Hangleiter, and Michael J. Gullans. *Polynomial-Time Classical Simulation of Noisy Quantum Circuits with Naturally Fault-Tolerant Gates*, pages 1309–1331. SIAM, 2026. doi:10.1137/1.9781611978971.50.

38 Thomas Schuster, Chao Yin, Xun Gao, and Norman Y. Yao. A polynomial-time classical algorithm for noisy quantum circuits, 2024. doi:10.48550/arXiv.2407.12768.

39 Oles Shtanko and Kunal Sharma. Complexity of local quantum circuits under nonunital noise, 2024. arXiv:2411.04819.

40 Brian Skinner, Jonathan Ruhman, and Adam Nahum. Measurement-induced phase transitions in the dynamics of entanglement. *Phys. Rev. X*, 9:031009, July 2019. doi:10.1103/PhysRevX.9.031009.

41 Fabian Ballar Trigueros and José Antonio Marín Guzmán. Nonstabilizerness and error resilience in noisy quantum circuits, 2025. arXiv:2506.18976.

42 Su un Lee, Soumik Ghosh, Changhun Oh, Kyungjoo Noh, Bill Fefferman, and Liang Jiang. Classical simulation of noisy random circuits from exponential decay of correlation, 2025. arXiv:2510.06328.

43 John Watrous. cs.uwaterloo.ca. https://cs.uwaterloo.ca/~watrous/QIT-notes/QIT-notes.pdf. [Accessed 11-09-2025].

44 Mark Wilde. markwilde.com. https://markwilde.com/teaching/2015-fall-qit/lectures/lecture-10.pdf. [Accessed 11-09-2025].

45 Mithuna Yoganathan, Richard Jozsa, and Sergii Strelchuk. Quantum advantage of unitary clifford circuits with magic state inputs. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 475(2225):20180427, May 2019. doi:10.1098/rspa.2018.0427.

Faux Determinism

Shalev Ben-David
Institute for Quantum Computing, Waterloo, Canada
University of Waterloo, Waterloo, Canada

M. H. Ebtehaj
Institute for Quantum Computing, Waterloo, Canada
University of Waterloo, Waterloo, Canada

Abstract

In the context of search problems, a pseudodeterministic algorithm is one which uses randomness but appears to be deterministic in its external behavior: it outputs the same answer when it is run on any one input multiple times. We introduce the relaxed notion of *computationally-secure pseudodeterminism*, which we call faux determinism. A faux-deterministic algorithm must appear to be deterministic to any external adversary with bounded resources. In other words, an algorithm is faux-deterministic if the task of "finding an input on which it fails to deterministically solve the search problem" is computationally intractable. In the black-box setting, we show that every verifiable search problem which has a randomized algorithm also has a faux-deterministic algorithm; this is not true of pseudodeterministic algorithms, which do not always exist. We highlight two reasons why this faux derandomization result is interesting: first, due to the computational indistinguishability, faux-deterministic algorithms may be used in place of pseudodeterministic ones in cryptographic settings. Second, from a conceptual point of view, our result helps explain why pseudodeterministic lower bounds are often hard to prove: such lower bounds must be inherently nonconstructive, since within $\mathsf{FZPP} \cap \mathsf{FNP}$, there is no efficient black-box algorithm for finding a hard input for a given faux-deterministic algorithm.

2012 ACM Subject Classification Theory of computation → Oracles and decision trees; Theory of computation → Pseudorandomness and derandomization; Theory of computation → Probabilistic computation

Keywords and phrases Decision Trees, Search Problems, Query Complexity, Pseudodeterminism, TFNP, Refutation

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.39

Funding This research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), DGECR-2019-00027 and RGPIN-2019-04804.[1]

Acknowledgements We thank Dima Gavinsky for helpful discussions, as well as anonymous reviewers for helpful comments on a draft of this paper.

1 Introduction

In computational complexity, the power of randomness is usually modeled via complexity classes such as BPP, ZPP, or RP; these allow the randomness to affect the probability of error or the runtime of the algorithm.

In the context of search problems, the output of the algorithm may additionally depend on the randomization, even when the output is correct (there are typically multiple correct outputs for a given input to a search problem). This leads naturally to the notion of *pseudodeterminism*, introduced in [6]. A pseudodeterministic algorithm is defined to be an

[1] Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), DGECR-2019-00027 et RGPIN-2019-04804.

© Shalev Ben-David and M. H. Ebtehaj;
licensed under Creative Commons License CC-BY 4.0

algorithm that may use randomness internally, but always outputs the same (correct) answer when run on the same input, even if run with different random seeds, except for a small probability of error.

Pseudodeterministic algorithms have been the subject of several lines of work in various computational models including query complexity [7, 11, 3], communication complexity [12], parallel and space-bounded computation [8, 10], and interactive proofs [9]. A major highlight is the recent result proving existence of a polynomial time pseudodeterministic algorithm that generates a canonical prime number for infinitely many input lengths [13, 5]. In the random oracle model, Yamakawa and Zhandry [14] also gave a search problem which can be solved efficiently using a quantum algorithm, but (assuming the Aaronson-Ambainis conjecture) cannot be solved efficiently by a pseudodeterministic quantum algorithm; they used this property for generating certified random bits, relying on the fact that any quantum algorithm which correctly solves the problem must give an output that has nontrivial entropy.

In this work, we introduce a variant of pseudodeterminism which we call faux determinism. A faux-deterministic algorithm need not succeed on all inputs; instead, it may fail to have good behavior on some inputs, so long as it is computationally difficult to find those inputs. In other words, while pseudodeterminism requires algorithms to be information-theoretically indistinguishable from deterministic algorithms in their output behavior, faux determinism merely requires algorithms to be computationally indistinguishable from deterministic algorithms relative to resource-bounded adversaries.

1.1 Definition and basic observations

We study faux determinism in the query (black-box) model.

Defining the model

We first clarify the definition. A faux-deterministic algorithm needs to have a random seed the adversary does not know about in advance, since otherwise finding an input on which the algorithm fails would be trivial (it could be hard-coded). For this reason, we define a faux-deterministic algorithm to be a probability distribution B over deterministic algorithms, with the property that for any computationally-bounded adversary A, if an algorithm B_r is sampled from this distribution, then the adversary A fails to find an input on which B_r fails when given oracle access to B_r.

In other words, our order of quantifiers is as follows: first, the faux-deterministic algorithm B is specified, except for a random seed r to be chosen later. Next, the adversary's algorithm A is chosen, in a way that may depend on the faux-deterministic algorithm B but not on its random seed. After this, the random seed r is chosen uniformly at random. Finally, A can make adaptive oracle calls to B_r, all with the same random seed r, which is now fixed. The goal of A is to find an input x to the search problem on which B_r fails, for this specific choice of random seed r. The algorithm B is called faux-deterministic if the task of A is impossible (except with superpolynomial resources).

In this work, we restrict our attention to the black-box query model; the algorithm B_r will be a decision tree, so B will be a probability distribution over decision trees, and the adversary A will be a (classical or quantum) query algorithm making queries to the function defined by B_r. In other words, the "truth table" of B_r will be considered the query input to the algorithm A, and strings x which are inputs to B_r are considered *indices* to the

exponentially long string B_r.[2] We often take the adversary A to be a randomized query algorithm, but we also consider the case where A is a quantum query algorithm, which makes queries to B_r in superposition. The formal definition is given in Definition 8.

To further clarify the definition, we make a few observations.

▶ **Lemma 1.** *A faux-deterministic algorithm is always a randomized algorithm.*

Proof. We show the contrapositive. If algorithm B fails as a randomized algorithm, there must exist a fixed input x such that $B_r(x)$ fails to give a valid output to the search problem with non-negligible probability over the randomness r. The adversary A can then deterministically output x without making any queries to B_r. With non-negligible probability over r, it will hold that B_r fails on the output x of A, which means that A refutes the proposed faux-deterministic algorithm B. ◀

While a faux-deterministic algorithm is always a randomized algorithm, the converse does not hold in general.

▶ **Lemma 2.** *A randomized algorithm, even with exponentially small error, need not satisfy the conditions of a faux-deterministic algorithm.*

Proof. Consider the FIND1 problem, the search problem in query complexity in which the input string x is promised to have Hamming weight at least $n/2$ and the goal is to output $i \in [n]$ such that $x_i = 1$. This has a randomized algorithm (output a random position i) whose error can be reduced by amplification (repeat k times to sample k random positions, query them to see if any of them are 1, and output the position of the first 1 found; this achieves error 2^{-k}).

This algorithm works on all inputs satisfying the Hamming weight promise. However, it is not faux-deterministic: when the randomness r is fixed, an adversary A can start by querying B_r on input x of high Hamming weight (such as $x = 1^n$), which gives a position i_1 such that $x_{i_1} = 1$; then it can construct an input y such that $y_{i_1} = 0$ and run $B_r(y)$, getting a different position i_2 such that $y_{i_2} = 1$; then find an input z with $z_{i_1} = z_{i_2} = 0$, querying it to find i_3; and so on. After k steps, the adversary will then find the entire set of k positions that the randomized algorithm B_r checks (which is a fixed set for any fixed choice of randomness r). If the adversary sets all those k positions to 0, the resulting string will be an input on which B_r fails. Finding this input requires only k queries to B_r, which is linear in the runtime of B_r, so B_r fails to be faux-deterministic (faux determinism requires superpolynomial security in the efficiency parameter). ◀

In fact, we can strengthen this result to show that some search problems which have a randomized algorithm lack *any* efficient faux-deterministic algorithm. However, we will later show in Theorem 5 that this does not happen for verifiable search problems such as FIND1.

▶ **Lemma 3.** *There is a black-box non-verifiable search problem which has an efficient randomized query algorithm but no efficient faux-deterministic algorithm.*

[2] One may object that the exponentially longer input string is not practical. For example, if the efficiency parameter is taken to be $\log n$, the adversary A must output strings of length n (and specify a string of length n whenever it makes a query to B_r). If one wants the adversary to specify those strings efficiently, one could switch to the "graybox" model, and represent each string x by a small circuit computing it. Our results should hold for this graybox model too. However, one must take care with graybox models, because depending on the allowed circuit sizes, some graybox models have FP = FBPP for verifiable search problems, causing a collapse of all the models we study in this work. We stick to the black-box model for simplicity.

Our proof of this lemma is very similar to the AOES construction in the concurrent work of [1].

Proof. Consider the following query problem on n bits: split the input into c blocks of n/c bits each, apply parity to each block, and output a string in $\{0,1\}^c$ purporting to solve the c parities; the output $z \in \{0,1\}^c$ is considered valid for an input x if it is *not all wrong*, that is, if for at least one $\ell \leq c$ it holds that z_ℓ is the value of the ℓ-th parity in the string x.

A randomized algorithm which makes no queries to x and guesses randomly will succeed except with error 2^{-c}. On the other hand, any faux-deterministic algorithm B can be refuted by a simple adversary A that employs the following strategy: first, A queries B_r on the input 0^n to get an output string $z \in \{0,1\}^c$. Next, A picks $i \in [n/c]$ at random, and flips the i-th bit of the ℓ-th parity block in the input 0^n for each $\ell \in [c]$ such that $z_\ell = 0$. In other words, A attempts to make all parities be wrong by changing only the i-th bit of each one. This results in a new input x (which has Hamming weight between 0 and c). Now, if B_r makes k queries to its input, then when i is chosen at random, with probability at least $1 - ck/n$, none of the i-th bits of any of the c parities in 0^n are queried by B_r. Therefore, with probability at least $1 - ck/n$, $B_r(x)$ still outputs the same output string z as before, which gets every parity wrong. Hence B_r fails on x, as desired.

Setting $c = \text{polylog}(n)$, the randomized algorithm has worst-case error $1/\text{ quasi-poly}(n)$ using 0 queries; on the other hand, the adversary A refutes B even if B uses $n^{1-o(1)}$ queries, with probability at least $1 - 1/\text{ poly}(n)$, using only a single query to B_r. Our definition of A was randomized, but it can also be made deterministic by choosing i to be the fixed bit in $[n/c]$ which works for as many random seeds r as possible. ◀

So far, we have seen that faux-deterministic algorithms are a proper subset of randomized algorithms. We next show that every pseudodeterministic algorithm, suitably amplified, is a faux-deterministic algorithm. It will also follow from Theorem 5 that not every faux-deterministic algorithm is pseudodeterministic; faux determinism is an intermediate condition between pseudodeterminism and randomized algorithms.

▶ **Lemma 4.** *A pseudodeterministic algorithm with negligible failure probability is always also a faux-deterministic algorithm.*

Proof. To see this, consider such an algorithm B, and let B_r denote the pseudodeterministic algorithm B with the randomness set to the fixed string r. Suppose there is an adversary strategy A such that (with non-negligible probability when r is chosen at random) A efficiently queries B_r to find an input x on which B_r fails. Now, since B is pseudodeterministic, it defines a deterministic function f from inputs to outputs. Moreover, since the failure rate of B is assumed to be negligible, the event that $B_r(z)$ outputs something different from $f(z)$ is negligible. Therefore, unless A makes exponentially many queries, the union bound dictates that with high probability over choice of r, the outputs A sees to its queries are identical to the outputs of f on those same queries.

Suppose then that the final output of A is a string x on which B_r fails. We can then produce x without any queries to B_r, simply by "querying" f instead of B_r, which is free (it has no dependence on r); this produces the same final output x on all but a negligible fraction of the random strings r. If A succeeds with non-negligible probability (over r), it means that the fixed string x is such that B fails on it with non-negligible probability (over r), which contradicts the assumption that B is a pseudodeterministic algorithm with negligible failure probability. ◀

To summarize, we have

$\mathsf{FP} \subseteq \mathsf{PseudoP} \subseteq \mathsf{FauxP} \subseteq \mathsf{FBPP}$.

In query complexity, for promise problems, all these inclusions are strict. The strictness of the first inclusion follows from the fact that randomized and deterministic algorithms can be separated for partial functions: a partial function can be interpreted as a search problem with one valid output for each input in the promise, and in that setting randomized algorithms are always pseudodeterministic.

However, the story changes for *verifiable* search problems: it actually holds that each randomized algorithm can be converted to a faux deterministic algorithm in a nontrivial way; this is Theorem 5. Therefore, the classes FauxP and FBPP (and also FZPP) collapse in the black-box model when restricted to verifiable search problems.

1.2 Faux derandomization

Our main contribution, other than introducing the notion of faux determinism, is the following "faux derandomization" theorem.

▶ **Theorem 5** (Faux derandomization). *In the query (black box) model, every search problem in* $\mathsf{FNP} \cap \mathsf{FBPP}$ *has a faux-deterministic algorithm, even in the setting of promise problems. In other words, if a search problem can be both verified efficiently and solved by a randomized algorithm, the randomized algorithm can be made faux-deterministic.*

More specifically, if the search problem has a verifier which uses v queries and a randomized algorithm which uses k queries, then for any t it has a faux-deterministic algorithm which uses $O((v+k)t)$ queries and cannot be refuted by an adversary making fewer than $2^{\Omega(\sqrt{t})}$ queries except with probability at most $2^{-\Omega(\sqrt{t})}$.

Moreover, this holds even with respect to a quantum adversary that makes quantum queries to the faux deterministic algorithm.

The statement of the theorem in its full generality can be found in Theorem 22. We highlight three ways to think about this theorem.

Computationally secure derandomization

Our theorem says that for verifiable search problems, any randomized algorithm can be "derandomized" in a computationally secure way: this is not true derandomization, but no polynomially-bounded adversary making black-box queries can distinguish it from true derandomization. More explicitly, starting with any randomized algorithm B, we can transform it to another randomized algorithm B' such that when the randomness r is fixed, B'_r solves the problem, even with *any subexponential number of (possibly adversarial) repeated uses*. The new algorithm has a "set it and forget it" random seed r; once it is fixed, even a computationally unbounded adversary which makes subexponentially many queries to the algorithm cannot find an instance on which the now-deterministic algorithm B'_r fails.

Since this "faux derandomized" algorithm is indistinguishable from a deterministic algorithm to computationally bounded adversaries, it can replace deterministic or pseudodeterministic protocols in cryptographic settings in which the latter are employed.[3]

[3] It should be noted, however, that there are no compelling cryptographic use cases for pseudodeterminism to date.

(To be sure, our algorithm still needs a random seed, so it may not look like derandomization; however, once the random seed is fixed, we get security against exponentially many adversarial inputs, giving us in some sense "exponentially long stretch" on the random seed length.)

Refutation paradigm

The refutation problems are studied in metacomplexity; see [4] and the references therein. Roughly speaking, if a task requires some amount of resources, the refutation task is to take as input an algorithm which uses fewer resources and demonstrate that it fails. Our result can be viewed as saying that in query complexity, the task of refuting pseudodeterministic algorithms for verifiable search problems is hard. Indeed, our faux-deterministic algorithms are just probability distributions over algorithms that pretend to be pseudodeterministic, but are not; yet when an algorithm is sampled from this distribution, it is hard to find an input on which it fails to be pseudodeterministic.

Nonconstructivity in pseudodeterministic lower bounds

Another interpretation of Theorem 5 is an explanation for the difficulty of proving pseudodeterministic lower bounds. The latter are notoriously difficult, even in the black-box setting: for instance, the pseudodeterministic complexity of the "find 1" problem from the proof of Lemma 2 is still open. Recall that in this task, the input is an n-bit string of Hamming weight at least $n/2$ and the goal is to output the position of a 1; this can be done in $O(1)$ randomized queries, requires $\Omega(n)$ deterministic queries, and its pseudodeterministic query complexity is only known to be between $\Omega(\sqrt{n})$ and $O(n)$, despite substantial attention from the community [11, 3]. What we show here is that the FIND1 problem has an $O(\text{polylog}\, n)$ faux-deterministic algorithm. This means that any lower bound must be nonconstructive: it must prove, for each algorithm B, the existence of an input string x on which it fails, but without outlining an efficient procedure for finding that string x (given oracle access to B). In other words, some natural lower bound techniques against pseudodeterminism also lower bound faux determinism, but since efficient faux-deterministic algorithms always exist (for $\mathsf{FBPP} \cap \mathsf{FNP}$), such techniques cannot rule out efficient pseudodeterministic algorithms.

One way to formalize the difficulty of proving a lower bound for FIND1 is via the following lemma.

▶ **Lemma 6.** *Consider the following search problem. The input is query access to a function f which solves the* FIND1 *problem: it maps strings of length n and Hamming weight at least $n/2$ to $[n]$, with the property that $x_{f(x)} = 1$ for every such x. The output is some input x of Hamming weight at least $n/2$ such that the block sensitivity of f at x is $\omega(\text{polylog}(n))$.*

Even though [11] show that every function solving FIND1 *has block sensitivity at least $\Omega(\sqrt{n})$ on some input, this search problem requires strictly more than $\text{poly}(n)$ queries to solve.*

Lemma 6 follows fairly straightforwardly from Theorem 5, as detailed in Section 5. But this consequence is perhaps surprising: combined with the existence of highly sensitive inputs ([11]), it gives a "meta" TFNP search problem which cannot be efficiently solved even with much weaker parameters.

Finally, we show that the parameters in Theorem 5 are quadratically tight (see Section 3.4).

▶ **Lemma 7.** *For any faux-deterministic algorithm for* FIND1 *which uses t queries, there exists an adversary strategy which uses 2^{t+1} queries and succeeds in refuting the faux-deterministic algorithm with probability at least $1/2$.*

1.3 Our techniques

We describe our approach for proving Theorem 5.

Reduction to FIND1

First, we use the existence of an FBPP algorithm B to reduce this task to the FIND1 problem. That is to say, on any input x, we can view the search problem as the task of finding a random seed r such that the output $B_r(x)$ passes verification by the verifier V; in other words, we want to find a random seed that causes the algorithm B to succeed at solving x, and we want to find this seed in a way that looks deterministic. We know that most seeds succeed (since B is a valid randomized algorithm), and we can check a seed (using the verifier for the search problem), so this amounts to a special case of the FIND1 problem in which we are given a black box which is known to accept most inputs r, and our goal is to "find a 1 in the string", that is, find a specific input r which is accepted. This reduction is somewhat standard for pseudodeterministic algorithms; see [11].

Attempts to faux-derandomize FIND1

Here is an example approach which does not work for designing a faux-deterministic algorithm for FIND1. We could try having B_r query a fixed (randomly chosen) subset S of $\log n$ bits of the input, and have it output the first index i for which $x_i = 1$. (This failed strategy also appeared earlier in Section 1.1; see Lemma 2.) For most x, there will be a 1 within those $\log n$ bits, so this algorithm will succeed. However, an adversary can proceed as follows: first it runs $B_r(x^1)$ on an arbitrary input x^1 to get an index $i_1 \in S$ with $x^1_{i_1} = 1$. Next, it picks an input x^2 close in Hamming distance to x^1 with $x^2_{i_1} = 0$, and runs $B_r(x^2)$ to get a different $i_2 \in S$. It repeats this, each time setting the previously chosen position of the input to 0. After around $|S|$ iterations, it will find an input on which B_r fails to find a 1.

The failure of the above non-adaptive approach to the design of B may suggest that adaptivity is important. We could try having B_r query adaptively (e.g. first query position 3, then based on its value, choose whether to query position 5 or 2, and then fork on the answer again, with paths growing exponentially and the positions being fixed in B_r but chosen randomly when r is random). The algorithm B_r can then pick the last 1 it saw down this path (alternatively, we could try the first 1 it saw). Unfortunately, this strategy also fails: the adversary can detect bits that are high up in the path (i.e. queried early), and set them to 0 successively, until the whole path is 0. This is easy for the adversary to do if $B_r(x)$ returns the first 1 it saw (it can just set that bit to 0 in the next input). If $B_r(x)$ returns the last 1 it saw, the adversary can detect indices i along the path taken by $B_r(x)$ by observing that flipping those bits changes the output of $B_r(x)$. It can binary search to find the bits along the path taken by $B_r(x)$, and just set them all to 0 in the next input. Repeating this process leads to inputs with more and more leading 0s in their path, until after around $\log^2 n$ queries the adversary can force B_r to fail.

A successful design

Our faux-deterministic algorithm works similarly to the fully-adaptive tree in the previous paragraph, but instead of forking on each bit we query, we fork on an OR of a batch of $\log n$ queried bits. In other words, the decision tree looks like a tree (of depth $\log n$) of ORs (of width $\log n$). We return the last 1 seen. The OR blocks are randomized when r is random.

This works because the OR function has very low sensitivity: on most inputs, the OR blocks will all evaluate to 1, and the path taken down the tree will simply be rightmost path. The adversary will be unable to find any input x on which the path of $B_r(x)$ is not the rightmost one until it has made some queries to B_r. Even then, it will only know bits in the rightmost block, and will be unable to find inputs x on which the path of $B_r(x)$ is not either the rightmost path, or the rightmost path with a left turn only at the very end. Indeed, after T queries to B_r, we show that the adversary can only find inputs x for which the path of $B_r(x)$ leads to the rightmost T leaves of the tree. To get the algorithm B_r to fail, however, requires all the OR blocks to evaluate to 0 (so that B_r never finds any 1 in the input at all); this means finding an input x for which the path of $B_r(x)$ is the leftmost path (consisting of all 0s for the OR blocks). Since there are 2^d leaves for depth d, the adversary requires exponentially many queries to B_r (where "exponential" is relative to the efficiency parameter $\log n$).

The main innovation here was the "tree of ORs" design: this lets the algorithm avoid being local (there are $\exp(d)$ many indices it might query for some input, where d is the number of queries it makes) while also letting it avoid having its path be sensitive to bit flips.

Quantum lower bound

Our faux-deterministic algorithm also works when the adversary is quantum, and makes superposition queries to B_r. Our proof of the quantum lower bound is based on a hybrid argument [2]; if Q is the adversary's quantum algorithm, we set $Q_0 = Q$ and make a sequence $Q_1, Q_2, \ldots$ where Q_1 is the quantum algorithm with the first query replaced by a "fake" quantum query, Q_2 is the quantum algorithm with the first two queries replaced by fakes, and so on, until all queries are replaced by fakes. We show that the final states (before measurement) are close between Q_t and Q_{t+1}, meaning that the fake initial queries are sufficiently accurate. This technique lets us show inductively that the first query of the real quantum algorithm must place extremely small amplitude on the paths of the tree that are not the rightmost path; the second query must place extremely small amplitude on the paths of the tree that are not the rightmost or second-rightmost paths; and so on.

2 Preliminaries

2.1 Notation

General notation

For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, ..., n\}$. By $\mathrm{poly}(n)$ and $\mathrm{polylog}(n)$, we denote the family of all functions growing at most polynomially and poly-logarithmically, that is $f(n) = O(n^c)$ and $f(n) = O(\log^c n)$ for some constant c, respectively. We also define $\mathrm{negl}(n)$ to denote the family of negligible functions; that is, $f(n) = \mathrm{negl}(n)$ if for every constant $c > 0$, we have $f(n) = O(n^{-c})$.

For a binary string $x \in \{0,1\}^n$, we use $|x| = |\{i : x_i = 1\}|$ to denote its Hamming weight, and for a set $B \subset [n]$ we use x^B to denote the string obtained from x by flipping precisely the bits in B. For a single index $j \in [n]$, we write x^j to denote $x^{\{j\}}$.

For a distribution $\mathcal{D}$, we use the notation $x \sim \mathcal{D}$ to indicate that x is sampled according to $\mathcal{D}$, and use $x \in \mathcal{D}$ to denote that x lies in the support of $\mathcal{D}$. Let X be a random variable and let $\hat{X}$ be a fixed value in its range. With a slight abuse of notation, we may write $\hat{X}$ to denote the event $\{X = \hat{X}\}$ whenever X is clear from the context. The same convention applies to $\Pr[\cdot|\hat{X}]$ and $\mathbb{E}[\cdot|\hat{X}]$.

Quantum computing

We will use the standard *bra-ket* notation for quantum computing. For a *quantum register* X of n qubits, we denote *quantum states* as unit vectors $|\phi\rangle_X$ in Hilbert space $\mathcal{H}_X = (\mathbb{C}^{2^n}, \langle\rangle)$, where $\langle\rangle$ is the standard complex inner product. We use the notation $|x\rangle_X$ for a binary string $x \in \{0,1\}^n$ to denote the corresponding elementary vector having a one in the entry x. We denote a *multi-register* system as $X = X_1 \dots X_k$, where $\mathcal{H}_X = \mathcal{H}_{X_1} \otimes \dots \otimes \mathcal{H}_{X_k}$. We may omit X whenever it is obvious from the context.

A *unitary transformation* U on register X is a linear transformation preserving ℓ_2-norm; that is, $\| U |\phi\rangle_X \|_2 = \| |\phi\rangle_X \|_2$ for all $|\phi\rangle_X$. With a slight abuse of notation, we additionally use U to denote unitary transformations $U \otimes I$ and $I \otimes U$ on larger registers XY and YX, respectively.

A *measurement* corresponding to the *computational basis* $\{|x\rangle_X \, | x \in \{0,1\}^n\}$ of register X on state $|\phi\rangle_{XY} = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle_X |\phi_x\rangle_Y$ collapses the state to $|x\rangle_X |\phi_x\rangle_Y$ with probability $|\alpha_x|^2$. The measurement of the second register would be defined similarly. We can also consider measurements in a more general setting of *projection operators*. Let $\mathcal{M} = \{\Pi_1, \dots, \Pi_k\}$ be a set of orthogonal projection matrices, meaning that $\Pi_i = \Pi_i^\dagger = \Pi_i^2$ and $\sum_{i=1}^k \Pi_i = I$. The outcome $i \in [k]$ of measurement $\mathcal{M}$ on state $|\phi\rangle$ occurs with probability $\| \Pi_i |\phi\rangle \|_2^2$, with the resulting state being $\frac{\Pi_i |\phi\rangle}{\| \Pi_i |\phi\rangle \|_2}$.

2.2 Search problems and query complexity measures

Let $\Sigma = (\Sigma_n)$ be a family of finite and non-empty sets. A *search problem* is a relation $S \subset \bigcup_{n\in\mathbb{N}} (\{0,1\}^n \times \Sigma_n)$. For $n \in \mathbb{N}$, we use $S_n := S \cap (\{0,1\}^n \times \Sigma_n)$ to denote the restriction of the problem to inputs of length n. For $x \in \{0,1\}^n$, the set of corresponding *certificates* (valid outputs) is $S_n(x) := \{y \in \Sigma_n : (x,y) \in S_n\}$; when n is clear from context we write $S(x)$ for $S_n(x)$. The *domain* and *range* of S are defined as $\mathrm{dom}(S) := \{x \in \{0,1\}^* : \exists y, (x,y) \in S\}$ and $\mathrm{ran}(S) := \{y : \exists x, (x,y) \in S\}$, respectively.

The problem S is called *total* if $\mathrm{dom}(S) = \{0,1\}^*$. Moreover, we say S is *Boolean* if $\mathrm{ran}(S) \subseteq \{0,1\}$, and *single-valued* if $|S(x)| = 1$ for every $x \in \mathrm{dom}(S)$. A Boolean single-valued problem is also called a *decision problem*.

Deterministic query complexity

A deterministic decision tree $T = (T_n)$ is a sequence of rooted binary trees T_n for input length $n \in \mathbb{N}$ where each internal node is labeled with an index $i \in [n]$, and has a right and a left child, each labeled as one of the possible values in $\{0,1\}$. Further, each leaf of T_n is labeled with a potential output in Σ_n.

For input $x \in \{0,1\}^n$, the value $T[x] := T_n[x]$ is computed as follows. Beginning with the root, if the current node is a leaf, output its label. Otherwise, the current node is labeled by some index $i \in [n]$. Query x_i and continue the computation from the corresponding child until a leaf is reached. The (deterministic) query complexity of T, denoted as $\mathsf{DEPTH}(T_n)$, is defined as its depth which is the maximum number of queries made on the worst-case input.

The *deterministic query complexity* of a search problem S, denoted by $D(S) = D(S_n)$, is the minimum depth of a decision tree T computing S, i.e., satisfying $T[x] \in S(x)$ for all $x \in \mathrm{dom}(S)$. We define the class FP to be the set of all search problems S such that $D(S_n) = \mathrm{polylog}(n)$.

Nondeterministic query complexity

A *verifier* V for a search problem S is a family of deterministic decision trees $V = (V_{n,y})$, where for each input length n and output $y \in \Sigma_n$, the tree $V_{n,y}$ outputs $V_{n,y}[x] = 1$ if $y \in S(x)$ and $V_{n,y}[x] = 0$ otherwise. We define the *nondeterministic query complexity* of V, denoted by $\mathsf{DEPTH}(V) = \mathsf{DEPTH}(V_n)$ as $\max_{y \in \Sigma_n} \mathsf{DEPTH}(V_{n,y})$.

The *nondeterministic query complexity* of a search problem, denoted by $N(S)$, is defined as the minimum depth of a verifier V for S. We define the class FNP to be the set of all search problems S such that $N(S_n) = \mathrm{polylog}(n)$.

Randomized and pseudodeterministic query complexity

A randomized decision tree $\mathcal{T} = (\mathcal{T}_n)$ is a sequence where each $\mathcal{T}_n$ is a probability distribution over deterministic decision trees of input length n. The query complexity of $\mathcal{T}$, denoted by $\mathsf{DEPTH}(\mathcal{T})$ is defined as the maximum depth over all decision trees T in the support of $\mathcal{T}_n$.

For an error parameter $\epsilon = \epsilon(n) \in (0, 1)$, the *randomized query complexity* of a search problem S, denoted by $R_\epsilon(S)$ is defined as the minimum depth of a randomized decision tree $\mathcal{T}$ such that $\Pr[\mathcal{T}_n[x] \notin S(x)] \leq \epsilon$ for all $x \in \mathrm{dom}(S_n)$. Whenever ϵ is omitted, we assume $\epsilon = 1/3$. We define the class FBPP to be the set of all search problems S such that $R(S_n) = \mathrm{polylog}(n)$.

Additionally, the *pseudodeterministic query complexity* of a search problem S, denoted by $D^{\mathrm{pseudo}}_\epsilon(S)$, is defined as the minimum depth of a randomized decision tree $\mathcal{T}$ such that for all $x \in \mathrm{dom}(S_n)$ there exists a canonical output $f(x) \in S(x)$ where $\Pr[\mathcal{T}_n[x] \neq f(x)] \leq \epsilon$. We define the class $\mathsf{PseudoP}$ to be the set of all search problems S such that $D^{\mathrm{pseudo}}_{1/3}(S_n) = \mathrm{polylog}(n)$.

Quantum query complexity

To define the quantum version of decision trees, we switch to the standard circuit model with query access. For a string $x \in \{0,1\}^n$, we define the *oracle unitary* O^x on *index* and *query* registers I and Q as the transformation $O^x |i\rangle_I |b\rangle_Q := |i\rangle_I |x_i \oplus b\rangle_Q$ for $i \in [n]$ and $b \in \{0,1\}$. A *quantum decision tree* $\mathcal{Q} = (\mathcal{Q}_n)$ is a sequence where each $\mathcal{Q}_n$ consists of unitaries $(U_0, \ldots, U_d)$ acting on index register I, query register Q, as well as an arbitrarily large work register W. We define the query complexity of $\mathcal{Q}$, denoted by $\mathsf{DEPTH}(\mathcal{Q}_n) := d(n)$. For an input $x \in \{0,1\}^n$, the execution of the algorithm $\mathcal{Q}[x]$ is defined as the measurement outcome of the work register (identifying a fixed set of computational-basis states with the elements of Σ_n) in the final state $|\psi\rangle = U_d O^x U_{d-1} O^x \ldots O^x U_0 |0\rangle_I |0\rangle_Q |0\rangle_W$.

Similar to the randomized case, the *quantum query complexity* of a search problem S, denoted by $Q_\epsilon(S)$ is defined as the minimum depth $d(n)$ of a quantum algorithm satisfying $\Pr[\mathcal{Q}[x] \notin S(x)] \leq \epsilon$ for all $x \in \mathrm{dom}(S)$. The class FBQP is defined as the set of search problems having query complexity $Q_{1/3}(S) = \mathrm{polylog}(n)$.

Block sensitivity

We recall the definition of block sensitivity for completeness. Let $f : \mathrm{dom}(f) \to \Sigma$ be a possibly partial function with $\mathrm{dom}(f) \subseteq \{0,1\}^*$, and let $x \in \mathrm{dom}(f) \cap \{0,1\}^n$. The *block sensitivity* of f on input x, denoted by $\mathrm{bs}(f,x)$, is defined as the maximum number b of disjoint blocks $B_1, \ldots, B_b \subset [n]$ of indices such that $x^{B_j} \in \mathrm{dom}(f)$ and $f(x) \neq f(x^{B_j})$ for all $j \in [b]$. The block sensitivity of f is defined as $\mathrm{bs}(f) = \max_{x \in \mathrm{dom}(f)} \mathrm{bs}(f,x)$.

2.3 Formal definition of faux determinism

In this section, we formalize the informal definition of faux determinism given in Section 1 using our query complexity notation.

For a search problem S and a decision tree T purporting to solve S on inputs of length n, we say $x \in \mathrm{dom}(S_n)$ is an *adversarial input* for T whenever $T[x] \notin S_n(x)$. We define the classical version of faux determinism as follows.

▶ **Definition 8** (faux determinism in query model)**.** *Let S be a search problem, $q = q(n) \in \mathbb{N}$ be a query upper bound, $\epsilon = \epsilon(n) \in [0,1)$ be an error parameter. We say that a distribution $\mathcal{T} = (\mathcal{T}_n)$ over decision trees is a* (q,ϵ)-faux-deterministic algorithm *for S if, for all $n \in \mathbb{N}$ and any deterministic*[4] *adversary A making fewer than $q(n)$ black-box queries of the form $x \in \mathrm{dom}(S_n)$ to a random decision tree $T \sim \mathcal{T}_n$, the probability that A finds an adversarial input is at most $\epsilon(n)$; i.e.,*

$$\Pr_{T \sim \mathcal{T}_n}[x \leftarrow A[T]; T[x] \notin S(x)] \leq \epsilon(n).$$

Similar to randomized decision trees, the complexity of a faux-deterministic algorithm is defined as its worst-case complexity over all the possible trees in the distribution:

$$\mathsf{DEPTH}(\mathcal{T}) := \max_{T \in \mathcal{T}} \mathsf{DEPTH}(T).$$

Next, we generalize the notion of faux-deterministic algorithms for quantum adversaries that are given query access to the oracle unitary

$$O^T |x\rangle_I |z\rangle_Z = \begin{cases} |x\rangle_I |T[x] \oplus z\rangle_Z & x \in \mathrm{dom}(S), \\ |x\rangle_I |z\rangle_Z & \text{otherwise}, \end{cases}$$

where we assume a fixed binary encoding of the output alphabet Σ_n, so that $T[x] \oplus z$ is well-defined.

▶ **Definition 9.** *We say that the distribution $\mathcal{T} = (\mathcal{T}_n)$ is a* quantum-secure (q,ϵ) faux-deterministic *algorithm for S whenever any quantum adversary $\mathcal{Q}$ making fewer than $q(n)$ many black-box queries to the oracle unitary O^T finds an adversarial input with probability at most $\epsilon(n)$; that is,*

$$\Pr[T \sim \mathcal{T}, x \leftarrow \mathcal{Q}[T]; x \in \mathrm{dom}(S_n) \text{ and } T[x] \notin S(x)] \leq \epsilon(n).$$

[4] In the above definition, the existence of deterministic and randomized adversaries are equivalent; i.e., if a randomized adversary succeeds with probability $\epsilon(n)$, we can choose the best deterministic adversary for each input length n to obtain a deterministic adversary that succeeds with probability at least $\epsilon(n)$ as well.

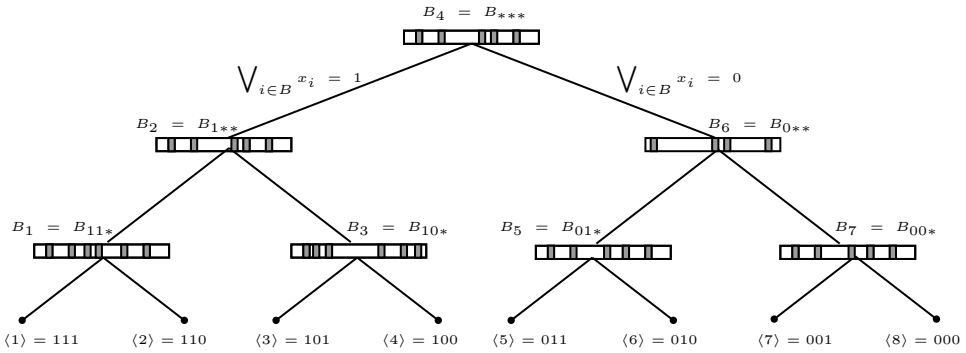


Figure 1 The Tree of ORs construction with depth $d = 3$. Extending the order $1 \prec * \prec 0$ induces a linear order on the leaves $\langle 1 \rangle, \ldots, \langle 8 \rangle$ and blocks $B_1, \ldots, B_7$..

We define FauxP as the class of search problems S for which there exists a distribution $\mathcal{T}$ over decision trees of depth polylog(n) such that, for every polynomial p, $\mathcal{T}$ is $(p(n), \text{negl}(n))$-faux-deterministic. The class QFauxP is defined analogously, with security against quantum adversaries.

As mentioned in Section 1.3, our main technical step is designing a faux-deterministic algorithm for a FBPP $\cap$ FNP-complete problem FIND1 defined as follows.

▶ **Definition 10** (FIND1). *Let $n \in \mathbb{N}$. We define* FIND1_n *to be the following search problem: given an input $x \in \{0,1\}^n$ with Hamming weight $|x| \geq n/2$, find an index i such that $x_i = 1$; i.e.,* $\text{FIND1}_n := \{(x, i) \in \{0,1\}^n \times [n] |\ |x| \geq \frac{n}{2} \text{ and } x_i = 1\}$.

3 Faux-deterministic algorithm for FIND1

In this section, we present our faux-deterministic algorithm for the FIND1 problem. More specifically, in Section 3.1 we formally introduce the *Tree of ORs* algorithm and establish a few of its structural properties required for our analysis. We then analyze the algorithm against deterministic adversaries in Section 3.2 and against quantum algorithms in Section 3.3.

3.1 Tree of ORs construction

As previously discussed in Section 1.3, the construction of the Tree of ORs algorithm will be based on a complete binary tree T of depth d. For the rest of this section, we identify each vertex with a string $z \in \{0,1\}^r *^{d-r}$ for some $0 \leq r \leq d$: the root is $*^d$ $(r = 0)$, each internal vertex has the form $b_1 \cdots b_r *^{d-r}$ with $0 \leq r < d$, and the leaves are exactly $\{0,1\}^d$ $(r = d)$. For an internal vertex $b_1 \cdots b_r *^{d-r}$, its left and right children are $b_1 \cdots b_r 1 *^{d-r-1}$ and $b_1 \cdots b_r 0 *^{d-r-1}$, respectively.

▶ **Definition 11** (Tree of ORs). *Let $n, w = w(n), d = d(n) \in \mathbb{N}$, and let T be a binary tree of depth d. For each internal node $z = b_1 \ldots b_k *^{d-k}$ of T with $0 \leq k < d$, we assign an independent and uniformly random block of indices $B_z \subset [n]$ of size $|B_z| = w$.*

Then, on input $x \in \{0,1\}^n$, the algorithm begins at root node $z = *^d$ and proceeds recursively as follows: on an internal node $z = b_1 \ldots b_k *^{d-k}$ with $k < d$: if any of the indices $i \in B_z$ is a 1-index $(x_i = 1)$, the evaluation of the node is defined to be $B_z(x) := 1$, and we move to the left child $b_1 \ldots b_k 1 *^{d-k-1}$. Otherwise, the logical OR $(\vee)$ of indices in B_z is defined to be $B_z(x) := 0$ and we move to the right child $b_1 \ldots b_k 0 *^{d-k-1}$.

Eventually, the algorithm arrives at a leaf node $\ell \in \{0,1\}^d$. If this is the right-most leaf 0^d, no 1-indices are found and we err. Otherwise, there exists a unique $k \in \{0,\dots,d-1\}$ such that $\ell = b_1 \dots b_k 10^{d-k-1}$. As a result, the last node where a 1-index was found is an internal node $z^* = b_1 \dots b_k *^{d-k}$, and we output the smallest index $i \in B_{z^*}$ with $x_i = 1$.

For each input length n, our distribution $\mathcal{T}_n$ is a uniform distribution over all possible such trees with depth $d(n)$ and width $w(n)$. We may drop n whenever it is obvious from the context.

3.1.1 Linear ordering

To help analyze the algorithm, we extend the ordering $1 \preceq * \preceq 0$ to a linear order on all internal and leaf labels $z \in \{0,1\}^r *^{d-r}$ with $0 \le r \le d$ by lexicographic order; i.e., $1^d \prec 1^{d-1}* \prec 1^{d-1}0 \prec \cdots \prec 0^{d-1}1 \prec 0^{d-1}* \prec 0^d$. This ordering induces a natural left-to-right order on the leaves of the tree. For $\ell \in [2^d]$ let $\langle \ell \rangle$ denote the ℓ-th leaf in this order. Also, for $\ell \in [2^d - 1]$, we use B_ℓ to denote the ℓ-th block in this order (see Figure 1).

Intuitively, when reaching leaf $\langle \ell \rangle$ the output of the algorithm will be from the block B_ℓ. The following proposition formalizes this correspondence.

▶ **Proposition 12.** *Let $\ell \in [2^d - 1]$, and for label z, let z^+ and z^- denote the successor and predecessor in the linear order defined above, whenever it exists.*

- **i.** *between any two consecutive leaves $\langle \ell \rangle$ and $\langle \ell + 1 \rangle$ there exists a unique internal label $z = \langle \ell \rangle^+ = \langle \ell + 1 \rangle^-$. Conversely, each internal node $z \in \{0,1\}^r *^{d-r}$ is between two consecutive leaves $z^+ \succ z \succ z^-$.*
- **ii.** *the ℓ-th block in this order, denoted by B_ℓ, is exactly the block corresponding to label $\langle \ell \rangle^+$; i.e. $B_\ell = B_{\langle \ell \rangle^+}$.*
- **iii.** *a tree $T \in \mathcal{T}$ fails to output a 1-index if and only if it reaches the right-most leaf $\langle 2^d \rangle$. Moreover, for $\ell < 2^d$ the tree outputs a 1-index from the block B_ℓ if and only if it reaches the leaf $\langle \ell \rangle$.*

Proof. For (i), let $\langle \ell \rangle = b_1 \dots b_i 10^{d-i-1}$ for some $i \in \{0,\dots,d-1\}$. Then, by definition of the linear order, $\langle \ell + 1 \rangle = b_1 \dots b_i 01^{d-i-1}$. So, the only possible string in between is $z = b_1 \dots b_i *^{d-i}$, which is exactly equal to $z = \langle \ell + 1 \rangle^- = \langle \ell \rangle^+$. Similarly, each internal node $z = b_1 \dots b_i *^{d-i}$ is between two consecutive leaves $z^- = b_1 \dots b_i 10^{d-i-1}$ and $z^+ = b_1 \dots b_i 01^{d-i-1}$. Part (ii) is a direct consequence of (i). For (iii), recall that the algorithm branches left if and only if a 1-index is found. So, for a leaf $\langle \ell \rangle = b_1 \dots b_i 10^{d-i-1}$, the last block on which a 1-index is found is exactly $\langle \ell \rangle^+ = b_1 \dots b_i *^{d-i}$. ◀

3.1.2 Hybrid Lemma

We now prove the central property of the algorithm for security against adversaries. For $x \in \{0,1\}^n$ where $|x| \ge n/2$ and tree $T \in \mathcal{T}$, let $\ell_T(x)$ denote the leaf reached when executing T on x. Then, given a tree $T \sim \mathcal{T}$, the main goal of an adversary is to find such an input x that reaches the right-most leaf $\ell_T(x) = \langle 2^d \rangle$.

Assume that the adversary has already found the first $\ell - 1$ blocks $B_1 = \hat{B}_1, \dots, B_{\ell-1} = \hat{B}_{\ell-1}$. As a result, the adversary can choose a string x with $x_i = 0$ for every $i \in \bigcup_{j=1}^{\ell-1} \hat{B}_j$, which by Proposition 12 implies that $\ell_T(x) \succeq \langle \ell \rangle$. In the following lemma, we argue that if the adversary has no information about the remaining blocks $B_{\ell+1}, \dots, B_{2^d-1}$, then any such input x reaches exactly the next leaf $\ell_T(x) = \langle \ell \rangle$ with high probability over the choice of T; i.e., the adversary cannot "skip" any of the blocks.

▶ **Lemma 13.** *Let $\mathcal{T}$ be the distribution given in Definition 11 on inputs of length n with width w and depth d. Then, for any input $x \in \{0,1\}^n$ with $|x| \geq n/2$, for $\ell \in [2^d]$, and for any sequences of subsets $\hat{B}_1, \ldots, \hat{B}_{\ell-1} \subset [n]$ of size w, we have*

$$\Pr_{T\sim\mathcal{T}_n}\left[\ell_T(x) \preceq \langle\ell\rangle \middle| \hat{B}_1, \ldots, \hat{B}_{\ell-1}\right] \geq (1-\frac{1}{2^w})^d.$$

To prove the lemma, we first provide a necessary condition for $\ell_T(x) \preceq \langle\ell\rangle$, and then prove that this condition is satisfied.

For $\ell \in [2^d-1]$, and $0 \leq r < d$, we call $y \in \{0,1\}^r *^{d-r}$ a *1-prefix* of $\langle\ell\rangle$ whenever there exists $\hat{y}, \hat{z} \in \{0,1\}^*$ such that $\langle\ell\rangle = \hat{y}1\hat{z}$ and $y = \hat{y}*^{d-|\hat{y}|}$; that is, internal node y is in the path to leaf $\langle\ell\rangle$ with evaluation of B_y being 1.

▷ Claim 14. Assume that for every 1-prefix y of $\langle\ell\rangle$ we have $B_y(x) = 1$. Then $\ell_T(x) \preceq \langle\ell\rangle$.

Proof. For the sake of contradiction, suppose $\ell_T(x) \succ \langle\ell\rangle$, and let $\hat{y}$ be the largest common prefix of $\langle\ell\rangle$ and $\ell_T(x)$ with length m. Since $\ell_T(x) \neq \langle\ell\rangle$, we know that $\langle\ell\rangle$ and $\ell_T(x)$ differ in at least one index. Additionally, we have $\ell_T(x) \succ \langle\ell\rangle$. Therefore, we can write $\langle\ell\rangle$ and $\ell_T(x)$ as $\langle\ell\rangle = \hat{y}1\hat{z}$ and $\ell_T(x) = \hat{y}0\hat{z}'$. Now consider the 1-prefix $y = \hat{y}*^{d-m}$ of $\langle\ell\rangle$; since $\hat{y}$ is also a prefix of $\ell_T(x)$, the execution of tree T on x must have visited the internal block B_y. Also, since $\ell_T(x)$ extends $\hat{y}$ by a 0 symbol, the subset B_y must not have found a 1-index in the input x. Therefore, $B_y(x) = 0$, which is a contradiction to the assumption of the claim. ◁

Proof of Lemma 13. We prove that with high probability for any $\langle\ell\rangle$ the condition of the Claim 14 holds. Let y be an arbitrary 1-prefix of $\langle\ell\rangle$ where $\langle\ell\rangle = \hat{y}1\hat{z}$ and $y = \hat{y}*^{d-|\hat{y}|}$. By the defined linear order, we have $y \succ \langle\ell\rangle$ as $* \succ 1$. Therefore, by the natural ordering of the leaves and Proposition 12 we have $y \succ \langle\ell\rangle \succ \langle\ell\rangle^- = \langle\ell-1\rangle^+ \succ \langle\ell-2\rangle^+ \succ \cdots \succ \langle 1\rangle^+$. So, the block variable B_y is not among the variables $B_1, \ldots, B_{\ell-1}$, and the set B_y is sampled independent of the first $\ell-1$ sets.

Let $y_1, \ldots, y_r$ be all the 1-prefixes of $\langle\ell\rangle$. Note that since $\langle\ell\rangle \in \{0,1\}^d$ we have $r \leq d$. Consequently,

$$\begin{aligned}
\Pr_{T\sim\mathcal{T}_n}\left[\ell_T(x) \preceq \langle\ell\rangle \middle| \hat{B}_1, \ldots, \hat{B}_{\ell-1}\right] &\geq \Pr_{T\sim\mathcal{T}_n}\left[B_{y_1}(x)=1, \ldots, B_{y_r}(x)=1 \middle| \hat{B}_1, \ldots, \hat{B}_{\ell-1}\right] \\
&= \Pr_{T\sim\mathcal{T}_n}\left[B_{y_1}(x)=1, \ldots, B_{y_r}(x)=1\right] \\
&= \Pr_{T\sim\mathcal{T}_n}\left[B_{y_1}(x)=1\right] \ldots \Pr_{T\sim\mathcal{T}_n}\left[B_{y_r}(x)=1\right] \\
&\geq (1-\frac{1}{2^w})^r \\
&\geq (1-\frac{1}{2^w})^d,
\end{aligned}$$

where the first line is by Claim 14, second and third lines are by independence of blocks, and the fourth line is due to $|x| \geq n/2$. ◀

3.2 Security against classical adversaries

In this section, we will prove the security of the Tree of ORs construction against classical adversaries.

▶ **Theorem 15.** *Let $\mathcal{T} = (\mathcal{T}_n)$ be the distribution in Definition 11 with width and depth parameters $w = w(n)$ and $d = d(n)$. Then, $\mathcal{T}_n$ is a $(2^d, d \cdot 2^{d-w})$-faux-deterministic algorithm of complexity $w \cdot d$ for the* FIND1 *problem.*

Proof. Clearly, the tree has query complexity $w \cdot d$. So, let A be an arbitrary deterministic adversary that has query access to a uniformly randomly chosen tree $T \sim \mathcal{T}_n$, and queries q inputs

$$x_{(1)}, x_{(2)}, \ldots, x_{(q)}$$

with $q \leq 2^d - 1$. Without loss of generality, we assume that the adversary outputs the string $x_{(q)}$ as its answer. Also, for the sake of simplicity, we assume that the oracle provides additional information to the adversary as well. More specifically, when querying $x_{(k)}$ with $\ell_T(x_{(k)}) = \langle j \rangle$ the oracle returns

- all of the blocks $B_1, \ldots, B_j$,
- and all of the first k blocks $B_1, \ldots, B_k$.

We can now lower bound the probability p that the algorithm does not err on $x_{(q)}$, i.e., $\ell_T(x_{(q)}) \neq 0^d$, as follows.

$$\begin{aligned}
p &:= \Pr_{T \sim \mathcal{T}_n} \left[\ell_T(x_{(q)}) \neq 0^d\right] \\
&\geq \Pr_{T \sim \mathcal{T}_n} \left[\ell_T(x_{(1)}) \preceq \langle 1 \rangle, \ldots, \ell_T(x_{(q)}) \preceq \langle q \rangle\right] \\
&= \prod_{\ell=1}^{q} \Pr_{T \sim \mathcal{T}_n} \left[\ell_T(x_{(\ell)}) \preceq \langle \ell \rangle \middle| \ell_T(x_{(1)}) \preceq \langle 1 \rangle, \ldots, \ell_T(x_{(\ell-1)}) \preceq \langle \ell - 1 \rangle\right] \\
&= \prod_{\ell=1}^{q} p_\ell,
\end{aligned}$$

where p_ℓ is the probability that the ℓ-th query reaches at most the ℓ-th leaf conditioned on the assumption that all of the previous $\ell - 1$ queries have resulted in a leaf before $\langle \ell \rangle$. Therefore, each of these queries has revealed at most the first $\ell - 1$ blocks $B_1 = \hat{B}_1, \ldots, B_{\ell-1} = \hat{B}_{\ell-1}$. Additionally, we assumed that after $\ell - 1$ queries, the oracle returns all of the first $\ell - 1$ blocks. Hence, $x_{(\ell)}$ depends only on the first ℓ blocks and we can write

$$p_\ell = \Pr_{T \sim \mathcal{T}_n} \left[\ell_T(x_{(\ell)}) \preceq \langle \ell \rangle \middle| \hat{B}_1, \ldots, \hat{B}_{\ell-1}\right] \geq (1 - \frac{1}{2^w})^d,$$

where we have used the Lemma 13 in the last inequality for $\ell \in [q]$. Therefore,

$$p = \prod_{\ell=1}^{q} p_\ell \geq \prod_{\ell=1}^{q} (1 - \frac{1}{2^w})^d \geq (1 - \frac{1}{2^w})^{dq} \geq 1 - \frac{dq}{2^w} \geq 1 - \frac{d2^d}{2^w}.$$ ◀

▶ **Corollary 16.** *Let $t = t(n) \in \mathbb{N}$ be an increasing function. The* FIND1 *problem has a $(2^{\frac{\sqrt{t}}{4}}, 2^{-\frac{\sqrt{t}}{4}})$-faux-deterministic algorithm of total depth t. Specifically, for any $\epsilon > 0$ and $t = \log^{2+\epsilon}(n)$, we have a $(\text{quasi-poly}(n), \text{negl}(n))$-faux-deterministic algorithm of depth $\text{polylog}(n)$.*

Proof. Letting $w = \frac{3}{4}\sqrt{t}$ and $d = \frac{1}{4}\sqrt{t}$ we obtain a tree with total depth $\frac{3}{16}t < t$. By Theorem 15, any attacker with less than $2^{\frac{1}{4}\sqrt{t}}$ queries will succeed at finding an invalid output with probability at most $\frac{d2^d}{2^w} \leq 2^{2d-w} = 2^{-\frac{1}{4}\sqrt{t}}$. For the second part, we have $2^{\frac{\sqrt{\log^{2+\epsilon}(n)}}{4}} = \text{quasi-poly}(n)$ and $2^{-\frac{\sqrt{\log^{2+\epsilon}(n)}}{4}} = \text{negl}(n)$. ◀

3.3 Security against quantum adversaries

In this section, we will prove the security of the Tree of ORs construction against quantum adversaries.

▶ **Theorem 17.** *Let $\mathcal{T} = (\mathcal{T}_n)$ be the distribution in Definition 11 with width and depth parameters $w = w(n)$ and $d = d(n)$. Then, $\mathcal{T}$ is a quantum-secure $(2^d - 1, \sqrt{d} \cdot 2^{d+2-w/2})$-faux-deterministic algorithm of depth $w \cdot d$ for* FIND1.

Proof. Let $\mathcal{Q} = (U_0, U_1, \ldots, U_q)$ be a quantum algorithm that makes q queries to a randomly sampled decision tree $T \sim \mathcal{T}_n$ to obtain $|\psi\rangle = U_q O^T \ldots U_1 O^T |\psi_0\rangle$, where $|\psi_0\rangle = U_0 |0\rangle$ is the initial state, and O^T is defined as

$$O^T |x\rangle_I |z\rangle_Z = \begin{cases} |x\rangle_I |T[x] \oplus z\rangle_Z & |x| \geq n/2, \\ |x\rangle_I |z\rangle_Z & \text{otherwise.} \end{cases}$$

Then, the algorithm performs a measurement on the final state $|\psi\rangle$ to obtain the output string $x = \mathcal{Q}[T] \in \{0,1\}^n$.

Let $B_1, \ldots, B_{2^d-1}$ denote the corresponding query blocks in the tree T according to Proposition 12. Similar to the Proof of Theorem 15, for the sake of simplicity, we assume that after the k-th query, the adversary has access to all of the first k blocks $B_1, \ldots, B_k \subset [n]$.

Hybrid argument. We are going to use a hybrid argument by constructing a series of quantum algorithms $\mathcal{Q}^0 = \mathcal{Q}$, $\mathcal{Q}^1$, …, and $\mathcal{Q}^q = \mathcal{Q}'$ as follows: for $k \in [q-1]$, the algorithm $\mathcal{Q}^k$ is defined by replacing the first k queries to O^T with the sequence of simulated queries $O^{T^1}, \ldots, O^{T^k}$ while maintaining the remaining $q - k$ queries as the original oracle O^T. Formally, we have the construction in Algorithm 1.

Algorithm 1 Construction of the hybrid $\mathcal{Q}^k$.

Defining the partial simulation T^k (see Figure 2). Assume that the first k blocks $\hat{B}_1, \ldots, \hat{B}_k$ and an input $x \in \{0,1\}^n$ are given. Then, $T^k[x]$ simulates the structure of T, beginning at the root node where at each step a block B_ℓ for some $\ell \in [2^d - 1]$ is queried.

- If the queried block is one of the given blocks $\ell \leq k$, we execute $\hat{B}_\ell[x]$ and traverse left ($\hat{B}_\ell[x] = 1$) or right ($\hat{B}_\ell[x] = 0$) accordingly. Otherwise, $\ell > k$, and we traverse left as if $B_\ell[x] = 1$.
- Finally, the algorithm reaches a leaf node. Let $L := \{\ell \in [k] : \hat{B}_\ell[x] = 1\}$. If $L = \emptyset$, we output an arbitrarily fixed index $i \in [n]$. Otherwise, L has a largest element ℓ^*, and we output an index $j \in \hat{B}_{\ell^*}$ satisfying $x_j = 1$.

The simulated unitary O^{T^k}. Naturally, we define the unitary O^{T^k} as

$$\forall x \in \{0,1\}^n \qquad O^{T^k} |x\rangle_I |z\rangle_Z = \begin{cases} |x\rangle_I |T^k[x] \oplus z\rangle_Z & |x| \geq \frac{n}{2}, \\ |x\rangle_I |z\rangle_Z & \text{otherwise.} \end{cases}$$

The hybrid $\mathcal{Q}^k$. Given blocks $\hat{B}_1, \ldots, \hat{B}_k$, the algorithm $\mathcal{Q}^k$ evolves the initial state $|\psi_0\rangle$ as follows.

- **Simulation phase:** For each $t \in [k]$, the algorithm replaces the t-th query with a simulated query to O^{T^t}. That is, we let $|\psi_0^k\rangle = |\psi_0\rangle$, and recursively define $|\psi_t^k\rangle := U_t O^{T^t} |\psi_{t-1}^k\rangle$ for $t = 1, \ldots, k$.
- **Faithful phase:** For $t = k+1, \ldots, q$, the algorithm proceeds with the remaining queries to the real oracle O^T: $|\psi_t^k\rangle := U_t O^T |\psi_{t-1}^k\rangle$.
- **Measurement:** finally, we perform the measurement in the computational basis on $|\psi_q^k\rangle$ to obtain the output string $x = \mathcal{Q}^k[T] \in \{0,1\}^n$.

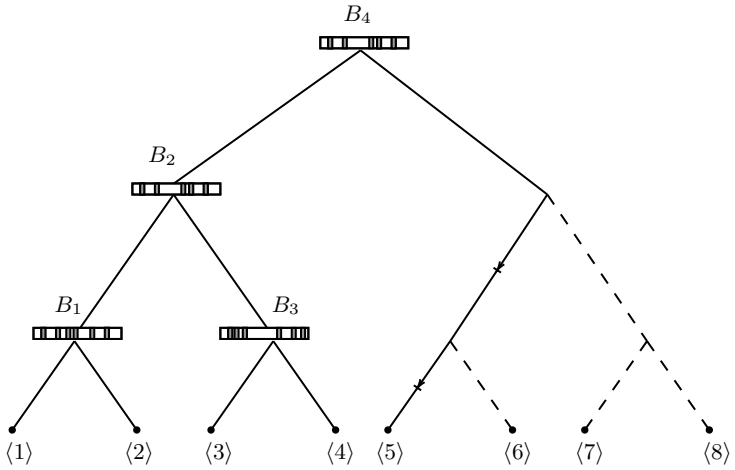


Figure 2 The simulated tree T^4 from T with the knowledge of B_1, B_2, B_3, and B_4. The quantum algorithm simulating them only computes the nodes of the tree for which the blocks are known. For instance, for any input x with $\ell_T(x) \preceq \langle 4\rangle$ the computed values of T and T^4 are equal; $T[x] = T^4[x]$.

Proof overview. Before proceeding into the details of the proof, we give an overview of the proof strategy. Let $|\psi\rangle := |\psi_q^0\rangle$ and $|\psi'\rangle := |\psi_q^q\rangle$ be the final states before measurement of the initial and final algorithms $\mathcal{Q}$ and $\mathcal{Q}'$. For a fixed tree $T \in \mathcal{T}$, we also define the projection $\Pi_1(T)$ to be the projection to states having input registers $|x\rangle_I$, where x is an *adversarial input* for T. I.e., valid inputs $|x| \geq n/2$ on which the tree T fails to find a 1-index. Similarly, we can define $\Pi_0(T)$ to be the projection to input registers on which the tree T succeeds or x is invalid. As a result, $\Pi_0(T) + \Pi_1(T) = I$. We can then obtain the probability that $\mathcal{Q}$ and $\mathcal{Q}'$ find an adversarial input (conditioned on tree T) as

$$p(T) := \|\Pi_1(T)\,|\psi\rangle\,\|^2 = \Pr_{\mathcal{Q}}\left[x \leftarrow \mathcal{Q}[T]; |x| \geq \frac{n}{2} \text{ and } T[x] \notin \mathsf{FIND1}(x)\right],$$
$$p'(T) := \|\Pi_1(T)\,|\psi'\rangle\,\|^2 = \Pr_{\mathcal{Q}'}\left[x \leftarrow \mathcal{Q}'[T]; |x| \geq \frac{n}{2} \text{ and } T[x] \notin \mathsf{FIND1}(x)\right].$$

We begin by bounding the "expected success probability" $p = \mathbb{E}_{T\sim\mathcal{T}}[\sqrt{p(T)}]$ through the following steps.

1. **Bounding the Euclidean distance.** First, we prove that, the expected Euclidean distance between the pre-measurement final states of any consecutive pairs $|\psi_q^{k+1}\rangle$ and $|\psi_q^k\rangle$ is small. As a result, we obtain an upper bound on $\mathbb{E}_T[\|\,|\psi'\rangle - |\psi\rangle\,\|]$.
2. **Bounding p'.** Next, we prove that $\mathbb{E}_T[p'(T)]$ is small with high probability as it does not make any queries to the real tree T.
3. **Putting the above together.** We then obtain an upper bound for p using the upper bounds on $\|\,|\psi\rangle - |\psi'\rangle\,\|$ and our bound on $\mathbb{E}_T[p'(T)]$.

Finally, we convert the bound on $\mathbb{E}_T[\sqrt{p(T)}]$ to the total success probability.

Bounding the Euclidean distance. Let $\langle\ell\rangle$ be the ℓ-th leaf in the tree as in Proposition 12 and $\ell_T(x)$ be the reached leaf when executing T on x. The strategy is as follows: for any valid input x if $\ell_T(x) \preceq \langle k\rangle$, the tree only queries among the blocks $B_1 = \hat{B}_1, \ldots, B_k = \hat{B}_k$ and outputs an index from one of these blocks. Thus, for any such input x, $T[x] = T^{k+1}[x]$ unless $\ell_T(x) \succ \langle k+1\rangle$. Furthermore, $\mathcal{Q}^k$ does not have any information about $B_{k+1}, \ldots, B_{2^d}$

before its first real query to O^T. Consequently, applying Lemma 13, we can prove that $\ell_T(x) \preceq \langle k+1 \rangle$ happens with high probability for every input $|x\rangle$ in the query of $\mathcal{Q}^k$. We conclude that the states in consecutive algorithms remain close to each other. Formally, we have the following.

▷ Claim 18. The following holds.

i. For $k \in \{0, \dots, q-1\}$, $\mathbb{E}_T \left[\left\| |\psi_q^{k+1}\rangle - |\psi_q^k\rangle \right\| \right] \leq \sqrt{\frac{4d}{2^w}}$;

ii. $\mathbb{E}_T \left[\left\| |\psi'\rangle - |\psi\rangle \right\| \right] \leq q\sqrt{\frac{4d}{2^w}}$.

Proof. By the unitary invariance of the Euclidean norm, we have

$$\begin{aligned} \left\| |\psi_q^{k+1}\rangle - |\psi_q^k\rangle \right\| &= \left\| U_q O^T \dots U_{k+1} O^{T^{k+1}} |\psi_k^{k+1}\rangle - U_q O^T \dots U_{k+1} O^T |\psi_k^k\rangle \right\| \\ &= \left\| \left(O^{T^{k+1}} - O^T \right) |\psi_k^k\rangle \right\|, \end{aligned}$$

where we also used the fact that $|\psi_k^{k+1}\rangle = |\psi_k^k\rangle$. For $x \in \{0,1\}^n$, let Π_x be the projection to the states having input register $|x\rangle_I$, where $\sum_{x\in\{0,1\}^n} \Pi_x = I$. Observe that for $x \in \{0,1\}^n$, if $T^{k+1}[x] = T[x]$ we get $\left(O^{T^{k+1}} - O^T \right) \Pi_x |\psi_k^k\rangle = 0$. So, we can write

$$\left\| |\psi_q^{k+1}\rangle - |\psi_q^k\rangle \right\| = \left\| \sum_{x\in\{0,1\}^n} \left(O^{T^{k+1}} - O^T \right) \Pi_x |\psi_k^k\rangle \right\| \tag{1}$$

$$\begin{aligned} &= \left\| O^{T^{k+1}} \sum_{\substack{x\in\{0,1\}^n \\ T^{k+1}[x]\neq T[x]}} \Pi_x |\psi_k^k\rangle - O^T \sum_{\substack{x\in\{0,1\}^n \\ T^{k+1}[x]\neq T[x]}} \Pi_x |\psi_k^k\rangle \right\| \\ &\leq \left\| O^{T^{k+1}} \sum_{\substack{x\in\{0,1\}^n \\ T^{k+1}[x]\neq T[x]}} \Pi_x |\psi_k^k\rangle \right\| + \left\| O^T \sum_{\substack{x\in\{0,1\}^n \\ T^{k+1}[x]\neq T[x]}} \Pi_x |\psi_k^k\rangle \right\| \\ &= \left\| \sum_{\substack{x\in\{0,1\}^n \\ T^{k+1}[x]\neq T[x]}} \Pi_x |\psi_k^k\rangle \right\| + \left\| \sum_{\substack{x\in\{0,1\}^n \\ T^{k+1}[x]\neq T[x]}} \Pi_x |\psi_k^k\rangle \right\| \\ &= 2 \left\| \sum_{\substack{x\in\{0,1\}^n \\ T^{k+1}[x]\neq T[x]}} \Pi_x |\psi_k^k\rangle \right\|. \end{aligned} \tag{2}$$

Now, consider the state $|\psi_k^k\rangle$; by definition, this state has been obtained by applying unitaries $T^1, \dots, T^k$, and therefore all the coefficients in $|\psi_k^k\rangle$ only depend on sets $\hat{B}_1, \dots, \hat{B}_k$. Consequently, conditioned on $B_1 = \hat{B}_1, \dots, B_k = \hat{B}_k$, we can write

$$|\psi_k^k\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle_I |\phi_x\rangle_{Q,W,O},$$

where all coefficients α_x are fixed complex numbers. As a result, we have

$$\mu_{\hat{B}_1,\dots,\hat{B}_k} := \mathop{\mathbb{E}}_T \left[\left\| |\psi_q^{k+1}\rangle - |\psi_q^k\rangle \right\|^2 \middle| \hat{B}_1, \dots, \hat{B}_k \right] \leq \mathop{\mathbb{E}}_T \left[4 \sum_{\substack{x\in\{0,1\}^n \\ T^{k+1}[x]\neq T[x]}} |\alpha_x|^2 \middle| \hat{B}_1, \dots, \hat{B}_k \right]$$

$$\leq 4 \sum_{x\in\{0,1\}^n} |\alpha_x|^2 \Pr_T\left[T[x] \neq T^{k+1}[x] \middle| \hat{B}_1, \ldots, \hat{B}_k\right].$$

Recall that if $\ell_T(x) \preceq \langle k+1\rangle$ then $T^{k+1}[x] = T[x]$. Applying Lemma 13, we obtain

$$\begin{aligned}\Pr_T\left[T[x] \neq T^{k+1}[x] \middle| \hat{B}_1, \ldots, \hat{B}_k\right] &\leq 1 - \Pr_T\left[\ell_T(x) \preceq \langle k+1\rangle \middle| \hat{B}_1, \ldots, \hat{B}_k\right] \\ &\leq 1 - (1 - \frac{1}{2^w})^d \\ &\leq \frac{d}{2^w}.\end{aligned}$$

As a result, we get the upper bound $\mu_{\hat{B}_1,\ldots,\hat{B}_k} \leq \frac{4d}{2^w} \sum_{x\in\{0,1\}^n} |\alpha_x|^2 = \frac{4d}{2^w}$. Conditioning on $\hat{B}_1, \ldots, \hat{B}_k$, we compute the expectation as

$$\mathop{\mathbb{E}}_T\left[\left\||\psi_q^{k+1}\rangle - |\psi_q^k\rangle\right\|^2\right] = \sum_{\hat{B}_1,\ldots,\hat{B}_k} \mu_{\hat{B}_1,\ldots,\hat{B}_k} \cdot \Pr_T\left[\hat{B}_1 \ \ldots, \hat{B}_k\right] \leq \frac{4d}{2^w}.$$

Finally, we apply Jensen's inequality to get

$$\mathop{\mathbb{E}}_T\left[\left\||\psi_q^{k+1}\rangle - |\psi_q^k\rangle\right\|\right] \leq \sqrt{\mathop{\mathbb{E}}_T\left[\left\||\psi_q^{k+1}\rangle - |\psi_q^k\rangle\right\|^2\right]} \leq \sqrt{\frac{4d}{2^w}}.$$

For part (ii), we can bound the distance between these two states as

$$\mathop{\mathbb{E}}_T\left[\left\||\psi'\rangle - |\psi\rangle\right\|\right] = \mathop{\mathbb{E}}_T\left[\left\|\sum_{k=1}^{q} |\psi_q^k\rangle - |\psi_q^{k-1}\rangle\right\|\right] \leq \sum_{k=1}^{q} \mathop{\mathbb{E}}_T\left[\left\||\psi_q^k\rangle - |\psi_q^{k-1}\rangle\right\|\right] \leq q\sqrt{\frac{4d}{2^w}}. \qquad \triangleleft$$

Bounding $\mathbb{E}_T[p'(T)]$. Note that $\mathcal{Q}'$ depends only on the first q blocks, with $q < 2^d - 1$. As a result, block B_{2^d-1} is independent of the output of $\mathcal{Q}'$, and we can show the following.

▷ Claim 19. $\mathbb{E}_T[p'(T)] \leq \frac{1}{2^w}$.

Proof. Conditioning on values $\hat{B}_1, \ldots, \hat{B}_q$, we have

$$\mathop{\mathbb{E}}_T[p'(T)] = \sum_{\hat{B}_1,\ldots,\hat{B}_q} \mathop{\mathbb{E}}_T\left[p'(T) \mid \hat{B}_1, \ldots, \hat{B}_q\right] \Pr_T\left[\hat{B}_1, \ldots, \hat{B}_q\right]. \tag{3}$$

Conditioning further on output of $\mathcal{Q}'$, we have

$$\begin{aligned}&\mathop{\mathbb{E}}_T\left[p'(T)|\hat{B}_1, \ldots, \hat{B}_q\right] \\ =& \sum_{y\in\{0,1\}^*, |y|\geq\frac{n}{2}} \mathop{\mathbb{E}}_T\left[p'(T) \mid y \leftarrow \mathcal{Q}'[T], \hat{B}_1, \ldots, \hat{B}_q\right] \Pr\left[y \leftarrow \mathcal{Q}'[T] \mid \hat{B}_1, \ldots, \hat{B}_q\right] \\ =& \sum_{y\in\{0,1\}^*, |y|\geq\frac{n}{2}} \mathop{\mathbb{E}}_T\left[T[y] \notin \mathsf{FIND1}(y) | y \leftarrow \mathcal{Q}'[T], \hat{B}_1, \ldots, \hat{B}_q\right] \Pr[y \leftarrow \mathcal{Q}'[T] \mid \hat{B}_1, \ldots, \hat{B}_q] \\ \leq& \sum_{y\in\{0,1\}^*, |y|\geq\frac{n}{2}} \mathop{\mathbb{E}}_T\left[B_{2^d-1}(y) = 0 | y \leftarrow \mathcal{Q}'[T], \hat{B}_1, \ldots, \hat{B}_q\right] \Pr[y \leftarrow \mathcal{Q}'[T] \mid \hat{B}_1, \ldots, \hat{B}_q] \\ =& \sum_{y\in\{0,1\}^*, |y|\geq\frac{n}{2}} \mathop{\mathbb{E}}_T[B_{2^d-1}(y) = 0] \Pr[y \leftarrow \mathcal{Q}'[T] \mid \hat{B}_1, \ldots, \hat{B}_q] \\ \leq& \sum_{y\in\{0,1\}^*, |y|\geq\frac{n}{2}} \frac{1}{2^w} \Pr[y \leftarrow \mathcal{Q}'[T] \mid \hat{B}_1, \ldots, \hat{B}_q] \\ =& \frac{1}{2^w},\end{aligned}$$

where the fourth line is because $T[y] \notin \mathsf{FIND1}(y)$ implies $B_{2^d-1}(y) = 0$, the fifth line is because B_{2^d-1} is independent of the first q blocks, and the sixth line is because the probability that B_{2^d-1} misses all 1-indices of y is at most 2^{-w}, and the last line is because we assumed the algorithm always outputs a valid string $|y| \geq n/2$. Plugging 2^{-w} in (3) we get the desired result. ◁

Putting the above together. We are now ready to upper bound $\mathbb{E}_T[p(T)]$. Note that, for any tree $T \in \mathcal{T}_n$,

$$\begin{aligned}\sqrt{p(T)} &= \|\Pi_1(T)\,|\psi\rangle\| = \|\Pi_1(T)\,|\psi\rangle - \Pi_1(T)\,|\psi'\rangle + \Pi_1(T)\,|\psi'\rangle\| \\ &\leq \|\Pi_1(T)\,(|\psi\rangle - |\psi'\rangle)\| + \|\Pi_1(T)\,|\psi'\rangle\| \\ &\leq \||\psi\rangle - |\psi'\rangle\| + \sqrt{p'(T)},\end{aligned}$$

where we have used triangle inequality and the fact that all eigenvalues of Π_1 have a norm at most one. Getting expectations from both sides, we obtain

$$\mathop{\mathbb{E}}_{T}\left[\sqrt{p(T)}\right] \leq \mathop{\mathbb{E}}_{T}\left[\||\psi\rangle - |\psi'\rangle\|\right] + \mathop{\mathbb{E}}_{T}\left[\sqrt{p'(T)}\right]$$

By Claim 18 part (ii), the first expectation in the above is upper bounded by $q\sqrt{4d/2^w}$. Moreover, by Claim 19, we can bound the second part $\mathbb{E}_T[\sqrt{p'(T)}] \leq \sqrt{\mathbb{E}_T[p'(T)]} \leq \sqrt{1/2^w}$, where we once again used Jensen's inequality. Putting these together, we get $\mathbb{E}_T[\sqrt{p(T)}] \leq q\sqrt{4d/2^w} + \sqrt{1/2^w} = (2q\sqrt{d}+1)2^{-\frac{w}{2}}$. Moreover, since $0 \leq p(T) \leq 1$, we have $p(T) \leq \sqrt{p(T)}$, and therefore

$$\Pr_{T,\mathcal{Q}}\left[x \leftarrow \mathcal{Q}[T]; T[x] \notin \mathsf{FIND1}\right] = \mathop{\mathbb{E}}_{T}[p(T)] \leq \mathop{\mathbb{E}}_{T}\left[\sqrt{p(T)}\right] \leq (2q\sqrt{d}+1)2^{-\frac{w}{2}}.$$

Finally, since $q \leq 2^d - 1$, $p \leq 2 \cdot (2 \cdot 2^d\sqrt{d}+1) \cdot 2^{-w/2} \leq \sqrt{d} \cdot 2^{d+2-w/2}$. ◀

▶ **Corollary 20.** *Let $t = t(n) \in \mathbb{N}$ be an increasing function. Then, the* FIND1 *problem has a quantum-secure $(2^{\frac{\sqrt{t}}{8}}, 2^{-\frac{\sqrt{t}}{8}})$-faux-deterministic algorithm of complexity t. Specifically, for any $\epsilon > 0$ and $t = \log^{2+\epsilon}(n)$, we have a quantum-secure $(\text{quasi-poly}(n), \text{negl}(n))$-faux-deterministic algorithm of depth $\text{polylog}(n)$.*

Proof. For large enough t, letting $w = \frac{7}{8}\sqrt{t}$ and $d = \frac{1}{8}\sqrt{t}+1$ in Theorem 17, we get a faux-deterministic algorithm of depth $dw \leq t$ such that for any attacker with less than $2^{\frac{1}{8}\sqrt{t}} \leq 2^{\frac{1}{8}\sqrt{t}+1} - 1$ queries will succeed at finding an invalid output with probability at most $\sqrt{d} \cdot 2^{d+2-w/2} \leq 2^{\frac{2}{8}\sqrt{t}-\frac{7}{16}\sqrt{t}} \leq 2^{\frac{-\sqrt{t}}{8}}$. ◀

3.4 Partial converse

In this section, we present a partial converse to our faux-deterministic construction: while our construction of depth t is secure against adversaries making $2^{O(\sqrt{t})}$ many queries, we show that for any distribution over decision trees of depth t, a non-adaptive adversary can find an adversarial input in $2^{O(t)}$ many queries.

▶ **Proposition 21.** *Let $\mathcal{T}_n$ be any distribution over decision trees with input length n and total depth $t = t(n) \leq \sqrt{n}/2$. Then, for $k \in \mathbb{N}$, $\mathcal{T}_n$ cannot be a $(k, 1 - e^{\frac{k}{2^{t+1}}})$-faux-deterministic algorithm for the* FIND1 *problem. Specifically, there exists a classical adversary that makes $k = 2^{t+1}$ many queries and succeeds with probability at least $1/2$ over the choice of the tree from the distribution.*

Proof. Let $T \in \mathcal{T}$ be a fixed decision tree of depth $t \leq \sqrt{n}/2$, and let $I := \{i_1, \ldots, i_s\}$ with $s \leq t$ denote the set of indices queried by T on the input 0^n; that is, the branch where all query outcomes are 0. Thus, a string x with $|x| \geq n/2$ is an adversarial input ($T[x] \notin \mathsf{FIND1}(x)$) if and only if $x_i = 0$ for every $i \in I$. Let k be a natural number to be determined later. We consider a simple randomized adversary $\mathcal{A}$ that for $j \in [k]$, chooses uniformly random independent strings X^j with $|X^j| = n/2$ and queries these strings to obtain $T[X^j]$. Eventually, $\mathcal{A}$ outputs any X^j such that $T[X^j] \notin \mathsf{FIND1}(X^j)$, whenever one exists.

For a fixed $j \in [k]$, define $Z^j \subset [n]$ to be the set of indices in X^j with value 0. Therefore, X^j is an adversarial input if and only if $I \subset Z^j$. Consequently, we have

$$\Pr_{X^j}[T[X^j] \notin \mathsf{FIND1}(X^j)] = \Pr_{X^j}[I \subset Z^j] \geq \frac{\binom{n-t}{n/2-t}}{\binom{n}{n/2}} \geq \frac{(n/2-t)^t}{n^t} = \frac{1}{2^t}(1-\frac{2t}{n})^t$$
$$\geq \frac{1}{2^t}(1-\frac{2t^2}{n}) \geq \frac{1}{2^{t+1}},$$

where we have used $s \leq t$, Bernoulli's inequality and the assumption $t \leq \frac{\sqrt{n}}{2}$. Since $X^1, \ldots, X^k$ are independent, for a fixed tree T, the probability that the adversary succeeds in one of its queries is at least

$$\Pr_{X^1,\ldots,X^k}\left[\exists j : T[X^j] \notin \mathsf{FIND1}(X^j)\right] \geq 1 - (1-\frac{1}{2^{t+1}})^k \geq 1 - e^{-\frac{k}{2^{t+1}}}.$$

Note that the above probability is independent of the fixed tree $T \in \mathcal{T}$. As a result, we have $\Pr_{T\sim\mathcal{T},X^1,\ldots,X^k}\left[\exists j : T[X^j] \notin \mathsf{FIND1}(X^j)\right] \geq 1 - e^{-k/2^{t+1}}$. Maximizing over the choice of $X^1, \ldots, X^k$, we obtain a deterministic adversary which proves the desired result. ◀

4 Faux-deterministic algorithm for verifiable search problems

As previously discussed, for input length $n \in \mathbb{N}$, the $\mathsf{FIND1}$ problem has a simple randomized algorithm $\mathcal{R} = (\mathcal{R}_n)$, where $\mathcal{R}_n = \{T_1, \ldots, T_n\}$ and each tree T_i on input x queries x_i and, if $x_i = 1$, outputs i. Thus, each query in the tree construction of Definition 11 can be viewed as

- querying a random tree $T_i \sim \mathcal{R}_n$;
- verifying whether $(x, T_i[x]) \in \mathsf{FIND1}$;
- and continuing to branch according to the above verification.

With a similar idea, we can generalize our results to obtain faux-deterministic algorithms for all search problems that have efficient randomized algorithms and verifiers.

▶ **Theorem 22.** *Let $\Sigma = (\Sigma_n)$ and let $S \subset \bigcup_{n\in\mathbb{N}}(\{0,1\}^n \times \Sigma_n)$ be a search problem with verifier $V = (V_{n,y})_{n\in\mathbb{N},y\in\Sigma_n}$ of complexity $v(n)$ and a randomized algorithm $\mathcal{R} = (\mathcal{R}_n)$ with complexity $r(n)$ with error at most $1/3$. Let $t = t(n) \in \mathbb{N}$ be an increasing function. Then, there exists a faux-deterministic algorithm $\mathcal{F}$ of complexity $(r(n) + v(n)) \cdot t$ such that*

- *S is $(2^{\frac{\sqrt{t}}{4}}, 2^{-\frac{\sqrt{t}}{4}})$-secure against classical adversaries;*
- *S is $(2^{\frac{\sqrt{t}}{8}-1}, 2^{-\frac{\sqrt{t}}{8}})$-secure against quantum adversaries.*

The proof follows the strategy of [11], adapting their reduction to our setting. We start with the following folklore amplification lemma that intuitively reduces the number of decision trees in the randomized algorithm to a linear number.

▶ **Lemma 23.** *Let $S = (S_n)$ where $S_n \subset \{0,1\}^n \times \Sigma_n$ for some output alphabet $\Sigma = (\Sigma_n)$. There exists a large enough constant c such that the following holds. Given a randomized*

algorithm $\mathcal{R} = (\mathcal{R}_n)$ for S with $\Pr_{R\sim\mathcal{R}_n}[R[x] \notin S(x)] \leq 1/3$ for all $x \in \mathrm{dom}(S_n)$, there exists a randomized algorithm $\mathcal{R}' = (\mathcal{R}'_n)$ where $\mathcal{R}'_n$ consists of cn many decision trees, has the same complexity as $\mathcal{R}$, and satisfies $\Pr_{R\sim\mathcal{R}'_n}[R[x] \notin S(x)] \leq 1/2$.

For the sake of completeness, we provide the proof as follows.

Proof. Let c be a constant to be determined later, and let $R_1, \ldots, R_m \sim \mathcal{R}_n$ be uniformly random and independent decision trees sampled from $\mathcal{R}_n$ where $m = cn$. For each input $x \in \mathrm{dom}(S_n)$ and index $i \in [m]$, we define the random variable $X_i^x := V_{n,R_i(x)}[x]$ to denote whether R_i computes a correct output for input x or not. By definition, for each such x, the random variable $X^x := \sum_{i=1}^m X_i^x$ denotes the number of decision trees outputting a correct answer for x. Our goal is to show that with non-zero probability over the choice of R_i, for all inputs $x \in \mathrm{dom}(S_n)$, we have $X^x \geq m/2$.

Let $x \in \mathrm{dom}(S_n)$ be a fixed input. By linearity of expectation, we have $\mathbb{E}_{R_1,\ldots,R_m}[X^x] = \sum_{i=1}^m \mathbb{E}_{R_i}[X_i^x] \geq 2m/3$, where we used the fact that the randomized algorithm $\mathcal{R}_n$ succeeds with probability at least 2/3. Moreover, since $X_1^x, \ldots, X_m^x$ are independent, we can use the multiplicative form of Chernoff's bound to obtain

$$\Pr_{R_1,\ldots,R_m}\left[X^x \leq \frac{m}{2}\right] = \Pr_{R_1,\ldots,R_m}\left[X^x \leq (1-\frac{1}{4})\frac{2m}{3}\right] \leq e^{\frac{-\frac{2m}{3}(\frac{1}{4})^2}{2}} = e^{-\frac{m}{48}}.$$

Thus, by union bound we can get $\Pr_{R_1,\ldots,R_m}\left[\exists x : X^x \leq \frac{m}{2}\right] \leq 2^n e^{-m/48}$. For $c > 48\ln 2$, the above probability is less than 1. Consequently, there exists fixed $R_1, \ldots, R_m \in \mathcal{R}_n$ such that for every $x \in \mathrm{dom}(S_n)$ at least $m/2$ of the R_i output a correct answer. So, a uniform distribution over $\mathcal{R}' := \{R_1, \ldots, R_m\}$ yields the desired result. ◀

Proof of Theorem 22. Let c be the constant in Lemma 23, and let $R_1, \ldots, R_m$ be the corresponding decision trees in $\mathcal{R}'_n$ where $m = cn$. Given an input string x, the main idea is to define a binary string $y(x) \in \{0,1\}^m$ that assigns 0 or 1 to each decision tree R_i and run the faux-deterministic algorithm for FIND1 on $y(x)$. Formally, we set

$$(y(x))_i := \begin{cases} 1 & V_{n,R_i[x]}[x] = 1, \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $y(x)_i = 1$ if and only if R_i outputs a correct answer on x. Note that by Lemma 23, $y(x)$ is a binary string of length $m = cn$ and has Hamming weight $|y(x)| \geq \frac{m}{2}$.

Let $\mathcal{T} = (\mathcal{T}_m)$ be the faux-deterministic algorithm from Theorem 15 with depth parameter t, applied to inputs of length m, and let $T \sim \mathcal{T}_m$. For $T \in \mathcal{T}_m$ we construct an algorithm F_T that on input x simulates $T[y(x)]$ as follows: whenever T queries an index $i \in [m]$, we run $R_i[x]$ and output $y(x)_i \in \{0,1\}$ (namely 1 iff $V_{n,R_i[x]}[x]$ accepts). At the end, the tree T returns an index $i \in [m]$, and we output $R_i[x]$. Let $\mathcal{F} = (\mathcal{F}_n)$ be the distribution of F_T where $T \sim \mathcal{T}_m$. It is easy to verify that $\mathsf{DEPTH}(\mathcal{F}_n) \leq (v(n) + r(n)) \cdot t$.

It only remains to prove that $\mathcal{F}$ is a faux-deterministic algorithm for S. We do this by reducing an adversary for $\mathcal{F}$ to an adversary for $\mathcal{T}_m$ (on FIND1 inputs $y(x)$) as follows. For $n \in \mathbb{N}$, let A be an adversary (classical or quantum) that makes at most q queries and finds an adversarial input for $F_T \sim \mathcal{F}_n$, where the probability is over $T \sim \mathcal{T}_m$. We construct a corresponding adversary A' for $\mathcal{T}_m$ by running the algorithm A with a simulation of $\mathcal{F}_n$ as follows.

- **Classical setting.** In this case, for an input $x \in \{0,1\}^n$, we can directly simulate a query to $F_T[x]$ with a single query to $T[y(x)]$: using $R_1, \ldots, R_m$ and the verifier V, we construct the string $y(x)$; then, we query T on input $y(x)$ to obtain an index $i = T[y(x)]$, and return $R_i[x]$ to A.

- **Quantum setting.** In the quantum setting, we simulate each call to the oracle unitary $O^{F_T} : |x\rangle_I |z\rangle_Z \to |x\rangle_I |z \oplus F_T[x]\rangle_Z$ (where we use the fixed binary encoding of Σ_n in the output register Z) with *two* calls to the unitary O^T, defined on registers Y and Q (with I and Z being treated as workspace and left unchanged) by

$$O^T |x\rangle_I |y\rangle_Y |i\rangle_Q |z\rangle_Z = \begin{cases} |x\rangle_I |y\rangle_Y |i \oplus T[y]\rangle_Q |z\rangle_Z & |y| \geq m/2, \\ |x\rangle_I |y\rangle_Y |i\rangle_Q |z\rangle_Z & \text{otherwise.} \end{cases}$$

  First, we prepare two extra registers Y and Q consisting of m and $\log m$ qubits, respectively. Additionally, using the classical algorithms $R_1, \ldots, R_m$ and V, we construct two unitary maps U_y and U_R defined by their evolution $U_y |x\rangle_I |0\rangle_Y = |x\rangle_I |y(x)\rangle_Y$ and $U_R |x\rangle_I |i\rangle_Q |z\rangle_Z = |x\rangle_I |i\rangle |z \oplus R_i[x]\rangle_Z$. Then, we simulate O^{F_T} by applying the unitaries $U_y^\dagger O^T U_R O^T U_y$. More specifically, we have the following evolution.

$$\begin{aligned} |x\rangle_I |0\rangle_Y |0\rangle_Q |z\rangle_Z &\xrightarrow{U_y} |x\rangle_I |y(x)\rangle_Y |0\rangle_Q |z\rangle_Z \\ &\xrightarrow{O^T} |x\rangle_I |y(x)\rangle_Y |T[y(x)]\rangle_Q |z\rangle_Z \\ &\xrightarrow{U_R} |x\rangle_I |y(x)\rangle_Y |T[y(x)]\rangle_Q |z \oplus R_{T[y(x)]}\rangle_Z \\ &\xrightarrow{O^T} |x\rangle_I |y(x)\rangle_Y |0\rangle_Q |z \oplus R_{T[y(x)]}\rangle_Z \\ &\xrightarrow{U_y^\dagger} |x\rangle_I |0\rangle_Y |0\rangle_Q |z \oplus R_{T[y(x)]}\rangle_Z , \end{aligned}$$

  which is equal to $|x\rangle_I |0\rangle_Y |0\rangle_Q |z \oplus F_T[x]\rangle_Z$.

In either case, from the perspective of A', an algorithm $F_T \in \mathcal{F}$ corresponding to the tree T is being queried. So, if A finds an adversarial input for F_T, our simulation returns an input $\bar{x}$ for which $F_T[\bar{x}] \notin S(\bar{x})$. Hence, for the input $y(\bar{x})$ and the output $i = T[y(\bar{x})]$, we must have $(y(\bar{x}))_i = 0$ (equivalently, $V_{n,R_i[\bar{x}]}[\bar{x}] = 0$), which means $y(\bar{x})$ is an adversarial input for T. As a result, if A succeeds with probability δ (over $T \sim \mathcal{T}_m$), then A' succeeds with probability δ.

On the other hand $\mathcal{T}_m$ is $(2^{\sqrt{t}/4}, 2^{-\sqrt{t}/4})$-secure against classical adversaries (Corollary 16) and $(2^{\sqrt{t}/8}, 2^{-\sqrt{t}/8})$-secure against quantum adversaries (Corollary 20). Since the classical reduction uses exactly q queries to the FIND1 oracle, while the quantum reduction uses $2q$ queries, the theorem follows for classical adversaries making fewer than $2^{\sqrt{t}/4}$ queries and quantum adversaries making fewer than $2^{\sqrt{t}/8-1}$ queries. ◀

Similar to Corollary 16 and Corollary 20, from the above we immediately get the following corollary.

▶ **Corollary 24.** *If the search problem S has a verifier and a randomized algorithm of depth* $\mathrm{polylog}(n)$ *with error at most* $1/3$*, then S has a* $(\text{quasi-poly}(n), \mathrm{negl}(n))$*-faux-deterministic algorithm of depth* $\mathrm{polylog}(n)$.

Proof. For any $\epsilon > 0$, take $t = \log^{2+\epsilon}(n)$ and apply Theorem 22; see Corollary 16 and Corollary 20 for the security bounds. ◀

5 Sensitivity problem

In this section, we consider the meta-complexity problem of finding an input with high block sensitivity.

▶ **Definition 25.** *Let $s = s(n) \in \mathbb{N}$. We define the sensitivity problem* $\text{findBS}^s = (\text{findBS}^s)_n$ *as follows. The input is oracle access to a function $f : \{0,1\}^n \to [n]$ such that for every $x \in \{0,1\}^n$ with $|x| \geq n/2$, we have $f(x) \in \mathsf{FIND1}(x)$ (equivalently, $x_{f(x)} = 1$). The goal is to find an input x with $|x| \geq n/2$ such that f has block sensitivity at least $\text{bs}(f,x) \geq s(n)$ on x, whenever one exists.*

Goldwasser et al. [11] showed that every function f solving $\mathsf{FIND1}$ ($f(x) \in \mathsf{FIND1}(x)$ for all x with $|x| \geq n/2$) has block sensitivity at least $\text{bs}(f) = \Omega(\sqrt{n})$. We use this fact to prove the following theorem.

▶ **Theorem 26.** *Let $s = s(n)$ be an integer valued function satisfying $\omega(1) \leq s(n) \leq o(\sqrt{n})$. Then, any algorithm solving* findBS^s *requires $2^{\sqrt{s}/5}$ many queries to its oracle to solve this problem with success probability at least $2^{-\sqrt{s}/5}$. In other words, any solver using fewer than $2^{\sqrt{s}/5}$ oracle queries cannot succeed with probability more than $2^{-\sqrt{s}/5}$. Specifically, for $\omega(\text{polylog}(n)) \leq s \leq o(\sqrt{n})$, the required query bound $2^{\sqrt{s}/5}$ is super-polynomial, so any such algorithm for* findBS^s *requires more than* $\text{poly}(n)$ *queries.*

Proof. For the sake of contradiction, suppose there exists a randomized algorithm $\mathcal{A}$ that solves findBS^s with success probability at least $2^{-\sqrt{s}/5}$ while making fewer than $2^{\sqrt{s}/5}$ many queries to a given function f in the promise of findBS^s. We show how to refute any tree T of depth $s-2$ purporting to solve $\mathsf{FIND1}$ with success probability greater than $2^{-\sqrt{s}/5} > 2^{-\sqrt{s-2}/4}$ while making fewer than $2^{\sqrt{s}/5} < 2^{\sqrt{s-2}/4}$ queries to T. This yields a contradiction to $(2^{\sqrt{s-2}/4}, 2^{-\sqrt{s-2}/4})$-security in Corollary 16.

So suppose we are given access to a decision tree T. We implicitly define $f : \{0,1\}^n \to [n]$ as follows:

$$f(x) := \begin{cases} T[x] & T[x] \in \mathsf{FIND1}(x), \\ \min\{i \in [n] : x_i = 1\} & \text{otherwise.} \end{cases}$$

Note that $f(x) \in \mathsf{FIND1}(x)$ whenever $|x| \geq n/2$, and therefore f is a valid function in the promise of findBS^s. Moreover, each query to $f(x)$ can be simulated with exactly one query to $T[x]$ and checking whether $T[x] \in \mathsf{FIND1}(x)$.

To refute T, we simulate $\mathcal{A}$ on f by making $2^{\sqrt{s}/5} < 2^{\sqrt{s-2}/4}$ many queries to T. As a result, with probability at least $2^{-\sqrt{s}/5}$, we obtain an input x with $|x| \geq n/2$ that has block sensitivity at least $b \geq s$.

So, let $i := T[x]$. We show that $x_i \neq 1$, which means we found an adversarial input $T[x] \notin \mathsf{FIND1}(x)$. So suppose $x_i = 1$, and therefore $f(x) = T[x]$. Let $Q \subset [n]$ with $|Q| = s-2$ be the set of queries made by T when executed on x. By disjointness of the sensitive blocks and since $b > s \geq |Q| + 2$, we have $b - |Q| \geq 2$. Therefore, there exists at least two disjoint sensitive block $B_1, B_2 \subset [n]$ satisfying $f(x^{B_a}) \neq f(x)$ and $B_a \cap Q = \emptyset$ for $a \in \{1,2\}$. Consequently, T generates the same outputs index $T[x^{B_a}] = T[x] = f(x) \neq f(x^{B_a})$ when executed on these two new inputs. By definition of f, it must be the case that $(x^{B_b})_i \neq 1 = x_i$, and hence, $i \in B_b$. But we argued $B_1 \cap B_2 = \emptyset$ which is a contradiction, completing the proof. ◀

References

1 Hugo Aaronson, Tom Gur, and Jiawei Li. Pseudo-deterministic Quantum Algorithms, 2026. arXiv:2602.17647. doi:10.48550/arXiv.2602.17647.

2 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. doi:10.1137/S0097539796300933.

3 Arkadev Chattopadhyay, Yogesh Dahiya, and Meena Mahajan. Pseudo-Deterministic Query Complexity of Search Problems. *Computational Complexity*, 34(2):20, 2025. doi:10.1007/s00037-025-00266-7.

4 Lijie Chen, Ce Jin, Rahul Santhanam, and R. Ryan Williams. Constructive Separations and Their Consequences. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 646–657, 2022. doi:10.1109/FOCS52979.2021.00069.

5 Lijie Chen, Zhenjian Lu, Igor Oliveira, Hanlin Ren, and Rahul Santhanam. Polynomial-Time Pseudodeterministic Construction of Primes. *Journal of the ACM*, 73(2):14:1–14:43, 2026. doi:10.1145/3803408.

6 Eran Gat and Shafi Goldwasser. Probabilistic Search Algorithms with Unique Answers and Their Cryptographic Applications. Technical Report TR11-136, Electronic Colloquium on Computational Complexity, 2011. URL: https://eccc.weizmann.ac.il/report/2011/136/.

7 Oded Goldreich, Shafi Goldwasser, and Dana Ron. On the possibilities and limitations of pseudodeterministic algorithms. In *Proceedings of the 4th Innovations in Theoretical Computer Science Conference (ITCS)*, ITCS '13, pages 127–138. Association for Computing Machinery, 2013. doi:10.1145/2422436.2422453.

8 Shafi Goldwasser and Ofer Grossman. Perfect Bipartite Matching in Pseudo-Deterministic RNC. Technical Report TR15-208, Electronic Colloquium on Computational Complexity, 2015. URL: https://eccc.weizmann.ac.il/report/2015/208/.

9 Shafi Goldwasser, Ofer Grossman, and Dhiraj Holden. Pseudo-Deterministic Proofs. In Anna R. Karlin, editor, *Proceedings of the 9th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ITCS.2018.17.

10 Shafi Goldwasser, Ofer Grossman, Sidhanth Mohanty, and David P. Woodruff. Pseudo-Deterministic Streaming. In Thomas Vidick, editor, *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 79:1–79:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.79.

11 Shafi Goldwasser, Russell Impagliazzo, Toniann Pitassi, and Rahul Santhanam. On the Pseudo-Deterministic Query Complexity of NP Search Problems. In Valentine Kabanets, editor, *Proceedings of the 36th Conference on Computational Complexity (CCC)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.36.

12 Mika Göös, Nathaniel Harms, Artur Riazanov, Anastasia Sofronova, Dmitry Sokolov, and Weiqiang Yuan. Pseudodeterministic Communication Complexity, 2025. arXiv:2511.04794. doi:10.48550/arXiv.2511.04794.

13 Igor Carboni Oliveira and Rahul Santhanam. Pseudodeterministic Constructions in Subexponential Time. Technical Report TR16-196, Electronic Colloquium on Computational Complexity, 2016. URL: https://eccc.weizmann.ac.il/report/2016/196/.

14 Takashi Yamakawa and Mark Zhandry. Verifiable Quantum Advantage without Structure. *Journal of the ACM*, 71(3):20:1–20:50, 2024. doi:10.1145/3658665.

ETH-Hardness of Learning Monotone Circuits and Approximating Their Size

Bruno Cavalar
University of Oxford, UK

Susanna F. de Rezende
Lund University, Sweden

Matthew Gray
University of Oxford, UK

Rahul Santhanam
University of Oxford, UK

Abstract

We show the following hardness results for monotone learning and approximation of monotone circuit size:

1. Under the Randomised Exponential-Time Hypothesis (rETH), it requires time $n^{\Omega(\log n)}$ to PAC-learn monotone formulas with n input bits and size $s(n) = n$ by monotone circuits of size $n^{(\log n)^{1-\epsilon}}$, for every $\epsilon > 0$.
2. Under the Randomised Exponential-Time Hypothesis (rETH), for any $\delta > 0$, there is a polynomially bounded function m such that $m^{1-\delta}$-multiplicatively approximating the minimum monotone circuit size of a monotone function consistent with a sequence of $m(n)$ labelled examples $\{(x_i, b_i)\}$ over n-bit inputs requires time $m^{\Omega(\log(m))}$.

Our results are shown by a novel application of lifting arguments in proof and communication complexity to hardness of monotone learning, by building on the seminal result of Atserias and Müller [6] on hardness of automating Resolution proofs.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Machine learning theory; Theory of computation → Proof complexity

Keywords and phrases monotone circuit complexity, PAC learning, lifting theorems, Resolution, meta-complexity, hardness of approximation, minimum circuit size problem

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.40

Related Version *Full Version*: `https://arxiv.org/abs/2607.12331`

Funding Rahul Santhanam and Bruno Cavalar acknowledge support from the EPSRC project EP/Z534158/1 on "Integrated Approach to Computational Complexity: Structure, Self-Reference and Lower Bounds". Susanna de Rezende received funding from the Knut and Alice Wallenberg Foundation grant KAW 2023.0116, ELLIIT, and the Swedish Research Council grant 2021-05104.

Acknowledgements We thank Noel Arteche for helpful discussions, and the anonymous reviewers for their valuable comments and suggestions.

1 Introduction

1.1 Motivation from Meta-Complexity and Learning

Meta-Complexity. One of the main goals of the area of *meta-complexity* is to prove strong hardness results for computing complexity measures such as circuit size, formula size and monotone circuit size. While such hardness results can be derived in a standard way from cryptographic assumptions such as the existence of one-way functions [43, 32], it has proven much more challenging to show hardness results under *worst-case assumptions*, such as NP-hardness results. A good reason for this difficulty was identified by Kabanets and Cai [32],



© Bruno Cavalar, Susanna F. de Rezende, Matthew Gray, and Rahul Santhanam;
licensed under Creative Commons License CC-BY 4.0

41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 40; pp. 40:1–40:25
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



who showed that NP-hardness of the Minimum Circuit Size Problem (which asks whether a Boolean function F given by its truth table has circuits of size at most s, for a given parameter s) via polynomial-time honest reductions would imply hard-to-prove circuit lower bounds such as super-polynomial circuit lower bounds for EXP.

Despite this obstacle, there are now several hardness results for various notions of circuit size under worst-case hardness assumptions, including DNF size, Boolean formula size (under ETH), multi-output circuit size, oracle circuit size, etc., [38, 2, 24, 30, 27, 28, 29, 22, 26]. A common feature of these results, except for the recent inapproximability result of [26] for oracle circuit size, is that they do not show strong inapproximability results for circuit size. Here, by a strong inapproximability result, we mean the hardness of approximating the circuit size within a $N^{\Omega(1)}$ multiplicative factor, where N is the size of the function representation.

A major motivation for showing strong inapproximability result for circuit size under worst-case assumptions is that they offer a pathway to worst-case to average-case reductions for NP. It was shown in [44, 23], building on work of [8], that there are non-black-box reductions from strong inapproximability of problems such as circuit size and average circuit size to average-case hardness for these problems. Thus, NP-hardness of strong inapproximability for either of these problems would rule out Heuristica, the complexity world imagined by Impagliazzo [31] where NP is hard in the worst case but easy on average. A recent result of Mazor and Pass [39] rules out strong inapproximability of circuit size under Levin reductions from SAT, assuming a standard cryptographic assumption, but strong inapproximability might still hold under a more general notion of reduction such as Turing reductions. The strong inapproximability result of [26] for oracle circuit size is based on cryptographic techniques and seems difficult to apply to standard circuit size measures. It could potentially be very useful to find other techniques yielding strong inapproximability.

In terms of circuit size measures for which hardness under worst-case assumptions is known, monotone circuit size has been little-studied despite its central role in complexity theory. Showing NP-hardness for the Minimum Monotone Circuit Size Problem (which asks whether a monotone Boolean function given by its truth table has monotone circuit size at most s, for a given parameter s) would be a significant step towards NP-hardness for the Minimum Circuit Size Problem, as there is a well-known reduction from Minimum Circuit Size to Minimum Monotone Circuit Size.

Learning. One of the main goals of the area of *learning theory* is to identify which learning problems are hard in the PAC-learning model. When PAC-learning a concept class $\mathcal{C}$, we are given samples $(x, f(x))$ from a distribution $\mathcal{D}$, where f belongs to $\mathcal{C}$, and our task is to output with high probability a hypothesis c that approximates f well according to the distribution. If the output hypothesis c is in $\mathcal{C}$, the learning algorithm is said to be *proper*, otherwise it is said to be *improper*. We also consider a *semi-proper* version of learning $\mathcal{C}$-circuits by $\mathcal{C}'$-circuits, where $\mathcal{C}' \supseteq \mathcal{C}$.

It is known how to rule out even *improper learning* of various concept classes under cryptographic assumptions [33, 34, 9]. More recently, there has been exciting work on ruling out improper learning even of weak concept classes such as DNFs [10, 11, 12] based on average-case assumptions such as Feige's hypothesis [16] and its variants. However, just as in the case of meta-complexity problems, showing hardness under worst-case assumptions remains challenging. This was explained in a work of Applebaum, Barak and Xiao [3], who showed that NP-hardness of learning general Boolean circuits under Karp reductions or non-adaptive reductions would imply the collapse of the Polynomial Hierarchy. This leaves open the possibility of using more general reductions such as Turing reductions to show hardness.

Known worst-case hardness results are typically for proper learning of weak circuit classes such as DNFs, or for semi-proper learning where the output hypothesis class $\mathcal{D}$ is not too much more powerful than the concept class [1]. While hardness of learning even restricted subclasses of monotone circuits under the uniform distribution is known under cryptographic assumptions [9], and unconditional hardness results are known in the statistical query learning framework [17], very little seems to be known about the worst-case hardness of learning monotone circuits.

As can be seen from the discussion above, there are several parallels between the situation with hardness of meta-complexity problems and the situation with hardness of learning. These parallels are not coincidental: learning for a circuit class $\mathcal{C}$ can be seen as a *search version* of the problem of determining minimum $\mathcal{C}$-size of a Boolean function. Indeed, given access to a Boolean function f, outputting a circuit of size s that computes f (perhaps approximately) via a learning algorithm witnesses that f has (perhaps approximate) circuit size at most s. This connection is exploited to derive new learning algorithms in [8].

In this paper, we aim to establish strong results on the hardness of monotone learning based on worst-case assumption, and then to leverage the connection between monotone learning and meta-complexity to get strong inapproximability of monotone circuit size.

The meta-complexity problem we consider is the following: given $m = \mathsf{poly}(n)$ labelled examples (x_i, b_i), where each x_i is an n-bit string and b_i a bit, and a parameter s, is there a monotone function f of monotone circuit size at most s that is consistent with the labelled examples, i.e., $b_i = f(x_i)$ for each i. In the context of learning, this is known as the Occam Learning problem, and the non-monotone analogue of this problem was called Partial Minimum Circuit Size Problem in [30], where it was shown to be NP-hard. We are interested here in strong inapproximability results, which the techniques of [30] are incapable of proving, since they reduce from the well-approximable Set Cover Problem.

1.2 Our Results

Our first main result is on the hardness of PAC-learning monotone formulas. Proving hardness of improperly PAC-learning monotone formulas is at least as hard as showing NP-hardness of learning general Boolean circuits, and hence beyond reach at this point in time. However, we are able to prove a strong result in the *semi-proper* setting under worst-case assumptions, where our concept class is the class of monotone formulas, but our hypothesis class is the class of polynomially larger monotone circuits. In this setting, we are able to get a quasipolynomial lower bound on the running time of any PAC-learning algorithm, assuming the randomised Exponential Time Hypothesis. Recall that the randomised Exponential Time Hypothesis (rETH) postulates that any algorithm solving 3SAT on formulas of n variables must take time $2^{\Omega(n)}$. This is a standard hardness assumption in fine-grained complexity and parameterized complexity.

▶ **Theorem 1.1** (Informal). *Under rETH, for every $\alpha > 0$, monotone formulas of size n on n variables require time $n^{\Omega(\log n)}$ to be PAC-learned by monotone circuits of size $n^{(\log n)^{1-\alpha}}$.*

The formal statement of Theorem 1.1 is in the first item of Theorem 4.11 in Section 4.4. To the best of our knowledge, it was not known previously even how to get hardness of learning monotone circuits of size n by monotone circuits of size $n^{1+\alpha}$ under standard worst-case assumptions. In fact, our techniques yield a tradeoff, where we can show hardness even for learning monotone formulas of even slightly super-polylogarithmic monotone formula size by monotone circuits of polynomial size. This result is stated in the second item of Theorem 4.11.

Our second main result shows the strong inapproximability of monotone circuit size for the partial version of the Minimum Monotone Circuit Size Problem, where we are given a sequence of labelled examples as input. The assumption used here is again rETH.

▶ **Theorem 1.2.** *For each $c > 1$ the following holds under rETH: there is no algorithm running in time $N^{o(\log N)}$ which, given as input a sequence of $O(n^c \log n)$ labelled examples (x_i, b_i) representable by N bits in all, where each x_i is an n-bit string and b_i a bit, can distinguish between the case that these labelled examples are consistent with a monotone formula of size n (i.e., there is f of monotone formula size n for which $f(x_i) = b_i$ for each i) and the case that these labelled examples are inconsistent with a monotone circuit of size n^c. In particular, for arbitrary $\epsilon > 0$ there is no $N^{o(\log N)}$ time algorithm which $N^{1-\epsilon}$-multiplicatively approximates the partial monotone formula size or the partial monotone circuit size.*

The formal statement of Theorem 1.2 is in the fourth item of Corollary 4.9.

Both Theorem 1.1 and Theorem 1.2 are consequences of a more general hardness result for a monotone version of the Computational Gap Learning problem studied in [3]. In our problem, which we call Monotone Circuit-Formula Gap Learning, we are given a circuit which samples a distribution of labelled examples (x_i, b_i) where x_i is of length n and b_i a bit, and are asked to distinguish between 2 cases: the first where the distribution is completely consistent with a formula of size at most s_1, and the second where the distribution does not even non-trivially correlate with any circuit of size at most s_2. Here $s_1, s_2 : \mathbb{N} \to \mathbb{N}$ are parameters of the problem, which we are able to choose so that $s_2 \gg s_1$. Theorem 1.1 follows immediately from our result about Monotone Circuit-Formula Gap Learning, while Theorem 1.2 follows by a standard sampling argument together with a union bound.

To show our central result about Monotone Circuit-Formula Gap Learning, we use a novel technique in the domain of hardness of learning: *lifting*, together with known strong inapproximability results for Resolution proof size [6, 14, 20, 13, 7]. One of the main applications of lifting is to obtain monotone circuit size lower bounds in a generic way from Resolution proof width lower bounds [18, 19]. The generic nature of these reductions opens the possibility of using them to derive *reductions* from approximating proof size for various proof systems to approximating circuit size for various circuit models. This is yet another instance of the connection between learning and automatability of proof systems proposed in [1].

It may be useful to compare our results with the work of Hirahara [22]. Hirahara shows that a variant of the Computational Gap Learning problem is NP-hard (under randomised reductions), where the task is to distinguish between linear programs (Turing machines computing a linear function) of size s_1 and programs of size $s_2 = s_1 \cdot n^{o(1)}$. Besides the obvious difference that we consider monotone circuits vs. formulas whereas [22] considers programs, crucially our results allow for a superpolynomial gap between circuit size, whereas [22] obtain only a sublinear gap between program size (though under the weaker assumption of NP-hardness). Hirahara also shows NP-hardness of approximating partial circuit size, when one is given the entire truth table of a partial function (referred to as MCSP^*), again only with a $n^{o(1)}$ approximation gap. In Theorem 1.2, however, we consider a *succinct* version of this problem, in which we are only given inputs where the function is defined; we also obtain a stronger hardness of approximation gap here.

We next give a technical overview of the ideas behind our proofs.

1.3 Technical Overview

Overall approach. The main insight of the paper is to convert the hardness of *automating* Resolution [6] into the hardness of approximating monotone circuit size, using results from the literature (called *lifting theorems*) that connect the complexity of Resolution refutations with the monotone complexity of Boolean functions [18].

To understand how this works, let us review how the seminal paper of Atserias and Müller [6] proved that automating Resolution is NP-hard. Recall that an algorithm that automates Resolution must, given as input an unsatisfiable CNF formula ϕ, construct a Resolution refutation of ϕ in time $\mathsf{poly}(|\phi| + s)$, where s is the size of the shortest Resolution refutation of ϕ. Given a 3-CNF formula F on n variables as input, the proof of [6] consists in constructing in polynomial-time a formula $\mathsf{Ref}(F)$ with the following properties:

1. If F is satisfiable, then $\mathsf{Ref}(F)$ admits a polynomial-size Resolution refutation;
2. If F is unsatisfiable, then any Resolution refutation of $\mathsf{Ref}(F)$ must have size $2^{n^{\Omega(1)}}$.

In turn, this implies that if we can *automate* Resolution, then we can determine in which of the two cases we are in, and thus decide the satisfiability of F.

Another significant ingredient in our approach is the *DAG-like lifting theorem*, which converts Resolution *width* lower bounds for a formula ϕ into monotone circuit size lower bounds for a related monotone function f_ϕ [18]. Importantly, the monotone complexity of the resulting function f_ϕ is tightly connected to the Resolution width of ϕ, so that f_ϕ has large monotone complexity iff ϕ requires large Resolution width.

Ideally, we would now like to combine the properties of $\mathsf{Ref}(F)$ with the lifting theorem to reduce the satisfiability of F to deciding the monotone complexity of $f := f_{\mathsf{Ref}(F)}$ when given access to some succinct representation of f (e.g., access to random samples of f, or a succinct representation of its truth-table). Unfortunately, the connection between monotone complexity and Resolution goes only via Resolution *width*, whereas the complexity gap obtained by Atserias-Müller is only for Resolution *size*. Though Resolution size does indeed give *upper bounds* on the monotone complexity of a related function via *feasible interpolation* [36, 35, 42], the converse is not known to hold.

A potential way out would be to consider the notion of *block-width* of proofs [6], a generalisation of Resolution-width to the setting when the variables are split into blocks, as is the case with $\mathsf{Ref}(F)$. The proof of Atserias-Müller can be rephrased as first showing that the block-width of $\mathsf{Ref}(F)$ is $O(1)$ when F is satisfiable, and $n^{\Omega(1)}$ when F is unsatisfiable [20]. However, though the notion of block-width can be lifted to Resolution size, it's not known if it can be lifted to monotone circuit size, and indeed significant technical challenges have been found when trying to do so in at least one previous work [20].

A more promising approach comes from considering a different encoding of $\mathsf{Ref}(F)$ found in [14]. Their encoding admits a Resolution-width gap of $\mathsf{Ref}(F)$ of the following form[1]:

1. If F is satisfiable, $\mathsf{Ref}(F)$ has Resolution-width $O(n)$;
2. If F is unsatisfiable, $\mathsf{Ref}(F)$ has Resolution-width $\Omega(n^3)$.

This seems to be what we are aiming at: an efficiently constructible formula with a noticeable gap in Resolution-width. However, lifting gives us that the monotone complexity of f_ϕ is roughly $2^{\Theta(\mathsf{ResWidth}(\phi))} \cdot \mathsf{Res}(\phi)$. So we obtain a function on $n^{O(1)}$ input bits with $2^{O(n)}$ monotone complexity when F is satisfiable, and $2^{\Omega(n^3)}$ when F is unsatisfiable. Unfortunately, it is not clear how to distinguish between these two cases other than by guessing the *exponential size* circuit that computes the function. In other words, we have obtained a reduction from an NP problem to an NEXP problem, which is hardly surprising. Moreover, the reduction is ultimately useless, as we could just solve SAT directly in exponential time.

[1] The encoding of [6] cannot achieve this gap because it uses unary pointers.

We overcome this hurdle by constructing a different formula $\mathsf{Ref}^*(F)$ with the following properties:[2]

1. $\mathsf{Ref}^*(F)$ has $2^{O(\sqrt{n})}$ variables and $2^{O(\sqrt{n})}$ clauses.
2. $\mathsf{Ref}^*(F)$ admits Resolution proofs of depth at most $O(\sqrt{n})$ when F is satisfiable;
3. $\mathsf{Ref}^*(F)$ requires Resolution proofs of width at least $2^{\Omega(\sqrt{n})}$ when F is unsatisfiable.

Note that we obtain not just a Resolution width upper bound in the satisfiable case, but the strictly more powerful upper bound on Resolution *depth*, which is strongly tied to the size of monotone *formulas* [41, 21]. This is how we obtain a gap between monotone formula and circuit sizes in the final result. Note moreover that the resulting formula has *subexponential* size. This is a careful tradeoff that we exploit: by increasing the size of the formula, we are able to reduce its Resolution-width (and even its Resolution-depth). We now describe the construction of the formula in more detail.

The $\mathsf{Ref}^*(F)$ Formula. The formula $\mathsf{Ref}(F)$ essentially encodes the statement "F has a Resolution proof of small size s", where s is a fixed polynomial in the number of variables of F. When F is satisfiable, the formula $\mathsf{Ref}(F)$ is *unsatisfiable*, and this formula can be refuted with low *block-width* [6, 40].

Their strategy can be seen as low memory technique to achieve the following. Given any proposed proof for F, a *refuter*[3] navigates through it while maintaining the guarantee that the current clause is unsatisfied by the known satisfying assignment x^*. Beginning at $\bot$, eventually such a refuter will make it to a line of the proof that is a weakening of an axiom. Since the line must be unsatisfied by x^*, it cannot be a weakening of any axiom and a contradiction has been found. While this technique can be used to show an $O(1)$ block-width upper bound, the depth of the proof strategy is $\Omega(s \cdot n)$, since it will involve learning every variable of $\Omega(s)$ lines of the proposed proof. Adapting this technique to achieve a $\sqrt{n}$ depth upper bound seems like a nearly hopeless endeavour on first blush. Achieving depth $\sqrt{n}$ requires us to refute $\mathsf{Ref}(F)$ while querying in total fewer variables than the number of variables of F. This presents two barriers:

- The refuter needs to get to a line of the proof that is claimed to be a weakening of an axiom while only seeing variables from $\sqrt{n}$ total blocks, and
- The refuter needs to learn some information about all n variables while being able to query far fewer than n variables.

To overcome the first of these issues, we can force the proof to have depth $\sqrt{n}$, as done in [13]. However, if we are only resolving over one literal at a time, the clauses at depth-$\sqrt{n}$, which are weakening of axioms of F, would have width $\sqrt{n}$. This would break the lower bound of Atserias and Müller because their strategy requires that the clauses in the purported proof that are claimed to be a weakening of an axiom must have width n.

To retrieve the lower bound, we can change the underlying proof system to one which resolves on $\sqrt{n}$ variables at a time. This results in each block, instead of having just a left and a right child – which are the two clauses from which it was derived – now having $2^{\sqrt{n}}$ children, and thus having one pointer point_z for each possible setting $z \in \{0,1\}^{\sqrt{n}}$ of the variables being resolved over (we also importantly fix the order in which variables are resolved

[2] For simplicity of exposition, at this introduction we only explain the construction with the parameters that give a polynomial gap between linear-size formulas and monotone circuits. The construction for "juntas" (i.e., functions that do not depend on all inputs) is the same, but with a different parametrisation.

[3] The "Refuter" is typically referred to as "Prover" in a Prover-Adversary game. Since F is satisfiable, the goal of the Prover is to find a contradiction to the statement that F admits a Resolution refutation, and so its task can be referred to as "refuting" a proposed proof of unsatisfiability of F.

over, with the first segment of $\sqrt{n}$ variables, then the second segment, etc). Unfortunately, this immediately means that the resulting $\mathsf{Ref}(F)$ formula will have to be of size at least $2^{\sqrt{n}}$ just to manage these pointer variables. On the positive side, this gives us a hint for how to overcome the second barrier.

In standard resolution refutations, if you know that a clause B' is the left child of B which is the result of resolving over x_i, you know that x_i is in B', and more importantly that $\neg x_i$ is *not* in B'; and if B' is the right child you know that $\neg x_i$ is in B', and more importantly that x_i is *not* in B'. Consequently, knowing that x^* is a satisfying assignment, if the prover always queries whichever child is not satisfied by x^*, they will eventually find themselves in a leaf which cannot be a valid weakening. In our new multi-variable resolution system, if you know that B' is the z^{th} child of B, you know $\sqrt{n}$ bits of information about the clause that B' claims to have. Using x^* as above, one could in principle get to a leaf which cannot be a valid weakening in depth $O(\sqrt{n})$.

We can extend this density of information from the pointers to a new class of summary variables $\mathsf{summ}^B_{i,z}$ which give us information about all the variables from $x_{(i-1)\sqrt{n}}$ to $x_{i\sqrt{n}-1}$. This idea of using summary variables appeared before in [7]. It is important that our summary variables encode which literals are *not* present (which is ultimately what we are concerned with for the upper bound). Formally, $\mathsf{summ\text{-}unlit}^B_{i,z}$ is set equal to 1 to indicate that for each $z_j = 1$, literal $x_{(i-1)d+j}$ is not present in block B, and that for each $z_j = 0$, literal $\neg x_{(i-1)d+j}$ is not present in block B. We can ensure that a Prover can learn what all $\sqrt{n}$ of the $\mathsf{summ\text{-}unlit}^B_{i,z}$ variables for a leaf node should be while only querying the right set of $O(\sqrt{n})$ variables on their way from the root to a leaf.

To make our bounds as tight as possible, we add some additional complexity, including binary pointers for axioms, split variables to reduce clause width, and a low degree expander between each layer of the proof. This low-degree expander, which was previously used in [13, 7], allows us to keep down to a constant the number of $\mathsf{point}^B_{B',z}$ variables a prover needs to query for each B, z before finding one that is set to one. While the expander does somewhat complicate the proof of the lower bound, we are able to derive it by reducing the retraction pigeonhole principle (rPHP) instance to finding a violated clause, similarly to what was done in [13].

Hardness of approximating circuit size and learning. To see how to join our new Ref^* formula with a lifting theorem to obtain the claimed hardness results, it will be helpful to understand the function that comes from lifting. Roughly speaking, "lifting" usually consists in composing some original computational object (e.g., a propositional formula or a function) with a gadget, so as to hide the original object in a stronger computational model in such a way that the only way to compute/solve the composed problem is by solving the original problem.

In our case, given a formula F with n variables and width k, together with a parameter m, the lifted function $f := f_{F,m}$ from [18] will have $|F| \cdot m^k$ input bits, where $|F|$ is the number of clauses of F. To each $i \in [|F| \cdot m^k]$, we associate a pair (C, α), where C is a clause of F and $\alpha \in [m]^k$. We interpret (C, α) as the k-width clause over the variables $\{y_{ij} : i \in [n], j \in [m]\}$ obtained by replacing the variable $x_i \in \mathsf{vars}(C)$ in C with $y_{i,\alpha(i)}$. We can thus see each input to f as a collection of clauses (C, α). The task of the partial function $f_{F,m}$ is to distinguish between the following two cases:

1. Accept (Output 1): there exists $x : [n] \to [m]$ such that the input bit corresponding to $(C, x^{\mathsf{vars}(C)})$ is set to 1 for every clause C of F, and everything else is set to 0;
2. Reject (Output 0): there exists $y \in \{0,1\}^{mn}$ such that all input bits associated with clauses (C, α) satisfied by y are set to 1, and everything else is set to 0.

The partial function is clearly well-defined, as in the first case the collection of clauses is unsatisfiable, and in the second case they are satisfiable. For this same reason, no input to the function which contains all the clauses of Case (1) will ever be in Case (2), so the function is monotone. One can think of the function f as "hiding" several variable renamings of the the formula F in its input bits; to compute f is to "sniff out" if F has been completely embedded among the renamed clauses.

Most importantly for our purposes, we can easily sample from the distribution which half of the time gives a uniform accepting input of f, and in the other half gives a uniform rejecting input. Indeed, first sample a bit $b \in \{0,1\}$, then, if $b = 1$, sample $n \log m$ bits and interpret it as $x : [n] \to [m]$, otherwise sample a string $y \in \{0,1\}^{mn}$. With x (resp. y), it's then easy to set the relevant bits so as to output an accepting (resp. rejecting) input of $f_{F,m}$. Let us abuse notation and denote by $\mathcal{D}^{F,m}$ both a circuit that samples from this distribution of binary strings (together with the bit b as a prefix) and the distribution itself. We thus see samples from $\mathcal{D}^{F,m}$ as a pair (x, b) where $x \in \{0,1\}^{|F|m^k}$ and $b \in \{0,1\}$. Given F and m as input, it's also easy to see that we can construct the circuit sampling from $\mathcal{D}^{F,m}$ in time $\mathsf{poly}(|F|, m^k)$.

Going back to the approach above, given an input F to 3SAT, our reduction will first construct $\mathcal{D} := \mathcal{D}^{\mathsf{Ref}^*(F),m}$, so as to obtain a distribution with the following properties. If F is satisfiable, then

$$\text{there is a monotone formula } L \text{ of size} \leq s_1 \text{ s.t. } L(x) = b \text{ for every } (x,b) \in \mathrm{supp}(\mathcal{D}); \quad (1)$$

if F is unsatisfiable, then

$$\text{every monotone circuit } K \text{ of size} \leq s_2 \text{ satisfies } \mathop{\mathbb{P}}_{(x,b)\leftarrow\mathcal{D}}[K(x) = b] < 1/2 + \gamma. \quad (2)$$

The value of s_1 is $2^{O(\sqrt{n}\log m)}$, and the value of s_2 is $2^{\Omega(m/\sqrt{n})}$; these come from a refined version of the DAG-like lifting theorem [15] with "small gadgets" that allows us to essentially pick any value of m we would like. However, lifting theorems typically are only stated for $\gamma = 1/2$ (corresponding to a worst-case lower bound). Fortunately, generalising to $\gamma \approx m^{-1}$ is an easy task, requiring only the modification of certain numbers and parameters in the proof of [15]. This is essential for our purposes, as in a learning problem we only expect to approximate the target concept.

We are now ready to state the computational problem for which we show a lower bound. The *Monotone Circuit-Formula Gap Learning problem* $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ is the problem of distinguishing between cases (1) and (2) when given $\mathcal{D}$ as input. Given that our $\mathsf{Ref}^*(F)$ formula has size $2^{O(\sqrt{n})}$, the reduction from SAT to this problem takes this time. Under ETH, we thus obtain an $N^{\Omega(\log N)}$ lower bound for this problem, since the size N of the input is $2^{O(\sqrt{n})}$. To reduce to the case $s_1 = n$, we pad the sampled strings; this takes time $2^{\widetilde{O}(\sqrt{n})}$, so our final lower bound is $N^{\widetilde{\Omega}(\log N)}$.

We can further reduce this problem to a gap version of Partial-Monotone-MCSP we call $\mathsf{mMCFSP}^{s_2}_{s_1}[\gamma]$[4]. In this problem, instead of $\mathcal{D}$, the algorithm is given t examples $(x_1, b_1), \ldots, (x_t, b_t)$, and must distinguish whether there is a monotone formula L of size s_1 such that $L(x_i) = b_i$ for all $i \in [t]$, or whether every monotone circuit C of size s_2 is such that $|\{i \in [t] : C(x_i) \neq b_i\}| \geq (1/2 + \gamma)t$. The reduction is randomised, naturally coming from sampling the distribution $\mathcal{D}$. Moreover, we need to sample $O(s_2 \log s_2)$ examples so

[4] Sometimes the literature reserves the term "Partial-MCSP" to the problem where one is given the entire truth table of a partial function, but we follow here the convention of [30].

that, with high probability, any circuit of size at most s_2 will err on a proportion larger than $1/2 + \gamma$ of the examples (in case (2)), and so the reduction takes time $O(s_2 \log s_2)$. This means that we get a tradeoff between the choice of s_2 and the runtime lower bound coming from rETH. The same reduction also gives us lower bounds on PAC-learning.

2 Preliminaries

For $f, g : \mathbb{N} \to \mathbb{N}$, we write $f \sim g$ if $\lim f/g \to 1$, and $f \asymp g$ if $f = \Theta(g)$. We say that a function Boolean f is γ-approximated by g over a distribution $\mathcal{D}$ if $\mathbb{P}_{x \leftarrow \mathcal{D}}[f(x) = g(x)] \geq 1/2 + \gamma$.

2.1 Basic complexity

Monotone circuits are Boolean circuits without $\neg$ gates. *Formulas* are Boolean circuits whose gates have fan-out 1. *Monotone formulas* are analogously defined.

The *Exponential-Time Hypothesis (ETH)* postulates that there exists a constant $\varepsilon > 0$ such that any algorithm solving 3SAT on n-variate formulas must take time $2^{\varepsilon n}$. The *randomised* ETH *(rETH)* postulates the same lower bound for randomised algorithms.

2.2 Expander graphs

A (m, n, δ, r, e)-*bipartite expander graph* is a bipartite graph $G = ((U \cup V), E)$ with $|U| = m$, $|V| = n$, with every vertex in U having degree at most δ, and where for every subset $S \subseteq [m]$ such that $|S| \leq r$, the neighbourhood of S, denoted $N(S)$, satisfies $|N(S)| \geq e|S|$.

▶ Remark 2.1. A complete bipartite graph $G = ((U \cup V), E)$ with $|U| = m$, $|V| = n$, is a (m, n, δ, r, c)-bipartite expander graph for, e.g., $\delta = n$, $r = n/2$ and $c = 2$.

A standard calculation [25] shows that random graphs are good expanders. We record one parameter regime that we use in our applications.

▶ **Lemma 2.2.** *Let $m, n \in \mathbb{N}$ be such that $m = n^2$. Then there exists $\delta = O(1)$ and $r = \Omega(\sqrt{n})$ such that with high probability a random graph $G \sim \mathbf{G}(m, n, \delta)$ is an $(m, n, \delta, r, \delta/2)$-bipartite expander graph.*

2.3 Proof complexity

Resolution. A *literal* is a propositional atom or its negation, a *clause* is a disjunction of literals, and a *CNF formula* is a conjunction of clauses. We see clauses as sets of literals, and write simply $C \subseteq D$ to express that C is a subclause of D. We denote the set of variables in a clause C by $\mathsf{vars}(C)$ (if x_i or $\neg x_i$ are present in C, we have $x_i \in \mathsf{vars}(C)$). A *Resolution refutation* of an unsatisfiable CNF formula $\varphi = C_1 \wedge \cdots \wedge C_m$ over variables $x_1, \ldots, x_n$ is a sequence $D_1, \ldots, D_s$ of clauses over $x_1, \ldots, x_n$ such that $D_s = \bot$, denoting the empty clause, and for every $i \in [s-1]$, the clause D_i either (a) is one of the clauses $C_1, \ldots, C_m$ of φ, or (b) is a *weakening* of a previous clause, meaning that $D_i \supseteq D_j$ for some $1 \leq j < i$, or (c) has been obtained from two previous clauses $D_j = A \vee x$ and $D_k = B \vee \neg x$, for $j, k < i$, by an application of the *Resolution rule*:

$$\frac{A \vee x \qquad B \vee \neg x}{A \vee B}$$

We say that $A \vee B$ was obtained by *resolving* over x. The *size* of the refutation is s, the number of clauses appearing in the refutation.

To every Resolution refutation we can associate a directed acyclic graph (dag) in a natural way. We define the *depth* of the proof as the length of the longest path in the dag, starting from the root labeled by the empty clause $\perp$. The proof is said to be *tree-like* if the associated graph is a tree rooted at $\perp$.

Let F be an unsatisfiable k-CNF formula. We denote by $\mathsf{ResWidth}(F)$ the minimum integer such that F admits a Resolution refutation whose clauses all have width at most $\mathsf{ResWidth}(F)$. Let $\mathsf{ResDepth}(F)$ be the minimum integer such that F admits a Resolution refutation with depth at most $\mathsf{ResDepth}(F)$. We denote by $\mathsf{Res}(F)$ the size of the smallest Resolution refutation of F, and by $\mathsf{TreeRes}(F)$ the size of the smallest tree-like such refutation. We denote by $|F|$ the number of clauses in F.

Prover-adversary game. An equivalent characterisation of Resolution width and Resolution depth can be given by notions from *query complexity*. Suppose that F is an n-variate unsatisfiable CNF formula. Given any assignment $x \in \{0,1\}^n$ to the variables of F, some clause of C is necessarily falsified. The minimum number k of queries to the variables of F needed to discover one such clause is equivalent to $\mathsf{ResDepth}(F)$.

To characterise $\mathsf{ResWidth}$, it is helpful to modify the above setup in what is commonly called a "Prover-Adversary game". As before, the Prover is querying variables, but now the Prover is allowed to forget the value of some queried variables. The adversary is answering the queries, but may respond with different values when a variable whose value was forgotten is queried again. The minimum number k of variables that the Prover needs to store in order to discover a falsified clause is equal to $\mathsf{ResWidth}(F) \pm 1$ [5].

2.4 Lifting for average-case lower bounds

Given a *gadget* $g : \mathcal{X} \times \mathcal{Y}$, let $g^n : \mathcal{X}^n \times \mathcal{Y}^n$ denote the function

$$g^n(x, y) = (g(x_1, y_1), \ldots, g(x_n, y_n)).$$

Let $\mathsf{Ind}_m : [m] \times \{0,1\}^m \to \{0,1\}$ denote the "index gadget", defined as the Boolean function such that $\mathsf{Ind}_m(x, y) = y[x]$.

▶ **Definition 2.3.** *Let F be an unsatisfiable formula with n variables. Let $f_{F,m}$ be the partial Boolean function over $|F| \cdot m^k$ bits defined as follows. To each $i \in [|F| \cdot m^k]$, associate a pair (C, α), where C is a clause of F and $\alpha \in [m]^k$. We interpret (C, α) as the k-width clause over the variables $\{y_{ij} : i \in [n], j \in [m]\}$ obtained by replacing the variable $x_i \in \mathsf{vars}(C)$ in C with $y_{i,\alpha(i)}$. Each binary string of length $|F| \cdot m^k$ is thus in 1-to-1 correspondence with the set*

$$F \circ \mathsf{Ind}_m^n := \{(C, \alpha) : C \text{ is a clause of } F,\ \alpha : \mathsf{vars}(C) \to [m]\}.$$

The function $f_{F,m}$ has the following behaviour:

1. *Accept the binary string whose support is $\left\{(C, \alpha) \in F \circ \mathsf{Ind}_m^n : x^{\mathsf{vars}(C)} = \alpha\right\}$, for every $x : [n] \to [m]$.*
2. *Reject the binary string whose support is $\{(C, \alpha) \in F \circ \mathsf{Ind}_m^n : (C, \alpha)(y) = 1\}$, for every $y \in \{0,1\}^{nm}$.*

The following theorem shows that the monotone circuit complexity of $f_{F,m}$ is roughly $m^{\Theta(\mathsf{ResWidth}(F))}$ for some $m \gg \mathsf{ResWidth}(F)$; moreover, a formula upper bound of the form $m^{O(\mathsf{ResDepth}(F))}$ also holds. We say that a circuit $D : \{0,1\}^r \to \{0,1\}^N$ *represents* a distribution $\mathcal{D}$ over N-input bits if the random binary string $D(\mathbf{r})$ where $\mathbf{r} \leftarrow \{0,1\}^r$ is distributed according to $\mathcal{D}$.

▶ **Theorem 2.4** ([15, Theorem 11, modified]). *There exists an algorithm $A(F, 1^m)$ which receives as input an unsatisfiable k-CNF formula F on N variables, runs in time $\mathsf{poly}(|F|\cdot m^k)$, and outputs a circuit representing a distribution $\mathcal{D}^{F,m} = (\mathcal{D}_1 + \mathcal{D}_0)/2$ over $|F| \cdot m^k$ bits such that:*

1. *There is a monotone circuit K of size $m^{O(\mathsf{ResWidth}(F))} \cdot \mathsf{Res}(F)$ and a monotone formula L of size $m^{O(\mathsf{ResDepth}(F))}$ such that $K(x) = L(x) = b$ for every $x \in \mathrm{supp}(\mathcal{D}_b)$ and $b \in \{0,1\}$.*
2. *There exists an absolute constant c such that for any $\ell \leq \mathsf{ResWidth}(F)$, any monotone circuit K that satisfies $\mathbb{P}_{b\leftarrow\{0,1\},x\leftarrow\mathcal{D}_b}[K(x) = b] \geq 1/2 + \gamma$, for $\gamma \geq c \cdot \ell \cdot \log(mN)/m$, has size at least*

$$\left(\frac{m}{c \cdot \ell \cdot \log(mN)}\right)^{\ell-1}.$$

Moreover, the distribution $\mathcal{D}_0$ (resp. $\mathcal{D}_1$) is uniformly distributed over $f^{-1}_{F,m}(0)$ (resp. $f^{-1}_{F,m}(1)$).

Proof sketch. The algorithm A constructs a circuit $\mathcal{D}^{F,m} : \{0,1\}^r \to \{0,1\}^{|F|\cdot m^k}$ where $r = mN + 1$ that behaves in the following way. If the first bit of r is 0, then let $y \in \{0,1\}^{mN}$ be the remaining mN bits. In the output bit (C, α), the circuit outputs 1 iff y satisfies the clause (C, α). Moreover, if the first bit of r is 1, then let $x \in [m]^N$ be given the first $m \log n$ of the remaining bits. In the output bit (C, α), the circuit outputs 1 if $x^{\mathsf{vars}(C)} = \alpha$. The construction of D, and D itself, can be clearly be done in time $\mathsf{poly}(|F| \cdot m^k)$.

Property (1) of the resulting distribution is folklore (see, e.g., [37]). Property (2) is proved in [15, Theorem 11] with $\gamma = 1/2$; the result with smaller γ can be easily obtained by tweaking a few parameters. We sketch how to do this in the full version of this paper. ◀

3 Shallow Refutation formula

Let F be a 3-CNF formula over n variables. The $\mathsf{Ref}_s(F)$ formula encodes the statement "F has a resolution proof of length s". Slightly more formally, the variables of $\mathsf{Ref}_s(F)$ are divided into s blocks, each of which encodes a potential clause in the purported refutation of F and how this clause was derived, and the clauses of $\mathsf{Ref}_s(F)$ enforce that all derivation steps are sound.

The known resolution refutation of $\mathsf{Ref}_s(F)$ when F is satisfiable, given by Pudlák's upper bound [40], has large width $\Omega(n)$. Under ETH, a large width is unavoidable, since it is possible to find width w refutations of an unsatisfiable N-variate CNF formula in time $N^{O(w)}$. While it is possible to show a gap between the Resolution width of the formula $\mathsf{Ref}_s(F)$ in the satisfiable and unsatisfiable cases of F [6, 14] ($O(n)$ vs. $\Omega(s/n)$), for our applications we need a better upper bound. In this section, we define a formula related to $\mathsf{Ref}_s(F)$ which substantially decrease the width upper bound at the cost of an increase in the size of the formula. Moreover, we show that our new formula has small Resolution *depth* when F is satisfiable.

3.1 The new formula $\mathsf{Ref}^G_d(F)$

Our new formula is denoted by $\mathsf{Ref}^G_d(F)$, and it depends on a bipartite expander graph G and an integer parameter d. We somewhat follow the notation established in "The Proof Analysis Problem" by Arteche et al. [4], but modify the structure of the purported resolution refutation and introduce some extension variables, similar to what was done in [13] and [7].

As was done in [13] we structure the blocks in d layers (instead of n), and as in [7], we add "summary" variables that give information about segments of n/d variables. These changes allow us to turn the $O(1)$ block-width upper bound and corresponding $O(n)$ resolution width upper bound in [40] into a $O(d)$ upper bound on depth, at the cost of increasing the number of variables and clauses in $\mathsf{Ref}_d^G(F)$ to $2^{\Theta(n/d)}$.

Since it is crucial that these summary variables encode that literals are absent, we also switch focus and define "un-literal" variables encoding that single literals are *not present* in a block as opposed to previous encoding of Ref which has variables encoding the *inclusions* of literals. This is done mostly for exposition purposes since our unlit variables are just the negation of the usual literal variables. We also include binary variable for pointers (as done in [14]) to make it possible to query pointers more efficiently.

Finally, we also include "split" variables to turn our formula into a 3-CNF formula using a standard transformation. This is done in order to minimise the blow-up in size of the function we obtain after lifting $\mathsf{Ref}_d^G(F)$.

We start by explaining the structure of the formula $\mathsf{Ref}_d^G(F)$. We then introduce the main variables of $\mathsf{Ref}_d^G(F)$ and the clauses enforcing that the purported proof is sound and is structured as we would like. Finally, we introduce the split variables and explain how they are used to transform the formula into a 3-CNF formula.

Structure of $\mathsf{Ref}_d^G(F)$ and notation. Let F be a CNF formula over n variables and $\mathsf{poly}(n)$ clauses, let d be an integer, and let G be a bipartite graph with $t2^{n/d}$ nodes on left side, t nodes on the right, and outdegree δ from each node on the left. We define $s := t \cdot d$. For the sake of convenience in our presentation, we assume that d divides n, and that $|F|$ (the number of clauses of F) and δ, the left outdegree of the expander G, are powers of 2, so they can be cleanly indexed into by binary pointers. We later explain why we can make these assumptions without loss of generality (see Remark 3.2).

The formula $\mathsf{Ref}_d^G(F)$ is defined over $s+1$ blocks arranged in $d+1$ layers. The block $s+1$ is the root and it is alone in layer 0. The other s blocks are arranged in decreasing order in layers 1 to d with s/d blocks per layer. Formally, for $B \in [s]$, we have $\mathsf{layer}(B) = 1+d-\lceil B/t \rceil$ while $\mathsf{col}(B) = B - (\lceil B/t \rceil - 1)t$. So for $B = s$ we get that $\mathsf{layer}(s) = 1, \mathsf{col}(s) = t$, while for $B = 1$ we get $\mathsf{layer}(1) = d, \mathsf{col}(1) = 1$.

Given an integer S which is a power of 2 and an integer $X \in [S]$, we denote by $\mathsf{bin}_S(X)$ the binary string of length $\log S$ encoding X. When S is clear from the context, we simply write $\mathsf{bin}(X)$. For instance, for $A \in [|F|]$, we write $\mathsf{bin}(A) \in \{0,1\}^{\log |F|}$ to be the binary string of length $\log |F|$ encoding A.

Finally, we need a way of managing binary indices for the at most δ valid z^{th} children B' of B. To do this, we need a surjection (which we will label $\mathsf{sur}_{z,\mathsf{col}(B)}$) from the set of at most δ valid z^{th} children to $[\delta]$. It is easy to see that, if $\mathsf{sur}_{z,\mathsf{col}(B)}$ is the lexicographical order over the valid B's, then $\mathsf{sur}_{z,\mathsf{col}(B)}$ can be efficiently computed when needed while constructing $\mathsf{Ref}_d^G(F)$ in all the parameter regimes considered in this paper. Note that there may be parameter regimes in which this is not the case.

The variables of $\mathsf{Ref}_d^G(F)$. Let $x_1, .., x_n$ be the variables of F. For any variable x_i, the variable $(x_i)^0$ corresponds to $\neg x_i$ and $(x_i)^1$ corresponds to x_i. We define the set $\mathsf{Unlit}_n = \{x_1, ..., x_n, \bar{x}_1, ..., \bar{x}_n\}$. Let $\mathsf{Leaves} := [s/d]$ be the set of blocks at layer d. Some variables (like $\mathsf{weak}, \mathsf{weakbin}$, which encode what axiom in F the block is a weakening of, in unary and in binary, respectively) are only defined for blocks in Leaves and some ($\mathsf{point}, \mathsf{pointbin}$, which encode the blocks that were used to derive the current block, also in unary and in binary, respectively) are only defined for blocks that are not in Leaves.

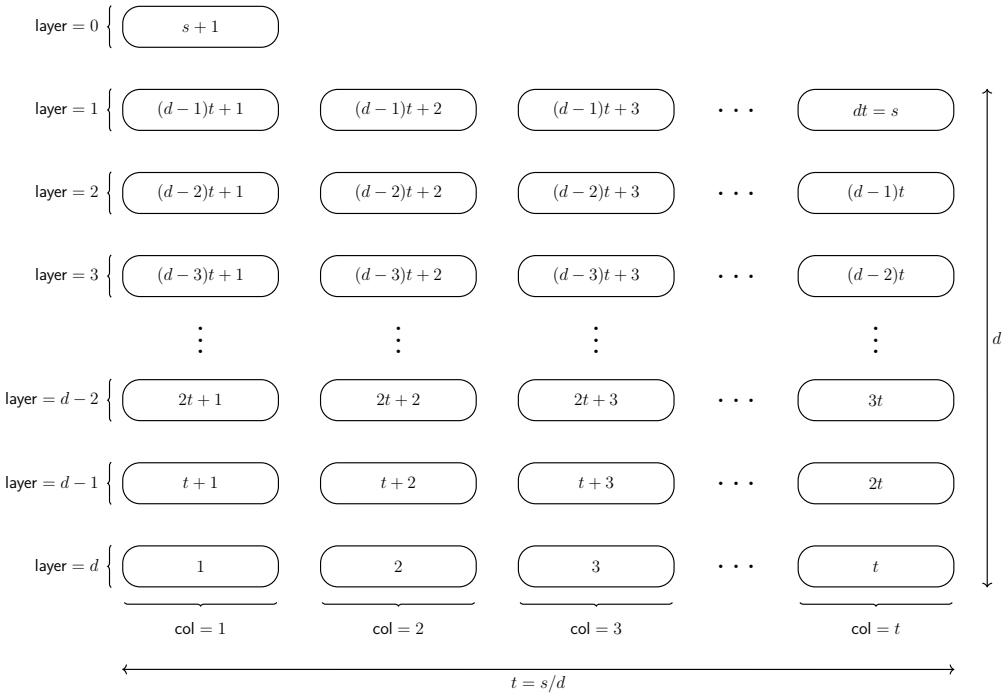


Figure 1 The layout of blocks within the purported refutation, demonstrating the columns and layers.

Recall that, in our setting, a block is obtained by resolving n/d variables at once. Therefore, the z^{th} child, for $z \in \{0,1\}^{n/d}$, of a block at layer i should not contain literals x_{id+j} for each $z_j = 1$, nor literals $\neg x_{id+j}$ for each $z_j = 0$. This will be enforced by using the variables summ-unlit. In fact, except for blocks that are leaves, the only information about what literals are contained in each block is given by these summ-unlit variables. Since the layer of a block determines what variables can be present in the block, only the relevant summ-unlit variables are defined for each block. In particular, the root does not contain any summ-unlit variable and the leaves contain all summ-unlit variables. Moreover, the leaves are the only blocks that contain the unlit variables, as only then are these needed to connect the clause in the block with the axiom of F that it is supposedly a weakening of.

We are now ready to introduce the variables of our formula.

▶ Remark 3.1. For each class of variables listed above there are fewer than $O\left(|F| \cdot n \cdot s^2 \cdot 2^{n/d}\right)$ variables of the class. Assuming $|F| = n^{O(1)}$, $d \leq n$, and $s = 2^{O(n/d)}$, this is $2^{O(n/d)} \cdot n^{O(1)}$.

▶ Remark 3.2. We now explain why we can make the divisibility assumptions we made about $|F|, d, n$ and δ.

- When d does not divide n, we can add one final layer with a corresponding set of summary and pointer variables to manage those x_i with $i > \lfloor n/d \rfloor d$.
- When $|F|$ and δ are not powers of two, we can add clauses which rule out each invalid binary pointer or interpret each invalid pointer as some canonical value.

The clauses of $\mathsf{Ref}_d^G(F)$. We first introduce the unsplit versions of the clauses which make up $\mathsf{Ref}_d^G(F)$. These clauses enforce the basic structure of the Shallow-Resolution formula ensuring particularly that our summary and pointer variables interact correctly. The clauses are defined for arbitrary G but in our proofs we focus on two cases for G: either G is a bipartite expander with t nodes on the left side, $t \cdot 2^{n/d}$ on the right side, and outdegree δ from each node the left side; or G is a complete graph such that all $(\mathsf{col}(B), (\mathsf{col}(B'), z)) \in E(G)$.

unlit^B_ℓ : literal $\ell \in \mathsf{Unlit}_n$ is not present in block $B \in \mathsf{Leaves}$;

weak^B_A : block $B \in \mathsf{Leaves}$ was obtained by weakening axiom $A \in [|F|]$;

$\mathsf{weakbin}^B_j$: for $B \in \mathsf{Leaves}$ and $j \in [\log |F|]$, the value of the j^{th} bit of $\mathsf{bin}(A) \in \{0,1\}^{\log |F|}$, where $A \in [|F|]$ is the axiom that block B was obtained from by weakening;

$\mathsf{summ\text{-}unlit}^B_{i,z}$: for block $B \in [s]$, $i \in [\mathsf{layer}(B)]$, and $z \in \{0,1\}^{n/d}$, for each $z_j = 1$, literal $x_{(i-1)d+j}$ is not present in block B, and for each $z_j = 0$ literal $\neg x_{(i-1)d+j}$ is not present in block B;

$\mathsf{point}^B_{B',z}$: for blocks $B, B' \in [s+1]$, the z^{th} child of block B is B' (and therefore $\mathsf{summ\text{-}unlit}^{B'}_{i,z} = 1$ for $i = \mathsf{layer}(B') = \mathsf{layer}(B) + 1$,)

$\mathsf{pointbin}^B_{z,j}$: for block $B \in [s+1]$, $j \in [\log \delta]$, and $z \in \{0,1\}^{n/d}$, the value of the j^{th} bit of $\mathsf{bin}(\mathsf{sur}_{z,\mathsf{col}(B)}(B')) \in \{0,1\}^{\log \delta}$ where B' is the z^{th} child of B (and, in particular, $\mathsf{layer}(B') = \mathsf{layer}(B) + 1$);

enable^B : block $B \in [s+1]$ is enabled.

We divide the clauses into three categories depending on whether the block in the superscript is in layer 0 (meaning the block is the root), in layer d (meaning the block is a leaf), or between layers 1 and $d-1$ (meaning the block is an internal block).

Root case – final block is enabled.

$$\mathsf{enable}^{s+1} \tag{Ref-1}$$

Leaf case – managing axioms. The following clauses are defined for each $B \in \mathsf{Leaf}$.

$$\left(\mathsf{enable}^B \wedge \mathsf{weak}^B_A\right) \to \neg \mathsf{unlit}^B_\ell \qquad \text{for } A \in [|F|],\ \ell, \text{ in clause } A; \tag{Ref-2}$$

$$\left(\bigwedge_{j \in [\log |F|]} \left(\mathsf{weakbin}^B_j\right)^{\mathsf{bin}(A)_j}\right) \to \mathsf{weak}^B_A \qquad \text{for } A \in [|F|]; \tag{Ref-3}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{summ\text{-}unlit}^B_{i,z}\right) \to \mathsf{unlit}^B_{(x_k)^{z_j}} \qquad \text{for } i \in [d], j \in [n/d], \ k = (i-1)n/d + j\,. \tag{Ref-4}$$

Internal Step – pointer and summary enforcement. The following clauses are defined for each B that is not a leaf, i.e., for B such that $\mathsf{layer}(B) < d$; for each $z \in \{0,1\}^{n/d}$; and for each B' which is a possible zth child for B, i.e., for $B' \in P_B \coloneqq \{B' : \mathsf{layer}(B') = \mathsf{layer}(B) + 1$ and $((\mathsf{col}(B), z), \mathsf{col}(B')) \in E(G)\}$.

$$\left(\bigwedge_{j \in [\log \delta]} \left(\mathsf{pointbin}^B_{z,j}\right)^{\mathsf{bin}(\mathsf{sur}(B'))_j}\right) \to \mathsf{point}^B_{B',z} \tag{Ref-5}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{point}^B_{B',z}\right) \to \mathsf{enable}^{B'} \tag{Ref-6}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{point}^B_{B',z'} \wedge \mathsf{summ\text{-}unlit}^B_{i,z}\right) \to \mathsf{summ\text{-}unlit}^{B'}_{i,z} \qquad \text{for } z' \in \{0,1\}^{n/d},\ i \le \mathsf{layer}(B); \tag{Ref-7}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{point}^B_{B',z}\right) \to \mathsf{summ\text{-}unlit}^{B'}_{i+1,z} \qquad \text{for } i = \mathsf{layer}(B). \tag{Ref-8}$$

▶ Remark 3.3. For each class of clauses listed above, there are fewer than $O(|F| \cdot n \cdot s^2 \cdot 2^{2n/d})$ clauses of that class, and each clause has fewer than $O(\log(|F|) + \log(\delta))$ literals in them.

We now introduce a final set of variables which refer to as "split variables". These are used to transform the formula into 3-CNF using a standard transformation. For every clause $C_i = v^i_1 \vee v^i_2 \vee ... \vee v^i_{|C_i|}$ so far introduced we introduce a collection of split variables $\mathsf{split}^j_{C_i}$ for each $j \in [|C_i| - 3]$ and replace C_i with a collection of clauses $(v^i_1 \vee v^i_2 \vee \mathsf{split}^1_{C_i}) \wedge (\neg\mathsf{split}^1_{C_i} \vee v^i_3 \vee \mathsf{split}^2_{C_i}) \wedge ... \wedge (\neg\mathsf{split}^{|C_i|-3}_{C_i} \vee v^i_{|C_i|-1} \vee v^i_{|C_i|})$.

▶ Remark 3.4. By Remark 3.3 if $s = 2^{O(n/d)}$, $d \leq n$, $|F| = n^{O(1)}$, and $\delta = O(s)$ then before splitting, the formula has $2^{O(n/d)} \cdot n^{O(1)}$ clauses containing at most $O(\log n + \log s)$ variables each. As a consequence, after splitting, $\mathsf{Ref}^G_d(F)$ has at most $O(\log n + \log s) \cdot 2^{O(n/d)} \cdot n^{O(1)} = 2^{O(n/d)} \cdot n^{O(1)}$ clauses, each of which contains at most 3 variables.

3.2 Main Theorem

We now show the following theorem.

▶ **Theorem 3.5.** *Given a 3-CNF formula F on n variables, with $\mathsf{poly}(n)$ clauses, and positive integers $d \leq n$ and $t \leq 2^{n/d}$, and a $(t \cdot 2^{n/d}, 2^{n/d}, \delta, r, c)$-bipartite expander graph G there is a deterministic algorithm running in time $2^{O(n/d)} \cdot n^{O(1)}$ that constructs a 3-CNF formula $\mathsf{Ref}^G_d(F)$ which satisfies the following properties:*

1. *$\mathsf{Ref}^G_d(F)$ has $2^{\Theta(n/d)} \cdot n^{\Theta(1)}$ variables and $2^{\Theta(n/d)} \cdot n^{\Theta(1)}$ clauses.*
2. *$\mathsf{Ref}^G_d(F)$ has width $O(1)$;*
3. *$\mathsf{Ref}^G_d(F)$ admits Resolution proofs of depth at most $O(d \log \delta + \log n)$ when F is satisfiable;*
4. *$\mathsf{Ref}^G_d(F)$ requires Resolution proofs of width at least $2^{\Omega(r(c-1)/n)}$ when F is unsatisfiable.*

Proof. The first two claims follow directly from Remark 3.4. The fourth claim follows from essentially the same argument given in [7] (reducing a hard for resolution pigeonhole principle to finding a collision on the expander G), which we defer to the full-length version of this paper.

The third claim follows from the guaranteed success of the following $O(d \log \delta + \log n)$ depth prover in the standard Prover-Delayer game (see Section 2.3) used to analyse bounds for resolution proofs. We lay out the strategy the prover takes in each of the three cases: root where $B = s + 1$, internal where $B \leq s$ and $\mathsf{layer}(B) < d$, leaf where $\mathsf{layer}(B) = d$, and backchecking which is done after finding a leaf which is a valid weakening. For each of these, we specify the types of clauses the prover could identify as being violated if the delayer gave answers that do not allow the prover to proceed in each step. In all cases it should be clear that the prover can identify a violated clause of one of the listed classes Ref-1–Ref-8 immediately. Moreover, since all non-split clauses have at most $O(\log \delta + \log |F|) = O(\log \delta + \log n)$ variables, finding a split clause which is violated can be done in additional depth $O(\log \delta + \log n)$ by querying all the variables in the clause and all the split variables associated with the clause.

The prover critically knows an assignment x^* which satisfies F and will use this knowledge in implementing their strategy. In discussing the strategy below, we apply subscripts to assignment strings x^*_i and summary strings z^{j+1}_i with these subscripts serving as indices specifying the i^{th} bit in the string.

Invariance condition. After applying the root case and k internal steps of the strategy (but not having yet applied the leaf case or the backchecking step), the following invariance conditions are maintained:

- for all $j \in 0 \cup [k]$ the prover knows the variables enable^{B_j} and $\mathsf{point}^{B_j}_{B_{j+1},z^{j+1}}$ which are all equal to 1;
- $B_0 = s+1$;
- $\forall j \in [d], \forall i \in [n/d] : z_i^{j+1} = x^*_{jn/d+i}$.

Root Case. Learn that $s+1$ is enabled by querying enable^{s+1} which will equal 1, otherwise (Ref-1) is violated. Learn the index B_1 of the child which is unsatisfied by x^* by querying the $\log\delta$ variables $\mathsf{pointbin}^{s+1}_{z^1,j}$ where $z_i^1 = x_i^*$. Finally, query the variable $\mathsf{point}^{s+1}_{B_1,z}$ which must be 1 otherwise (Ref-5) is violated.

Internal Step. Learn that B_j is enabled by querying enable^{B_j} which will equal 1 (Ref-6). Learn the index B_{j+1} of the child of B_j which is unsatisfied by x^* by querying the $\log\delta$ variables $\mathsf{pointbin}^{B_j}_{z^{j+1},j'}$ where $z_i^{j+1} = x^*_{jn/d+i}$. Finally, query the variable $\mathsf{point}^{B_j}_{B_{j+1},z}$ which must be 1 otherwise right side of (Ref-5) is violated.

Leaf Case. Learn that B_j is enabled by querying enable^{B_j} which will equal 1, otherwise (Ref-6) is violated. Query all $\mathsf{summ\text{-}unlit}^{B_j}_{i,z^i}$ where each z^i is specified according to the system in the internal step. If they are all 1 then, learn the axiom $A \in [|F|]$ which B_j is a weakening of by querying the $\log|F|$ variables $\mathsf{weakbin}^{B_j}_{j'}$. Then query the variable $\mathsf{weak}^{B_j}_A$ which must be 1 otherwise it would violate (Ref-3). Since x^* is a satisfying assignment, one of (Ref-2) or (Ref-4) will be violated. We are left with the case where at least one of the $\mathsf{summ\text{-}unlit}^{B_j}_{i,z^i}$ variables is 0.

Backchecking. For the indices i and j for which $\mathsf{summ\text{-}unlit}^{B_j}_{i,z^i}$ was found to be 0, query all $\mathsf{summ\text{-}unlit}^{B_\kappa}_{i,z^i}$ for $\kappa \in [i+1, j-1]$. If $\mathsf{summ\text{-}unlit}^{B_{i+1}}_{i,z^i} = 0$ then the appropriate (Ref-8) is violated. Otherwise the prover finds a κ such that $\mathsf{summ\text{-}unlit}^{B_\kappa}_{i,z^i} = 1$ but $\mathsf{summ\text{-}unlit}^{B_{\kappa+1}}_{i,z^i} = 0$, which implies the appropriate (Ref-7) is violated.

In the root case, and each of the d internal steps, the prover queries $O(\log\delta)$ variables; in the leaf case, the prover queries $O(\log|F|)$ variables; and in the backchecking case, the prover queries at most d variables. In total, it makes at most $O(d\log\delta + \log|F|)$ queries and is guaranteed to find a violated clause. ◀

From Theorem 3.5 we obtain two corollaries. First, using the fact that an unbalanced complete bipartite graph is an expander (Remark 2.1), we obtain the following.

▶ **Corollary 3.6.** *Given a 3-CNF formula F on n variables, with $\mathsf{poly}(n)$ clauses, and positive integers $d \le n$ and $t \le 2^{n/d}$, there is a deterministic algorithm running in time $2^{O(n/d)} \cdot n^{O(1)}$ that constructs a 3-CNF formula $\mathsf{Ref}^G_d(F)$ which satisfies the following properties:*

1. *$\mathsf{Ref}^G_d(F)$ has $2^{O(n/d)} \cdot n^{O(1)}$ variables and $2^{O(n/d)} \cdot n^{O(1)}$ clauses;*
2. *$\mathsf{Ref}^G_d(F)$ has width $O(1)$;*
3. *$\mathsf{Ref}^G_d(F)$ admits Resolution proofs of depth at most $O(d\log t + \log n)$ when F is satisfiable;*
4. *$\mathsf{Ref}^G_d(F)$ requires Resolution proofs of width at least $2^{\Omega(t/n)}$ when F is unsatisfiable.*

Using a randomised construction of expanders (Lemma 2.2), we can improve the above construction in the following way.

▶ **Corollary 3.7.** *Given a 3-CNF formula F on n variables, with $\mathsf{poly}(n)$ clauses, and a positive integer $d \le n$, there is a randomized algorithm running in time $2^{O(n/d)} \cdot n^{O(1)}$ that constructs a 3-CNF formula $\mathsf{Ref}^G_d(F)$ which satisfies the following properties:*

1. $\mathsf{Ref}_d^G(F)$ *has* $2^{O(n/d)} \cdot n^{O(1)}$ *variables and* $2^{O(n/d)} \cdot n^{O(1)}$ *clauses;*
2. $\mathsf{Ref}_d^G(F)$ *has width* $O(1)$*;*
3. $\mathsf{Ref}_d^G(F)$ *admits Resolution proofs of depth at most* $O(d + \log n)$ *when* F *is satisfiable;*
4. $\mathsf{Ref}_d^G(F)$ *requires Resolution proofs of width at least* $2^{2^{\Omega(n/d)}}$ *when* F *is unsatisfiable.*

4 Learning lower bounds under ETH

In this section, we formally prove Theorems 1.1 and 1.2. We begin by defining a gap learning decision problem to which we will reduce the SAT problem, and from which we will be able to reduce to other learning problems.

▶ **Definition 4.1.** *Let* $s_1, s_2, \varepsilon : \mathbb{N} \to \mathbb{N}$ *be time-constructible functions. The computational problem* $\mathsf{mCFGL}_{s_1}^{s_2}[\varepsilon]$ *(referred to as* Monotone Circuit-Formula Gap Learning*) receives as input* $(1^n, \mathcal{D})$*, where* $\mathcal{D}$ *is a circuit with* $n+1$ *output bits and size* n^2 *(here, we identify the circuit with its representation for simplicity of notation). We interpret* $\mathcal{D}$ *as a distribution of pairs* $(x, b) \in \{0,1\}^n \times \{0,1\}$*, obtained by sampling a uniformly random input* r *and outputting* $\mathcal{D}(r)$*. The task is to decide if there exists a monotone formula* C *(with* n *input bits and single output bit) of size* s_1 *such that* $\mathbb{P}_{(x,b)\leftarrow\mathcal{D}}[C(x) = b] = 1$*, or if every monotone circuit* C' *such that* $\mathbb{P}_{(x,b)\leftarrow\mathcal{D}}[C'(x) = b] \geq 1/2 + \varepsilon$ *must have size at least* s_2*. The number* n *is called the* dimension *of the input.*

▶ Remark 4.2. We observe that the size of the sampling circuit $\mathcal{D}$ is chosen to be n^2 arbitrarily but without loss of generality. Indeed, given any family of inputs where the n-bit samplers have size $s = n^{O(1)}$, if $s < n^2$ we can just add dummy gates to the circuit; if $s > n^2$ we can pad the output x with $\sqrt{s} - n$ 0's; in both cases, we reduce to the case $s = n^2$ in polynomial-time.

We now implement the strategy outlined in the introduction. We take a 3CNF formula F on n variables, construct the formula $\mathsf{Ref}_d^G(F)$ satisfying either Corollary 3.6 (in case of a deterministic reduction) or Corollary 3.7 (in case of a randomised reduction). Using a deterministic reduction, taking m such that $n \leq m \ll 2^{n/d}$, $t \leq 2^{n/d}$ and $\ell = \varepsilon dm/n \ll 2^{t/n}$ for small enough ε in Theorem 2.4 gives that the function $f_{\mathsf{Ref}_s^G(F),m}$ has $N = 2^{O(n/d)} \cdot n^{O(1)}$ input bits and the following monotone complexity:

1. At most $m^{O(d \log t + \log n)} = 2^{O((d \log t + \log n) \log m)}$ when F is satisfiable, and
2. At least $2^{\Omega(\ell)} \geq 2^{\Omega(m/\log N)}$ when F is unsatisfiable.

Using a randomised reduction, the value of t is irrelevant and the resulting monotone complexity is

1. At most $m^{O(d+\log n)} = 2^{O((d+\log n)\log m)}$ when F is satisfiable, and
2. At least $2^{\Omega(\ell)} \geq 2^{\Omega(m/\log N)}$ when F is unsatisfiable.

Since we are reducing from ETH, the value $m = \mathsf{poly}(n)$ is already good enough, as a monotone circuit lower bound beyond $2^{\Omega(n)}$ (for an N-variate function) is not of much use for reductions to Partial-MCSP or PAC-learning (since we would need to output a circuit of this size or generate an exponential number of examples). The value d is then the main parameter we choose. There are two parameter regimes we will care about:

1. *($d = \log n$).* We obtain a formula upper bound which is smaller than the number N of input bits (a "junta"). The runtime lower bound based on ETH becomes $N^{\Omega(\log \log N)}$.
2. *($d \approx \sqrt{n}$).* This gives us a polynomial vs. quasipolynomial gap. The runtime lower bound becomes $N^{\widetilde{\Omega}(\log N)}$.

4.1 ETH-hardness of Monotone Circuit-Formula Gap Learning

We first show the result for juntas.

▶ **Theorem 4.3** (Junta Gap-Learning). *Let $\gamma > 0$ be a constant. For large enough $n \in \mathbb{N}$, the 3SAT problem on n variables reduces to $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ in time $2^{O(n/\log n)}$, where $s_1(N) = 2^{O(\log\log N)^3}$ and $s_2(N) = N^{\Omega(\log N)}$. Consequently, the problem $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ needs time $N^{\Omega(\log\log N)}$ to be solved under ETH.*

Proof. Let F be a given 3-CNF over n variables. Fix $d = \log n$. Note that $\mathsf{Ref}^G_s(F)$ is a formula of width $O(1)$ over $N = 2^{\Theta(n/\log n)}$ variables and with $N^{O(1)}$ clauses. Let

$$m := n^3/(\log n)^3, \quad t := n^2.$$

and let $\ell = \varepsilon dm/n$ for a sufficiently small constant $\varepsilon > 0$. The function $f^* := f_{\mathsf{Ref}^G_d(F),m}$ is defined on $n_0 := N^{O(1)}m^{O(1)} = 2^{O(n/\log n)}$ variables (Definition 2.3). Theorems 2.4 and 3.5 imply that f^* can be computed with monotone formulas of size $s := m^{O(d\log t+\log n)} = 2^{O(\log^3 n)} = 2^{O(\log\log N)^3}$ when F is satisfiable. Moreover, when F is unsatisfiable, the function requires monotone circuits of size at least

$$s' := 2^{\Omega(\ell)} = N^{\Omega(\log N)}$$

to be γ-approximated over the distribution $\mathcal{D} := \mathcal{D}^{\mathsf{Ref}^G_s(F),m}$ given by Theorem 2.4, since we can take ε sufficiently smaller than γ. This distribution can be constructed in time $2^{O(n/\log n)}$ given F and m. A postprocessing taking $2^{O(n/\log n)}$-time may be needed to adjust the description of the circuit to the required length (see Remark 4.2). The ETH lower bound follows by noticing that $N^{\log\log N} = 2^{O(n)}$. ◀

The polynomial vs. quasipolynomial gap follows similarly.

▶ **Theorem 4.4** (Polynomial vs. quasipolynomial gap). *Let $\gamma > 0$ be a constant. For large enough $n \in \mathbb{N}$, the 3SAT problem on n variables reduces to $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ in time $2^{O(\sqrt{n}\log n)}$, where $s_1(N) = N$ and $s_2(N) = N^{\Omega(\log N)}$. Consequently, the problem $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ needs time $N^{\Omega((\log N)/(\log\log N)^2)}$ to be solved under ETH.*

Proof. Let F be a given 3-CNF over n variables. Fix $d = \alpha\sqrt{n}/\log n$ where $\alpha > 0$ is small enough constant. Note that $\mathsf{Ref}^G_s(F)$ is a formula of width $O(1)$ over $N = 2^{\Theta(\sqrt{n}\log n)}$ variables and with $N^{O(1)}$ clauses. Let

$$m := (\log N)^2 \asymp n\log^2 n, \quad t := n^2.$$

and let $\ell = \varepsilon dm/n$ for a sufficiently small constant $\varepsilon > 0$. The function $f^* := f_{\mathsf{Ref}^G_d(F),m}$ is defined on $n_0 := N^{\Theta(1)}m^{\Theta(1)} = 2^{\Theta(\sqrt{n}\log n)} = N^{\Theta(1)}$ variables (Definition 2.3). Theorems 2.4 and 3.5 imply that f^* can be computed with monotone formulas of size

$$s := m^{O(d\log t+\log n)} = 2^{O(d\log^2 n+\log^2 n)} = 2^{O(\alpha\sqrt{n}\log n)} \le n_0,$$

when F is satisfiable and α is small enough. Moreover, when F is unsatisfiable, the function requires monotone circuits of size at least

$$s' := 2^{\Omega(\ell)} = N^{\Omega(\log N)}$$

to be γ-approximated over the distribution $\mathcal{D} := \mathcal{D}^{\mathsf{Ref}^G_s(F),m}$ given by Theorem 2.4, since we can take ε sufficiently smaller than γ. This distribution can be constructed in time $2^{O(\sqrt{n}\log n)}$ given F and m. The ETH lower bound follows by noticing that $N^{(\log N)/(\log\log N)^2} = 2^{O(n)}$. ◀

4.2 rETH-hardness of Monotone Circuit-Formula Gap Learning

We now show slightly stronger results but under rETH.

▶ **Theorem 4.5** (Junta Gap-Learning under rETH). *Let $\gamma > 0$ be a constant. For large enough $n \in \mathbb{N}$, the 3SAT problem on n variables randomly reduces to $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ in time $2^{O(n/\log n)}$, where $s_1(N) = (\log N)^{O(\log\log N)}$ and $s_2(N) = N^{\Omega(\log N)}$. Consequently, the problem $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ needs time $N^{\Omega(\log\log N)}$ to be solved under rETH.*

Proof. Let F be a given 3-CNF over n variables. Fix $d = \log n$. Note that $\mathsf{Ref}^G_s(F)$ is a formula of width $O(1)$ over $N = 2^{\Theta(n/\log n)}$ variables and with $N^{O(1)}$ clauses. Let

$$m := n^3/(\log n)^3,$$

and let $\ell = \varepsilon dm/n$ for a sufficiently small constant $\varepsilon > 0$. The function $f^* := f_{\mathsf{Ref}^G_d(F),m}$ is defined on $n_0 := N^{O(1)}m^{O(1)} = 2^{O(n/\log n)}$ variables (Definition 2.3). Theorems 2.4 and 3.5 imply that f^* can be computed with monotone formulas of size $s := m^{O(d+\log n)} = 2^{O(\log^2 n)} = 2^{O(\log\log N)^2}$. when F is satisfiable. The rest of the proof is the same as in Theorem 4.3. ◀

▶ **Theorem 4.6** (Polynomial vs. quasipolynomial gap under rETH). *Let $\gamma > 0$ be a constant. For large enough $n \in \mathbb{N}$, the 3SAT problem on n variables reduces to $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ in time $2^{O(\sqrt{n})}$, where $s_1(N) = N$ and $s_2(N) = N^{\Omega(\log N)}$. Consequently, the problem $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ needs time $N^{\Omega(\log N)}$ to be solved under rETH.*

Proof. Let F be a given 3-CNF over n variables. Fix $d = \alpha\sqrt{n}$ where $\alpha > 0$ is small enough constant. Note that $\mathsf{Ref}^G_s(F)$ is a formula of width $O(1)$ over $N = 2^{\Theta(\sqrt{n})}$ variables and with $N^{O(1)}$ clauses. Let

$$m := (\log N)^2 \asymp n,$$

and let $\ell = \varepsilon dm/n$ for a sufficiently small constant $\varepsilon > 0$. The function $f^* := f_{\mathsf{Ref}^G_d(F),m}$ is defined on $n_0 := N^{\Theta(1)}m^{\Theta(1)} = 2^{\Theta(\sqrt{n}\log n)}$ variables (Definition 2.3). Theorems 2.4 and 3.5 imply that f^* can be computed with monotone formulas of size

$$s := m^{O(d+\log n)} = 2^{O(d)} = 2^{O(\alpha\sqrt{n})} \le n_0,$$

when F is satisfiable and α is small enough. Moreover, when F is unsatisfiable, the function requires monotone circuits of size at least

$$s' := 2^{\Omega(\ell)} = N^{\Omega(\log N)}$$

to be γ-approximated over the distribution $\mathcal{D} := \mathcal{D}^{\mathsf{Ref}^G_s(F),m}$ given by Theorem 2.4, since we can take ε sufficiently smaller than γ. This distribution can be constructed in time $2^{O(\sqrt{n})}$ given F and m. The ETH lower bound follows by noticing that $N^{\log N} = 2^{O(n)}$. ◀

4.3 From Gap Learning to Partial-Gap-MCFSP

We now define the partial MCSP problem we will consider. Instead of the entire truth table of a partial function, we consider a more succinct encoding where we are only given locations where the function is defined.

▶ **Definition 4.7.** *Let $s_1, s_2, \varepsilon : \mathbb{N} \to \mathbb{N}$ be time-constructible functions. Let $\mathsf{mMCFSP}^{s_2}_{s_1}[\varepsilon]$ be the promise problem which receives as input 1^n and a list of m pairs $((x_1, b_1), \ldots, (x_m, b_m))$ (called* examples*), where $x_i \in \{0,1\}^n$ and $b_i \in \{0,1\}$ for all $i \in [m]$, and must distinguish between the following two cases:*

- *there is a monotone formula C of size at most s_1 such that, for all $i \in [m]$, $C(x_i) = b_i$;*
- *for all circuits C' of size less than s_2, we have $|\{i \in [m] : C'(x_i) \neq b_i\}| \geq m(1/2 - \varepsilon)$.*

The number n is called the dimension *of the input.*

To show hardness of mMCFSP, it suffices to reduce from mCFGL. As described in the introduction, it suffices to sample from the distribution $\mathcal{D}$ given as input to mCFGL, In the small formula case, we clearly generate examples which can be computed by a small formula. In the large circuit case, with enough examples we can conclude by the Chernoff inequality that no circuit of small size can compute most of those examples correctly.

▶ **Theorem 4.8.** *For every growing functions $s_1, s_2 : \mathbb{N} \to \mathbb{N}$ such that $s_1 \leq s_1' < s_2' \leq s_2$ (for large enough n) and $\gamma : \mathbb{N} \to [0,1]$, the problem $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$ with input $(1^n, \mathcal{D})$ randomly reduces to $\mathsf{mMCFSP}^{s_2'}_{s_1'}[2\gamma]$ in time $\mathsf{poly}(n) + O(\frac{1}{\gamma^2} \cdot s_2' \log s_2')$, producing $O(\frac{1}{\gamma^2} \cdot s_2' \log s_2')$ examples of dimension n.*

Proof. Let $\mathcal{D}$ be given as input for $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma]$. Take $m = \frac{4}{\gamma^2} \cdot K s_2' \log s_2'$ samples from $\mathcal{D}$, and call them $(x_1, b_1), ..., (x_m, b_m)$.

In the case that there exists a formula of size $s_1 < s_1'$ computing $\mathcal{D}$ then that same small circuit will agree with $(x_1, b_1), ..., (x_m, b_m)$ for all i. So "small" instances map to "small" instances. It will then suffice to prove the theorem to show that "large" instances map to "large" instances. In particular, it suffices to show that, with all but negligible probability, for every circuit of C' of size less than s_2', we have $|\{i \in [m] : C'(x_i) \neq b_i\}| \geq m(1/2 - \varepsilon - \gamma)$.

Let $a_{C'}$ be the number of $i \in [m]$ such that $C'(x_i) = b_i$. By definition of mCFGL, we know $\mathbb{P}_{(x,b) \leftarrow \mathcal{D}}[C'(x) = b] \leq 1/2 + \gamma$, and thus $\mathbb{E}[a_{C'}] \leq m(1/2 + \gamma)$. Therefore, for each circuit C', by Chernoff-Hoeffding's inequality we know that

$$\mathop{\mathbb{P}}_{(x_1,b_1),\ldots,(x_m,b_m) \leftarrow \mathcal{D}}[a_{C'} - m(1/2 + \gamma) \geq m\gamma] \leq e^{-2m^2\gamma^2/m} = e^{-2m\gamma^2} = (s_2')^{-4s_2'}, \tag{9}$$

or, equivalently,

$$\mathop{\mathbb{P}}_{(x_1,b_1),\ldots,(x_m,b_m) \leftarrow \mathcal{D}}[a_{C'} \geq m(1/2 + 2\gamma)] \leq (s_2')^{-4s_2'}.$$

There are less than $(s_2')^{3s_2'}$ monotone circuits with n bit inputs, s_2' gates, fan-in 2, and arbitrary fan-out. By union bounding over these circuits, we can see that the probability of any circuit satisfying $a_{C'} \geq m(1 + 2\gamma)$ is less than $1/3$ for large enough n. This completes the proof. ◀

We now employ the hardness results obtained for mCFGL under rETH in Section 4.2 to obtain hardness of $\mathsf{mMCFSP}^{s_2}_{s_1}$ using the reduction of the previous theorem. We consider 4 choice of parameters which are in a way best possible:

1. *Junta vs. quasipolynomial.* Here we take the largest gap possible between formula and circuit size. This gives us a weaker runtime lower bound of $N^{\Omega(\log\log N)^{1-\varepsilon}}$. This is the weakest runtime lower bound we obtain.
2. *Linear vs. quasipolynomial.* Here we obtain a weaker gap but against superpolynomial-size circuits. This gives us a better runtime lower bound of $N^{\Omega(\log N)^{1-\varepsilon}}$.

3. *Junta vs. polynomial.* We then consider the setting where the gap is largest possible subject to $s_2 = \mathsf{poly}(n)$. This gives us a runtime lower bound of $N^{\Omega(\log\log N)}$, slightly better than (1).
4. *Linear vs. polynomial.* We weaken the gap to an arbitrary polynomial n^c. This gives us a better runtime lower bound of $N^{\Omega(\log N)}$, slightly better than (2). This is the strongest runtime lower bound we obtain.

▶ **Corollary 4.9.** *Let $\gamma > 0$ be a constant. Under rETH, the following holds:*

1. ***(Junta vs. quasipolynomial)*** *For every constant $\varepsilon > 0$, the problem $\mathsf{mMCFSP}^{s_2}_{s_1}[\gamma]$ requires time $N^{\Omega(\log\log N)^{\varepsilon}}$ to be decided, where $s_1(n) = (\log N)^{\log\log N}$ and $s_2(n) = n^{(\log\log n)^{1-\varepsilon}}$. The hard instances consist of $m = \Theta(s_2 \log s_2)$ examples, and thus have size $N = nm = n^{\Theta(\log\log n)^{1-\varepsilon}}$.*
2. ***(Linear vs. quasipolynomial)*** *For every constant $\varepsilon \in (0,1)$, the problem $\mathsf{mMCFSP}^{s_2}_{s_1}[\gamma]$ requires time $N^{\Omega((\log N)^{2/(2-\varepsilon)})}$ to be decided, where $s_1(n) = n$, and $s_2(n) = n^{(\log n)^{1-\varepsilon}}$. The hard instances consist of $m = \Theta(s_2 \log s_2)$ examples, and thus have size $N = nm = n^{\Theta(\log n)^{1-\varepsilon}}$.*
3. ***(Junta vs. polynomial)*** *For every constant $\varepsilon > 0$, the problem $\mathsf{mMCFSP}^{s_2}_{s_1}[\gamma]$ requires time $N^{\Omega(\log\log N)}$ to be decided, where $s_1(n) = (\log N)^{\log\log N}$ and $s_2(n) = n^c$. The hard instances consist of $m = \Theta(s_2 \log s_2)$ examples, and thus have size $N = \Theta(nm) = \mathsf{poly}(n)$.*
4. ***(Linear vs. polynomial)*** *For every constant $c > 0$, the problem $\mathsf{mMCFSP}^{s_2}_{s_1}[\gamma]$ requires time $N^{\Omega(\log N)}$ to be decided, where $s_1(n) = n$, and $s_2(n) = n^c$. The hard instances consist of $m = \Theta(s_2 \log s_2)$ examples, and thus have size $N = \Theta(nm) = \mathsf{poly}(n)$.*

Proof. We first prove Item (1). By Theorem 4.5, for large enough $n \in \mathbb{N}$ the problem $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma/2]$ requires time $n^{\Omega(\log n)}$ to be computed under ETH for some $s_1 = 2^{O(\log\log n)^2}$ and $s_2 = n^{\Omega(\log n)}$. Take $s_1' = s_1$ and $s_2' = n^{(\log\log n)^{1-\varepsilon}}$, thus satisfying $s_1 < s_1' < s_2' < s_2$.

If $\mathsf{mMCFSP}^{s_2'}_{s_1'}[\gamma]$ can be computed in time $T(N)$ on an instance of $m = Ks_2' \log s_2' = n^{O(\log\log n)^{1-\varepsilon}}$ examples and dimension n (and thus input size $N = nm$), where K is a large enough constant, we obtain by Theorem 4.8 that $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma/2]$ can be solved in randomised time $\mathsf{poly}(n) + O(m) + T(N)$. This must be at least $n^{\Omega(\log\log n)}$ which implies $T(N) = n^{\Omega(\log\log n)}$. Since $N = n^{\Theta(\log n \log n)^{1-\varepsilon}}$, we obtain $\log n \log n = (1 - o(1)) \log\log N$ and

$$T(N) \geq N^{\Omega(\log\log N)^{\varepsilon}}.$$

We now prove Item (2). The proof is similar. Let $s_1' = n$ and $s_2' = n^{(\log n)^{1-\varepsilon}}$ in the argument above. We have that $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma/2]$ requires time $n^{\Omega(\log n)}$ to be computed by Theorem 4.6. Since $N \asymp ns_2' \log s_2' = n^{\Theta(\log n)^{1-\varepsilon}}$, we obtain $\log n \asymp (\log N)^{1/(2-\varepsilon)}$ and thus

$$T(N) \geq 2^{\Omega(\log n)^2} \geq 2^{\Omega(\log N)^{2/(2-\varepsilon)}}.$$

We repeat the argument for Item (3). Take $s_1' = (\log n)^{\log\log n}$ and $s_2' = n^c$. We have that $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma/2]$ requires time $n^{\Omega(\log\log n)}$ to be computed by Theorem 4.5. Since $N \asymp ns_2' \log s_2' \asymp n^{c+1} \log n$, we obtain $\log N \asymp \log n$ and $\log\log N \sim \log\log n$, and thus

$$T(N) \geq n^{\Omega(\log\log n)} \geq N^{\Omega(\log\log N)}.$$

Finally, we prove Item (4). Let $s_1' = n$ and let $s_2' = n^c$ in the argument above. We have that $\mathsf{mCFGL}^{s_2}_{s_1}[\gamma/2]$ requires time $n^{\Omega(\log n)}$ to be computed by Theorem 4.6. Since $N \asymp ns_2' \log s_2' \asymp n^{c+1} \log n$, we obtain $\log N \asymp \log n$ and thus

$$T(N) \geq 2^{\Omega(\log n)^2} \geq 2^{\Omega(\log N)^2}.$$

◀

4.4 From Gap Learning to Improper PAC-Learning

Given a function $c : \{0,1\}^n \to \{0,1\}$ and a distribution $\mathcal{D}$ with support in $\{0,1\}^n$, we denote by $\mathsf{Ex}(c, \mathcal{D})$ the oracle that runs in unit time and outputs a sample (x, b) where $x \leftarrow \mathcal{D}$ and $b = c(x)$. We also define $\mathsf{err}_{c,\mathcal{D}}(h) := \mathbb{P}_{x \in \mathcal{D}}[h(x) \neq c(x)]$.

▶ **Definition 4.10.** *Let $\mathcal{C}_n, \mathcal{H}_n$ be sets of Boolean functions over $\{0,1\}^n$, and let $\mathcal{C} = \bigcup_{n\geq 1} \mathcal{C}_n$ and $\mathcal{H} = \bigcup_{n\geq 1} \mathcal{H}_n$ be such that $\mathcal{C} \subseteq \mathcal{H}$. Let $T : \mathbb{N} \to \mathbb{N}$. We say that the* concept class $\mathcal{C}$ *is* PAC-learnable in time T *by the* hypothesis class $\mathcal{H}$ *if there is a randomised algorithm such that, for every $c \in \mathcal{C}$, given $(1^n, 1^{1/\varepsilon}, 1^{1/\delta})$ and oracle access to $\mathsf{Ex}(c, \mathcal{D})$, outputs in time $T(n + 1/\varepsilon + 1/\delta)$, with probability $1 - \delta$, a concept $h \in \mathcal{H}$ such that $\mathsf{err}_{c,\mathcal{D}}(h) \leq \varepsilon$.*

For $s : \mathbb{N} \to \mathbb{N}$, let $\mathcal{C}_s$ (resp. $\mathcal{L}_s$) be the set of monotone Boolean circuits (resp. formulas) on n bits of size $s(n)$. Let $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_{s(n)}$ and $\mathcal{L} = \bigcup_{n\in\mathbb{N}} \mathcal{L}_{s(n)}$. Hardness of PAC-learning $\mathcal{L}$ by $\mathcal{C}$ follows from the results of Section 4.2 by a proof similar to the one employed in the previous section.

▶ **Theorem 4.11.** *Under rETH, we have:*

1. *For every constant $\alpha > 0$, monotone formulas of size n require time $n^{\Omega(\log n)}$ to be PAC-learned by monotone circuits of size $s(n)$, for any $s(n)$ such that $n \ll s(n) \leq n^{(\log n)^{1-\alpha}}$.*
2. *For every constant $\alpha > 0$, monotone formulas of size $(\log n)^{O(\log\log n)}$ require time $n^{\Omega(\log\log n)}$ to be PAC-learned by monotone circuits of size $s(n)$, for any $s(n)$ such that $n \ll s(n) \leq n^{(\log\log n)^{1-\alpha}}$.*

Proof. We first prove the first item. The proof of the second item is the same, so we omit it. Let $\varepsilon = \delta = 1/3$. Let $(1^n, \mathcal{D})$ be an input to $\mathsf{mCFGL}^{s_2}_{s_1}[\varepsilon]$, where s_1, s_2 are as in Theorem 4.4. Let $s'_1 = n$ and $s'_2 = s(n)$. Suppose there is an PAC-learner $\mathcal{A}$ for $\mathcal{L}_n$ outputting a hypothesis in $\mathcal{C}_{s'_2}$ in time $T = T(n + 1/\varepsilon + 1/\delta)$.

Run the algorithm $\mathcal{A}$, replacing oracle calls to Ex with samples from $\mathcal{D}$, and obtaining a circuit C' at the end. Since $|\mathcal{D}| = \mathsf{poly}(n)$, this can be done in time $T \cdot \mathsf{poly}(n) + O(s'_2 \log s'_2)$, as we also need $O(s'_2 \log s'_2)$ bits to describe C'. We now take $O(\frac{1}{\gamma^2})$ samples from $\mathcal{D}'$, and accept $\mathcal{D}$ if C' computes correctly $2/3$ of the examples. In the YES case, we will be correct with probability $2/3$ since there is a formula of size s_1 computing correctly with probability 1. In the NO case, we know that $\mathbb{P}_{(x,b)\leftarrow\mathcal{D}}[C'(x) = b] \leq 1/2+\gamma$, and thus the number of examples $a_{C'}$ that C' computes correctly is less than $2/3$ with high probability by inequality (9). The total runtime is $T\mathsf{poly}(n) + O(s'_2 \log s'_2 + 1/\gamma^2)$, implying that $T = n^{\Omega(\log n)}$. ◀

References

1 Michael Alekhnovich, Mark Braverman, Vitaly Feldman, Adam R. Klivans, and Toniann Pitassi. The complexity of properly learning simple concept classes. *J. Comput. Syst. Sci.*, 74(1):16–34, 2008. doi:10.1016/J.JCSS.2007.04.011.

2 Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. *SIAM J. Comput.*, 38(1):63–84, 2008. doi:10.1137/060664537.

3 Benny Applebaum, Boaz Barak, and David Xiao. On basing lower-bounds for learning on worst-case assumptions. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science FOCS '08*, pages 211–220, 2008. doi:10.1109/FOCS.2008.35.

4 Noel Arteche, Albert Atserias, Susanna F. de Rezende, and Erfan Khaniki. The proof analysis problem. In *Proceedings of the 66th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 2555–2576, 2025. doi:10.1109/focs63196.2025.00133.

5 Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, 2008. Preliminary version in *CCC '03*. doi:10.1016/J.JCSS.2007.06.025.

6 Albert Atserias and Moritz Müller. Automating resolution is NP-hard. *Journal of the ACM*, 67(5), 2020. Preliminary version in *FOCS '19*. doi:10.1145/3409472.

7 Gaia Carenini and Susanna F. de Rezende. On the automatability of tree-like k-dnf resolution. In *Proceedings of the 40th Computational Complexity Conference (CCC '25)*, volume 339 of *LIPIcs*, pages 14:1–14:21, 2025. doi:10.4230/LIPIcs.CCC.2025.14.

8 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In Ran Raz, editor, *Proceedings of the 31st Conference on Computational Complexity (CCC '16)*, volume 50 of *LIPIcs*, pages 10:1–10:24, 2016. doi:10.4230/LIPIcs.CCC.2016.10.

9 Dana Dachman-Soled, Homin K. Lee, Tal Malkin, Rocco A. Servedio, Andrew Wan, and Hoeteck Wee. Optimal cryptographic hardness of learning monotone functions. *Theory Comput.*, 5(1):257–282, 2009. doi:10.4086/TOC.2009.V005A013.

10 Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing (STOC 16)*, pages 105–117, 2016. doi:10.1145/2897518.2897520.

11 Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning DNF's. In *Proceedings of the 29th Conference on Learning Theory (COLT '16)*, pages 815–830, 2016. URL: http://proceedings.mlr.press/v49/daniely16.html.

12 Amit Daniely and Gal Vardi. From local pseudorandom generators to hardness of learning. In *Proceedings of the 34th Conference on Learning Theory (COLT '21)*, pages 1358–1394. PMLR, 2021. URL: http://proceedings.mlr.press/v134/daniely21a.html.

13 Susanna F. de Rezende. Automating tree-like resolution in time $n^{o(\log n)}$ is ETH-hard. In *Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium, (LAGOS '21)*, volume 195, pages 152–162, 2021. doi:10.1016/J.PROCS.2021.11.021.

14 Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is NP-hard. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21)*, pages 209–222, 2021. doi:10.1145/3406325.3451080.

15 Susanna F. de Rezende and Marc Vinyals. Lifting with colourful sunflowers. In *Proceedings of the 40th Computational Complexity Conference (CCC 25)*, pages 36:1–36:19, 2025. doi:10.4230/LIPIcs.CCC.2025.36.

16 Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing (STOC 02)*, pages 534–543, 2002. doi:10.1145/509907.509985.

17 Vitaly Feldman, Homin K. Lee, and Rocco A. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. In *Proceedings of the 24th Conference on Learning Theory (COLT '11)*, pages 273–292, 2011. URL: http://proceedings.mlr.press/v19/feldman11a/feldman11a.html.

18 Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory of Computing*, 16(1):1–30, 2020. Preliminary version in *STOC '18*. doi:10.4086/toc.2020.v016a013.

19 Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 38:1–38:19, 2019. doi:10.4230/LIPIcs.ITCS.2019.38.

20 Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi. Automating cutting planes is NP-hard. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC 20)*, pages 68–77, 2020. doi:10.1145/3357713.3384248.

21 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. *SIAM J. Comput.*, 47(6):2435–2450, 2018. Preliminary version in FOCS '15. doi:10.1137/16M1059369.

22 Shuichi Hirahara. NP-hardness of learning programs and partial MCSP. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science FOCS 22*, pages 968–979, 2022. doi:10.1109/FOCS54457.2022.00095.

23 Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. *SIAM J. Comput.*, 52(6):S18–349, 2023. doi:10.1137/19M124705X.

24 Shuichi Hirahara, Igor C. Oliveira, and Rahul Santhanam. NP-hardness of minimum circuit size problem for OR-AND-MOD circuits. In *Proceedings of the 33rd Computational Complexity Conference (CCC 18)*, pages 5:1–5:31, 2018. doi:10.4230/LIPIcs.CCC.2018.5.

25 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43(04):439–562, August 2006. doi:10.1090/s0273-0979-06-01126-8.

26 Yizhi Huang, Rahul Ilango, and Hanlin Ren. NP-hardness of approximating meta-complexity: A cryptographic approach. *SIAM J. Comput.*, 54(4):819–886, 2025. doi:10.1137/23M1608483.

27 Rahul Ilango. Approaching MCSP from above and below: Hardness for a conditional variant and $\mathrm{AC}^0[p]$. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS '20)*, pages 34:1–34:26, 2020. doi:10.4230/LIPIcs.ITCS.2020.34.

28 Rahul Ilango. Constant depth formula and partial function versions of MCSP are hard. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS '20)*, pages 424–433, 2020. doi:10.1109/FOCS46700.2020.00047.

29 Rahul Ilango. The minimum formula size problem is (ETH) hard. In *Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS '21)*, pages 427–432, 2021. doi:10.1109/FOCS52979.2021.00050.

30 Rahul Ilango, Bruno Loff, and Igor C. Oliveira. NP-hardness of circuit minimization for multi-output functions. In *Proceedings of the 35th Computational Complexity Conference (CCC '20)*, pages 22:1–22:36, 2020. doi:10.4230/LIPIcs.CCC.2020.22.

31 Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the 10th Annual IEEE Structure in Complexity Theory Conference (SCT 95)*, pages 134–147, 1995. doi:10.1109/SCT.1995.514853.

32 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC '00)*, pages 73–79, 2000. doi:10.1145/335305.335314.

33 Michael J. Kearns and Leslie G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *J. ACM*, 41(1):67–95, 1994. doi:10.1145/174644.174647.

34 Michael Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC '93)*, pages 372–381, 1993. doi:10.1145/167088.167197.

35 Jan Kraíček. Interpolation by a game. *Mathematical Logic Quarterly*, 44(4):450–458, 1998. doi:10.1002/MALQ.19980440403.

36 Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997. doi:10.2307/2275541.

37 Shachar Lovett, Raghu Meka, Ian Mertz, Toniann Pitassi, and Jiapeng Zhang. Lifting with sunflowers. In *Proceedings of 13th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 104:1–104:24, 2022. doi:10.4230/LIPIcs.ITCS.2022.104.

38 William Masek. Some NP-complete set covering problems. Unpublished, 1979.

39 Noam Mazor and Rafael Pass. Gap MCSP is not (Levin) NP-complete in Obfustopia. In *Proceedings of the 39th Computational Complexity Conference, (CCC '24)*, pages 36:1–36:21, 2024. doi:10.4230/LIPIcs.CCC.2024.36.

40 Pavel Pudlák. On reducibility and symmetry of disjoint NP pairs. *Theoretical Computer Science*, 295:323–339, 2003. doi:10.1016/S0304-3975(02)00411-5.

41 Ran Raz and Pierre McKenzie. Separation of the Monotone NC Hierarchy. *Combinatorica*, 19(3):403–435, 1999. Preliminary version in *STOC* 97. doi:10.1007/S004930050062.

42 Alexander A Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya: mathematics*, 59(1):205, 1995.

43 Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997. doi:10.1006/JCSS.1997.1494.

44 Rahul Santhanam. Pseudorandomness and the minimum circuit size problem. In Thomas Vidick, editor, *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS '20)*, pages 68:1–68:26, 2020. doi:10.4230/LIPIcs.ITCS.2020.68.

Systematic Data Structure Lower Bounds via the Query-With-Sketch Model

Sumegha Garg
Rutgers University, New Brunswick, NJ, USA

Songhua He
Rutgers University, New Brunswick, NJ, USA

Yuanzhi Li
Carnegie Mellon University, Pittsburgh, PA, USA

Periklis A. Papakonstantinou
Rutgers University, New Brunswick, NJ, USA

Xin Yang
Meta, Seattle, WA, USA

Abstract

We study data structure lower bounds for the *Approximate Matrix Powering* (AMP) problem. Given a substochastic, symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ and parameters k and α, the goal is to preprocess $\mathbf{M}$ so as to answer entry queries $(u, v) \mapsto \mathbf{M}^k[u, v]$ up to additive error $1/n^\alpha$. We focus on AMP in the *succinct and systematic* regime, in which the data structure stores $\mathbf{M}$ verbatim, uses an additional r bits of redundancy, and must answer queries by probing only a small number of entries of $\mathbf{M}$.

Our main conceptual contribution is a general framework for proving probe–redundancy trade-offs for systematic data structures. We introduce the *query-with-sketch* model and develop a min-entropy-based approach that lifts conditional min-entropy bounds in the absence of redundancy to probe lower bounds in the presence of redundancy. We then establish these min-entropy bounds using problem-specific analytic and algebraic tools, for the downstream applications to AMP and its variants. As a consequence, our results provide new unconditional evidence toward a conjecture of Pătraşcu and Roditty on the space required for constant-time set-disjointness queries [31].

2012 ACM Subject Classification Theory of computation → Data structures design and analysis

Keywords and phrases systematic data structure, query-with-sketch model, approximate matrix powering

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.41

Acknowledgements We thank the anonymous reviewers for their valuable suggestions.

1 Introduction

Unconditional lower bounds are among the clearest windows we have into the limits of computation. In the data structure world this promise is especially tangible: computation is free, and the only scarce resources are *information* – how many bits one stores, and how many locations one probes. A rich line of work has explored the tradeoff between these two resources across a variety of models, including the standard cell-probe model [43], as well as more restrictive frameworks such as succinct data structures [23] – under which the space usage is close to the information-theoretic minimum. While the strongest known lower bounds on the number of probes per query are logarithmic in the cell probe model [25], linear lower bounds are known for succinct data structures (e.g., [13]). In this paper, we develop a new framework for proving such tradeoffs in the succinct data structure regime, under an additional restriction: the data structure stores the input in read-only memory, together

© Sumegha Garg, Songhua He, Yuanzhi Li, Periklis A. Papakonstantinou, and Xin Yang; licensed under Creative Commons License CC-BY 4.0

with an auxiliary data structure of at most r bits. Data structures of this form are known as *systematic* data structures[1]. Lower bounds in this model have been studied extensively in prior work [27, 13, 18, 19, 20, 6, 7].

A guiding application of our lower-bound framework is the *Approximate Matrix Powering* (AMP) problem. Given a substochastic, symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ and parameters α and k, the goal is to preprocess $\mathbf{M}$ so as to answer entry queries $(u, v) \mapsto \mathbf{M}^k[u, v]$ up to additive error $1/n^\alpha$. AMP is a fundamental computational task that arises in the analysis of the long-term behavior of Markov chains [30], in iterative methods from numerical linear algebra [17], and in PageRank and related random-walk-based algorithms [5]. AMP subsumes, as special cases, the counting set intersection problem as well as its decision version, set disjointness. These set problems are central throughout algorithms and complexity theory (e.g., [8, 31, 14, 15, 24]). In this work, we obtain (to our knowledge) the first lower bounds for set intersection and set disjointness in the systematic and static data structure model.

We model a static data structure problem as a function $f : \mathcal{X}^N \times \mathcal{Q} \to \mathcal{Y}$, where $\mathcal{X}^N$ denotes the space of possible inputs (data), $\mathcal{Q}$ denotes the set of queries, and $f(X, \phi)$ is the answer to query $\phi \in \mathcal{Q}$ on input $X \in \mathcal{X}^N$. The computation proceeds in two phases. In the preprocessing phase, a systematic data structure with unbounded computational power, given access to the input X, stores X verbatim together with an additional $r = o(N)$ bits of auxiliary information, called the *redundancy*. Subsequently, upon receiving a query $\phi \in \mathcal{Q}$, the data structure is given the redundancy bits for free and must output $f(X, \phi)$ while making only a bounded number of probes to the input X. After answering the query, the data structure is allowed to update its redundancy bits[2].

To prove general lower bounds for systematic data structures, we formalize and analyze a query-with-sketch model using a novel min-entropy lemma (Lemma 8). The key power of this lemma is that it shifts the quantifier in the analysis: instead of reasoning about the information gained from a sequence of t probes conditioned on r bits of redundancy, we reason about the entropy loss in output after the algorithm sees t arbitrary input elements (without any redundancy). This reduction is the core of our framework, as it allows us to set aside the data structure algorithm itself and focus purely on the problem's inherent properties, and it can be summarized as follows.

▶ **Theorem 1** (Lifting min-entropy to data structure lower bounds). *Fix a data structure problem $f : \mathcal{X}^N \times \mathcal{Q} \to \mathcal{Y}$. Fix the redundancy parameter r. If there exists a sequence of m queries $Q \in \mathcal{Q}^m$, and an input distribution μ over $\mathcal{X}^N$ such that:*

> *For* every *partial assignment σ to at most $o(N)$ elements of the input, the min-entropy of the answers to these m queries remains high. That is, the probability of* any *single answer vector $\mathbf{y}$ is exponentially small:*
>
> $$\max_{\mathbf{y} \in \mathcal{Y}^m} \Pr_{X \leftarrow \mu} \left[(f(X, Q_1), \ldots, f(X, Q_m)) = \mathbf{y} \mid \sigma \right] \leq 2^{-2r}.$$

[1] It is well known that data structure lower bounds have deep connections to other areas of theoretical computer science, including rigidity bounds [10, 33], circuit lower bounds [41, 11], cryptography [16], and fine-grained complexity [21, 9]. Importantly, [9] show that improving the state-of-the-art data structure lower bounds for the function inversion problem – under systematic and non-adaptive data structures – would also imply new circuit lower bounds in Valiant's common-bits model [38, 39].

[2] Previous works [13, 7] on succinct and systematic data structures assume probe access to the redundancy bits, and do not explicitly allow the redundancy to be updated after each query. Consider the Prefix Sum problem over a bitstring $X \in \{0, 1\}^N$, where the data structure should support querying the parity of a prefix $\bigoplus_{i=1}^{k} X_i$, given k. [13] proved that any succinct and systematic data structure with r bits of redundancy requires $q = \Omega(N/(r+1))$ probes to answer parity queries. However, if we allow the data structure to update its redundancy between queries, this trade-off disappears. For the sequential query order $k = 1, \ldots, N$, a data structure can simply maintain the running parity, achieving constant costs even in the worst-case. This simple example demonstrates the separation between read-only and read-write access to redundancy.

Then, any (possibly randomized) succinct and systematic data structure (with read-write access to r bits of redundancy) that answers all queries in Q correctly with $\geq 99/100$ probability, must make $\Omega(N/m)$ probes per query on average. Here, a probe returns an element of $\mathcal{X}$.

This theorem provides a unifying template for all of our results. For a given data structure problem, we partition the query set $\mathcal{Q}$ into a sequence of blocks, each containing m queries. Let t denote the amortized probe complexity of a systematic data structure D for answering all the queries. To establish a tradeoff of the form $r \cdot t = \tilde{\Omega}(N)$, we will show that the problem satisfies the min-entropy condition with block size $m = \tilde{\Theta}(r)$. This approach yields an amortized lower bound even when D knows the entire query sequence in advance and is allowed to adaptively update its redundancy after each query. Note that the min-entropy condition must hold for *every* partial assignment: even if a benevolent adversary reveals the $o(N)$ most informative input elements, knowing the m queries, there must still remain sufficient entropy in the answers. This is a strong guarantee and is technically challenging to establish for our downstream applications to AMP. See Figure 1 for outline of our results.

Comparison with previous works

Previous lower bounds for succinct and systematic data structures have been obtained for problems such as rank and fully indexable dictionaries [27, 18, 19, 20], succinct sampling from discrete distributions [6], online matrix–vector multiplication [7], and substring search, prefix sum, polynomial evaluation, and related problems [13]. In particular, Gal and Miltersen [13] established a general lower bound of the form $(r+1)t \geq \Omega(N)$ for problems satisfying a list-decoding property: namely, that the entire list of answers can be recovered from any sufficiently large subset. While this property holds for problems with strong global algebraic structure – such as polynomial evaluation, which serves as the main example in [13] – it fails for many natural problems in which answers are more local and largely independent, including the approximate matrix powering problem studied in this work.

Our approach departs from these reconstruction-based arguments. Many previous bounds, including those of [13, 7], proceed by encoding information about the input from the behavior of a low-probe data structure on a carefully chosen sequence of queries, and then compare the resulting encoding length with an entropy lower bound for the input distribution. Such arguments naturally lead to inequalities on the total transcript length, and hence to probe–redundancy tradeoffs after the query sequence is optimized. By contrast, our query-with-sketch reduction does not attempt to reconstruct the input. Instead, it reduces lower bounds to showing that the answer vector to a block of queries retains large conditional min-entropy even after an arbitrary small set of input locations has been revealed. Thus the information-theoretic object in our framework is the residual uncertainty of the answers under partial information, rather than the entropy of the input itself.

1.1 Applications to Approximate Matrix Powering

In this section, we detail our results for the fundamental problem of Approximate Matrix Powering.

▶ **Definition 2** (Approximate Matrix Powering (AMP)). *Fix parameters $\alpha > 0$ (a large enough constant) and an integer k ($2 \leq k \leq n$). The data structure must solve the following task:*

- ***Input:*** *A substochastic and symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$.*
- ***Query:*** *A pair of indices $(u, v) \in [n] \times [n]$.*
- ***Output:*** *An estimation $\hat{y} \in \mathbb{R}$ that approximates the corresponding entry of the k-th matrix power within a specified additive error:*

$$\left|\hat{y} - \mathbf{M}^k[u, v]\right| \leq \frac{1}{n^\alpha}$$

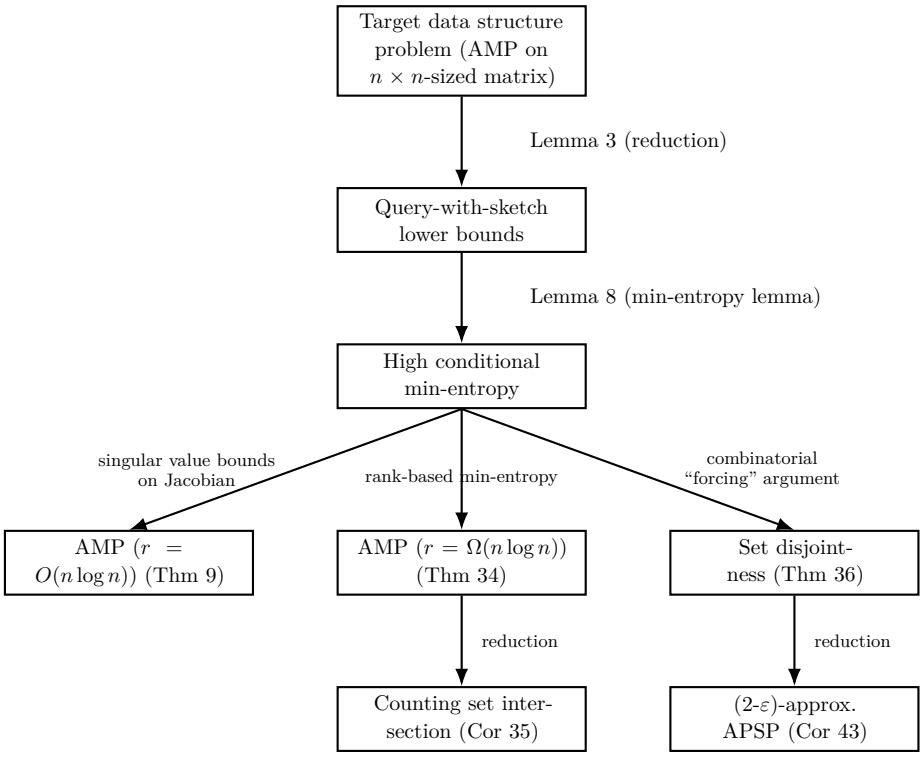


Figure 1 Our proof strategy. We reduce data structure bounds to a query-with-sketch bound (Lemma 3), which we then reduce to an information-theoretic min-entropy question (Lemma 8). We solve the latter by applying a different analytical technique for each problem.

This problem is closely related to the online matrix-vector multiplication (OMV) problem studied by [7]. That work establishes a tradeoff of $r \cdot t = \Omega(n^3)$ for any succinct or systematic data structure with r bits of redundancy and amortized probe complexity t, for answering Boolean matrix-vector multiplication queries. We emphasize that OMV and AMP are incomparable problems. On the one hand, AMP can be viewed as potentially harder than OMV, since it requires computing entries of higher powers of the matrix. On the other hand, AMP is easier in the following aspect: in OMV, the vector is part of the query and is not stored in the data structure, leading to 2^n possible queries, whereas in AMP the queries range over matrix entries and there are only n^2 possible queries.

Our data structure lower bound for the AMP problem is as follows. Since AMP involves real-valued matrices, to avoid discussion on word size, we assume that each probe returns an entry of the matrix.

Theorem (Informal statement for Theorem 9 and Theorem 34)

Fix parameters α, k. Any (possibly randomized) succinct and systematic data structure of r bits of redundancy that supports answering approximated elements of the k-th power of a substochastic and symmetric matrix $\mathbf{M}$ should make $\Omega(n^2/r)$ probes per query on average. This holds for $r \in [\Omega(n \log n), O(n^2)]$ for $k = 2$, and $r \in [\Omega(\log^2 n), O(n^2)]$ for every $k \in [3, n]$.

Since computing elements of a matrix square is a special case of matrix powering, can always be performed using $O(n)$ probes, the above lower bound holds for a full range of probe complexity. Interestingly, the cases of sublinear-in-n (Theorem 9) and superlinear-in-n

(Theorem 34) redundancy use different hard input distributions and proof techniques, where either does not seem to extend to the full range. We give a proof overview in the next subsection.

For general k, the naive algorithms give only straightforward upper bounds obtained by reading enough entries of M to compute the requested entry. We are not aware of nontrivial succinct systematic data structures for AMP that match the above probe–redundancy tradeoff across the full parameter range. Thus our result should be viewed as a lower bound establishing a new barrier for this problem, rather than as a tight characterization.

We next state our results for data structure problems on sets and graphs.

Set Intersection

First, our proof of Theorem 34 (AMP with superlinear-in-n redundancy) also yields an identical lower bound for the set intersection problem and its counting variant. Set intersection is a fundamental data structure problem, which has received significant attention in fine-grained complexity [8, 31, 14, 15, 24]. In this problem, the input consists of n subsets of the universe $[n]$, and the goal is to answer queries asking for the size of the intersection between the u-th and v-th sets. We show that, under the same parameter regime, answering each query requires $\Omega(n^2/r)$ probes on average. See Corollary 35 for the full statement.

To the best of our knowledge, data structure lower bounds for the set intersection problem were previously known only for restricted data structures that are in comparable to the succinct and systematic setting. For instance, in the pointer machine model [35], which disallows random access and forces data to be accessed by traversing pointers stored in cells, [1] showed tradeoff of $S(N)Q(N) = \Omega(N^{2-o(1)})$, where N is the total size of the input sets, $S(N)$ is the total space of the data structure, and $Q(N)$ is the probe cost in pointers followed.

Set disjointness

In 2010, Pătraşcu and Roditty [31] conjectured that any cell-probe data structure which preprocesses n sets $A_1, \ldots, A_n \subseteq [\text{polylog}(n)]$ and answers set disjointness queries, that is, whether the u-th and v-th sets intersect, using constant probe complexity must use $\tilde{\Omega}(n^2)$ space. We study a related version of the set disjointness problem in which the sets $A_1, \ldots, A_n$ are subsets of $[n]$, and make the first progress toward this conjecture in the systematic data structure setting. In particular, we prove a time–space tradeoff of the form $r \cdot t = \Omega\left(\frac{n^2}{\log^3 n}\right)$, for any succinct and systematic data structure solving set disjointness.

Theorem (Informal statement for Theorem 36)

Any (possibly randomized) succinct and systematic data structure using r bits of redundancy, that answers set disjointness queries, should make $\Omega(n^2/(r \log^3 n))$ probes on average per query. Here, $r \in [\Omega(n \log n), O(n\sqrt{n}/\log^3 n)]$.

We note that the range of r is optimal for our hard input distribution, in which the total size of all sets is $O(n\sqrt{n})$. Moreover, the probe complexity is always at most $O(n)$, since a set disjointness query can be answered by probing only the two sets involved in the query.

All Pair Shortest Paths (APSP)

By a reduction from set disjointness, we also obtain a data structure lower bound for $(2-\varepsilon)$-approximate all-pairs shortest paths (APSP). Specifically, any succinct and systematic data structure that answers a sequence of distance queries within a multiplicative factor strictly

less than 2, must either use r bits of redundancy or incur an average probe complexity of $\Omega(n^2/(r \log^3 n))$. This tradeoff holds for $r \in [\Omega(n \log n), O(n\sqrt{n}/\log^3 n)]$. See Corollary 43 for the full statement.

Data structures for answering approximate distance queries are commonly referred to as *distance oracles*. Thorup and Zwick [37] showed that there exists a distance oracle with $O(n^{3/2})$ space that answers distance queries within a factor of 3 in constant time. Subsequently, Pătraşcu and Roditty [31] constructed an $O(n^{5/3})$-space data structure that returns an estimate of at most $2d + 1$ in constant time, where d is the true distance. On the lower-bound side, [31] established a conditional result showing that any cell-probe data structure achieving a 2-approximation in constant time must use $\tilde{\Omega}(n^2)$ space. Our result provides the first *unconditional* time-space tradeoff for approximate APSP. Moreover, the tradeoff is tight: when $r = \Omega(n^{3/2})$, the distance oracles from prior work can be stored entirely within the redundancy.

1.2 Proof Overview and Technical Contributions

In this section, we outline the proofs of our main results. We introduce the *query-with-sketch* model, which captures data structures that store a short sketch of the input and answer a fixed batch (or sequence) of queries using limited probes to the raw input. We then show a general *min-entropy lemma* that reduces lower bounds in this model to a conditional min-entropy bound. Finally, we instantiate this template for AMP using different analytic tools for different parameter regimes, obtaining strong probe–redundancy trade-offs.

1.2.1 The query-with-sketch model and the min-entropy lemma

First, we introduce the query-with-sketch model. The final data structure lower bounds consist of a reduction to this model, and a min-entropy lemma that transfers min-entropy upper bounds to query-with-sketch lower bounds. These definitions and reductions are formalized at Section 2.

The query-with-sketch model can be thought of as a succinct and systematic data structure with its queries fixed beforehand[3]. An algorithm in the query-with-sketch model aims to compute a multi-bit function $f : \{0,1\}^N \to \{0,1\}^M$ (or more generally, $f : \mathcal{X}^N \to \mathcal{Y}$) on x with the help of a short piece of information that we call "sketch", and limited number of coordinate probes to x. The sketch size should be smaller than N and M for the model to make sense. A more detailed definition and some notation are deferred to Section 2.1.

Capabilities and limits of algorithms in the query-with-sketch

The query-with-sketch model, as a natural information-theoretic model, is of independent interest. We use the Hamming weight problem to illustrate that sketch can help reduce the query (probe) complexity in a non-trivial way. In the Hamming weight problem, the input is a string from $\{0,1\}^n$ and the output is the number of 1s. In the standard query model, n probes are needed to compute, with high probability, the exact answer to this problem. But, in the query-with-sketch, given a small sketch (shorter than $\log n$ bits) of the lower-order

[3] We adopt this name from the query complexity literature, where a "query" typically refers to a low-level read from the input. However, to maintain clarity and align with data structure conventions, this paper will consistently use "probe" for a low-level read from the input X and "query" for the high-level request. Thus, what the query complexity literature calls "query complexity" corresponds directly to what we call "probe complexity" in this paper.

digits in the hamming weight, it suffices to sample n^{ε}-many bits, $\varepsilon < 1$, to compute the higher-order digits. The intuition is that the least significant digits are harder to compute than the more significant ones.

Min-entropy lemma

All of our main results are built upon the following lemma, which provides a powerful connection between probe complexity and information. This min-entropy lemma formalizes a simple yet effective intuition: an r-bit sketch is useless if the answer would have remained more than r bits uncertain, for any set of probes the algorithm could have made.

It translates the task of proving a time-space tradeoff into a more manageable, information-theoretic question: how much uncertainty (min-entropy) about the answer $f(X)$ remains, even after we are given the values of any q locations in the input X?

Lemma (Informal statement of Lemma 8)

Let $f : \mathcal{X}^N \to \mathcal{Y}$ be a function over an adversarial distribution $\mu_{\mathcal{X}^N}$. Let Π^S be a deterministic algorithm for f using an r-bit sketch S. If for some integer q, the min-entropy of $f(X)$ remains $> r$ conditioned on any partial assignment on X of length q, then Π^S must make $> \Omega(q)$ probes on average, even given sketch S.

Here is the high-level intuition of the min-entropy lemma. The proof (see Section 2.2) views any query-with-sketch algorithm as a collection of 2^r decision trees. The r-bit sketch $S(X)$ simply tells the algorithm which one of these trees to execute. A leaf at depth q in any tree corresponds to the answers from q specific probes.

Our lemma's condition says that for any such leaf, the final answer $f(X)$ is still highly ambiguous – there are more than 2^r possible correct answers consistent with that probe path. Since the r-bit sketch (which was fixed from the start) cannot possibly resolve this ambiguity, the algorithm cannot halt and must make more probes.

This lemma is powerful, but our main results require a technical strengthening. We show that the same lower bound holds even if the high min-entropy condition is only guaranteed to apply to a large, high-probability set of "good inputs". See Lemma 8 for the full, formal statement.

Connection to succinct and systematic data structures

The query-with-sketch complexity was studied for the direct sum of parity problems [29, 2], and was applied to obtain data structure lower bounds for the prefix sum problem [13]. However, the reduction in [13] was restricted to the prefix sum problem. We defer the discussion on this reduction to Section 2.3. Below, we give a reduction that is generally applicable to many data structure problems.

▶ **Lemma 3** (Reduction from data structures to query-with-sketch algorithms). *Fix a data structure problem $f : \mathcal{X}^N \times \mathcal{Q} \to \mathcal{Y}$. Suppose there is a (possibly randomized) succinct and systematic data structure D with the following properties:*

- *It has a redundancy of r bits that one can update for free.*
- *For a fixed sequence of queries $Q = (\kappa_1, \ldots, \kappa_l) \in \mathcal{Q}^l$, it answers the entire sequence correctly with probability at least $99/100$.*
- *The total number of probes made to answer the entire sequence Q is at most tl in the worst case.*

Then for any integer $m \in [l]$ that divides l, there exists a starting index $i \in \{1, m+1, 2m+1, \ldots, l-m+1\}$ and a (randomized) query-with-sketch algorithm Π^S for answering the subsequence $(\kappa_i, \ldots, \kappa_{i+m-1})$ that has:

- *An r-bit sketch S.*
- *An average-case probe complexity of at most tm.*
- *A success probability of at least $99/100$.*

While our reduction applies to randomized data structures, our min-entropy lemma applies to deterministic query-with-sketch algorithms. This gap can be fixed by applying Yao's minimax principle, while we are studying the distributional complexity of data structures and query-with-sketch algorithms.

We note that our reduction applies to common variants of data structure problems in the literature, including those where the goal is to answer a single query (e.g., [13]) and those that answer, or are motivated by, a sequence of queries (e.g., [7]).

Furthermore, the lemma's requirement for success on all queries simultaneously is not a strong restriction. An algorithm that only guarantees to answer each of the l queries correctly with high probability (e.g., $\geq 1 - o(1)$) can be converted to one that satisfies our stricter requirement. This is achieved via standard probability amplification, which increases the total probe complexity by a multiplicative factor of $O(\log l)$.

We also note that, as a general lower bounding tool, our approach inherently cannot prove time-space trade-offs stronger than $r \cdot t = \Omega(N \log |\mathcal{Y}|)$. A nontrivial query-with-sketch algorithm always has $r \leq m \log |\mathcal{Y}|$, since $m \log |\mathcal{Y}|$ bits are sufficient to encode all the answers of the m consecutive queries. In addition, a nontrivial query-with-sketch algorithm never makes more than N probes to the input, which implies that $t \leq N/m$ in our reduction. The two facts together imply that there always exists query-with-sketch algorithms achieving $r \cdot t \leq N \log |\mathcal{Y}|$, for every $r \geq 1$. To the best of our knowledge, among previous works [26, 19, 18, 13, 20, 6, 7], only a recent breakthrough [7] overcame this barrier to achieve a stronger trade-off for the OuMv (online vector-matrix-vector multiplication) problem. They achieved it by showing that any efficient algorithm for OuMv would implicitly function as a compression scheme that violates Shannon's source coding theorem. While powerful, this argument is highly tailored to the algebraic properties of matrix multiplication, and the massive 2^n probe space. It does not provide a general template for breaking the $r \cdot t$ trade-off barrier.

1.2.2 Approximate matrix powering with sublinear-in-n redundancy

Warm up: a lower bound to approximating the matrix square

Let us begin our proof outline for a restricted form of the full lower bound. Our lower bound holds for symmetric matrices and for arbitrary matrix powers $k > 2$. In the proof outline we will only discuss the lower bound for squaring, $\mathbf{M}^2$, not-necessarily symmetric matrices $\mathbf{M}$. These are the only simplifying assumptions made for this high-level exposition. Specifically, in this proof overview we will show that every succinct and systematic data structure for approximating the matrix square $\mathbf{M}^2$ either requires $r = 0.5n$ bits of redundancy or $t = \Omega(n)$ number of probes for every query. After that, we show how we extend it to an $r \cdot t = \Omega(n^2)$ lower bound for general sublinear-in-n r, for computing $\mathbf{M}^k$.

While we accept every possible output within an $1/n^\alpha$ additive error, we restrict the AMP problem to a version where the output is fixed, which enables us to apply the query-with-sketch reduction. Specifically, fix a large enough constant α', we let the output to be

$[\mathbf{M}^k]_{\alpha'}$, a rounded matrix power where each element is rounded to its nearest multiple of $1/n^{\alpha'}$. We will show in Lemma 13 that the reduction to this rounded version will only incur negligible error probability, when α' is large enough compared to α.

Our two-step reduction (Lemma 3 and Lemma 8) implies that, one only needs (i) a nemesis input distribution $\mathcal{M}$; (ii) a sequence Q of n^2 queries; and (iii) to show that for every $m = r$ consecutive queries from Q, even when conditioned on knowing any $0.001n^2$ elements from $\mathbf{M}$, the answers remain an $\Omega(r)$ min-entropy.

We construct a nemesis input distribution $\mathcal{M}_{\beta,\gamma}$ of the form $\mathbf{M} = \frac{n^2-1}{n^2}\mathbf{I} + \frac{1}{n^\beta}\mathbf{M_U}$, where $\mathbf{M_U}$ is a random matrix where each element is i.i.d. (independent and identically distributed) uniformly random from the range $[\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]$. For the reader's convenience, we may consider an instance where $\beta = 2$ and $\gamma = 4$. In the actual proof, $\mathbf{M_U}$ is a uniformly random symmetric matrix, and a similar proof applies. In addition, we construct the sequence of queries Q in a way that every r consecutive queries span r different rows and columns. To illustrate our technique in this warm-up, focus on an illustrative block of r diagonal queries of approximating $\mathbf{M}^2[1,1], \mathbf{M}^2[2,2], \ldots, \mathbf{M}^2[r,r]$.

Notice that $\mathbf{M}^2 = (\frac{n^2-1}{n^2})^2\mathbf{I} + \frac{2}{n^2} \cdot \frac{n^2-1}{n^2}\mathbf{M_U} + \frac{1}{n^4}{\mathbf{M_U}}^2$. We shall assume that the elements $\mathbf{M}[1,1], \mathbf{M}[2,2], \ldots, \mathbf{M}[r,r]$ are revealed to the algorithm for free, since this will only decrease the probe complexity. We will show that, however, the second-order term ${\mathbf{M_U}}^2$ remains highly uncertain.

More formally, we use σ to denote an arbitrary *partial assignment* to $0.001n^2$ elements of $\mathbf{M}$. Our reductions (the reduction from data structure lower bounds to query-with-sketch lower bounds, and the reduction to the min-entropy of the problem) help us get rid of analyzing the algorithms, and reduce it to a task of showing that the output has a high conditional min-entropy:

$$\max_{\sigma,(y_i)_{i\in[r]}} \Pr_{\mathbf{M}\leftarrow\mathcal{M}}[[\mathbf{M}^2[1,1]]_{\alpha'} = y_1, \ldots, [\mathbf{M}^2[r,r]]_{\alpha'} = y_r|\sigma] < 2^{-2r}. \tag{1}$$

Our strategy is to select $\Omega(r)$ many input elements that are not fixed by σ, so that their *Jacobian matrix* with the output elements is a diagonal matrix. That is, (if given all the other $n - \Omega(r)$ elements fixed), each output element is an increasing function of its corresponding input element, but does not depend on other elements. We call these selected elements *core* elements. We can achieve it because each element of the matrix square only depends on input elements of the same row or the same column.

More precisely, for each output element $\mathbf{M}^2[u,u]$, we try to find an input element $\mathbf{M}[u,i]$ such that $i > r$, and $\mathbf{M}[u,i]$ is not fixed by σ. That is, the element shares the same row or column with $\mathbf{M}^2[u,u]$, but does not share the same row or column with any other output element. We use $\mathbf{J}$ to denote its Jacobian matrix with the output elements, where each row corresponds to a core element $\mathbf{M}[u_i,i]$, each column corresponds to an output element $\mathbf{M}^2[u,u]$, and each $\mathbf{J}[i,u] := \frac{\partial \mathbf{M}^2[u,u]}{\partial \mathbf{M}[u_i,i]}$ quantifies how the rate $\mathbf{M}^2[u,u]$ will increase if we perturb $\mathbf{M}[u_i,i]$ by a little bit. Intuitively, the higher the rate is, the less likely the output will evaluate to a fixed value, which yields a min-entropy bound.

$$\mathbf{J} = \begin{pmatrix} \Omega(1/n^3) & 0 & \ldots & 0 \\ 0 & \Omega(1/n^3) & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & \ldots & \Omega(1/n^3) \end{pmatrix}$$

We can always select $r' = \Omega(r)$ core elements since a partial assignment σ of length $0.001n^2$ can only fix a small proportion of elements at the first $r = 0.5n$ rows and the first r columns.

Given these core elements $\mathbf{M}[i_1, j_1], \ldots, \mathbf{M}[i_{r'}, j_{r'}]$, we may discard all the other elements of $\mathbf{M}$. Specifically, upper bounding the conditional probability in inequality (1) can be reduced to upper bound

$$\max_{\sigma', (y_i)_{i\in[r]}} \Pr_{\mathbf{M}\leftarrow\mathcal{M}}[[\mathbf{M}^2[1,1]]_{\alpha'} = y_1, \ldots, [\mathbf{M}^2[r,r]]_{\alpha'} = y_r | \sigma'] \tag{2}$$

where σ' is any partial assignment for all the other elements of $\mathbf{M}$ except core elements, such that σ' is consistent with σ. This is because the previous probability is a weighted sum of the latter one.[4]

The Jacobian matrix tells us that given σ', the output elements are mutually independent. For each core element $\mathbf{M}[i,j]$, its corresponding output element $\mathbf{M}^2[u,u]$, and each possible output y_u, given that $\frac{\partial \mathbf{M}^2[u,u]}{\partial \mathbf{M}[i,j]} = \Omega(1/n^3)$, we have

$$\Pr_{\mathbf{M}\leftarrow\mathcal{M}}[[\mathbf{M}^2[u,u]]_{\alpha'} = y_u | \sigma'] \leq O(1/n^{\alpha'+9})$$

which is obtained by dividing the output length $(1/n^{\alpha'})$ by the partial derivative $(\Omega(1/n^3))$ and the range length that the core element is uniformly selected from $(1/n^{\gamma+\beta} = 1/n^6)$.

Therefore, the probability in (2) is upper bounded by $O((1/n^{\alpha'+9})^{r'}) = O(1/2^{\Omega(r\log n)})$, which implies a desired $\Omega(r \log n)$ conditional min-entropy.

From $k = 2$ to $k \geq 3$: singular value bounds to the Jacobian matrix

While each element of the matrix square only depends on $2n-1$ elements of the input matrix, one cannot get a stronger probe lower bound for a smaller redundancy r. Therefore, we generalize the above arguments to $3 \leq k \leq n$, where an $r \cdot t = \Omega(n^2)$ lower bound still holds.

We use the same input distribution $\mathcal{M}$ and the sequence of queries Q as above. When $k \geq 3$, the expansion $\mathbf{M}^k = \sum_{l=0}^{k} \binom{k}{l} (\frac{n^2-1}{n^2})^{k-l} (\frac{1}{n^\beta})^l {\mathbf{M}_\mathbf{U}}^l$ is still dominated by the terms where t are small. Again, we fix and focus on a sequence of output elements $\mathbf{M}^k[1,1], \ldots, \mathbf{M}^k[r,r]$ for some $r = o(n)$.

Given a partial assignment of length $\leq 0.001n^2$, unlike the previous case, all the elements of the first r rows and columns may be fixed by σ. However, we are still going to show that the output remains a high min-entropy. Our proof aligns with the proof outline above, but with the following edits:

First, we partition the set of input elements of $\mathbf{M}$ from the last $n-r$ columns and $n-r$ rows into disjoint subsets of size r, such that each pair of elements from the same subset does not share the same row or the same column. In this way, given a partial assignment σ of length $\leq 0.001n^2$, there will be at least one subset where $r' = \Omega(r)$ elements in it are not fixed by σ. We choose these elements as our core elements.

Then, denote by $(i_1, j_1), \ldots, (i_{r'}, j_{r'})$ the core elements. We look into the Jacobian matrix between the core elements and the output elements again. Since the r output elements and r' core (input) elements span different rows and columns, their Jacobian matrix is of the following form for $K := \binom{k}{3} \frac{(n^2-1)^{k-3}}{n^{2k-6+2\beta}}$.

---

[4] For a discrete probability space and two events A, B, $\Pr[A|B] = \sum_{B'} \Pr[A|B' \cap B] \cdot \Pr[B'|B]$, where the summation runs over a partition of B space; and thus, $\Pr[A|B] \leq \Pr[A|B']$, for some $B' \subseteq B$.

$$\mathbf{J} = \begin{pmatrix} K\mathbf{M_U}[1,i_1]\mathbf{M_U}[j_1,1] + O(k^5/n^{2+3\beta}) & \dots & \dots & K\mathbf{M_U}[r,i_1]\mathbf{M_U}[j_1,r] + O(k^5/n^{2+3\beta}) \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ K\mathbf{M_U}[1,i_{r'}]\mathbf{M_U}[j_{r'},1] + O(k^5/n^{2+3\beta}) & \dots & \dots & K\mathbf{M_U}[r,i_{r'}]\mathbf{M_U}[j_{r'},r] + O(k^5/n^{2+3\beta}) \end{pmatrix}$$

This matrix $\mathbf{J}$ is a perturbed random matrix. The dominant term for each entry is a product of two random $\mathbf{M_U}$ elements, while the $O(k^5/n^{2+3\beta})$ term is a small, deterministic-like perturbation. Critically, because all $r + r'$ input and output indices are distinct, each $\mathbf{M_U}$ element appears at most once in the dominant terms of $\mathbf{J}$. This makes the dominant terms i.i.d. random variables.

We apply a concentration bound on its singular values [42] to show that the largest $0.9r'$ singular values of $\mathbf{J}$ are high with high probability. Intuitively, the singular values of a matrix measure the "stretch" it applies to its input vectors. A matrix with $\Omega(r')$ large singular values is robustly high-rank.

This high-rank property is the key to our min-entropy bound. We use it to bound the pre-image of a fixed output. If the map "stretches" the input space in $\Omega(r')$ dimensions, then to map to a tiny output region (our rounded answer box of side length $1/n^{\alpha'}$), the input must originate from an exponentially small-volume region. This small volume, under our uniform distribution $\mathcal{M}$, translates directly to an exponentially small probability, which is the high conditional min-entropy we need.

By a union bound over the (polynomially many) possible choices for the core element set, we can define our "good input set" $\mathcal{M}$ for which the conditional min-entropy can be bounded as the set where this high-rank property holds for all relevant Jacobians.

More specifically, given σ', we show that if two core element assignments $\mathbf{x}, \mathbf{x}'$ map to the same output $\mathbf{y}$ (within precision $1/n^{\alpha'}$), they must be geometrically close, if a constant proportion of singular values of $\mathbf{J}$ are large.

In fact, for any fixed output $\mathbf{y}$, the set of all consistent core inputs $\mathbf{x}$ is *concentrated* into an exponentially small-volume region. Under our uniform input distribution, this small volume implies an exponentially small probability, which confirms the high conditional min-entropy.

We omitted some details that will be clarified in the full proof.

1.2.3 Approximate matrix powering with superlinear-in-n redundancy

The above arguments do not easily extend to the case $r = \omega(n \log n)$, since the Jacobian matrix is no longer controlled when there are more than n core elements and more than n output elements. The singular value concentration bound [42] only works for random matrices whose elements are independent. To that end, we construct and study the following combinatorial distribution of matrices.

To better illustrate the distribution, we describe its transition graph $G = (V, E, W)$ instead, which is an undirected graph with edge weight to be its transition probability from both sides. Consider a vertex set $[n]$ that is partitioned to three parts V_1, V_2, V_3, each of size exactly $n/3$. For every pair of vertices from $V_1 \times V_2$ or $V_2 \times V_3$, we connect them by an edge of weight $1/n^3$ with independent probability $1/2$. In addition, we connect each vertex to itself by an edge of weight $1 - 1/n^2$. Note that as a substochastic matrix, we do not require each row sum of the matrix to be 1, but less than or equal to 1.

We construct our sequence of queries as the sequence of all pairs of vertices between V_1 and V_3. The sequence is constructed in a way that among every r consecutive queries, each vertex is mentioned by $\Theta(r/n)$ times.

Intuitively, the graph is a 3-layered graph with heavy lazy transitions. We will show in Observation 29 that approximating the matrix power of the transition graph reduces to counting the number of length-2 walks from each $u \in V_1$ to $v \in V_3$. Due to the heavy lazy transitions, the matrix power is dominated by the length-2 walks. In fact, we will show that even computing the *parity* of the number of walks is hard.

Although the nemesis input distribution is combinatorial, the analysis remains linear-algebraic in spirit. We will try to find a basis of the input to the output that preserves the min-entropy.

Given a partial assignment of length $\leq 0.001n^2$, we remove those vertices whose $\geq 0.01n$ neighbors are revealed. For each of the remaining vertices v from V_3, there are $O(r/n)$ queries on v. We denote by $(u_1, v), \ldots, (u_m, v) \in V_1 \times V_3$ the queries on v, for $m = O(r/n)$. We select m corresponding vertices $w_1, \ldots, w_m \in V_2$ such that none of the edges $\forall i \in [m], (w_i, v)$ are revealed by σ.

As above, we fix a partial assignment σ' that reveals all the input edges other than the selected edges above, and is consistent with σ. Lower-bounding the min-entropy conditioned on σ reduces to bounding the min-entropy conditioned on every such σ'. Given that the whole subgraph between V_1 and V_2 is revealed by σ', queries on different vertices $v \in V_3$ are independent, and the min-entropy is the sum of min-entropy on each individual v.

Fix a vertex $v \in V_3$, denote by $\hat{\mathbf{M}} \in \mathbb{F}_2^{m \times m}$ the indicator matrix of whether each $u_i \in V_1$ is connected to w_j. Denote by $\mathbf{x} \in \mathbb{F}_2^m$ the indicator vector of whether each $(w_i, v) \in E$ or not, which is a uniformly random vector conditioned on σ'. The parity of the output (the number of length-2 paths between each u_i and v) is exactly $\hat{\mathbf{M}}\mathbf{x}$. Using a classic result that a random binary matrix has a high rank with high probability ([3]), we show that with high probability every $\hat{\mathbf{M}}$ with $m \geq 10 \log n$ has a high rank, which implies a high min-entropy of the output.

1.2.4 Lower bounds to $(2 - \varepsilon)$-approximate APSP

In APSP, we study the *existence* instead of the parity of the number of length-2 paths between pairs of vertices. The graph distribution defined above is too dense, forcing the distance between vertices from V_1 and V_3 to be 2 almost surely. To that end, we work on its sparse variant instead. We still consider a graph $G = (V, E)$ of three layers V_1, V_2, V_3 with $n/3$ vertices each. However, for each pair of vertices $(u, w) \in V_1 \times V_2$ or $(w, v) \in V_2 \times V_3$, we connect them with independent probability $1/\sqrt{n} \log n$. For each pair of vertices $(u, v) \in V_1 \times V_3$, they have a length-2 path if and only if there exists a vertex $w \in V_2$ such that $(u, w), (w, v) \in E$, i.e., whether the neighbor sets of u and v intersect. Since the entropy of the graph is $\Theta(n\sqrt{n}/\log n)$, we prove an $\Omega(n^2/\log^3 n)$ time-space trade-off for every $\Omega(n \log^2 n) \leq r \leq O(n\sqrt{n}/\log^3 n)$, where the range of r is nearly tight.

A similar proof to the above can no longer apply here. While the output can still be formulated as a matrix-vector product of the form $\hat{\mathbf{M}}\mathbf{x}$, the arithmetic is over semirings (i.e., ANDs and ORs). There does not exist a natural notion of a "basis" in this setting.

Instead, we introduce a combinatorial *forcing argument*. We show that for a sparse graph, a single "disjoint" (0) answer for a query (u, v) *forces* a large number of potential edges to be non-existent (0). The high min-entropy comes from two facts: (i) with high probability a large proportion $(1 - 1/\text{polylog}(n))$ of answers are 0 (Lemma 42); and (ii) only with exponentially small probability that *many* such forcing events (from a block of queries) can be simultaneously satisfied by a random graph.

We omitted details on how to apply the forcing argument to the input graph conditioned on a partial assignment σ, and how we deal with partial assignments that reveal too much information about the answer. We deal with these problems formally in Section 5.

2 Query-with-sketch model and the min-entropy lemma

In this section we give the two abstract ingredients used throughout the paper: the query-with-sketch model and a min-entropy lemma for proving lower bounds in this model. We then show how lower bounds in the query-with-sketch model imply lower bounds for succinct and systematic data structures.

Because the query-with-sketch model will serve as the target of our lower bounds, we make its execution model explicit: the sketch is fixed from the input, subsequent probes may be chosen adaptively from the transcript, and the algorithm eventually halts with an output.

2.1 Query-with-sketch model

We define the query-with-sketch (QwS) model for a fixed function $f : \mathcal{X}^N \to \mathcal{Y}$; in contrast, a data structure is typically specified by a query problem $f : \mathcal{X}^N \times \mathcal{Q} \to \mathcal{Y}$, where the query $Q \in \mathcal{Q}$ determines which value $f(X, Q)$ must be returned.

▶ **Definition 4** (Query-with-sketch algorithms). *Let $f : \mathcal{X}^N \to \mathcal{Y}$ be a function. A deterministic query-with-sketch algorithm Π^S with sketch length r consists of two objects.*

- *A sketching map $S : \mathcal{X}^N \to \{0,1\}^r$. On input X, the algorithm is given the sketch $S(X)$.*
- *A deterministic transition rule Π. For every $l \geq 0$, given a transcript*

$$C_\ell = \big(S(X), (i_1, a_1), \ldots, (i_\ell, a_\ell)\big),$$

where $i_j \in [N]$ is a probed coordinate and $a_j \in \mathcal{X}$ is the value returned by the probe, the rule Π either outputs a new probe $i_{\ell+1} \in [N]$ or halts with an answer $y \in \mathcal{Y}$.

The computation on input X starts from $C_0 = (S(X))$. If $\Pi(C_\ell) = i_{\ell+1} \in [N]$, the algorithm probes coordinate $i_{\ell+1}$, receives $a_{\ell+1} = X_{i_{\ell+1}}$, and continues with $C_{\ell+1}$. If $\Pi(C_\ell) = y \in \mathcal{Y}$, the algorithm halts and outputs y.

We allow repeated probes, and count them with multiplicity. A transcript of length ℓ induces a partial assignment to at most ℓ distinct input coordinates. Let $\mathsf{T}_{\Pi^S}(X)$ denote the number of probes made before halting, with $\mathsf{T}_{\Pi^S}(X) = \infty$ if the computation does not halt. The worst-case and average-case probe complexities are

$$\mathrm{Cost}(\Pi^S) := \sup_{X \in \mathcal{X}^N} \mathsf{T}_{\Pi^S}(X), \qquad \text{Avg-Cost}_\mu(\Pi^S) := \mathbb{E}_{X \sim \mu}\big[\mathsf{T}_{\Pi^S}(X)\big].$$

▶ **Definition 5** (Randomized query-with-sketch algorithms). *A randomized query-with-sketch algorithm is a distribution over deterministic query-with-sketch algorithms. Equivalently, a random tape ρ is sampled before the sketch is formed, and the algorithm then uses the deterministic pair (S_ρ, Π_ρ) throughout the computation.*

The worst-case sketch length and worst-case probe complexity are the maximum over all random tapes and all inputs. The average-case probe complexity with respect to an input distribution μ is

$$\mathbb{E}_\rho \mathbb{E}_{X \sim \mu}\big[T_{\Pi_\rho^{S_\rho}}(X)\big].$$

The success probability is taken over both $X \sim \mu$ and the random tape ρ.

Query-with-sketch and related models

As a natural information-theoretic model of computation, variants of the query-with-sketch model have been introduced and studied in different scenarios. The query-with-sketch model is a direct generalization of the decision tree model with *help bits* [29, 2]. In that model, algorithms compute the direct sum of k instances of a boolean function using $k-1$ help bits, which can be viewed as a sketch of length $k-1$. In addition, each output bit is computed using an individual decision tree in their case, whereas in our setting, all the output bits are decided in a single decision tree. The lower bound to succinct but unrestricted data structures for the partial sums problem [32] also reduces to lower bounds to data structures with *published bits*. However, the model used in this work is substantially different from ours. The queries are made to a data structure instead of the input elements.

2.2 The min-entropy lemma

We show that lower bounds to query-with-sketch algorithms can be reduced to lower bounding the conditional joint min-entropy of the output.

▶ **Definition 6** (joint min-entropy)**.** *Let X be a random variable and Y an event. The* joint min-entropy *of X and Y is defined as $H_{\min}(X,Y) := \min_x -\log \Pr[X = x \wedge Y]$. When $\Pr[Y] = 0$, $H_{\min}(X,Y)$ is defined as $+\infty$.*

We relate the complexity of query-with-sketch algorithms with a special form of conditional joint min-entropy of the output. To formally state our result, let us give necessary definitions.

▶ **Definition 7.** *A* partial assignment *to a function $f : \mathcal{X}^N \to \mathcal{Y}$ of length q is an event of the form $(X_{i_1} = x_{i_1}, \ldots, X_{i_q} = x_{i_q})$, where $i_1, \ldots, i_q \in [N]$ are distinct coordinates and $x_{i_j} \in \mathcal{X}$ for every $j \in [q]$. We use $\mathcal{E}_{\leq q}$ to denote the set of all partial assignments of length at most q.*

It is important to note that a partial assignment is an event about the function itself and is irrelevant to the algorithm.

▶ **Lemma 8.** *Let $f : \mathcal{X}^N \to \mathcal{Y}$ be an arbitrary function, and $\mu_{\mathcal{X}^N}$ an distribution over $\mathcal{X}^N$. Fix Π^S to be a deterministic query-with-sketch algorithm for f with average-case probe complexity $\leq 0.1q$.*

In addition, suppose there exists a set $\mathsf{X} \subseteq \mathcal{X}^N$ of "good" inputs such that

$$\Pr_{X \leftarrow \mu_{\mathcal{X}^N}}[X \in \mathsf{X}] \geq 99/100.$$

Furthermore, assume there is a positive integer $r \geq 2$ such that for every partial assignment $\sigma \in \mathcal{E}_{\leq q}$ of length $\leq q$, there exists a subset of inputs $\mathsf{X}_\sigma \subseteq \mathcal{X}^N$ such that

$$\Pr_{X \leftarrow \mu_{\mathcal{X}^N}}[X \in \mathsf{X} \backslash \mathsf{X}_\sigma | \sigma] \leq 2^{-2r}$$

and the output of f remains highly uncertain:

$$H_{\min}(f(X), X \in \mathsf{X} \cap \mathsf{X}_\sigma | \sigma) > 2r$$

Then, for Π^S to achieve $\geq 99/100$ success probability given a random input from $\mu_{\mathcal{X}^N}$, its sketch length must be $|S| > r$.

For our purpose of proving a query-with-sketch lower bound for randomized algorithms given a fixed sequence of queries, we could prove distributional lower bounds for deterministic query-with-sketch algorithms instead, by the standard Yao's minimax principle.

The sets $\mathsf{X}, \mathsf{X}_\sigma$ describe good inputs where the min-entropy can be well-bounded. We note that X and X_σ are optional. If we set $\mathsf{X} = \mathsf{X}_\sigma = \mathcal{X}^N$ for every σ, there is no constraint on the inputs.

Proof of Lemma 8. Assume for the sake of contradiction that the sketch length is $\leq r$. Without loss of generality, we let the sketch length be r. We will show that the success probability of Π^S cannot achieve $\geq 99/100$, which is a contradiction.

We characterize Π^S as a set of 2^r decision trees. Each input X is assigned to a decision tree T_S indexed by its sketch S. $f(X)$ is computed correctly if T_S given input X produces the correct answer $f(X)$. Each leaf of T_S can be characterized by a pair (y, σ), corresponding to the output at the leaf and the probe history. Let E_S denote the event that "X is assigned with sketch S", and E_q the event that "Π^S given input X makes no more than q probes". Then $\Pr[E_q] \geq 0.9$, or the average probe complexity would exceed $0.1q$. We have

$$
\begin{aligned}
&\Pr_{X \leftarrow \mu_{\mathcal{X}^N}}[\Pi^S(X) = f(X)] \\
=&\sum_{S \in \{0,1\}^r} \sum_{(y,\sigma) \text{ is a leaf of } T_S} \Pr[E_S \text{ and } \sigma \text{ and } f(X) = y] \\
\leq& 0.11 + \sum_{S \in \{0,1\}^r} \sum_{\substack{(y,\sigma) \text{ is a leaf of } T_S \\ \text{of depth } \leq q}} \Pr[E_S \text{ and } \sigma \text{ and } f(X) = y \text{ and } X \in \mathsf{X} \text{ and } E_q] \\
\leq& 0.11 + \sum_{S \in \{0,1\}^r} \sum_{\substack{(y,\sigma) \text{ is a leaf of } T_S \\ \text{of depth } \leq q}} \Pr[\sigma] \cdot \Pr[f(X) = y \text{ and } X \in \mathsf{X} \text{ and } E_q | \sigma] \\
\leq& 0.11 + \sum_{S \in \{0,1\}^r} \sum_{\substack{(y,\sigma) \text{ is a leaf of } T_S \\ \text{of depth } \leq q}} \Pr[\sigma] \cdot (\Pr[f(X) = y \text{ and } X \in \mathsf{X} \cap \mathsf{X}_\sigma \text{ and } E_q | \sigma] + 2^{-2r}) \\
<& 0.11 + \sum_{S \in \{0,1\}^r} \sum_{\substack{(y,\sigma) \text{ is a leaf of } T_S \\ \text{of depth } \leq q}} \Pr[\sigma] \cdot 2^{-2r+1} \\
\leq& 0.11 + \sum_{S \in \{0,1\}^r} 2^{-2r+1} \\
\leq& 2^{-r+1} + 0.11 < 0.99
\end{aligned}
$$

The second line follows our assumption that $X \in \mathsf{X}$ with $\geq 99/100$ probability. The fourth and the fifth line follow our assumption to the conditional joint min-entropy and $\Pr[X \in \mathsf{X}_\sigma | \sigma]$. The sixth line follows from the fact that the partial assignments σ of different leaves of the same decision tree are disjoint events of the input. ◀

2.3 Reduction from data structures

We prove Lemma 3, which reduces data structure lower bounds to query-with-sketch lower bounds.

Proof to Lemma 3. The data structure D can be characterized as a sequence of l/m query-with-sketch algorithms for computing each block of m queries. The sketch to each algorithm is the (updated) redundancy after the end of its previous algorithm. The probe made by all the query-with-sketch algorithms are exactly the probe made by D to answer Q. In

addition, the error probability of each query-with-sketch algorithm is at most the error probability of the data structure D. Besides, in both algorithms, the succinct space (sketch) exclusively depends on the input $X \in \mathcal{X}^n$. Among these algorithms, at least one of them has average-case probe complexity below average $\leq tm$. ◀

Through the above reduction and min-entropy lemma, to obtain a data structure lower bound one only need to (i) construct an input distribution $\mu_{\mathcal{X}^n}$, a subset X of $\mathcal{X}^N$, and subsets X_σ for every σ; (ii) show that for every partial assignment $\sigma \in \mathcal{E}_{\leq q}$, $H_{\min}(f(X), X \in \mathsf{X} \cap \mathsf{X}_\sigma | \sigma) > 2r$; and (iii) show that with high probability a random input falls in X (resp., falls in X_σ conditioned on σ, for every σ).

Comparison to a previous reduction

The query-with-sketch complexity of the *direct sum* of boolean problems was introduced and studied in [29, 2]. They showed that any l decision trees for computing the direct sum of the parity of m bits each, with $l-1$ help bits (i.e., the sketch), at least one of the decision trees must have a depth $\geq m$. Later, [13] got a data structure lower bound for prefix sum by a reduction from the help-bits lower bound for direct-sum parity. However, their reduction is specific to the prefix sum problem. Our reduction goes in a different way and is generally applicable to succinct and systematic data structures whose min-entropy can be bounded. In addition, our reduction is applicable to every problem f whose conditional min-entropy is bounded. The remainder of this paper will be devoted to showing the min-entropy bounds on the aforementioned problems. To compare our approach with the previous work, we explain the lower bound in [13] below.

In the prefix sum problem, the input is a length-n bit string X. The data structure, given an arbitrary index $k \in [n]$, should output the parity of $\sum_{i=1}^{k} X_i$. In [13], they showed that every succinct and systematic data structure for the prefix sum with r bits of redundancy requires $\Omega(n/(r+1))$ probes to the input bits. Their proof idea is as follows. Assume for contradiction that there exists a data structure with r bits of redundancy that makes $< \frac{n}{2(r+1)}$ bit probes for every given index $k \in [n]$. Split the input string X into $r+1$ blocks of about equal length, the parity of each block can be determined using $< n/(r+1)$ probes. However, [29] shows that even with r help bits (i.e., the sketch), to compute $r+1$ instances of parity of $n/(r+1)$ bits, at least one of the $r+1$ decision trees has depth $\geq n/(r+1)$.

3 Lower bounds to approximate matrix powering with sublinear-in-n redundancy

In this section, we study the approximate matrix powering problem. For this problem, we establish an $r \cdot t = \Omega(n^2)$ trade-off for data structures with redundancy $r < 0.25n \ln n$ and amortized probe complexity t. The subsequent section addresses the case where $r \in [10n \log n, n^2/1000]$, proving the same trade-off with a different technique. Let us begin by defining our rounding notation and then formally state the problem.

Approximate matrix powering (AMP) problem

Fix a constant $\alpha > 0$ and a number $k \geq 2$. The input is a substochastic symmetric matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$. The succinct and systematic data structure, given a pair of indices $(u, v) \in [n] \times [n]$, must produce a value ans such that $|\mathbf{M}^k[u, v] - ans| \leq 1/n^\alpha$.

▶ **Theorem 9** (Lower bound to AMP with sublinear-in-n redundancy). *Fix $3 \le k \le n$ and a sufficiently large constant α. For every (possibly randomized) succinct and systematic data structure with $r \in (c\log^2 n, 0.25n\ln n)$ bits of redundancy that answers the whole approximated matrix power $\mathbf{M}^k$ with additive error $1/n^\alpha$ with $\ge 9/10$ probability, its total number of probes to $\mathbf{M}$ is $\Omega(n^4/r)$ at the worst case, where c is a large enough constant.*

We note that we prove a strong statement where in average, answering each query requires $\Omega(n^2/r)$ probes.

We give a distributional lower bound to the following nemesis input distribution.

Nemesis input distribution

Fix a large enough integer $n \ge 1$, and β, γ large enough constant parameters that depends on α. Let $\mathcal{M}_{\beta,\gamma}$ be a distribution of $n \times n$ matrices $\mathbf{M} = \frac{n^2-1}{n^2}\mathbf{I} + \frac{1}{n^\beta}\mathbf{M_U}$, where $\mathbf{M_U}$ is a random symmetric matrix whose upper triangle elements $\mathbf{M}[i,j]$ are independent and identically distributed (i.i.d.) uniformly from $[\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]$, for every $i \le j$. We write $\mathbf{M} \leftarrow \mathcal{M}_{\beta,\gamma}$ (or simply $\mathbf{M} \leftarrow \mathcal{M}$) for $\mathbf{M}$ sampled from distribution $\mathcal{M}_{\beta,\gamma}$.

To apply Lemma 3, we also need to fix a sequence of queries (i.e., a permutation to the set of possible queries to the data structure $[n] \times [n]$). We will show that to answer each of the consecutive $3m := 3r/\ln n$ queries, a query-with-sketch algorithm either requires r bits of redundancy or $\Omega(n^2)$ probes to the input matrix.

Specifically, we fix the following sequence of queries, which is a concatenation of $(n-1)$ shifted main diagonals of $\mathbf{M}^k$ (excluding the main diagonal itself):

$$Q = (\underbrace{(1,2),(2,3),\ldots,(n,1)}_{\text{Shift 1: } v=u+1}, \underbrace{(1,3),(2,4),\ldots,(n,2)}_{\text{Shift 2: } v=u+2}, \ldots, \underbrace{(1,n),(2,1),\ldots,(n,n-1)}_{\text{Shift } n-1\text{: } v=u-1})$$

Formally, the sequence Q consists of $n-1$ blocks, where for every $i \in [n-1]$, the i-th block contains n queries (u,v) defined by:

- $u \in [n]$;
- $v = ((u+i-1) \bmod n) + 1$.

All we need from the construction is the following property.

▶ **Observation 10.** *For any window of $3j$ consecutive queries from Q (where $3j < n$), one can select a subset of j queries from this window such that all the $2j$ indices (that is, both the row u and column v from each selected query) are pairwise distinct.*

Proof. First, we show that any $3j$ consecutive queries span $3j$ different rows and $3j$ different columns.

Since $3j < n$, the $3j$ consecutive queries must belong to at most 2 different shifted main diagonals. For queries belong to the same diagonal, they clearly belong to different rows and columns.

Given two queries (u,v), (u',v') that respectively belong to the i-th and the $(i+1)$-th diagonals. If they belong to a window of length $3j < n$, we have $u' \le u-2$. They do not share the same row. In addition,

$$(v - v') \bmod n = (u - u' - 1) \bmod n \ne 0$$

when $u' \le u-2$. They do not share the same column as well.

The existence of the j queries follows from a greedy construction: we greedily pick unselected queries that have distinct rows and columns, until we cannot pick any more queries. Since the $3j$ elements belong to different rows and columns, picking one query (u,v) from it will disable at most 2 other queries from the window: at most one from column u, and at most one from row v. Therefore, one can always pick j many queries. ◀

Our query-with-sketch framework is built for exact problems. To apply it to AMP, which is an approximate problem (allowing additive error), we first prove our lower bound on an exact variant we call *rounded AMP*. In this problem, the algorithm must output the true value rounded to a precision of $1/n^{\alpha'}$.

A lower bound for this rounded problem implies a lower bound for the original AMP problem because the two problems are almost identical. They only differ on a small set of "boundary" inputs–inputs where the true value is too close to a rounding point to be distinguished. We formalize this by first showing that this property holds in each dimension. Then we apply a union bound over all the dimensions.

▶ **Definition 11** (rounding notation). *Fix a rounding parameter $\alpha' > 0$. For a real number $x \in \mathbb{R}$, we denote its rounded value to its nearest multiple of $1/n^{\alpha'}$ as $[x]_{\alpha'}$. For a real matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$, we use $[\mathbf{M}]_{\alpha'}$ to denote the matrix where this rounding is applied element-wise.*

Rounded AMP problem

Fix a rounding constant $\alpha' > 0$ and a number $k \geq 2$. The input is a substochastic symmetric matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$. The succinct and systematic data structure, given a pair of indices $(u, v) \in [n] \times [n]$, must output $[\mathbf{M}^k[u, v]]_{\alpha'}$.

Given Lemma 3, we only need to show that for every $3m = 3r/\ln n$ consecutive queries, every query-with-sketch algorithm with r bits of sketch that computes the $3m$ queries with probability $\geq 99/100$, the algorithm makes $\Omega(n^2)$ probes in the worst case.

By Lemma 8, proving such a lower bound reduces to lower bounding the conditional min-entropy of the m query answers by $2r$, conditioned on assignments to the input matrix $\mathbf{M}$ of length $q := 10^{-4}n^2$. For this input distribution, we will construct a nontrivial good input set $\mathsf{M} \subseteq \mathbb{R}^{n\times n}$. And we will trivially set $\mathsf{M}_\sigma = \mathbb{R}^{n\times n}$ for every σ. Formally, the remainder of this section will be devoted to proving the following technical lemma.

▶ **Lemma 12.** *Fix $k \geq 3$ and constants α', β, γ such that $\gamma \geq 2$, $\beta \geq \gamma + 2$ and $\alpha' \geq 3\beta + 2\gamma + 3.6$. For every large enough integer n, there exists a set of "good" matrices $\mathsf{M} \subseteq \mathbb{R}^{n\times n}$ such that*

$$\Pr_{\mathbf{M}\leftarrow\mu_{\mathcal{X}^n}}[\mathbf{M} \in \mathsf{M}] \geq 1 - o(1).$$

In addition, fix $r \in (c\log^2 n, n\ln n/2)$, for every $3m = 3r/\ln n$ consecutive queries $(u_1, v_1), \ldots, (u_{3m}, v_{3m})$ from Q, and every partial assignment σ of length $q = 10^{-4}n^2$ to $\mathbf{M}$, we have

$$H_{\min}(([\mathbf{M}^k[u_1, v_1]]_{\alpha'}, \ldots, [\mathbf{M}^k[u_{3m}, v_{3m}]]_{\alpha'}), \mathbf{M} \in \mathsf{M}|\sigma) > 2r$$

where c is a constant that depends only on α', β, γ.

In addition, the following lemma shows that the outputs close to the boundary has a negligible measure. Its proof is deferred to Appendix A.

▶ **Lemma 13.** *Fix $k \geq 3$ and constants $\alpha, \alpha', \beta, \gamma$ such that $\gamma \geq 2$, $\beta \geq \gamma+2$, $\alpha' \geq 3\beta+2\gamma+3.6$ and $\alpha > \alpha' + 2$. For every large enough integer n, we have*

$$\Pr_{\mathbf{M}\leftarrow\mathcal{M}}\left[\exists u, v \in [n], a \in \mathbb{N}, \left|\mathbf{M}^k[u, v] - a/2n^{\alpha'}\right| \leq 1/n^{\alpha}\right] < o(1)$$

Given Lemma 12 and Lemma 13, our lower bound to approximate matrix powering follows.

Proof to Theorem 9. Let us first show that any data structure for rounded AMP with success probability $\geq 99/100$ either requires $\geq r$ redundancy or $\geq \Omega(n^4/r)$ worst-case probe complexity to answer queries in Q.

By Lemma 3, given the constructed sequence Q of queries, we only need to show that for every $3m$ consecutive queries from Q, every randomized query-with-sketch algorithm that computes the $3m$ queries with probability $\geq 99/100$ must pay $\geq r$ bits of sketch or $\geq 0.1q = 0.3tm$ average-case probe complexity.

By Yao's minimax principle, given our input distribution $\mathcal{M}$, we instead show that for every m queries and every deterministic query-with-sketch algorithm with success probability $\geq 99/100$ given a random input from $\mathcal{M}$, either r bits of sketch or $0.1q = 0.3tm$ average-case probe complexity is required. The lower bound follows from Lemma 8 and Lemma 12.

Now, we show that the same lower bound applies to the original AMP. We achieve it by a reduction from the rounded AMP. That is, we design a data structure D' for the rounded AMP given a data structure D for the AMP, with the same redundancy, probe complexity, and only an $o(1)$ increase in fail probability.

D' simply outputs the output of D rounded to the nearest multiple of $1/n^{\alpha'}$. By the correctness of D, for every query (u, v),

$$|\mathbf{M}^k[u, v] - D(u, v)| \leq 1/n^{\alpha}.$$

$\mathbf{M}^k[u, v]$ rounds to another value only if there exists a number $a \in \mathbb{N}$ such that one of the following two cases happen:

$$D(u, v) \leq a/2n^{\alpha'} \leq \mathbf{M}^k[u, v]$$

or

$$D(u, v) \geq a/2n^{\alpha'} \geq \mathbf{M}^k[u, v],$$

which happens only if

$$|\mathbf{M}^k[u, v] - a/2n^{\alpha'}| \leq |\mathbf{M}^k[u, v] - D(u, v)| \leq 1/n^{\alpha}.$$

However, by Lemma 13, this happens with $< o(1)$ probability. Therefore, the success probability of D' is at most $o(1)$ worse than the probability of D. The above lower bound also applies to AMP. ◀

3.1 Partial derivatives between the inputs and outputs

We are going to show that, for every $3m = 3r/\ln n$ consecutive queries $((u_1, v_1), \ldots, (u_m, v_m))$ that span m different rows and columns, and every partial assignment σ of length $q = 0.001n^2$, the query answers conditioned on the partial assignment remain a high min-entropy for good inputs from M. At a high level, every partial assignment fixes a proportion of input elements. We will show that, given every partial assignment, one can always pick $\Theta(r/\log n)$ elements from the unfixed input elements that form a "basis" to the output elements. That is, each fixed and possible output only corresponds to a small subset of inputs. We capture this by studying the Jacobian matrix between the output elements and the input elements. Note that such a basis does not always exist, where we construct a large enough good input set M that guarantees the existence of the basis.

To start with, we regard each output element $\mathbf{M}^k[u,v]$ as a function of n^2 input elements $\mathbf{M}[i,j]$, and study their partial derivatives $\frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[i,j]}$. Recall that $\mathbf{M} = \frac{n^2-1}{n^2}\mathbf{I} + \frac{1}{n^\beta}\mathbf{M_U}$ where each element $\mathbf{M_U}[i,j] = \mathbf{M_U}[j,i]$ of $\mathbf{M_U}$ is uniformly random from $[\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]$. We ignore for now the fact that $\mathbf{M}$ is a symmetric matrix. The partial derivative under the symmetric constraint is simply $\frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[i,j]} + \frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[j,i]}$ when $i \neq j$.

We will use the following observation multiple times in the proof.

▶ **Observation 14.** *For every integer $n > 1$,*

$$\left(1 - \frac{1}{n}\right)^n \leq e^{-1} \leq \left(1 - \frac{1}{n}\right)^{n-1}$$

Proof. For simplicity, we let $f(n) = (1 - \frac{1}{n})^n$, $g(n) = (1 - \frac{1}{n})^{n-1}$ for now. It is known that

$$\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = e^{-1}.$$

Given this fact, to show the inequalities hold whenever $n > 1$, we show that (i) the inequalities hold true when $n = 2$, and (ii) $f(n)$ is an increasing function and $g(n)$ is a decreasing function.

One can verify that the inequalities hold for the case $n = 2$. To show the monotonicity of $f(n)$ and $g(n)$,

$$\begin{aligned} f'(n) &= \left(e^{n\ln(1-1/n)}\right)' = \left(1 - \frac{1}{n}\right)^n \left(\ln(1 - 1/n) + \frac{1}{n-1}\right) \\ &= \left(1 - \frac{1}{n}\right)^n \left(\left(-\frac{1}{n} - \frac{1}{2n^2} - \frac{1}{3n^3} - ...\right) + \frac{1}{n-1}\right) > 0 \end{aligned}$$

when $n \geq 2$.

$$\begin{aligned} g'(n) &= \left(e^{(n-1)\ln(1-1/n)}\right)' = \left(1 - \frac{1}{n}\right)^{n-1} \left(\ln(1 - 1/n) + \frac{1}{n}\right) \\ &= \left(1 - \frac{1}{n}\right)^n \left(\left(-\frac{1}{n} - \frac{1}{2n^2} - \frac{1}{3n^3} - ...\right) + \frac{1}{n}\right) < 0 \end{aligned}$$

when $n \geq 2$. ◀

To bound the partial derivatives $\frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[i,j]}$, we bound $\frac{\partial \mathbf{M_U}^k[u,v]}{\partial \mathbf{M_U}[i,j]}$ first.

▶ **Lemma 15.** *Let $\gamma \geq 2$, and the values of elements of $\mathbf{M_U}$ range from $[\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]$. For every positive integer $3 \leq k \leq n$, for every element $[i,j]$ of $\mathbf{M_U}$ and every element $[u,v]$ of $\mathbf{M_U}^k$ such that $u \neq i$ and $j \neq v$. When $k \geq 3$, we have:*

$$\frac{\partial \mathbf{M_U}^k[u,v]}{\partial \mathbf{M_U}[i,j]} = \Theta\left(\frac{k-2}{n^2}\right)$$

Specifically, when $k = 3$,

$$\frac{\partial \mathbf{M_U}^k[u,v]}{\partial \mathbf{M_U}[i,j]} = \mathbf{M_U}[u,i]\mathbf{M_U}[j,v]$$

Proof of Lemma 15. Note that

$$\mathbf{M_U}^k[u,v] = \sum_{u=h_0,h_1,\ldots,h_{k-1},h_k=v} \prod_{t=0}^{k-1} \mathbf{M_U}[h_t, h_{t+1}]$$

When $u \neq i$ and $j \neq v$, the partial derivative is

$$\frac{\partial {\mathbf{M_U}}^k[u,v]}{\partial \mathbf{M_U}[i,j]} = \sum_{g=1}^{k-2} \left(\sum_{u=h_0,h_1,\ldots,h_g=i,h_{g+1}=j,\ldots,h_{k-1},h_k=v} \left(\prod_{t=0,t\neq g}^{k-1} \mathbf{M_U}[h_t,h_{t+1}] \right) \right)$$

In this equation the appearance of $\mathbf{M_U}[i,j]$ is canceled at position g in the summation of the product. This directly gives the expression for $k = 3$. Since for all $i,j \in [n]$, $\mathbf{M_U}[i,j] = \mathbf{M_U}[j,i] \in [\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]$, which is a quite small range, $\prod_{t=0,t\neq g}^{k-1} \mathbf{M_U}[h_t,h_{t+1}]$ can be lower-bounded by setting all elements in $\mathbf{M_U}$ equal to $\frac{1}{n} - \frac{1}{n^\gamma}$ and upper bounded by setting all elements in $\mathbf{M_U}$ equal to $\frac{1}{n}$. We only need to count the number of sequences $u = h_0, h_1, ..., h_{k-1}, h_k = v$ in which two adjacent indices are i and j, while in both the lower and upper bounds all elements are equivalent. Since g can only be picked from $\{1, 2, ..., k-2\}$, there are $(k-2)n^{k-3}$ distinct sequences $h_0, \ldots, h_k$. We get

$$\begin{aligned}(k-2)n^{-2} = (k-2)n^{k-3}\left(\frac{1}{n}\right)^{k-1} \geq \frac{\partial {\mathbf{M_U}}^k[u,v]}{\partial \mathbf{M_U}[i,j]} \\ \geq (k-2)n^{k-3}\left(\frac{1}{n} - \frac{1}{n^\gamma}\right)^{k-1} \geq e^{-\frac{k-1}{n^{\gamma-1}-1}}(k-2)n^{-2}.\end{aligned}$$

Recall that $\gamma \geq 2$, so $\frac{\partial {\mathbf{M_U}}^k[u,v]}{\partial \mathbf{M_U}[i,j]} = \Theta(\frac{k-2}{n^2})$. ◀

▶ **Lemma 16.** *Let $\gamma, \beta \geq 2$, and the values of elements of $\mathbf{M_U}$ range from $[\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]$. For every positive integer $3 \leq k \leq n$, every element (i,j) of $\mathbf{M}$ and every element (u,v) of $\mathbf{M}^k$ such that u,v,i,j are pairwise distinct, we have*

$$\frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[i,j]} = \binom{k}{3} \frac{(n^2-1)^{k-3}}{n^{2k-6+2\beta}} (\mathbf{M_U}[u,i]\mathbf{M_U}[j,v] + \mathbf{M_U}[u,j]\mathbf{M_U}[i,v]) + O(k^5/n^{2+3\beta}).$$

Proof. Notice that

$$\mathbf{M}^k = \sum_{t=0}^{k} \binom{k}{t} \left(\frac{n^2-1}{n^2}\right)^{k-t} \left(\frac{1}{n^\beta}\mathbf{M_U}\right)^t$$

and

$$\begin{aligned}\frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[i,j]} &= \sum_{t=0}^{k} \binom{k}{t} \left(\frac{n^2-1}{n^2}\right)^{k-t} \left(\frac{1}{n^\beta}\right)^t \frac{\partial {\mathbf{M_U}}^t[u,v]}{\partial \mathbf{M}[i,j]} \\ &= \sum_{t=0}^{k} \binom{k}{t} \left(\frac{n^2-1}{n^2}\right)^{k-t} \left(\frac{1}{n^\beta}\right)^{t-1} \frac{\partial {\mathbf{M_U}}^t[u,v]}{\partial \mathbf{M_U}[i,j]}\end{aligned}$$

While u,v,i,j are pairwise distinct, $\frac{\partial {\mathbf{M_U}}^t[u,v]}{\partial \mathbf{M_U}[i,j]}$ is 0 when $t \leq 2$. Therefore,

$$\frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[i,j]} = \binom{k}{3} \frac{(n^2-1)^{k-3}}{n^{2k-6+2\beta}} \mathbf{M_U}[u,i]\mathbf{M_U}[j,v] + O(k^5/n^{2+3\beta}).$$ ◀

3.2 Jacobian matrix and singular value concentration bounds

While the previous section studies the partial derivatives between individual input elements and output elements of different rows and columns. We extend it to the interplay between multiple input elements and multiple output elements, which formalizes as the Jacobian

matrix. We will show that, fix a number $m \leq n/4$, for every m input elements and m output elements whose column indices and row indices are pairwise distinct, the largest $0.9m$ singular values of their Jacobian matrix are controlled with high probability. We use a recent work [42] on singular values concentration bounds to achieve it.

▶ **Definition 17.** *Fix $m \leq n/4$, m input indices $(i_1, j_1), \ldots, (i_m, j_m)$, and m output indices $(u_1, v_1), \ldots, (u_m, v_m)$. When $i_1 \neq j_1, \ldots, i_m \neq j_m$, their Jacobian matrix is defined as*

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \mathbf{M}^k[u_1,v_1]}{\partial \mathbf{M}[i_1,j_1]} + \frac{\partial \mathbf{M}^k[u_1,v_1]}{\partial \mathbf{M}[j_1,i_1]} & \cdots & \cdots & \frac{\partial \mathbf{M}^k[u_m,v_m]}{\partial \mathbf{M}[i_1,j_1]} + \frac{\partial \mathbf{M}^k[u_m,v_m]}{\partial \mathbf{M}[j_1,i_1]} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial \mathbf{M}^k[u_1,v_1]}{\partial \mathbf{M}[i_m,j_m]} + \frac{\partial \mathbf{M}^k[u_1,v_1]}{\partial \mathbf{M}[j_m,i_m]} & \cdots & \cdots & \frac{\partial \mathbf{M}^k[u_m,v_m]}{\partial \mathbf{M}[i_m,j_m]} + \frac{\partial \mathbf{M}^k[u_m,v_m]}{\partial \mathbf{M}[j_m,i_m]} \end{pmatrix}$$

Given an $m \times m$ Hermitian matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$, we denote by $\lambda_1(\mathbf{A}) \geq \cdots \geq \lambda_m(\mathbf{A})$ eigenvalues of $\mathbf{A}$.

The singular values $\sigma_1(\mathbf{A}), \ldots, \sigma_m(\mathbf{A})$ of a (possibly non-Hermitian) matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$, are the eigenvalues of $\sqrt{\mathbf{A}^T\mathbf{A}}$ arranged in non-increasing order.

We will use the following standard tools from matrix analysis.

▶ **Proposition 18** (Weyl's inequality for singular values [22])**.** *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$ be general matrices. For every $k \in [m]$,*

$$|\sigma_k(\mathbf{A} + \mathbf{B}) - \sigma_k(\mathbf{A})| \leq \|\mathbf{B}\|_{op}$$

where $\|\mathbf{B}\|_{op} = \sigma_1(\mathbf{B})$ is the operator norm.

▶ **Proposition 19** ([36])**.** *Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$ where $\mathrm{rank}(\mathbf{B}) = 1$. For every $k \in [m]$,*

$$\sigma_{k+1}(\mathbf{A}) \leq \sigma_k(\mathbf{A} + \mathbf{B}) \leq \sigma_{k-1}(\mathbf{A})$$

where we denote $\sigma_0(\mathbf{A}) = +\infty$ and $\sigma_{m+1}(\mathbf{A}) = 0$.

▶ **Proposition 20** (Geršgorin circle theorem [22])**.** *Let $\mathbf{A} \in \mathbb{C}^{m \times m}$. Let $R_i(\mathbf{A}) = \sum_{j \neq i} |\mathbf{A}[i,j]|$ denote the absolute row sums of $\mathbf{A}$ other than $\mathbf{A}[i,i]$. And consider the m Geršgorin discs*

$$\{z \in \mathbb{C} : |z - \mathbf{A}[i,i]| \leq R_i(\mathbf{A})\}$$

for $i \in [m]$. The eigenvalues of $\mathbf{A}$ are in the union of Geršgorin discs

$$G(A) = \bigcup_{i=1}^{m} \{z \in \mathbb{C} : |z - \mathbf{A}[i,i]| \leq R_i(\mathbf{A})|\}.$$

We will use the following concentration bound on the intermediate singular values of a random matrix [34, 42]. We adapt their result in our setting and show the following in Appendix B.

▶ **Lemma 21.** *For $i \in \{1, 2\}$, let X_i, Y_i, Z_i, W_i be i.i.d. uniform on some symmetric interval (the interval may depend on i). Define*

$$U_1 := X_1 + Y_1 + Z_1 + W_1, \qquad U_2 := X_2Y_2 + Z_2W_2.$$

Let μ_i be the distribution of U_i.

Fix μ to be either μ_1 or μ_2. Let m a large enough number, and $\mathbf{A} \in \mathbb{R}^{m\times m}$ be a matrix whose elements are i.i.d. drawn from μ. There exists constants $0 < C_1 < C_2$ and $C_3 > 0$ such that for all ℓ between 1 and m,

$$\Pr\left[\frac{C_1\ell}{\sqrt{m}} \le \sigma_{m+1-\ell}(\mathbf{A}) \le \frac{C_2\ell}{\sqrt{m}}\right] \ge 1 - \exp(-C_3\ell)$$

Basically, a random matrix of the above form has high singular values to a large proportion of its dimensions with high probability. Our Jacobian matrix is a combination of scaled, shifted and perturbed random matrices of the above form. Formally, we give the following concentration bound on our Jacobian matrix.

▶ **Lemma 22.** *Let $\beta, \gamma \ge 2$ be constants such that $\beta \ge \gamma + 2$. For every large enough n and every $m \le n/4$. Fix a sequence of indices $(i_1, j_1), \ldots, (i_m, j_m)$ of $\mathbf{M}$ and a sequence of indices $(u_1, v_1), \ldots (u_m, v_m) \in [n] \times [n]$ of $\mathbf{M}^k$ such that $i_1, j_1, \ldots, i_m, j_m, u_1, v_1, \ldots, u_m, v_m$ are $4m$ pairwise distinct numbers. With probability $\ge 1 - \exp(-\Omega(m))$, we have $\sigma_{0.9m}(\mathbf{J}) = \Omega(k^3\sqrt{m}/n^{2\beta+\gamma+1})$.*

In addition, the above holds true even if each element of $\mathbf{J}$ is perturbed additively by at most $O(k^5/n^{2+3\beta})$.

Proof. Let $K := \binom{k}{3}\frac{(n^2-1)^{k-3}}{n^{2k-6+2\beta}} = \Theta(k^3/n^{2\beta})$. We define four auxiliary matrices whose singular values are bounded, and their combination is exactly $\mathbf{J}$. Notice that the expectation of each element of $\mathbf{M_U}$ is exactly $E := \frac{1}{n} - \frac{1}{2n^\gamma}$. Let $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B} \in \mathbb{R}^{m\times m}$ such that for every $a, b \in [m]$,

$$\mathbf{A}_1[a,b] := 6\sqrt{2}n^{2\gamma}\left((\mathbf{M_U}[u_b, i_a] - E)(\mathbf{M_U}[j_a, v_b] - E) + (\mathbf{M_U}[u_b, j_a] - E)(\mathbf{M_U}[i_a, v_b] - E)\right),$$

$$\mathbf{A}_2[a,b] := \sqrt{3}n^{\gamma}\left(\mathbf{M_U}[u_b, i_a] + \mathbf{M_U}[j_a, v_b] + \mathbf{M_U}[u_b, j_a] + \mathbf{M_U}[i_a, v_b] - 4E\right),$$

$$\mathbf{B}[a,b] := \frac{\partial \mathbf{M}^k[u_b, v_b]}{\partial \mathbf{M}[i_a, j_a]} + \frac{\partial \mathbf{M}^k[u_b, v_b]}{\partial \mathbf{M}[j_a, i_a]} - K(\mathbf{M_U}[u_b, i_a]\mathbf{M_U}[j_a, v_b] + \mathbf{M_U}[u_b, j_a]\mathbf{M_U}[i_a, v_b]).$$

By Lemma 16, we can rewrite the Jacobian matrix as

$$\mathbf{J} = \frac{K}{6\sqrt{2}n^{2\gamma}}\mathbf{A}_1 + \frac{K}{\sqrt{3}n^{\gamma}}E\mathbf{A}_2 + 2KE^2\mathbf{1}\mathbf{1}^T + \mathbf{B}$$

where $\mathbf{1}$ denotes the all-1 vector of length m. One can verify that both $\mathbf{A}_1, \mathbf{A}_2$ are random matrices satisfying the condition in Lemma 21. Both $\mathbf{A}_1, \mathbf{A}_2$ have i.i.d. random elements because $\mathbf{M_U}$ is a uniformly random symmetric matrix, and the indices i, j, u, v do not repeat.

By Geršgorin circle theorem (Proposition 20), for every $i \in [m]$, we have

$$\sigma_i(\mathbf{B}) = \sqrt{\lambda_i(\mathbf{B}^T\mathbf{B})} \le O(mk^5/n^{2+3\beta}). \tag{3}$$

We know from Lemma 21 that for every $\mathbf{A} = \mathbf{A}_1, \mathbf{A}_2$.

$$\Pr\left[C_1\sqrt{m} \le \sigma_1(\mathbf{A}) \le C_2\sqrt{m}\right] \ge 1 - \exp(-C_3 m)$$

and for every $l \in [m]$,

$$\Pr\left[C_1 l/\sqrt{m} \le \sigma_{m+1-l}(\mathbf{A}) \le C_2 l/\sqrt{m}\right] \ge 1 - \exp(-C_3 l).$$

By Proposition 19,

$$\sigma_{0.9m}\left(\frac{K}{\sqrt{3}n^{\gamma}}E\mathbf{A}_2 + 2KE^2\mathbf{1}\mathbf{1}^T\right) \ge \frac{K}{\sqrt{3}n^{\gamma}}E\sigma_{0.9m+1}(\mathbf{A}_2)$$

By Weyl's inequality (Proposition 18) and triangle inequality, we have

$$\sigma_{0.9m}(\mathbf{J}) \geq \frac{K}{\sqrt{3}n^{\gamma}} E \cdot \sigma_{0.9m+1}(\mathbf{A}_2) - \frac{K}{6\sqrt{2}n^{2\gamma}} \cdot \sigma_1(\mathbf{A}_1) - \sigma_1(\mathbf{B}).$$

By inequality (3) and the concentration bounds on the singular values of $\mathbf{A}_1, \mathbf{A}_2$, the following holds with high probability

$$\sigma_{0.9m}(\mathbf{J}) \geq \frac{K}{\sqrt{3}n^{\gamma}} E \cdot 0.1C_1\sqrt{m} - \frac{K}{6\sqrt{2}n^{2\gamma}} \cdot C_2\sqrt{m} - O(mk^5/n^{2+3\beta})$$

which implies that $\sigma_{0.9m}(\mathbf{J}) = \Omega(K\sqrt{m}/n^{\gamma+1}) = \Omega(k^3\sqrt{m}/n^{2\beta+\gamma+1})$ when $\beta \geq \gamma + 2$.

The above holds true even if each element of $\mathbf{J}$ is perturbed additively by at most $O(k^5/n^{2+3\beta})$, since the absolute values of elements of $\mathbf{B}$ remains $O(k^5/n^{2+3\beta})$. ◀

3.3 From singular values to the min-entropy: proof to Lemma 12

We have prepared to prove our technical lemma, which is to show that there exists a good input set M such that the joint min-entropy conditioned on any partial assignment of length $\leq q = 10^{-4}n^2$ is high. Our good input set M will be defined as inputs where the Jacobian matrices have well-controlled singular values. Before that, we partition the set of input elements into groups of *core* elements of size m, so that for every partial assignment σ, we can always pick one such group of core elements that form a basis of the output elements.

▶ **Definition 23.** *For every window of $3m$ consecutive queries from Q, fix once and for all one subset of m queries whose $2m$ row and column indices are pairwise distinct. Such a subset exists by Observation 10. Let $(u_1, v_1), \ldots, (u_m, v_m)$ denote this fixed subset for the window. We define $\mathcal{C}$ to be an arbitrary collection of disjoint sequences from*

$$([n] \setminus \{u_1, \ldots, u_m, v_1, \ldots, v_m\}) \times ([n] \setminus \{u_1, \ldots, u_m, v_1, \ldots, v_m\})$$

such that, for every $((i_1, j_1), \ldots, (i_m, j_m)) \in \mathcal{C}$, the indices $i_1, j_1, \ldots, i_m, j_m$ are pairwise distinct. Moreover, we choose $\mathcal{C}$ so that

$$|\mathcal{C}| \geq \left\lfloor \frac{(n-2m)(n-2m-1)}{3m} \right\rfloor.$$

We call $\mathcal{C}$ the candidate core set *associated with this window.*

The existence of such a partition is guaranteed by the same construction as the query sequence Q, where by Observation 10, the permutation of indices we defined as Q guarantees that no m consecutive elements share the same column or the same row. Given $\mathcal{C}$, we can define our good input set M.

▶ **Definition 24.** *We define M as the set of inputs satisfying the following condition. For every window of $3m$ consecutive queries from Q, let $(u_1, v_1), \ldots, (u_m, v_m)$ be the fixed subset of queries chosen in Definition 23, and let $\mathcal{C}$ be its candidate core set. Then for every sequence $((i_1, j_1), \ldots, (i_m, j_m)) \in \mathcal{C}$, the Jacobian matrix between $\mathbf{M}[i_1, j_1], \ldots, \mathbf{M}[i_m, j_m]$ and $\mathbf{M}^k[u_1, v_1], \ldots, \mathbf{M}^k[u_m, v_m]$ satisfies*

$$\sigma_{0.9m}(J) = \Omega\left(\frac{k^3\sqrt{m}}{n^{2\beta+\gamma+1}}\right).$$

▶ **Lemma 25.** *There exists a constant c such that for every $m \in [c \log n, n/4]$, $\Pr_{\mathbf{M} \leftarrow \mathcal{M}}[\mathbf{M} \in \mathsf{M}] > 1 - o(1)$.*

Proof. Fix a window of $3m$ consecutive queries from Q, and consider the fixed subset of m queries and the candidate core set $\mathcal{C}$ from Definition 23. By Lemma 22, for every fixed sequence in $\mathcal{C}$, the corresponding Jacobian matrix has the desired singular-value bound with probability at least $1 - \exp(-\Omega(m))$.

It remains to union bound over all events that appear in the definition of $\mathcal{M}$. There are $O(n^2)$ windows of $3m$ consecutive queries in Q. For each such window, the candidate core set $\mathcal{C}$ contains at most $O(n^2/m)$ disjoint sequences. Hence the total number of Jacobian matrices over which we union bound is at most $O(n^4/m)$, which is polynomial in n. Therefore,

$$\Pr[\mathbf{M} \notin \mathcal{M}] \leq O(n^4/m) \cdot \exp(-\Omega(m)) = o(1),$$

provided $m \geq c \log n$ for a sufficiently large constant c. ◀

Given guarantees that the Jacobian matrices have high singular values, given an arbitrary partial assignment σ of length $\leq q = 10^{-4}n^2$, there always exists a sequence of input indices from $\mathcal{C}$ where a large proportion of these input elements are unfixed by σ. Those elements, due to their nice properties in their Jacobian matrix with the output, form a basis to the output. In this way, the entropy of the input preserves to the output. Before proving Lemma 12, we need the following definition and basic tools.

▶ **Definition 26.** *Given an arbitrary partial assignment σ, the* core indices *of $\mathbf{M}$ is a length-$(m' := 0.9m)$ subsequence $((i_1, j_1), \ldots, (i_{m'}, j_{m'}))$ of an arbitrary sequence from $\mathcal{C}$, such that $\mathbf{M}[i_1, j_1], \ldots, \mathbf{M}[i_{m'}, j_{m'}]$ and $\mathbf{M}[j_1, i_1], \ldots, \mathbf{M}[j_{m'}, i_{m'}]$ are not fixed by σ. We use $(\mathbf{M})_{\text{core}} := (\mathbf{M}[i_1, j_1], \ldots, \mathbf{M}[i_{m'}, j_{m'}])$ to denote the vector representation of core elements. In addition, we use $(\mathbf{M}^k)_{\text{out}} := (\mathbf{M}^k[u_1, v_1], \ldots, \mathbf{M}^k[u_m, v_m])$ to denote the vector representation of the selected output elements. Also, we always use $[\mathbf{v}]_{\alpha'}$ to denote the vector where each element is rounded to its nearest multiple of $1/n^{\alpha'}$.*

We will show that, for every fixed (and rounded) output, the large singular values of the Jacobian matrix between core elements and output elements forces $\Omega(m)$ dimensions of the input to be concentrated. The min-entropy will be reduced to measuring the volume of a hyperrectangle in the uniform input space.

▶ **Proposition 27** (Cauchy interlacing theorem for singular values [22])**.** *Let $\mathbf{J} \in \mathbb{R}^{m \times n}$ a matrix where $m \leq n$, and $\mathbf{J}'$ a matrix obtained by deleting one row from it. Then*

$$\sigma_1(\mathbf{J}) \geq \sigma_1(\mathbf{J}') \geq \sigma_2(\mathbf{J}) \geq \sigma_2(\mathbf{J}') \geq \cdots \geq \sigma_{m-1}(\mathbf{J}) \geq \sigma_{m-1}(\mathbf{J}').$$

▶ **Proposition 28** (Lagrange's mean value theorem)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous and differentiable function. For every two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, there exists $\mathbf{c} = (1 - t_0)\mathbf{a} + t_0\mathbf{b}$ for some $t_0 \in (0, 1)$ such that*

$$f(\mathbf{b}) - f(\mathbf{a}) = \nabla f(\mathbf{c}) \cdot (\mathbf{b} - \mathbf{a})$$

where $\nabla f(\mathbf{c})$ is the gradient of f evaluated at the point $\mathbf{c}$.

Proof to Lemma 12. We have shown in Lemma 25 that a random input drawn from $\mathcal{M}$ is a good input with high probability. What remains is to show that the conditional joint min-entropy of the output is high for every partial assignment σ of bounded length.

Given a partial assignment σ of length $\leq q = 10^{-4}n^2$, there exists a sequence $((i_1, j_1), \ldots, (i_m, j_m)) \in \mathcal{C}$ where at most $0.1m$ elements of $\mathbf{M}$ or their symmetric elements are revealed by σ. Otherwise there would be $> \frac{1}{2} \cdot \lfloor \frac{(n-2m)(n-2m-1)}{3m} \rfloor \cdot 0.1m > q$ elements revealed by σ. We

fix an arbitrary such sequence, and the core indices $(i_1, j_1), \ldots, (i_{m'}, j_{m'})$ be its subsequence obtained by removing all the elements and the symmetric elements fixed by σ. For every possible output vector $\mathbf{y}$, we have

$$\Pr_{\mathbf{M}\leftarrow\mathcal{M}}\left[[(\mathbf{M}^k)_{\text{out}}]_{\alpha'} = \mathbf{y} \text{ and } \mathbf{M} \in \mathsf{M} \middle| \sigma\right] \leq \max_{\sigma'} \Pr_{\mathbf{M}\leftarrow\mathcal{M}}\left[[(\mathbf{M}^k)_{\text{out}}]_{\alpha'} = \mathbf{y} \text{ and } \mathbf{M} \in \mathsf{M} \middle| \sigma'\right] \quad (4)$$

where the maximum is taken over all partial assignments σ' on all the input elements other than core elements and their symmetric elements, such that σ' is consistent with σ. We denote $f : \mathbb{R}^{m'} \to \mathbb{R}^m$ as the function of $(\mathbf{M}^k)_{\text{out}}$ given core elements, and all the other input elements fixed by σ'. And we denote by $f_i(\cdot)$ the i-th element of the output vector.

Let $\|\cdot\|_\infty$ denote max norm, i.e., the maximum absolute value of all elements. Fix an arbitrary instance $\mathbf{M}'$ that is consistent σ' such that $[(\mathbf{M}'^k)_{\text{out}}]_{\alpha'} = \mathbf{y}$. Then, for matrices $\mathbf{M}$ consistent with σ' and with the same output, we have

$$\|f((\mathbf{M})_{\text{core}}) - f((\mathbf{M}')_{\text{core}})\|_\infty \leq 1/n^{\alpha'}.$$

By the mean value theorem, there exists m vectors $\mathbf{c}_1, \ldots, \mathbf{c}_m$, such that for every $\ell \in [m]$,

$$f_\ell((\mathbf{M})_{\text{core}}) - f_\ell((\mathbf{M}')_{\text{core}}) = \nabla f_\ell(\mathbf{c}_\ell) \cdot ((\mathbf{M})_{\text{core}} - (\mathbf{M}')_{\text{core}})$$

where $\nabla f_\ell(\mathbf{c}_\ell)$ is exactly the ℓ-th column of the Jacobian matrix evaluated at $\mathbf{c}_\ell$. We let $\mathbf{J} \in \mathbb{R}^{m'\times m}$ be the matrix where the ℓ-th column is exactly $\nabla f_\ell(\mathbf{c}_\ell)$ for each $\ell \in [m]$. Then we have

$$1/n^{\alpha'} \geq \|f((\mathbf{M})_{\text{core}}) - f((\mathbf{M}')_{\text{core}})\|_\infty = \|\mathbf{J}^T((\mathbf{M})_{\text{core}} - (\mathbf{M}')_{\text{core}})\|_\infty. \quad (5)$$

Because by Lemma 16, when given σ' fixed, the Jacobian matrix is almost fixed but the $O(k^5/n^{2+3\beta})$ term. We have

$$\|\mathbf{J} - \mathbf{J}(\mathbf{M})\|_\infty, \|\mathbf{J} - \mathbf{J}(\mathbf{M}')\|_\infty \leq O(k^5/n^{2+3\beta})$$

for $\mathbf{J}(\mathbf{M}), \mathbf{J}(\mathbf{M}')$ respectively the Jacobian matrix evaluated at $\mathbf{M}$ and $\mathbf{M}'$. The singular value bounds given in Lemma 21 also apply to $\mathbf{J}$, since, the as stated in Lemma 21, the bound applies even when $\mathbf{J}$ is a matrix obtained by perturbing every element of a Jacobian matrix by at most $O(k^5/n^{2+3\beta})$.

Let $m'' := 0.8m$, by Lemma 21 and Cauchy interlacing theorem, $\mathbf{J} \in \mathbb{R}^{m'\times m}$ between the core elements in $\mathbf{M}$ and $(\mathbf{M}^k)_{\text{out}}$ satisfies: for every $\ell \in [0.8m]$,

$$\sigma_\ell(\mathbf{J}) = \Omega(k^3\sqrt{m}/n^{2\beta+\gamma+1}). \quad (6)$$

Consider the singular value decomposition $\mathbf{J} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ for orthogonal matrices $\mathbf{U} \in \mathbb{R}^{m'\times m'}, \mathbf{V} \in \mathbb{R}^{m\times m}$ and a diagonal matrix $\mathbf{\Sigma}$ where $\mathbf{\Sigma}[\ell,\ell] = \sigma_\ell(\mathbf{J})$ for every $\ell \in [m']$. Denote by $\mathbf{u}_1, \ldots, \mathbf{u}_{m'}$ the columns of $\mathbf{U}$, which are orthonormal vectors. Let $((\mathbf{M})_{\text{core}} - (\mathbf{M}')_{\text{core}}) = \sum_{\ell=1}^{m'} \alpha_\ell \mathbf{u}_\ell$ for some $\alpha_1, \ldots, \alpha_{m'} \in \mathbb{R}$. Then we have

$$\begin{aligned}
\|\mathbf{J}^T((\mathbf{M})_{\text{core}} - (\mathbf{M}')_{\text{core}})\|_\infty &\geq \frac{1}{\sqrt{m}}\|\mathbf{J}^T((\mathbf{M})_{\text{core}} - (\mathbf{M}')_{\text{core}})\|_2 \\
&= \frac{1}{\sqrt{m}}\|\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T(\sum_{\ell=1}^{m'}\alpha_\ell\mathbf{u}_\ell)\|_2 \\
&= \frac{1}{\sqrt{m}}\|\mathbf{\Sigma}^T\mathbf{U}^T(\sum_{\ell=1}^{m'}\alpha_\ell\mathbf{u}_\ell)\|_2 \\
&= \sqrt{\frac{\sum_{\ell=1}^{m'}\sigma_\ell(\mathbf{J})^2\alpha_\ell^2}{m}}
\end{aligned}$$

where the first inequality is by Cauchy-Schwarz, the third line is because the L_2 norm remains the same under orthogonal transformations.

Notice that for every $\ell \in [m']$, $\alpha_\ell = \mathbf{u}_\ell \cdot ((\mathbf{M})_{\text{core}} - (\mathbf{M}')_{\text{core}})$. Combining the above with inequality (5) and equation (6), for every $\ell \in [m'']$, we have

$$((\mathbf{M})_{\text{core}} - (\mathbf{M}')_{\text{core}})^T \mathbf{u}_\ell \leq \frac{\sqrt{m}}{n^{\alpha'} \sigma_\ell(\mathbf{J})} \leq O(n^{2\beta+\gamma-\alpha'+1}/k^3).$$

Therefore, given $\mathbf{y}$ and σ', the possible input set is contained in the solution set of the above m'' inequalities, whose volume is proportional to the volume of the hyperrectangle with m'' dimensions concentrated in the uniform input space.

Notice that the total volume of possible inputs $\mathbf{M}$ given σ' is $1/n^{(\beta+\gamma)m'}$. For every pair of different possible matrices consistent to σ', their maximum L_2 distance is at most $\sqrt{m'}/n^{\beta+\gamma}$. Therefore, the maximum volume of the hyperrectangle is upper-bounded by

$$(O(n^{2\beta+\gamma-\alpha'+1}/k^3))^{m''} \cdot (\sqrt{m'}/n^{\beta+\gamma})^{m'-m''}.$$

Recall that $m'' = 0.8m$, and $m' = 0.9m$. By inequality (4),

$$\begin{aligned}
&\Pr_{\mathbf{M} \leftarrow \mathcal{M}} \left[[(\mathbf{M}^k)_{\text{out}}]_{\alpha'} = \mathbf{y} \text{ and } \mathbf{M} \in \mathsf{M} \middle| \sigma \right] \\
\leq &(O(n^{2\beta+\gamma-\alpha'+1}/k^3))^{m''} \cdot (\sqrt{m}/n^{\beta+\gamma})^{m'-m''} \cdot n^{(\beta+\gamma)m'} \\
\leq &\exp(m''(2\beta+\gamma-\alpha'+1)\ln n) \cdot \exp((m'-m'')(0.5-\beta-\gamma)\ln n) \cdot \exp(m'(\beta+\gamma)\ln n) \\
\leq &\exp((2.4\beta+1.6\gamma-0.8\alpha'+0.85)m \ln n) \\
\leq &\exp(-2r)
\end{aligned}$$

while $\alpha' \geq 3\beta + 2\gamma + 3.6$ and $m = \frac{r}{\ln n} < n$.

Since the conditional min-entropy of m selected output elements is at least $2r$, the conditional min-entropy of all the $3m$ output elements is also at least $2r$. ◀

4 Lower bounds to approximate matrix powering with superlinear-in-n redundancy

The above approach does not easily extend to proofs to $r \cdot t = \Omega(n^2)$ lower bounds for $r = \omega(n \ln n)$ because the Jacobian matrix of $\omega(n)$ core elements and output elements can not be easily controlled. Our analysis above relies on the fact that the core elements, output elements, and their symmetric elements do not share rows and columns. To that end, we use a different input distribution which leads to the same trade-off for $r \in [10n \log n, n^2/1000]$. Notably, the lower bound derived using this distribution also implies a lower bound for the fundamental *counting set intersection* problem in data structures.

Nemesis input distribution

Fix $n \geq 1$ a large enough integer. Fix a partition $V_1 := \{1, 2, \ldots, n/3\}, V_2 := \{n/3 + 1, \ldots, 2n/3\}, V_3 := \{2n/3 + 1, \ldots, n\}$ to $[n]$. Let $\mathcal{M}$ be a distribution of $n \times n$ symmetric matrices $\mathbf{M}$ with the following conditions.

1. For every $(u, v) \in (V_1 \times V_2) \cup (V_2 \times V_3)$,

$$\mathbf{M}[u, v] = \mathbf{M}[v, u] = \begin{cases} 1/n^3 & \text{with independent probability } 1/2 \\ 0 & \text{with independent probability } 1/2. \end{cases}$$

2. For every $u \in [n]$, $\mathbf{M}[u,u] = 1 - 1/n^2$.
3. For every other $(u,v) \in [n] \times [n]$, $\mathbf{M}[u,v] = 0$.

In other words, its underlying graph is a three-layered undirected graph with heavy lazy transitions. For each pair of vertices of consecutive layers, we connect them with independent probability $1/2$. We have the following observation.

▶ **Observation 29.** *For every pair of vertices $(u,v) \in V_1 \times V_3$ and every $2 \le k \le n$, if there are exactly a length-2 simple paths from u to v,*

$$\mathbf{M}^k[u,v] = \binom{k}{2}\left(\frac{n^2-1}{n^2}\right)^{k-2}\frac{a}{n^6} + O(k^4/n^9).$$

Proof. Let $a_2, \ldots, a_k$ respectively denote the number of length-$2, \ldots, k$ walks from u to v without self loops. Then each $a_i = O(n^{i-1})$. Specifically, for every odd i, $a_i = 0$. Because the underlying graph of M without self-loops is a bipartite graph where u, v belong to the same part. We get

$$\mathbf{M}^k[u,v] = \binom{k}{2}\left(\frac{n^2-1}{n^2}\right)^{k-2}\frac{a_2}{n^6} + \sum_{i=4}^{k}\binom{k}{i}\left(\frac{n^2-1}{n^2}\right)^{k-i}\frac{a_i}{n^{3i}}$$

where the summation is upper bounded by the sum of a geometric series, and is dominated by the case $i = 4$. ◀

Since $k^4/n^9 = o(k^2/n^6)$, approximating the power of $\mathbf{M}$ can be used to answer the number of length-2 paths between vertices from V_1 and V_3 exactly. In this way, we get rid of the matrix, and are able to focus on a clean combinatorial task. In fact, we will show that even computing the parity of $\mathbf{M}^k[u,v]$ requires high data structure complexity.

This problem can also be rephrased in terms of counting set intersection: for every pair of vertices $(u,v) \in V_1 \times V_3$, the number of length-2 paths from u to v is exactly the size of the intersection of their neighbor sets.

The sequence of queries that we construct is similar to the above. We let it to be a concatenation of $n/3$ disjoint bijections between V_1 and V_3:

$$Q = (\underbrace{(1,2n/3+1),(2,2n/3+2),\ldots,(n/3,n)}_{\text{identity bijection}},\ldots,\underbrace{(1,n),(2,2n/3+1),\ldots,(n/3,n-1)}_{\text{shifted bijection}})$$

Formally, for every $i \in [n/3]$, the i-th bijection is exactly a sequence of pairs (u,v) where

- $u \in V_1$;
- $v = (u + i - 2) \bmod (n/3) + 2n/3 + 1$.

The following observation is immediate from our construction.

▶ **Observation 30.** *For every consecutive $m \ge n$ queries in Q, there does not exist an index $v \in [n]$ that is covered more than $\lceil \frac{m}{n/3} \rceil$ times.*

Even though the input distribution is a clean uniform distribution, the conditional min-entropy remains non-trivial to analyze. Our strategy, analogous to the previous section, is to find a "basis" from the input elements to the outputs. We show that we can always find such a basis for a large set M of "good inputs".

▶ **Definition 31.** *Let* $\hat{\mathbf{M}} \in \mathbb{F}_2^{n\times n}$ *denote the binary matrix where each element* $\hat{\mathbf{M}}[i,j] = 1$ *if and only if* $\mathbf{M}[i,j] \neq 0$.

Let $\mathsf{M} \subseteq \mathbb{R}^{n\times n}$ *be a set of "good" input matrices* $\mathbf{M}$ *such that for every* $d \in [10\log n, n/3]$, *and every* $V_1' \subseteq V_1, V_2' \subseteq V_2$, $|V_1'| = |V_2'| = d$, *the submatrix has a high rank:*

$$\operatorname{rank}(\hat{\mathbf{M}}[V_1', V_2']) \geq d/2.$$

Notice that the submatrix is a uniformly random square matrix, where there are classic results bounding its rank.

▶ **Proposition 32** (page 38 of [3]; see also page 1 of [12])**.** *Let* $\hat{\mathbf{M}}$ *be a uniformly random* $n \times n$ *matrix over* $\mathbb{F}_2$. *For every* $k \in \{0, \ldots, n\}$,

$$\Pr[\operatorname{rank}(\hat{\mathbf{M}}) = n-k] = \frac{1}{2^{k^2}} \frac{\prod_{i=1}^{n}(1-1/2^i)\prod_{i=k+1}^{n}(1-1/2^i)}{\prod_{i=1}^{n-k}(1-1/2^i)\prod_{i=1}^{k}(1-1/2^i)}.$$

Formally, the following lemma shows that the input matrix is good with high probability.

▶ **Lemma 33.** *With probability* $\geq 1 - o(1)$, *a random input matrix from* $\mathcal{M}$ *is good.*

Proof. For every $d \in [10\log n, n/3]$, there are $\binom{n/3}{d}^2 \leq (\frac{en/3}{d})^{2d}$ submatrices of dimension $d \times d$. For each submatrix $\hat{\mathbf{M}}[V_1', V_2']$, and every $k \in [d/2+1, \ldots, d]$, by Proposition 32 we have

$$\Pr[\operatorname{rank}(\hat{\mathbf{M}}[V_1', V_2']) = d-k] = \frac{1}{2^{k^2}} \frac{\prod_{i=1}^{d}(1-1/2^i)\prod_{i=k+1}^{d}(1-1/2^i)}{\prod_{i=1}^{d-k}(1-1/2^i)\prod_{i=1}^{k}(1-1/2^i)} \leq 2^{d-d^2/4}.$$

Therefore,

$$\Pr[\operatorname{rank}(\hat{\mathbf{M}}[V_1', V_2']) \leq d/2] \leq (d/2)2^{d-d^2/4}.$$

By union bound, the probability that at least one of the submatrices have a low rank is

$$\Pr[\mathbf{M} \text{ is bad}] \leq \sum_{d=10\log n}^{n/3} (\frac{en/3}{d})^{2d}(d/2)2^{d-d^2/4} \leq \sum_{d=10\log n}^{n/3} 2^{-5\log^2 n} = o(1). \qquad \blacktriangleleft$$

Equipped with the above tools, we are ready to prove the lower bound to AMP with high redundancy.

▶ **Theorem 34** (Lower bound to AMP with superlinear-in-n redundancy)**.** *Fix* $2 \leq k \leq n$ *and a constant* $\alpha > 6$. *For every (possibly randomized) succinct and systematic data structure with* $r \in [10n\log n, n^2/1000]$ *bits of redundancy that answers the whole approximated matrix power* $[\mathbf{M}^k]_\alpha$ *correctly with* $\geq 99/100$ *probability, its total number of probes is* $\Omega(n^4/r)$ *in the worst case.*

Proof. For each $(u,v) \in V_1 \times V_3$, to approximate $[\mathbf{M}^k]_\alpha$ for $\alpha > 6$, by Observation 29, the algorithm should also be able to answer exactly the number of length-2 paths from u to v.

By Yao's minimax principle, there is a randomized data structure that answer the fixed sequence of queries Q with high probability efficiently for every possible input only if there exists a deterministic data structure that with the same success probability and the same complexity given a random input drawn from $\mathcal{M}$. In below, we focus on deterministic data structures and query-with-sketch algorithms.

By Lemma 3 and Lemma 8, the lower bound reduces to showing that for every block of $m = 20r$ consecutive queries from Q, and for every partial assignment σ of length at most $q := 0.001n^2$ to $\mathbf{M}$, the joint min-entropy of the output conditioned on σ is $\geq 2r$. We will focus on the min-entropy of the unrounded elements of $\mathbf{M}^k$ in below, as it is equivalent to the min-entropy of the rounded elements of $[\mathbf{M}^k]_\alpha$, given our input distribution $\mathcal{M}$. Besides, in applying Lemma 8, we only pick a good input set M constructed above, and set $\mathsf{M}_\sigma = \mathbb{R}^{n\times n}$ for every σ.

By our construction, the input matrix $\mathbf{M}$ is fixed other than two submatrices, $\mathbf{M}[V_1 \times V_2]$ and $\mathbf{M}[V_2 \times V_3]$. Given a partial assignment of length at most q, there are at most $0.1n$ vertices from $V_1 \cup V_3$, where each of their corresponding rows (if in V_1) or columns (if in V_3) has $\geq 0.01n$ elements revealed from σ. We call these vertices "bad vertices", and other vertices "good vertices". Let $Q' = ((u_1, v_1), \ldots, (u_{m'}, v_{m'}))$ be the subsequence of the m consecutive queries containing only pairs of good vertices. By Observation 30, we have

$$m' \geq m - 0.1n \cdot (\frac{m}{n/3} + 1) \geq 0.69m$$

There exists a sequence $(w_1, \ldots, w_{m'})$ such that the sequence of triples $(u_1, v_1, w_1), \ldots, (u_{m'}, v_{m'}, w_{m'})$ are *edge-disjoint* triples where none of the edges $(u_i, w_i), (w_i, v_i)$ are revealed by σ, which means that none of pairs of indices $(u, v), (v, w)$, or (u, w) belong to multiple triples. Its existence directly follows from a construction: we decide each w_i one by one. For each (u_i, v_i), given $w_{<i}$ fixed, among $n/3$ indices from V_2, at least $n/3 - 2\lceil \frac{m}{n/3} \rceil - 0.02n \geq 0.19n$ elements of $\mathbf{M}[u_i, \cdot]$ and $\mathbf{M}[\cdot, v_i]$ are unfixed by σ and $w_{<i}$, which means there always exists an w_i that satisfies our condition.

Observe that for every partial assignment σ of length $\leq q$,

$$\begin{aligned}&H_{\min}(\mathbf{M}^k[u_1, v_1], \ldots, \mathbf{M}^k[u_{m'}, v_{m'}], \mathbf{M} \in \mathsf{M}|\sigma)\\ \geq &\min_{\sigma'} H_{\min}(\mathbf{M}^k[u_1, v_1], \ldots, \mathbf{M}^k[u_{m'}, v_{m'}], \mathbf{M} \in \mathsf{M}|\sigma'])\end{aligned}$$

where σ' is selected from all possible partial assignments to $\mathbf{M}$ that is consistent to σ and reveals all the elements other than the m' elements $\mathbf{M}[w_1, v_1], \ldots, \mathbf{M}[w_{m'}, v_{m'}]$.

Fix a vertex $v \in V_3$ such that there are $\geq 10 \log n$ queries on v in Q'. Since $m' \geq 0.69m \geq 13.8r \geq 138n \log n$, at least $\geq 202m'/207$ queries from Q' are made on such vertices $v \in V_3$. Or otherwise there will be less than

$$\frac{n}{3} \cdot 10 \log n + \frac{202m'}{207} \leq m'$$

queries in Q'.

Let $d \geq 10 \log n$ be the number of queries on v in Q', and $(u'_1, v, w'_1), \ldots, (u'_d, v, w'_d)$ the triples. By our construction of w and Q, none of the vertices $u'_i \in V_1$ and $w'_i \in V_2$ repeat in the d triples. In addition, conditioned on σ', different columns of the matrix power $\mathbf{M}^k[\cdot, v]$ are independent. The outputs $\mathbf{M}^k[u'_1, v], \ldots, \mathbf{M}^k[u'_d, v]$ only depend on variables $\mathbf{M}[w'_1, v], \ldots, \mathbf{M}[w'_d, v]$. More precisely, each $\mathbf{M}^k[u'_i, v]$ is a linear combination of the variables: for every $i \in [d]$,

$$\mathbf{M}^k[u'_i, v] = \underbrace{\sum_{w:\forall j, w\neq w'_j} \mathbf{M}[u'_i, w]\mathbf{M}[w, v]}_{\text{fixed given } \sigma'} + \sum_{j=1}^{d} \mathbf{M}[u'_i, w'_j]\mathbf{M}[w'_j, v]$$

We only need to show that the set of vectors $\{(\hat{\mathbf{M}}[u'_i, w'_j])_{j\in[d]}\}_{i\in[d]}$ form a basis of rank at least $d/2$. If it is true, the parity of the output elements, which can always be presented as a linear combination of these vectors, will have $\geq d/2$ min-entropy.

Formally, let $\mathbf{y} \in \mathbb{F}_2^d$ be the vector where $\mathbf{y}(i) = \hat{\mathbf{M}}^k[u'_i, v]$. Let $\mathbf{x} \in \mathbb{F}_2^d$ be the vector where $\mathbf{x}(i) = \hat{\mathbf{M}}[w_i, v]$. Let $V'_1 = \{u'_1, \ldots, u'_d\}$ and $V'_2 = \{w'_1, \ldots, w'_d\}$ Then for every σ', there exists a vector $\mathbf{w} \in \mathbb{F}_2^d$ such that

$$\mathbf{y} = \mathbf{w} + \hat{\mathbf{M}}[V'_1, V'_2]\mathbf{x}$$

where $\mathbf{x}$ is a uniformly random vector over $\mathbb{F}_2^d$, given σ'. Therefore,

$$\begin{aligned}
&H_{\min}(\mathbf{M}^k[u'_1, v], \ldots, \mathbf{M}^k[u'_d, v], \mathbf{M} \in \mathsf{M}|\sigma') \\
\geq &\min_{\sigma'} H_{\min}(\mathbf{y}|\sigma') \\
= &\min_{\sigma'} H_{\min}(\hat{\mathbf{M}}[V'_1, V'_2]\mathbf{x}|\sigma') \\
= &\min_{\sigma'} \operatorname{rank}(\hat{\mathbf{M}}[V'_1, V'_2]) \\
\geq &d/2.
\end{aligned}$$

Here, the min-entropy of the output is greater than or equal to the min-entropy of $\mathbf{y}$ because by Observation 29, $\mathbf{M}^k[u, v]$ determines the number of length-2 walks from u to v, and hence its parity.

Therefore, each vertex v of d queries contribute $\geq d/2$ min-entropy. In addition, different columns $\mathbf{M}^k[\cdot, v]$ of the output are independent given σ'. The total min-entropy of the output is at least $\frac{1}{2} \cdot \frac{202m'}{207}$, where $\frac{202m'}{207}$ is the least number of queries from Q' that are made on these vertices v. We conclude that

$$H_{\min}(\mathbf{M}^k[u_1, v_1], \ldots, \mathbf{M}^k[u_{m'}, v_{m'}], \mathbf{M} \in \mathsf{M}|\sigma) \geq \frac{101m'}{207} \geq 6.7r > 2r$$

By Lemma 3 and Lemma 8, the lower bound follows. ◀

The lower bound to the counting set intersection follows through a reduction from AMP.

▶ **Corollary 35** (Lower bound to counting set intersection). *Given a (possibly randomized) succinct and systematic data structure with $r \in [10n \log n, n^2/1000]$ bits of redundancy that given as input $2n/3$ subsets of $[n/3]$ answers their pairwise set intersection size correctly with $\geq 99/100$ probability, its total number of probes is $\Omega(n^4/r)$ at the worst case.*

We note that no super-linear-in-n amortized probe lower bound for counting set intersection can be obtained. Because the intersection size of every pair of sets can be answered in $O(n)$ probes.

Proof. Let the input sets follow i.i.d. uniform distribution from $\{0, 1\}^{n/3}$. Counting set intersection is exactly counting the number of length-2 paths given an input from $\mathcal{M}$, which we have established a lower bound above. ◀

5 Lower bounds to set disjointness and $(2 - \varepsilon)$-approximate APSP

In this section, we study the lower bound to the set disjointness, which is the decision version of counting set intersection. The same nemesis distribution and analysis also implies the same probe-redundancy trade-off for the $(2 - \varepsilon)$-approximation of APSP. We state our proof in the context of set disjointness.

Set disjointness problem

Given as input n sets $A_1, \ldots, A_n \subseteq [n]$. The succinct and systematic data structure, given a pair of indices $(u, v) \in [n] \times [n]$, must answer whether $|A_u \cap A_v| = 0$. For simplicity, we denote the answer as $Ans_{u,v}$, which is a binary bit that is 1 iff the answer is NO.

For the convenience of our analysis, each A_u also denotes a binary vector of length n, where the w-th bit denotes whether $w \in A_u$.

The uniform input distribution is no longer hard for this problem: when each vector is i.i.d. uniformly from $\{0,1\}^n$, with high probability for every u, v, $|A_u \cap A_v| \neq 0$. To that end, we construct the following nemesis distribution where 1s are sparse in the input vectors.

Nemesis input distribution

Let $\mathcal{A}$ denote the input distribution that, for every $u, i \in [n]$, $A_u(i) = 1$ with independent probability $1/\sqrt{n}\log n$.

One can see that $\Pr[Ans_{u,v} = 1] = \Theta(1/\log^2 n)$ for every $u \neq v$.

We use a similar sequence Q of queries as above. Let $V_1 := \{1, \ldots, n/2\}$ and $V_2 := \{n/2+1, \ldots, n\}$. We let Q to be a concatenation of $n/2$ disjoint bijections between V_1 and V_2:

$$Q = (\underbrace{(1, n/2+1), (2, n/2+2), \ldots, (n/2, n)}_{\text{identity bijection}}, \ldots, \underbrace{(1, n), (2, n/2+1), \ldots, (n/2, n-1)}_{\text{shifted bijection}})$$

Formally, for every $i \in [n/2]$, the i-th bijection is exactly a sequence of pairs (u, v) where

- $u \in V_1$;
- $v = (u + i - 2) \bmod (n/2) + n/2 + 1$.

The main result in this section is the following lower bound.

▶ **Theorem 36** (Lower bound to set disjointness). *Given a (possibly randomized) succinct and systematic data structure with $r \in [10n\log^2 n, n\sqrt{n}/1000\log^3 n]$ bits of redundancy that given as input n subsets of $[n]$ answers the sequence of queries Q correctly with $\geq 99/100$ probability, its total number of probes is $\Omega(n^4/(r\log^3 n))$ in the worst case.*

Different from the previous lower bounds to AMP, the answers cannot be formulated as (nearly-)linear combinations of selected input elements, where analytic or linear algebraic tools may apply. However, the sparsity of the sets still enables us to lower bounding the min-entropy. Formally, we define the following set of "good inputs".

▶ **Definition 37.** *Let* A *be the set of good inputs* $\{A = (A_1, \ldots, A_n)\}$ *satisfying the following conditions*

(1) *For every $u \in [n]$,*

$$0.99\frac{\sqrt{n}}{\log n} \leq |A_u| \leq 1.01\frac{\sqrt{n}}{\log n}.$$

(2) *For every $k \in [\log^2 n, \sqrt{n}]$, $u \in V_1$ and $v \in V_2$,*

$$\sum_{i=0}^{k-1} Ans_{u+i,v} \leq \frac{2k}{\log n}$$

where $Ans_{u+i,v}$ denotes $Ans_{u+i-n/2,v}$ if $u + i > n/2$.

(3) *For every* $k \in [\log^2 n, \sqrt{n}]$*, every* $u \in V_1$ *and every* $w \in [n]$*,*

$$\sum_{i=0}^{k-1} A_{u+i}(w) \leq 2 \log n$$

where $A_{u+i}(w)$ *denotes* $A_{u+i-n/2}(w)$ *if* $u + i > n/2$*.*

We will show that, for most of indices $v \in V_2$, for a fixed assignment to all the $A_1, \ldots, A_{n/2}$, and for a fixed sequence of answers $Ans_{\cdot,v}$, a large proportion of bits in A_v are forced to be 0 if the input is good, which results in a high conditional min-entropy.

We use the following form of Chernoff bound to show that a random input is good with high probability.

▶ **Proposition 38** (simplified Chernoff bound)**.** *Let* $X_1, X_2, \ldots, X_n$ *be independent random variables such that* $X_i \in [0, 1]$*. Let* $X = \sum_{i=1}^{n} X_i$*. Then, for every* $\delta > 0$*,*

$$\Pr[|X - \mathbb{E}[X]| \geq \delta \cdot \mathbb{E}[X]] \leq 2 \cdot \exp\left(-\frac{\delta^2}{2+\delta} \cdot \mathbb{E}[X]\right).$$

▶ **Lemma 39.** *For every large enough* n*, a random input from* $\mathcal{A}$ *is good with probability at least* $99/100$*.*

Proof. We will show that each condition holds with low probability. The desired probability bound follows from the union bound.

For condition (1), we know from our input construction that $\mathbb{E}[|A_u|] = \frac{\sqrt{n}}{\log n}$. By Chernoff bound,

$$\Pr\left[\left|A_u - \frac{\sqrt{n}}{\log n}\right| \geq 0.01\frac{\sqrt{n}}{\log n}\right] \leq \exp(-\Omega(\sqrt{n}/\log n)).$$

By union bound over all u, with $\geq 1 - o(1)$ probability condition (1) holds.

Then, we show that condition (2) holds with high probability conditioned on (1) is true. Fix an index $v \in V_2$ and an assignment α to A_v where $|\alpha| \in [0.99\frac{\sqrt{n}}{\log n}, 1.01\frac{\sqrt{n}}{\log n}]$. Conditioned on a fixed $A_v = \alpha$, for every u and i, $Ans_{u+i,v}$ are independent random variables where

$$\Pr[Ans_{u+i,v} = 0 | A_v = \alpha] = \left(1 - \frac{1}{\sqrt{n}\log n}\right)^{|\alpha|}.$$

By Observation 14 and the Taylor expansion $\exp(-x) = 1 - x + \frac{x^2}{2} - \ldots$, we have

$$\Pr[Ans_{u,v} = 1 | A_v = \alpha] \geq 1 - \exp\left(-\frac{|\alpha|}{\sqrt{n}\log n}\right) \geq \frac{|\alpha|}{\sqrt{n}\log n} - \frac{|\alpha|^2}{2n\log^2 n} \geq \frac{0.98}{\log^2 n}$$

and

$$\Pr[Ans_{u,v} = 1 | A_v = \alpha] \leq 1 - \exp\left(-\frac{|\alpha|}{\sqrt{n}\log n - 1}\right) \leq \frac{|\alpha|}{\sqrt{n}\log n - 1} \leq \frac{1.02}{\log^2 n}.$$

By Chernoff bound,

$$\Pr\left[\sum_{i=0}^{k-1} Ans_{u+i,v} > \frac{2k}{\log n} \middle| A_v = \alpha\right] \leq \exp(-1.9k/\log n) \leq O(n^{-2.7})$$

By union bound over u, v, and k, with $\geq 1 - o(1)$ probability condition (2) holds.

For condition (3), notice that $\mathbb{E}[\sum_{i=0}^{k-1} A_{u+i}(w)] = \frac{k}{\sqrt{n}\log n}$. By Chernoff bound, for every k, u, w,

$$\Pr\left[\sum_{i=0}^{k-1} A_{u+i}(w) > 5\log n\right] \leq O(n^{-2.7}).$$

Again, by union bound, condition (3) holds with $\geq 1 - o(1)$ probability. ◀

However, the above set of good inputs still does not guarantee a high conditional min-entropy. Imagine a magical algorithm that finds all the 1s in all the strings in $O(n\sqrt{n}/\log n)$ probes. No entropy would remain in the input. To that end, we construct good input sets A_σ for every partial assignment σ of length $\leq q$. Intuitively, the good input sets A_σ ensures that high uncertainty remains in the unprobed part of the input.

▶ **Definition 40.** *For every partial assignment σ of length $\leq 10^{-4}n^2$, let A_σ be the set of good inputs $\{A = (A_1, \ldots, A_n)\}$ such that for every shifted bijection $(1, v_1), \ldots, (n/2, v_{n/2})$ in the sequence of queries Q, no more than $0.15n$ indices v_i in V_2 satisfy the following condition: let $W_{i,v_i} \subseteq [n]$ denote the set of indices w such that $A_i(w) = 1$ and $A_{v_i}(w)$ is revealed by σ; then $|W_{i,v_i}| > 0.5\sqrt{n}/\log n$.*

To show that a random input falls in A_σ with high probability conditioned on σ, we will use the following standard Chernoff bound.

▶ **Proposition 41** (standard Chernoff bound [28])**.** *Let $X_1, \ldots, X_n$ be independent random variables with $X_i \in \{0, 1\}$ and $\Pr[X_i = 1] \in (0, 1)$ for $i \in [n]$. Then, for $X = \sum_{i=1}^n X_i, \mu = \mathbb{E}[X]$, and any $\delta > 0$,*

$$\Pr[X > (1+\delta)\mu] < \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right)^\mu.$$

▶ **Lemma 42.** *For every partial assignment σ of length $\leq 10^{-4}n^2$, we have*

$$\Pr_{A \leftarrow \mathcal{A}}[A \in \mathsf{A} - \mathsf{A}_\sigma | \sigma] \leq \exp(-0.0048n\sqrt{n}/\log n).$$

Proof. We show that for every fixed shifted bijection $(1, v_1), \ldots, (n/2, v_{n/2})$, the condition is satisfied with high probability. And we conclude with a union bound over all the n shifted bijections.

Given σ of length at most q, there are at most $0.01n$ vectors in which $\geq 0.01n$ bits are revealed from σ. Without loss of generality, we assume that the first $0.49n$ pairs $(1, v_1), \ldots, (0.49n, v_{0.49n})$ do not contain such vectors.

For each of these pairs (i, v_i) for $i \in [0.49n]$, there are $\geq 0.98n$ indices $w \in [n]$ such that neither of $A_i(w)$ nor $A_{v_i}(w)$ are revealed by σ. Without loss of generality, we assume these indices are $w = 1, \ldots, 0.98n$. When $A \in \mathsf{A}$ and $|W_{i,v_i}| > 0.5\sqrt{n}/\log n$, we should have $\sum_{w=1}^{0.98n} A_i(w) < 0.51\sqrt{n}/\log n$, by $|A_i| \leq 1.01\sqrt{n}/\log n$. We apply the simplified Chernoff bound (Proposition 38) with $\mathbb{E}[\sum_{w=1}^{0.98n} A_i(w)|\sigma] = 0.98\sqrt{n}/\log n$, and obtain

$$\Pr[\sum_{w=1}^{0.98n} A_i(w) < 0.51\sqrt{n}/\log n | \sigma] \leq 2\cdot\exp(-\frac{47}{243}\cdot 0.51\sqrt{n}/\log n) \leq 2\cdot\exp(-0.098\sqrt{n}/\log n).$$

Let $X_1, \ldots, X_{0.49n} \in \{0, 1\}$ be the indicators of whether $|W_{i,v_i}| > 0.5\sqrt{n}/\log n$. Then, conditioned on σ, these indicators are independent random bits, each evaluating to 1 with probability $\leq 2 \cdot \exp(-0.098\sqrt{n}/\log n)$. $A \in \mathsf{A} - \mathsf{A}_\sigma$ only if $X := \sum_{i=1}^{0.49n} X_i > 0.05n$.

By applying the standard Chernoff bound (Proposition 41) with $(1+\delta)\mu = 0.05n$ and $\mu \le 0.98n\exp(-0.098\sqrt{n}/\log n)$, we get

$$\Pr[X > 0.05n] < \exp(-0.0048n\sqrt{n}/\log n).$$

The desired inequality follows by a union bound over all $n/2$ shifted bijections. ◀

Given an input with the above conditions, we are able to prove the lower bound.

Proof to Theorem 36. Again, by Yao's minimax principle, we may focus on proving lower bounds for data structures given a random input from $\mathcal{A}$. By Lemma 3 and Lemma 8, the lower bound reduces to showing that for every $m := 120r\log^3 n$ consecutive queries $(u_1, v_1), \ldots, (u_m, v_m)$ from Q, and for every partial assignment σ of length at most $q := 10^{-4}n^2$ to $\mathbf{M}$, the joint min-entropy of the output conditioned on σ is $\ge 2r$. We may always assume that m is a multiple of $n/2$, as this will only increase m by at most $n/2 = o(r\log^3 n)$. In addition, this ensures that the consecutive m queries is a concatenation of shifted bijections. As shown in Lemma 39 and Lemma 42, the input falls in A with high probability, and falls in A_σ with high probability for every σ.

Observe that for every partial assignment σ of length $\le q$,

$$\begin{aligned}&H_{\min}(Ans_{u_1,v_1}, \ldots, Ans_{u_m,v_m}, A \in \mathsf{A} \cap \mathsf{A}_\sigma | \sigma)\\ \ge &\min_{\sigma'} H_{\min}(Ans_{u_1,v_1}, \ldots, Ans_{u_m,v_m}, A \in \mathsf{A} \cap \mathsf{A}_\sigma | \sigma')\end{aligned}$$

where σ' is selected from all possible partial assignments to $A = (A_1, \ldots, A_n)$ that is consistent to σ and reveals all the $A_1, \ldots, A_{n/2}$. We will show that the min-entropy remains high even conditioned on σ'.

Before bounding the min-entropy, let us further pick an analyzable part from the remaining input. Given σ of length at most q, there are at most $0.01n$ vectors in which $\ge 0.01n$ bits are revealed from σ. We call these vectors "bad vectors", and other vectors "good vectors".

By our construction to Q, for every m consecutive queries, each A_u for $u \in [n]$ is involved in exactly $m/(n/2)$ queries. Therefore, at least $0.98m$ queries are among good vectors. In addition, at most $0.3m$ queries (u, v) has $|W_{u,v}| > 0.5\sqrt{n}\log n$. Denote by $m' \ge 0.68m$ the number of remaining queries, and the sequence of remaining queries Q'.

Fix an index $v \in V_2$ such that it is contained in at least $m'/3n$ queries in Q'. There are at least $\ge 0.31n$ many such vertices, otherwise the total number of queries in Q' is

$$< 0.31n \cdot \frac{2m}{n} + \left(\frac{n}{2} - 0.31n\right) \cdot \frac{m'}{3n} \le m'$$

where $2m/n$ is the maximum number of queries that involve a vertex $v \in V_2$ in m consecutive queries.

We denote by $d \ge m'/3n > \log^2 n$ the number of queries that contain v, and $(u'_1, v), \ldots, (u'_d, v) \in Q'$ these queries. Given all the vectors in V_1 fixed by σ', these queries are independent of other queries in Q'.

Assuming a good input, by the condition (2) of Definition 37, at most $\frac{2\cdot 2m/n}{\log n} = \frac{4m}{n\log n}$ out of d queries have answers of 1. Recall that condition (2) gives an upper bound to the sum of answers of a consecutive sequence of queries to every fixed v. Note from the construction to Q that the m consecutive queries contain exactly $2m/n \le \sqrt{n}$ consecutive queries on v. While Q' is obtained by removing elements from the m queries, the sum of answers of queries in Q' is smaller than or equal to the consecutive sum in Q, and is also bounded by condition (2).

Without loss of generality, we assume that they are the first $d' \leq \frac{4m}{n \log n}$ queries:

$Ans_{u'_1,v} = \cdots = Ans_{u'_{d'},v} = 1$.

For each $u \in [n]$, we use A'_u to denote the vector A_u removed all the indices w where $A_v(w)$ is revealed by σ. We will show that the Hamming weight of $A'_{u'_{d'+1}} \vee \cdots \vee A'_{u'_d}$ is high. That is, given σ', for every fixed and possible answer to good inputs, a large enough proportion of bits of A'_v are forced to be 0, to avoid intersecting with any $A'_{u'_{d'+1}}, \ldots, A'_{u'_d}$. Therefore, the conditional probability this happens is always small.

Notice that by condition (1) of Definition 37 and Definition 40,

$$\sum_{i=d'+1}^{d} |A'_{u'_i}| \geq (d - d') \cdot 0.49 \frac{\sqrt{n}}{\log n} > \frac{9.7r \log^2 n}{\sqrt{n}}$$

where for each index $w \in [n]$, by condition (3) of Definition 37,

$$\sum_{i=d'+1}^{d} |A'_{u'_i}(w)| \leq 2 \log n.$$

Therefore, for the Hamming weight of the OR of these binary strings,

$$\left| A'_{u_{d'+1}} \vee \cdots \vee A'_{u_d} \right| \geq \frac{9.7r \log^2 n}{\sqrt{n}} \cdot \frac{1}{2 \log n} = \frac{4.8r \log n}{\sqrt{n}}.$$

Since given σ, A'_v is a random vector where each bit is 1 with independent probability $1/\sqrt{n} \log n$, we have

$$\Pr[A'_v \wedge (A'_{u_{d'+1}} \vee \cdots \vee A'_{u_d}) = 0 \text{ and } A \in \mathsf{A} \cap \mathsf{A}_\sigma | \sigma'] \leq (1 - 1/\sqrt{n} \log n)^{\frac{4.8r \log n}{\sqrt{n}}} \leq \exp(-4.8r/n).$$

That means that the min-entropy of queries on v is at least

$4.8 \log(e) r/n \geq 6.92r/n$.

Since the min-entropy on different $v \in V_2$ conditioned on σ' are independent, the total min-entropy is lower bounded by their sum:

$$H_{\min}(Ans_{u_1,v_1}, \ldots, Ans_{u_l,v_l}, A \in \mathsf{A} \cap \mathsf{A}_\sigma | \sigma) \geq 0.31n \cdot \frac{6.92r}{n} > 2r$$

Therefore, the lower bound holds for set disjointness. ◀

The above lower bound also applies for the $(2 - \varepsilon)$-approximate APSP, for every $\varepsilon > 0$. We obtain it via a reduction to set disjointness under the nemesis distribution $\mathcal{A}$.

$(2 - \varepsilon)$-approximate all-pairs shortest paths (APSP)

Fix a parameter $\varepsilon > 0$, Given as input an undirected graph $G = (V, E)$. The succinct and systematic data structure, given a sequence of all pairs of vertices $(u, v) \in \binom{V}{2}$, must answer a value $d \in [d_G(u, v), (2 - \varepsilon) d_G(u, v)]$ for each (u, v), where $d_G(u, v)$ denotes the shortest distance between u and v in G.

▶ **Theorem 43** (Lower bound to $(2 - \varepsilon)$-approximate APSP). *Fix $\varepsilon > 0$. Given a (possibly randomized) succinct and systematic data structure with $r \in [5n \log^2 n, n\sqrt{n}/4000 \log^3 n]$ bits of redundancy that given as input a graph G on n vertices $(2 - \varepsilon)$-approximates all pairs shortest paths distances correctly with $\geq 99/100$ probability, its total number of probes is $\Omega(n^4/(r \log^3 n))$ at the worst case.*

Proof. We prove the lower bound through a reduction from set disjointness. For every instance of set disjointness, we construct an input to APSP, such that data structures to APSP, through the reduction, can be revised into data structures for set disjointness with the same success probability and complexity.

Suppose for the sake of contradiction that there exists a data structure D of bounded complexity for $(2-\varepsilon)$-APSP. Given an input from $\mathcal{A}$. We construct a graph $G = (V, E)$ of $2n$ vertices, where $V = V_1 \cup W \cup V_2$ and $2|V_1| = |W| = 2|V_2| = n$. For every $u \in V_1$ and $w \in [n]$, u connects to the w-th vertex in W if and only if $A_u(w) = 1$. Analogously, for every $v \in V_2$ and $w \in [n]$, v connects to the w-th vertex in W if and only if $A_v(w) = 1$.

For each query (u, v) from Q, we ask D about the shortest path length between $u \in V_1$ and $v \in V_2$ in G. If the answer $d \in [2, 4)$, we answer that $|A_u \cap A_v| \geq 1$; otherwise, we answer $|A_u \cap A_v| = 0$.

To see the correctness, by our construction, $d_G(u, v) = 2$ if and only if $|A_u \cap A_v| \geq 1$; otherwise $d_G(u, v) \geq 4$. The constructed graph is a bipartite graph on two parts $V_1 \cup V_2$ and W, and there cannot be paths of length 3 between u and v. We get a data structure for set disjointness with the same redundancy and the same probe complexity. Therefore, the same lower bound holds for $(2-\varepsilon)$-approximate APSP. ◀

References

1. Peyman Afshani and Jesper Sindahl Nielsen. Data structure lower bounds for document indexing problems. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, Rome, Italy, July 11-15, 2016*, volume 55 of *LIPIcs*, pages 93:1–93:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.93.
2. Richard Beigel and Tirza Hirst. One help bit doesn't help. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 124–130. ACM, 1998. doi:10.1145/276698.276720.
3. Eric David Belsley. *Rates of convergence of Markov chains related to association schemes*. Harvard University, 1993.
4. Vladimir I Bogachev. Measures on topological spaces. In *Measure theory*, pages 476–583. Springer, 2007.
5. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998. doi:10.1016/S0169-7552(98)00110-X.
6. Karl Bringmann and Kasper Green Larsen. Succinct sampling from discrete distributions. In *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, pages 775–782, 2013. doi:10.1145/2488608.2488707.
7. Diptarka Chakraborty, Lior Kamma, and Kasper Green Larsen. Tight cell probe bounds for succinct boolean matrix-vector multiplication. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1297–1306, 2018. doi:10.1145/3188745.3188830.
8. Hagai Cohen and Ely Porat. Fast set intersection and two-patterns matching. *Theoretical Computer Science*, 411(40-42):3795–3800, 2010. doi:10.1016/j.tcs.2010.06.002.
9. Henry Corrigan-Gibbs and Dmitry Kogan. The function-inversion problem: Barriers and opportunities. In *Theory of Cryptography Conference*, pages 393–421, 2019. doi:10.1007/978-3-030-36030-6_16.
10. Zeev Dvir, Alexander Golovnev, and Omri Weinstein. Static data structure lower bounds imply rigidity. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 967–978, 2019. doi:10.1145/3313276.3316348.

11 Pavel Dvorák, Michal Koucký, Karel Král, and Veronika Slívová. Data structures lower bounds and popular conjectures. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms, ESA 2021, Lisbon, Portugal (Virtual Conference), September 6-8, 2021*, volume 204 of *LIPIcs*, pages 39:1–39:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ESA.2021.39.

12 Jason Fulman and Larry Goldstein. Stein's method and the rank distribution of random matrices over finite fields. *The Annals of Probability*, 43(3):1274–1314, 2015.

13 Anna Gál and Peter Bro Miltersen. The cell probe complexity of succinct data structures. *Theoretical computer science*, 379(3):405–417, 2007. doi:10.1016/j.tcs.2007.02.047.

14 Isaac Goldstein, Tsvi Kopelowitz, Moshe Lewenstein, and Ely Porat. Conditional lower bounds for space/time tradeoffs. In *Workshop on Algorithms and Data Structures*, pages 421–436, 2017. doi:10.1007/978-3-319-62127-2_36.

15 Isaac Goldstein, Moshe Lewenstein, and Ely Porat. On the hardness of set disjointness and set intersection with bounded universe. In Pinyan Lu and Guochuan Zhang, editors, *30th International Symposium on Algorithms and Computation, ISAAC 2019, Shanghai University of Finance and Economics, Shanghai, China, December 8-11, 2019*, volume 149 of *LIPIcs*, pages 7:1–7:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ISAAC.2019.7.

16 Alexander Golovnev, Siyao Guo, Thibaut Horel, Sunoo Park, and Vinod Vaikuntanathan. Data structures meet cryptography: 3sum with preprocessing. In *Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing*, pages 294–307, 2020. doi:10.1145/3357713.3384342.

17 Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

18 Alexander Golynski. Optimal lower bounds for rank and select indexes. *Theoretical Computer Science*, 387(3):348–359, 2007. doi:10.1016/j.tcs.2007.07.041.

19 Alexander Golynski, Roberto Grossi, Ankur Gupta, Rajeev Raman, and S. Srinivasa Rao. On the size of succinct indices. In Lars Arge, Michael Hoffmann, and Emo Welzl, editors, *Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*, volume 4698 of *Lecture Notes in Computer Science*, pages 371–382. Springer, 2007. doi:10.1007/978-3-540-75520-3_34.

20 Alexander Golynski, Rajeev Raman, and S Srinivasa Rao. On the redundancy of succinct data structures. In *Scandinavian Workshop on Algorithm Theory*, pages 148–159, 2008. doi:10.1007/978-3-540-69903-3_15.

21 Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 21–30, 2015. doi:10.1145/2746539.2746609.

22 Roger A. Horn and Charles R. Johnson. *Matrix Analysis, 2nd Ed.* Cambridge University Press, 2012. doi:10.1017/CBO9781139020411.

23 Guy Joseph Jacobson. *Succinct static data structures*. Carnegie Mellon University, 1988.

24 Tsvi Kopelowitz and Virginia Vassilevska Williams. Towards optimal set-disjointness and set-intersection data structures. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, Saarbrücken, Germany (Virtual Conference), July 8-11, 2020*, volume 168 of *LIPIcs*, pages 74:1–74:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.74.

25 Kasper Green Larsen. Higher cell probe lower bounds for evaluating polynomials. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 293–301. IEEE, 2012. doi:10.1109/FOCS.2012.21.

26 Peter Bro Miltersen. Lower bounds on the size of selection and rank indexes. In *SODA*, volume 5, pages 11–12, 2005. URL: http://dl.acm.org/citation.cfm?id=1070432.1070435.

27 Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 103–111, 1995. doi:10.1145/225058.225093.

28 Rajeev Motwani and Prabhakar Raghavan. Randomized algorithms. *ACM Computing Surveys (CSUR)*, 28(1):33–37, 1996. doi:10.1145/234313.234327.

29 Noam Nisan, Steven Rudich, and Michael Saks. Products and help bits in decision trees. *SIAM Journal on Computing*, 28(3):1035–1050, 1998. doi:10.1137/S0097539795282444.

30 James R. Norris. *Markov chains*. Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, 1998.

31 Mihai Patrascu and Liam Roditty. Distance oracles beyond the thorup-zwick bound. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 815–823. IEEE, 2010. doi:10.1109/FOCS.2010.83.

32 Mihai Patrascu and Emanuele Viola. Cell-probe lower bounds for succinct partial sums. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 117–122. SIAM, SIAM, 2010. doi:10.1137/1.9781611973075.11.

33 Sivaramakrishnan Natarajan Ramamoorthy and Cyrus Rashtchian. Equivalence of systematic linear data structures and matrix rigidity. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, Seattle, Washington, USA, January 12-14, 2020*, volume 151 of *LIPIcs*, pages 35:1–35:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.35.

34 Stanislaw J Szarek. Spaces with large distance to ℓ_∞^n and random matrices. *American Journal of Mathematics*, 112(6):899–942, 1990.

35 Robert Endre Tarjan. A class of algorithms which require nonlinear time to maintain disjoint sets. *Journal of computer and system sciences*, 18(2):110–127, 1979. doi:10.1016/0022-0000(79)90042-4.

36 Robert C Thompson. The behavior of eigenvalues and singular values under perturbations of restricted rank. *Linear Algebra and its Applications*, 13(1-2):69–78, 1976.

37 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM (JACM)*, 52(1):1–24, 2005. doi:10.1145/1044731.1044732.

38 Leslie G Valiant. Graph-theoretic arguments in low-level complexity. In *International Symposium on Mathematical Foundations of Computer Science*, pages 162–176, 1977. doi:10.1007/3-540-08353-7_135.

39 Leslie G Valiant. Why is boolean complexity theory difficult. *Boolean Function Complexity*, 169(84-94):4, 1992.

40 Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

41 Emanuele Viola. Lower bounds for data structures with space close to maximum imply circuit lower bounds. *Theory of Computing*, 15(1):1–9, 2019. doi:10.4086/toc.2019.v015a018.

42 Feng Wei. Upper bound for intermediate singular values of random matrices. *Journal of Mathematical Analysis and Applications*, 445(2):1530–1547, 2017.

43 Andrew Chi-Chih Yao. Should tables be sorted? *Journal of the ACM (JACM)*, 28(3):615–628, 1981. doi:10.1145/322261.322274.

A Missing proof to Lemma 13

The probability we are going to bound is the probability that any of the dimension of the $\mathbf{M}^k[u, v]$ is not close to $a/2n^{\alpha'}$, for integers a. It is non-trivial to bound since $\mathbf{M}^k$ does not follow a uniform distribution.

Specifically, we will show that each dimension of $\mathbf{M}^k$ is close to such a value with probability $\leq 1/\text{poly}(n)$. The desired probability bound follows by a union bound. For each dimension $[u, v]$ of $\mathbf{M}^k$, we are going to show that, even when all the other elements of $\mathbf{M_U}$ are fixed by an arbitrarily partial assignment σ, the function $\phi_{k,\sigma} : \mathbb{R} \to \mathbb{R}$ that maps from $\mathbf{M_U}[u, v]$ to $\mathbf{M}^k[u, v]$ evaluates to a value that is close to $a/2n^{\alpha'}$ with low probability. This is because $\phi_{k,\sigma}$ is an increasing function with its derivative well-controlled everywhere.

First, we need the following partial derivative bounds that generalizes Lemma 15 and Lemma 16, and works for every pair of indices $[u, v], [i, j]$. Recall that we assume that each ${\mathbf{M_U}}^k[u, v]$ is a function on n^2 variables $\mathbf{M_U}[i, j]$. For simplicity, we study for now their partial derivatives without the constraint of being a symmetric matrix. Given the symmetric constraint, their partial derivative is exactly $\frac{\partial {\mathbf{M_U}}^k[u,v]}{\partial \mathbf{M_U}[i,j]} + \frac{\partial {\mathbf{M_U}}^k[u,v]}{\partial \mathbf{M_U}[j,i]}$ when $i \neq j$, and is $\frac{\partial {\mathbf{M_U}}^k[u,v]}{\partial \mathbf{M_U}[i,j]}$ otherwise.

▶ **Lemma 44.** *Let $\gamma > 2$, and the values of elements of $\mathbf{M_U}$ range from $[\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]$. For every integer $2 \leq k \leq n$, for every element $[i, j]$ of $\mathbf{M_U}$ and every element $[u, v]$ of ${\mathbf{M_U}}^k$, we have the following bounds of $\frac{\partial {\mathbf{M_U}}^k[u,v]}{\partial \mathbf{M_U}[i,j]}$:*

$$\frac{\partial {\mathbf{M_U}}^k[u, v]}{\partial \mathbf{M_U}[i, j]} = \begin{cases} \Theta(\frac{n}{n^2}), & u = i \text{ or } v = j \\ \Theta(\frac{k-2}{n^2}), & \text{otherwise} \end{cases}$$

If $k = 1$, we have:

$$\frac{\partial {\mathbf{M_U}}^k[u, v]}{\partial \mathbf{M_U}[i, j]} = \begin{cases} 1, & u = i \text{ and } v = j \\ 0, & \text{otherwise} \end{cases}$$

Proof. We generalize Lemma 15 to every pair of vertices $[u, v]$ and $[i, j]$. Recall that

$$\frac{\partial {\mathbf{M_U}}^k[u, v]}{\partial \mathbf{M_U}[i, j]} = \sum_{g=0}^{k-1} \left(\sum_{u=h_0,h_1,\ldots,h_g=i,h_{g+1}=j,\ldots,h_{k-1},h_k=v} \left(\prod_{t=0,t\neq g}^{k-1} \mathbf{M_U}[h_t, h_{t+1}] \right) \right)$$

Again, $\prod_{t=0,t\neq g}^{k-1} \mathbf{M_U}[h_t, h_{t+1}]$ can be lower-bounded by setting all elements in $\mathbf{M_U}$ equal to $\frac{1}{n} - \frac{1}{n^\gamma}$ and upper bounded by setting all elements in $\mathbf{M_U}$ equal to $\frac{1}{n}$. We only need to count the number of sequences $u = h_0, h_1, ..., h_{k-1}, h_k = v$ in which two adjacent indices are i and j, while in both the lower and upper bounds all elements are equivalent. When g is picked from $\{1, 2, ..., k-2\}$, there are $(k-2)n^{k-3}$ distinct sequences $h_0, \ldots, h_k$ for each summation. g can be 0 (resp., $k-1$) only if u (resp., v) coincides with i (resp., j). More precisely, the number of extra and distinct sequences for both cases $u = i$ and $j = v$ are exactly n^{k-2}.

Therefore, when $u \neq i$ and $v \neq j$,

$$\frac{\partial {\mathbf{M_U}}^k[u, v]}{\partial \mathbf{M_U}[i, j]} \leq (k-2)n^{k-3}\frac{1}{n^{k-1}} = \frac{k-2}{n^2}$$

and

$$\frac{\partial {\mathbf{M_U}}^k[u, v]}{\partial \mathbf{M_U}[i, j]} \geq (k-2)n^{k-3}\left(\frac{1}{n} - \frac{1}{n^\gamma}\right)^{k-1} \geq e^{-\frac{k-1}{n^{\gamma-1}-1}} \cdot \frac{k-2}{n^2} \geq (1 - o(1)) \cdot \frac{k-2}{n^2}.$$

by Observation 14 and $\gamma > 2$.

Similarly, for $u = i$ or $v = j$ but not both,

$$(1 - o(1))\frac{n + k - 2}{n^2} \leq \frac{\partial \mathbf{M_U}^k[u, v]}{\partial \mathbf{M_U}[i, j]} \leq \frac{n + k - 2}{n^2}$$

When $u = i$ and $v = j$,

$$(1 - o(1))\frac{2n + k - 2}{n^2} \leq \frac{\partial \mathbf{M_U}^k[u, v]}{\partial \mathbf{M_U}[i, j]} \leq \frac{2n + k - 2}{n^2}$$ ◀

The following lemma generalizes Lemma 16.

▶ **Lemma 45.** *Let $\gamma > 2$ and $\beta \geq \gamma + 2$, and the values of elements of $\mathbf{M_U}$ range from $[\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]$. For every integer $3 \leq k \leq n$, for every element $[i, j]$ of $\mathbf{M}$ and every element $[u, v]$ of $\mathbf{M}^k$, we have the following bound of $\frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[i,j]}$:*

$$\frac{\partial \mathbf{M}^k[u, v]}{\partial \mathbf{M}[i, j]} = \begin{cases} \Theta(k), & u = i \text{ and } v = j \\ \Theta(\frac{k^2}{n^{1+\beta}}), & u = i \text{ or } v = j, \text{ but not both} \\ \Theta(\frac{k^3}{n^{1+2\beta}}), & \text{otherwise} \end{cases}.$$

Proof. We generalize Lemma 16 to every pair of indices $[u, v]$ and $[i, j]$. Recall that

$$\frac{\partial \mathbf{M}^k[u, v]}{\partial \mathbf{M}[i, j]} = \sum_{t=0}^{k} \binom{k}{t} \left(\frac{n^2 - 1}{n^2}\right)^{k-t} \left(\frac{1}{n^\beta}\right)^{t-1} \frac{\partial \mathbf{M_U}^t[u, v]}{\partial \mathbf{M_U}[i, j]}$$

which is dominated by $t = 1$ when $u = i$, $v = j$; dominated by $t = 2$ when $u = i$ or $v = j$, but not both; and dominated by the term of $t = 3$ otherwise.

Combined with Lemma 44, we have the following bounds. When $u = i, v = j$,

$$\Omega(k) \leq \frac{\partial \mathbf{M}^k[u, v]}{\partial \mathbf{M}[i, j]} \leq k$$

When $u = i$ or $v = j$, but not both,

$$\Omega\left(\frac{k^2}{n^{1+\beta}}\right) \leq \frac{\partial \mathbf{M}^k[u, v]}{\partial \mathbf{M}[i, j]} \leq \frac{k^2(n + k - 2)}{n^{2+\beta}}$$

When $u \neq i$ and $v \neq j$,

$$\Omega\left(\frac{k^3}{n^{1+2\beta}}\right) \leq \frac{\partial \mathbf{M}^k[u, v]}{\partial \mathbf{M}[i, j]} \leq \frac{k^3(2n + k - 2)}{n^{2+2\beta}}$$ ◀

Now, we are ready to prove Lemma 13.

Proof to Lemma 13. Recall that we are going to show that the following happens with high probability: for every $a \in \mathbb{N}$ and every $u, v \in [n]$, $|\mathbf{M}^k[u, v] - a/2n^{\alpha'}| > 1/n^\alpha$. We bound its fail probability for every fixed u, v. The desired bound follows by a union bound.

Let σ denote a partial assignment to all the other elements of $\mathbf{M_U}$ except $\mathbf{M_U}[u, v], \mathbf{M_U}[v, u]$. And we let $\phi_{k,\sigma} : \mathbb{R} \to \mathbb{R}$ a function that maps $\mathbf{M_U}[u, v] = \mathbf{M_U}[v, u]$ to $\mathbf{M}^k[u, v]$. We have

$$\Pr_{\mathbf{M} \leftarrow \mathcal{M}}[\exists a \in \mathbb{N}, |\mathbf{M}^k[u, v] - a/2n^{\alpha'}| \leq 1/n^\alpha] \leq \max_\sigma \Pr_{x \in [\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]}[\exists a \in \mathbb{N}, |\phi_{k,\sigma}(x) - a/2n^{\alpha'}| \leq 1/n^\alpha].$$

There are two cases. When $u = v$,

$$\frac{\partial \phi_{k,\sigma}}{\partial x} = \frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}_{\mathbf{U}}[u,v]} = \frac{1}{n^\beta} \cdot \frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[u,v]} = \Theta(k/n^\beta).$$

When $u \neq v$,

$$\frac{\partial \phi_{k,\sigma}}{\partial x} = \frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}_{\mathbf{U}}[u,v]} + \frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}_{\mathbf{U}}[v,u]} = \frac{1}{n^\beta} \cdot \left(\frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[u,v]} + \frac{\partial \mathbf{M}^k[u,v]}{\partial \mathbf{M}[v,u]} \right) = \Theta(k/n^\beta).$$

Let us call $D(a/2n^{\alpha'}, 1/n^\alpha)$ a disc, which is the set of real numbers that is within $1/n^\alpha$ distance from $a/2n^{\alpha'}$. We also need to lower bound the range length of $\mathbf{M}^k[u,v]$, which gives the lower bound of how many discs centered at $a/2n^{\alpha'}$ that f will meet. Since the range length of $\mathbf{M}_{\mathbf{U}}[u,v]$ is $1/n^\gamma$, the range length of $\mathbf{M}^k[u,v]$ is $\Theta(k/n^{\gamma+\beta})$. Therefore, there will be $\Theta(kn^{\alpha'-\gamma-\beta})$ discs.

For each disc, the range length of $\mathbf{M}_{\mathbf{U}}[u,v]$ such that $\phi_{k,\sigma}$ evaluates to this disc is at most $\Theta(n^\beta/kn^\alpha)$. Therefore, the total volume of $\mathbf{M}_{\mathbf{U}}[u,v]$ such that $\phi_{k,\sigma}$ falls in at least one discs is at most $\Theta(n^{\alpha'-\gamma-\alpha})$. Therefore, by union bound,

$$\begin{aligned}
&\Pr_{\mathbf{M} \leftarrow \mathcal{M}}[\exists u, v \in [n], \exists a \in \mathbb{N}, |\mathbf{M}^k[u,v] - a/2n^{\alpha'}| \leq 1/n^\alpha] \\
\leq & n^2 \max_\sigma \Pr_{x \in [\frac{1}{n} - \frac{1}{n^\gamma}, \frac{1}{n}]}[\exists a \in \mathbb{N}, |\phi_{k,\sigma}(x) - a/2n^{\alpha'}| \leq 1/n^\alpha] \\
\leq & n^2 \cdot \Theta(n^{\alpha'-\gamma-\alpha})/(1/n^\gamma) \\
= & \Theta(n^{\alpha'-\alpha+2})
\end{aligned}$$

which is $o(1)$ when $\alpha > 2 + \alpha'$. ◀

B Missing proof to Lemma 21

We prove Lemma 21 in this section, which is a direct application of the concentration bound on intermediate singular values given in [34, 42]. Below are definitions used in presenting the concentration bound.

▶ **Definition 46.** *Let Z be a random variable. Then the ψ_2-norm (subgaussian norm) of Z is defined as*

$$\|Z\|_{\psi_2} := \inf \left\{ \lambda > 0 : \mathbb{E}\left[\exp\left(\frac{|Z|^2}{\lambda^2} \right) \right] \leq 2 \right\}$$

▶ **Lemma 47** (Corollary 1.11. of [42])**.** *Let $\mathbf{A}$ an $m \times m$ random matrix with i.i.d. entries that have mean 0, variance 1 and ψ_2-norm K. Assume also that there exist a constant $c(K) > 0$ that depends only on K, and positive numbers $p > 0$, $s \leq c(K)\min(1/p, 1)$ such that for every $i, j \in [m]$,*

$$\sup_{u \in \mathbb{C}} \Pr[|\mathbf{A}[i,j] - u| \leq s] \leq ps. \tag{7}$$

Then there exist $0 < C_1 < C_2$ and $C_3 > 0$ that depend only on K and p such that for all $\ell \in [m]$,

$$\Pr\left[\frac{C_1 \ell}{\sqrt{m}} \leq \sigma_{m+1-\ell}(\mathbf{A}) \leq \frac{C_2 \ell}{\sqrt{m}} \right] \geq 1 - \exp(-C_3 \ell).$$

Lemma 21 follows by showing that both p and K are constants in our distribution. We also use the following inequality.

▶ **Proposition 48** (Young's convolution inequality [4]). *Suppose f is in the Lebesgue space $L^p(\mathbb{R})$ and g is in $L^q(\mathbb{R})$ and $1/p + 1/q = 1/r + 1$ with $1 \le p, q, r \le \infty$. Then*

$$\|f * g\|_r \le \|f\|_p \|g\|_q$$

where $(f*g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$ is the convolution of f and g.

Proof to Lemma 21. Recall that μ is either μ_1 or μ_2, the distributions of $X_1+Y_1+Z_1+W_1$ or $X_2Y_2 + Z_2W_2$, where X_i, Y_i, Z_i, W_i are i.i.d. uniform on some symmetric intervals chosen so that the resulting distribution has variance 1. Concretely,

$$X_1, Y_1, Z_1, W_1 \leftarrow \mathrm{Unif}\left[-\frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2}\right] \qquad X_2, Y_2, Z_2, W_2 \leftarrow \mathrm{Unif}\left[-\left(\frac{9}{2}\right)^{1/4}, \left(\frac{9}{2}\right)^{1/4}\right]$$

We will apply Lemma 47 to $\mathbf{A}$ and show that C_1, C_2, C_3 are constants, i.e., in both cases $\mu = \mu_1$ or $\mu = \mu_2$, p, K are constants.

To show that the ψ_2-norm K of $\mathbf{A}[i,j]$ is a constant, note that μ is supported on a constant-bounded interval in both cases. $K = \Theta(1)$ follows directly from the fact that bounded random variables are subgaussian [40].

What remains is to show that there exists p such that for a possibly small constant $c(K)$, inequality (7) holds. To that end, we give upper bounds to the probability density functions (PDF) f_{μ_1} and f_{μ_2}. Specifically, $\|f_\mu\|_\infty \le M$ for an absolute constant M implies that

$$\sup_{u\in\mathbb{R}} \Pr[|\mathbf{A}[i,j] - u| \le s] = \sup_{u\in\mathbb{R}} \int_{u-s}^{u+s} f_\mu(t)dt \le 2s\|f_\mu\|_\infty \le (2M)s \tag{8}$$

where it suffices to take $u \in \mathbb{R}$ since $\mathbf{A}[i,j]$ is real-valued. Thus (7) holds with $p := 2M$.

Case $\mu = \mu_1$. Let $a := \sqrt{3}/2$. The uniform density on $[-a, a]$ is $f_0(t) = \frac{1}{2a}\mathbf{1}_{\{|t|\le a\}}$, hence $\|f_0\|_\infty = \frac{1}{2a} = \frac{1}{\sqrt{3}}$. The density of $X_1 + Y_1 + Z_1 + W_1$ is the 4-fold convolution $f_{\mu_1} = f_0 * f_0 * f_0 * f_0$. Note that $\|f_0\|_1 = 1$ since f_0 is a PDF. By repeatedly applying Young's convolution inequality $\|f*g\|_\infty \le \|f\|_\infty \cdot \|g\|_1$,

$$\|f_{\mu_1}\|_\infty = \|(f_0 * f_0 * f_0) * f_0\|_\infty \le \|f_0 * f_0 * f_0\|_\infty \cdot \|f_0\|_1 \le \|f_0\|_\infty \cdot 1 = \frac{1}{\sqrt{3}}.$$

Case $\mu = \mu_2$. Let $b := \left(\frac{9}{2}\right)^{1/4}$ and define $U := X_2Y_2$. Let $f_{X,Y}(x,y) = \frac{1}{4b^2}$ be the joint PDF of X_2, Y_2, for $|x|, |y| \le b$.

To obtain a clean formula of the PDF $f_U(u)$, we use a change of variables. Let $(U, V) = (X_2Y_2, X_2)$. Then $x = v, y = u/v$. Notice the following Jacobian

$$\left|\frac{\partial(x,y)}{\partial(u,v)}\right| = \begin{vmatrix} \partial x/\partial u & \partial x/\partial v \\ \partial y/\partial u & \partial y/\partial v \end{vmatrix} = \begin{vmatrix} 0 & 1 \\ 1/v & -u/v^2 \end{vmatrix} = \frac{1}{|v|},$$

which implies that $f_{U,V}(u,v) = f_{X,Y}(x,y) \cdot \left|\frac{\partial(x,y)}{\partial(u,v)}\right| = f_{X,Y}(v, \frac{u}{v}) \cdot \frac{1}{|v|}$. Hence,

$$\begin{aligned} f_U(u) &= \int_{|v|\le b,\ |u/v|\le b} f_{U,V}(u,v)dv = \int_{|v|\le b,\ |u/v|\le b} f_{X,Y}\left(v, \frac{u}{v}\right)\frac{1}{|v|}dv \\ &= \int_{|v|\in[|u|/b, b]} \frac{1}{4b^2}\frac{1}{|v|}dv = \frac{1}{2b^2}\ln\frac{b^2}{|u|} \end{aligned}$$

for $0 < |u| \leq b^2$, and

$$\|f_U\|_2^2 = \int_{-b^2}^{b^2} |f_U(u)|^2 du = \frac{1}{2b^4} \int_0^{b^2} \ln^2\left(\frac{b^2}{u}\right) du = \frac{1}{2b^2} \int_0^{\infty} t^2 e^{-t} dt = \frac{1}{b^2}$$

where we substitute $u = b^2 e^{-t}$ and use the fact that $\Gamma(3) = \int_0^{\infty} t^2 e^{-t} dt = 2$.

Let $U' := Z_2 W_2$. Note that U and U' are i.i.d. By applying Young's convolution inequality, we have

$$\|f_{\mu_2}\|_\infty = \|f_U * f_{U'}\|_\infty \leq \|f_U\|_2 \cdot \|f_{U'}\|_2 = \frac{1}{b^2} = \frac{\sqrt{2}}{3}$$

In both cases, $M := \frac{\sqrt{3}}{3} \geq \|f_\mu\|_\infty$. By (8), we have

$$\sup_{u \in \mathbb{R}} \Pr[|\mathbf{A}[i,j] - u| \leq s] \leq ps$$

for $p = \frac{2\sqrt{3}}{3}$ and $s \leq c(K)/p$. By Lemma 47, C_1, C_2, C_3 are constants. ◀

Randomized Separations in Black-Box TFNP

Fedor Kiselev
MIPT, Moscow, Russia

Abstract

We study the relationship between deterministic and randomized black-box reducibility between problems in TFNP. Our main contribution is a general technique that establishes equivalence between these reducibility types from specific TFNP problems to any TFNP problem. In particular, we show that this equivalence holds for reductions from complete problems in PPP, PPAD, PPA, and t-PPP. In turn, it strengthens all known black-box separations, originating from these classes, to randomized separations.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Complexity classes

Keywords and phrases TFNP, Pigeonhole Principle

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.42

1 Introduction

TFNP is the class of search problems such that for any input there is at least one correct solution, and the correctness of solutions can be checked in polynomial time. It contains many important problems, such as integer factorization and finding a Nash equilibrium. Being a semantic class, TFNP is conjectured to have no complete problems [15]. Consequently, most of the research is focused on studying subclasses of TFNP, defined by complete problems in them. The most well-studied of these classes are PLS [10], PPP, PPA, PPAD, and PPADS [15].

It is straightforward to show that PPAD $\subset$ PPADS $\subset$ PPP and PPAD $\subset$ PPA [15], and it is conjectured that there are no other inclusions between them. However, proving such separations unconditionally would imply P $\neq$ NP. Instead, we can look at black-box analogs of these classes. Since all known inclusions between TFNP classes also hold for their black-box analogs, separations between black-box classes rule out known techniques for showing inclusion.

After years of progress, for every pair of the above mentioned classes, it has been shown that either the inclusion holds unconditionally or it does not hold in the black-box setting [1, 14, 2, 6]. Subsequently, this result was extended to all possible intersections of these classes [7, 13]. Naturally, a question arises whether these black-box separations could be further strengthened. One possible way to do this is to show that these separations persist even if we consider randomized reductions between complete problems in classes instead of deterministic ones. Such stronger separations were achieved for some pairs of classes: in [12] it was shown that in the black-box setting there are no randomized reductions from any of the above mentioned TFNP subclasses to PWPP (introduced in [9]), and in [8] it was shown that there is no randomized reduction from PPP to the newly introduced class n-PWPP. In addition, for most of the studied TFNP classes, it is easy to show that there is no randomized reduction from them to FP. A notable exception is the problem Lossy, introduced in [11], which admits a randomized reduction to FP but no deterministic one. It demonstrates that these two relations do not always coincide. Overall, randomized reducibility in the black-box TFNP remains largely unexplored.


© Fedor Kiselev;
licensed under Creative Commons License CC-BY 4.0
41st Computational Complexity Conference (CCC 2026).
Editor: Dana Moshkovitz; Article No. 42; pp. 42:1–42:17

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


1.1 Our Results

We investigate the conditions under which it is possible in the black-box model to strengthen a deterministic separation to a randomized one. Our main result, stated in Theorem 13, shows that for certain well-structured problems, if there is no reduction to some $S \in \mathsf{TFNP}$, there is also no randomized reduction to S. Notably, S is an arbitrary problem in TFNP with no additional restrictions. The proof technique applies to highly symmetric problems. In addition, it requires that, under particular restrictions, the problem can be reduced to a smaller instance of the same problem. Many central TFNP problems possess these required structural properties.

We then demonstrate that all the requirements of Theorem 13 are met for certain complete problems in classes $\mathsf{PPP}, \mathsf{PPA}, \mathsf{PPAD}$ [1], and t-PPP [8]. In particular, it automatically strengthens all known separations originating from them to randomized separations.

1.2 Connection to Proof Complexity

Black-box TFNP has a very tight connection with proof complexity. For many TFNP subclasses it has been shown that they are characterized by some proof system [6]. That is, problem $S \in \mathsf{TFNP}$ is in the subclass if and only if the totality of S has an easy proof in the proof system corresponding to that subclass. Moreover, in [4] it was shown that this connection can be extended to randomized reductions, using randomized versions of proof systems. More specifically, problem $S \in \mathsf{TFNP}$ is randomized-reducible to a complete problem in some subclass if and only if the totality of S has an easy proof in the randomized version of the corresponding proof system. This connection can be used to translate our results to proof complexity. Let us call a proof system well-behaved if it has corresponding proof system (see more on that in [3]). Then for any problem S such that it meets the conditions of Theorem 13, if the totality of S is hard to prove in some well-behaved proof system, then it is also hard to prove in its randomized version.

Finally, it should be mentioned that the result, similar to ours, was achieved in [16], where it was essentially shown that PPP is not randomized-reducible to PLS.

1.3 Paper Organization

The remainder of the paper is structured as follows. In Section 2 we introduce most of the required definitions. In Section 3 we prove our main theorem, as well as a few auxiliary results. In Section 4 we define classes $\mathsf{PPP}, \mathsf{PPAD}, \mathsf{PPA}$ and t-PPP and demonstrate that the main theorem is applicable to them. In Section 5 we briefly discuss limitations of the main theorem and future directions.

2 Preliminaries

2.1 Black-Box TFNP

Black-box classes are usually distinguished from their white-box analogs by using superscript "dt", which stands for "decision tree". Since our work is dedicated to black-box classes only, we will omit this superscript.

▶ **Definition 1.** *A search problem is a sequence $R = \{R_N\}_{N \in \mathbb{N}}$, where $R_N \subset \Sigma_N^{l_N} \times O_N$ for some finite alphabet Σ_N, a finite set of solutions O_N, and input length l_N. Any $x \in \Sigma_N^{l_N}$ is called the input of R_N. A solution o is correct for an input x if $(x, o) \in R_N$. A search problem is considered total if, for any input, there is a correct solution. Search problem R belongs to* TFNP *if it is total and there is a family of decision trees $T = \{T_{N,o}\}_{N \in \mathbb{N}, o \in O_N}$ of depth* $\mathrm{poly}(\log N)$ *such that $(x, o) \in R_N \Leftrightarrow T_{N,o}(x) = 1$.*

Note that this definition is more general than the commonly used one, where $\Sigma_N = \{0,1\}$ and $l_N = N$. We believe that this generalization makes reasoning easier since it allows us to omit the binary encoding of input. All problems defined in Section 4 can be transformed into equivalent problems with $\Sigma_N = \{0,1\}$ and $l_N = N$.

We will sometimes abuse the notation by calling R_N a search problem. Also, if there are several search problems in the context, we will distinguish related objects by subscript letters.

▶ **Definition 2.** *Given search problems R_N and S_M, a pseudo-reduction from R_N to S_M is a pair $(f = \{f_i\}_{i\in[l_M]}, g = \{g_o\}_{o\in O_M})$ of families of decision trees over inputs of R_N. Each f_i defines a mapping $\Sigma_N^{l_N} \to \Sigma_M$; each g_o defines a mapping $\Sigma_N^{l_N} \to O_N$. We say that the pseudo-reduction is successful on input x of R_N if:*

$$\forall o \in O_M : [(x, g_o(x)) \in R_N \Leftarrow (f(x), o) \in S_M],$$

where $f(x) = f_1(x)f_2(x)\dots f_{l_M}(x)$. The depth d of the pseudo-reduction is the maximum depth of decision trees in f and g. The complexity of the pseudo-reduction is $\max(d, \log M)$. *A reduction from R to S is a sequence of pseudo-reductions from R_N to S_M for $M = M(N)$, such that N-th pseudo-reduction is successful on all inputs of R_N and has complexity* $\mathrm{poly}(\log N)$.

Note that pseudo-reduction is not an established term. We introduce it to explicitly distinguish the cases where we have no guarantee that (f, g) has small complexity and is successful on all inputs.

▶ **Definition 3.** *Given search problems R_N and S_M, a randomized pseudo-reduction from R_N to S_M is a distribution D over some finite set of pseudo-reductions from R_N to S_M. The success rate on input x is:*

$$\mathbb{P}_{(f,g)\sim D}[\forall o \in O_M : (x, g_o(x)) \in R_N \Leftarrow (f(x), o) \in S_M].$$

The depth of randomized pseudo-reduction is equal to the maximum depth of pseudo-reductions in D. Complexity is defined in the same way. A randomized reduction from R to S is a sequence of randomized pseudo-reductions from R_N to S_M for $M = M(N)$, such that N-th randomized pseudo-reduction has a success rate of at least $\frac{1}{\mathrm{poly}(\log N)}$ and complexity $\mathrm{poly}(\log N)$.

This definition can be seen as one-round communication with an adversary, who, given an arbitrary x and sampled (f, g), tries to provide worst possible solution for $f(x)$. Note that the adversary knows which (f, g) was sampled, which makes it similar to public coin communication. As far as we know, this definition of randomized reduction between TFNP problems is most common, so we are going to use it. Nonetheless, it would be interesting to see if this result holds for private coin definition.

Also note that some definitions of randomized reductions require a higher success rate threshold. We have chosen one with a weaker requirements. Since we are going to show that there is no randomized reduction, our result will also work for more demanding definitions.

We are interested in proving that there is no randomized reduction. For this purpose, it is more convenient to use an equivalence, which follows from Yao's minimax principle [17].

▶ **Lemma 4.** *There is a randomized reduction from a search problem R to a search problem S if and only if there are $p(N) = \mathrm{poly}(\log N)$ and $M = M(N)$ such that for any N and distribution $\mathcal{X}$ over input of R_N there is a pseudo-reduction (f, g) from R_N to S_M with complexity at most $p(N)$ and success rate at least $\frac{1}{p(N)}$.*

Finally, sometimes we limit our search problem R to some family of inputs $X = \{X_N \subset \Sigma_N^{l_N}\}_{N\in\mathbb{N}}$. We denote it as (R, X). In that case, for any pseudo-reduction (f, g) from (R_N, X_N) to (S_M, Y_M) we require that $x \in X \Rightarrow f(x) \in Y$; also, for a reduction, we require success only on $x \in X_N$.

2.2 Partial Assignment

▶ **Definition 5.** *A partial assignment on inputs of the search problem R_N is any element $\rho \in (\Sigma_N \cup \{*\})^{l_N}$. The size of ρ is $|\rho| = |\{i \mid \rho_i \neq *\}|$. We say that ρ_1 extends ρ_2 (denoted $\rho_1 \sqsupset \rho_2$) if they agree in all non-$*$ positions of ρ_2. Partial assignments ρ_1 and ρ_2 are consistent (denoted $\rho_1 \uparrow\uparrow \rho_2$) if there is an input that extends both of them; consistent relative to $X_N \subset \Sigma_N^{l_N}$ ($\rho_1 \overset{X_N}{\uparrow\uparrow} \rho_2$) if such an input exists in X_N.*

Partial assignments represent information we know about an input of a search problem. They are particularly useful when we are working with decision trees. A partial assignment naturally corresponds to every path from the root to a leaf in a decision tree. This partial assignment consists of answers to all queries made on this path of the decision tree.

▶ **Definition 6.** *Given the search problem R_N and a partial assignment ρ, we say that ρ witnesses a solution $o \in O_N$ if $(x, o) \in R_N$ for all inputs x extending ρ. A partial assignment ρ is called witnessing if it witnesses some solution.*

2.3 Properties of Pseudo-Reductions

In this section, we introduce definitions for honest and perceptive pseudo-reductions. To the best of our knowledge, we are the first to introduce these properties. They are useful for us because they simultaneously are easy to enforce and guarantee a nice structure for a set of inputs on which a pseudo-reduction is not successful.

Without loss of generality, we assume that all studied search problems contain a dedicated always incorrect solution $\perp$. This allows a pseudo-reduction to explicitly show that it does not know a correct solution.

▶ **Definition 7.** *We call a pseudo-reduction (f, g) from R_N to S_M honest if, for all $o' \in O_M$ and $x \in \Sigma_N^{l_N}$, either $g_{o'}(x) = \perp$ or the partial assignment corresponding to the computation of $g_{o'}$ on x witnesses the solution $g_{o'}(x)$.*

Intuitively, this property means that whenever the pseudo-reduction is unable to find a correct solution, it says so instead of trying to guess it. Any pseudo-reduction can be transformed into an honest one with an overhead of $\text{poly}(\log N)$ in depth. To achieve this, for every $o' \in O_M$ and for every leaf in $g_{o'}$, we modify $g_{o'}$ so that it additionally checks if the solution in this leaf is correct, and if not, it returns $\perp$ instead. Since $R \in \mathsf{TFNP}$, this check can be made by a decision tree of $\text{poly}(\log N)$ depth. Note that if the pseudo-reduction was successful on an input x, it will remain so.

▶ **Definition 8.** *Let (f, g) be a pseudo-reduction from R_N to S_M, where $R, S \in \mathsf{TFNP}$. Let us fix some family of decision trees $\mathcal{T}_M = \{T_{M,o}\}_{o\in O_M}$ that checks the correctness of solutions for S_M. We call a pseudo-reduction (f, g) perceptive (with respect to $\mathcal{T}_M$) if:*

1. *For any leaf l in g_o, if the corresponding partial assignment is witnessing, then l returns any witnessed solution.*
2. *Let $\overline{T}_{M,o}$ be the decision tree over the inputs of R_N, constructed from $T_{M,o}$ by substituting every query to i-th position of an input of S_M with the tree f_i. We require that g_o contains $\overline{T}_{M,o}$ in the sense that any path in g_o as a partial assignment extends some path in $\overline{T}_{M,o}$.*

This is complementary to the honesty property, which intuitively means that if a pseudo-reduction has enough information to give a correct solution, it does so.

We will usually omit the choice of the family of decision trees that checks the correctness of a solution. All we want from it is to check accurately and to have polylogarithmic depth. With respect to such a family, any pseudo-reduction can be transformed into a perceptive one with at most a polylogarithmic blowup in size. To achieve this, we apply the following operations:

1. We extend every g_o by replacing all leaves with the decision tree $\overline{T}_{M,o}$. New leaves inherit the value of the replaced leaf.
2. In every leaf in the new decision trees, if the corresponding path is witnessing, we replace the leaf's value with any witnessed solution.

Note that if the pseudo-reduction was successful on an input x, it will remain so. Moreover, this operation preserves honesty.

Now we are ready to show why these properties are useful. For a pseudo-reduction of R_N to S_M, we call an input of R_N *bad* if the pseudo-reduction is not successful on it. The following lemma shows how such inputs are distributed for an honest and perceptive pseudo-reduction.

▶ **Lemma 9.** *Let (f, g) be an honest and perceptive pseudo-reduction of R_N to S_M of depth d. Then, for any bad input x, there is a non-witnessing partial assignment ρ such that:*

- $|\rho| \le d$,
- $\rho \sqsubset x$,
- *Any x' such that $\rho \sqsubset x'$ is also bad.*

Proof. Since x is bad, there is $o \in O_M$ such that $(f(x), o) \in S_M$, but $(x, g_o(x)) \notin R_N$. Let ρ be the partial assignment corresponding to the execution of g_o on x. It is clear to see that $|\rho| \le d$ and $\rho \sqsubset x$. Since the pseudo-reduction is perceptive and $(x, g_o(x)) \notin R_N$, ρ must be non-witnessing. Also, since our pseudo-reduction is honest, $g_o(x) = \perp$. Then, for any $x' \sqsupset \rho$, we have $(x', g_o(x')) = (x', \perp) \notin R_N$. Lastly, $(f(x'), o) \in S_M$ since $T_{M,o}(f(x')) = T_{M,o}(f(x)) = 1$ due to perceptiveness. ◀

▶ **Corollary 10.** *We can apply this lemma to every bad input and combine the resulting ρ into one set B. Then B is a set of non-witnessing partial assignments with sizes at most d such that x is bad if and only if it extends some $\rho \in B$.*

2.4 Search Problem Restriction

▶ **Definition 11.** *Given a search problem R_N, a set of inputs X_N, and a partial assignment ρ, a restriction of the pair (R_N, X_N) by ρ (denoted as $(R_N, X_N)\restriction\rho$) is a pair (R'_N, X'_N), where $\Sigma'_N = \Sigma_N$, $l'_N = l_N - |\rho|$, $O'_N = O_N$, $X'_N \subset \Sigma_N^{l'_N}$ is a set of strings such that their substitution in ρ in place of all $*$ gives an element of X_N, $R'_N = \{(x', o) \mid$ substituting x' into ρ gives x such that $(x, o) \in R_N\}$.*

If we have a pseudo-reduction (f, g) from (R_N, X_N) to a search problem S_M, then we can also apply the restriction to (f, g) (denoted as $(f, g)\restriction\rho$) to get a pseudo-reduction from $(R_N, X_N)\restriction\rho$ to S_M. It is achieved simply by resolving in (f, g) all queries that are already specified in ρ. Moreover, the following relation holds:

▶ **Lemma 12.** *Given a pseudo-reduction (f, g) from (R_N, X_N) to a search problem S_M and a partial assignment ρ, if $x' \in X_N\restriction\rho$ is a bad input for pseudo-reduction $(f, g)\restriction\rho$, then $x \in X_N$, obtained by substituting x' into ρ, is bad for pseudo-reduction (f, g).*

Proof. Since x' is bad, there is o such that $((f\restriction\rho)(x'), o) \in S_M$, $(x', (g_o\restriction\rho)(x')) \notin R_N\restriction\rho$. But $(f\restriction\rho)(x') = f(x)$, which means $(f(x), o) \in S_M$; and $(g_o\restriction\rho)(x') = g_o(x)$, which means $(x', g_o(x)) \notin R_N\restriction\rho \Leftrightarrow (x, g_o(x)) \notin R_N$. ◀

3 Main Theorem

▶ **Theorem 13** (Main theorem). *Let $R, S \in$ TFNP, $X = \{X_N \subset \Sigma_N^{l_N}\}_{N\in\mathbb{N}}$. Let the following conditions be met:*

1. *There is no reduction from (R, X) to S.*
2. *For any $p(N) = \mathrm{poly}(\log N)$ and for any sufficiently large N, there are sets $\mathcal{P}_1 = \mathcal{P}_1(N)$ and $\mathcal{P}_2 = \mathcal{P}_2(N)$ of partial assignments over inputs of R_N such that:*
   - **a.** *For any non-witnessing partial assignment ρ over inputs of R_N with $|\rho| \le p(N)$ and such that ρ is extendable to some $x \in X_N$, there is $A_\rho \subset \mathcal{P}_1$ such that $\forall x \in X_N$ $(\rho \sqsubset x \Leftrightarrow \exists \tau \in A_\rho : \tau \sqsubset x)$.*
   - **b.** *For any sequence $\{\kappa_N \in \mathcal{P}_2(N)\}_N$, (R, X) is reducible to $\{(R_N, X_N)\restriction\kappa_N\}_N$.*
   - **c.** *For κ taken uniformly from $\mathcal{P}_2$, and then x taken uniformly from $\{x \in X_N : x \sqsupset \kappa\}$, the resulting distribution of x is uniform on X_N.*
   - **d.** *There is $\delta > 0$ such that for any $T \subset \mathcal{P}_1$:*

$$|\{\kappa \in \mathcal{P}_2 \mid \exists \tau \in T : \kappa \sqsupset \tau\}| \ge (1 - o(N^{-\delta}))|\{\kappa \in \mathcal{P}_2 \mid \exists \tau \in T : \kappa \overset{X_N}{\upuparrows} \tau\}|.$$

Then there is no randomized reduction from (R, X) to S.

Before we proceed with the proof, let us informally explain what sets $\mathcal{P}_1$ and $\mathcal{P}_2$ represent and how this proof works.

The set $\mathcal{P}_1$ consists of "small" partial assignments. Usually we want to choose $\mathcal{P}_1$ so that all its elements are non-witnessing and have the same size and structure. The purpose of this set is to refine the set of partial assignments, resulting from the application of Corollary 10, into the set $T \subset \mathcal{P}_1$ with the same properties. This allows us to work with small partial assignments of the same size and structure, which is convenient.

The set $\mathcal{P}_2$ consists of "big" partial assignments. As with $\mathcal{P}_1$, we usually want to choose $\mathcal{P}_2$ so that all its elements are non-witnessing and have the same size and structure. Moreover, from $\kappa \in \mathcal{P}_2$ we expect that restricting (R_N, X_N) by κ turns it into essentially the same problem, but smaller.

For a quick example of such $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N we refer to Figure 1.

In general, the proof is structured as follows:

- We are given (f, g) with small complexity and we want to show that it performs poorly under the uniform distribution over X_N.
- Assuming (f, g) is honest and perceptive, apply Corollary 10 to obtain the set B of partial assignments. For B holds that $x \in X_N$ is bad iff $\exists \rho \in B : x \sqsupset \rho$.
- Use condition 2a to refine B into $T \subset \mathcal{P}_1$ with the same property as B.
- Show that (for infinitely many N) any $\kappa \in \mathcal{P}_2$ must be consistent (relative to X_N) with some $\tau \in T$. Indeed, otherwise, we can restrict our pseudo-reduction by κ to kill all $\tau \in T$, which means that the resulting pseudo-reduction is always successful. But that would contradict condition 1, since a restriction by κ turns our problem into essentially the same problem, but smaller.
- Use condition 2d to show that almost all $\kappa \in \mathcal{P}_2$ must extend some $\tau \in T$.
- Use condition 2c to show that the same holds for almost all $x \in X_N$, which due to the property of T means that almost all $x \in X_N$ are bad. This concludes the proof.

Proof. Let us fix any $p(N) = \text{poly}(\log N)$. For arbitrary N let (f, g) be any pseudo-reduction of R_N to S_M with complexity $p(N)$. Our goal is to show that this pseudo-reduction performs poorly on x, taken uniformly from X_N. We can assume (f, g) to be honest and perceptive: as shown in Subsection 2.3, these properties can be achieved with polynomial blowup in complexity. Moreover, if the pseudo-reduction was successful on an input, it will remain so. Then, from Corollary 10, there is a set of non-witnessing partial assignments B such that $x \in X_N$ is bad if and only if it extends some $\rho \in B$. We exclude from B all ρ that are not extendable to any $x \in X_N$. Since all $\rho \in B$ have a size at most $p(N)$, we can use condition 2a to construct $T = \bigcup_{\rho \in B} A_\rho \subset \mathcal{P}_1$. By design, for such T we also have that $x \in X_N$ is bad if and only if it extends some $\tau \in T$.

Now, we proceed to show that for infinitely many N, any $\kappa \in \mathcal{P}_2$ is consistent relative to X_N with at least one $\tau \in T$. Assume that there is κ such that it is not consistent relative to X_N with any $\tau \in T$. Then the pseudo-reduction $(f_N, g_N)\restriction\kappa$ from $(R_N, X_N)\restriction\kappa_N$ to S_M is successful on all inputs of $(R_N, X_N)\restriction\kappa_N$. Indeed, if x' is a bad input, from Lemma 12 we have that x, obtained by substituting x' into κ, is also bad. But it is impossible, since a bad input must extend some $\tau \in T$, which would contradict inconsistency of κ. Now, if such κ exists for all sufficiently large N, then $\{(R_N, X_N)\restriction\kappa_N\}_N$ is reducible to S, which together with conditions 2b and 1 leads to a contradiction.

Finally, from condition 2d we can conclude that for some $\delta > 0$ for infinitely many N at least $1 - o(N^{-\delta})$ of all $\kappa \in \mathcal{P}_2$ are not just consistent, but even extend some $\tau \in T$. Since from condition 2c random x can be generated by extending random $\kappa \in \mathcal{P}_2$, at least $1 - o(N^{-\delta})$ of all $x \in X_N$ also extend some $\tau \in T$, which means that they are bad. So, at most $o(N^{-\delta})$ of all inputs are good. But in Lemma 4 for randomized reduction to exist we require a success rate of at least $\frac{1}{\text{poly}(\log N)}$. ◀

Auxiliary Results

The following results are useful for showing that the premise of the main theorem holds. In particular, the following lemma shows that under certain conditions on $R \in \mathsf{TFNP}$ and its inputs X_1 and X_2, for any $S \in \mathsf{TFNP}$, (R, X_1) is reducible to S if and only if (R, X_2) is reducible to S. This is useful to get condition 1 of the theorem.

▶ **Lemma 14.** *Let $R = (R_N)$ be any search problem from TFNP, let X and X' be inputs of R such that $X_N \subset X'_N$ for all N. Assume that for any $p(N) = \text{poly}(\log N)$ for all sufficiently large N for any non-witnessing partial assignment ρ of size at most $p(N)$ it holds that if ρ can be extended to $x' \in X'_N$, it can also be extended to $x \in X_N$. Then, for any $S \in \mathsf{TFNP}$, (R, X) is reducible to S if and only if (R, X') is reducible to S.*

Proof. It suffices to show that if (R, X') is not reducible to S, then (R, X) is also not reducible to S. In addition, we can consider only honest and perceptive pseudo-reductions. Let $p(N) = \text{poly}(\log N)$, let (f, g) be an arbitrary honest and perceptive pseudo-reduction from (R_N, X') to S_M with the complexity at most $p(N)$. For infinitely many N, there is a bad input $x' \in X'_N$ for any such pseudo-reduction. Then we can take the partial assignment $\rho \sqsubset x'$ from Lemma 9. For sufficiently large N, ρ can be extended to $x \in X_N$, which means that x is also bad. That concludes the proof. ◀

If we take $\mathcal{P}_2$ and X_N as in Theorem 13, we can associate with them graph G with $V(G) = \mathcal{P}_2$, and $E(G)$ being pairs of elements from $\mathcal{P}_2$ that are consistent relative to X_N. If $\mathcal{P}_2$ and X_N are very symmetric, this graph has useful properties. These properties are demonstrated in the following lemma:

▶ **Lemma 15.** *Let G be a graph, $V_0 \sqcup \ldots \sqcup V_n \subset V(G)$. Let there be $d_{0\to k}, d_{k\to 0} > 0$ for all $k \in [n]$ such that:*

- $\forall v \in V_0 : |N(v) \cap V_k| = d_{0\to k}$;
- $\forall v \in V_k : |N(v) \cap V_0| = d_{k\to 0}$.

There $N(\cdot)$ is the neighborhood function. Also, let $N(V_0) \subset V_0 \sqcup \ldots \sqcup V_n$. Then for any nonempty $V' \subset V_0$, it holds:

$$\frac{|N(V') \setminus V_0|}{|V'|} \geq \frac{|N(V_0) \setminus V_0|}{|V_0|}$$

Proof.

$$|N(V') \setminus V_0| = \sum_{k\in[n]} |N(V') \cap V_k| \geq \sum_{k\in[n]} |V'| \frac{d_{0\to k}}{d_{k\to 0}} = |V'| \sum_{k\in[n]} \frac{d_{0\to k}}{d_{k\to 0}}$$

$$|N(V_0) \setminus V_0| = \sum_{k\in[n]} |N(V_0) \cap V_k| = \sum_{k\in[n]} |V_0| \frac{d_{0\to k}}{d_{k\to 0}} = |V_0| \sum_{k\in[n]} \frac{d_{0\to k}}{d_{k\to 0}}$$

$$\frac{|N(V') \setminus V_0|}{|V'|} \geq \sum_{k\in[n]} \frac{d_{0\to k}}{d_{k\to 0}} = \frac{|N(V_0) \setminus V_0|}{|V_0|}.$$

◀

For nonempty $T \subset \mathcal{P}_1$ let us look at the ratio between the number of $\kappa \in \mathcal{P}_2$ that are consistent with some $\tau \in T$ relative to X_N and the number of $\kappa \in \mathcal{P}_2$ that extend some $\tau \in T$. The condition 2d of Theorem 13 requires this ratio to be sufficiently close to 1. The following lemma uses the previous result to show that under certain conditions, if we already showed that this ratio is close to 1 for $|T| = 1$, the same holds for arbitrary T.

▶ **Lemma 16.** *Let $X_N \subset \Sigma_N^{l_N}$, let $\mathcal{P}_1$ and $\mathcal{P}_2$ be sets of partial assignments over X_N. Let it be true for some $\gamma > 1$ that for every $\tau \in \mathcal{P}_1$, it holds:*

1. *For every $x \in X_N$ such that $\tau \sqsubset x$, there is $\kappa \in \mathcal{P}_2$ such that $\tau \sqsubset \kappa \sqsubset x$.*
2. $|\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| = \gamma |\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|$
3. *For graph G with $V(G) = \mathcal{P}_2$, $E(G) = \{(\kappa_1, \kappa_2) \mid \kappa_1 \overset{X_N}{\uparrow\uparrow} \kappa_2\}$, $V_0 = \{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}$, $V_k = \{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau$, κ and τ have $|\tau| - k$ common non-$*$ positions$\}$, all conditions of Lemma 15 are met.*

Then, for any $T \subset \mathcal{P}_1$, it holds that:

$$|\{\kappa \in \mathcal{P}_2 \mid \exists \tau \in T : \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| \leq \gamma |\{\kappa \in \mathcal{P}_2 \mid \exists \tau \in T : \kappa \sqsupset \tau\}|.$$

Proof. The first condition gives us the following equivalence:

▷ Claim 17. For any $\tau \in \mathcal{P}_1$ and $\kappa \in \mathcal{P}_2$: $\kappa \overset{X_N}{\uparrow\uparrow} \tau \Leftrightarrow \exists \kappa' \in \mathcal{P}_2 : \kappa' \sqsupset \tau \wedge \kappa \overset{X_N}{\uparrow\uparrow} \kappa'$.

Now, let us prove the claim of the lemma by induction on the size of T. For $|T| = 1$, we already know that the equality holds. Let $T = T_0 \cup \{\tau\}$ and for T_0 the required inequality holds. Define $F(\tau) = \{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}$, $F_w(\tau) = \{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}$, $N(\kappa') = \{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \kappa'\}$. For sets of τ or κ definitions are similar. Note that $F_w = N \circ F$ due to Claim 17. In these terms, we need to prove that:

$$|F_w(T)| \leq \gamma |F(T)|.$$

First, assume the case $F(T_0) \cap F(\tau) = \varnothing$. Then:

$$|F_w(T)| \leq |F_w(T_0)| + |F_w(\tau)| \leq \gamma(|F(T_0)| + |F(\tau)|) = \gamma |F(T)|.$$

Now assume that $F(T_0) \cap F(\tau) = V' \neq \varnothing$, $F(\tau) \setminus V' = V''$. Then we have:

$$
\begin{aligned}
&|F_w(T)| = \\
&|F_w(T_0)| + |F_w(\tau) \setminus F_w(T_0)| \leq && \text{(splitting } F_w(T) \text{ into 2 parts)} \\
&\gamma|F(T_0)| + |F_w(\tau) \setminus F_w(T_0)| = && \text{(applying induction)} \\
&\gamma|F(T_0)| + |F_w(\tau) \setminus N(F(T_0))| \leq && \text{(using that } F_w = N \circ F) \\
&\gamma|F(T_0)| + |F_w(\tau) \setminus N(V')| = && \text{(using that } V' \subset F(T_0)) \\
&\gamma|F(T_0)| + |F_w(\tau)| - |N(V')| = && (F_w(\tau) = N(F(\tau)) \supset N(V')) \\
&\gamma|F(T_0)| + \gamma|F(\tau)| - |N(V')| = && \text{(applying the second condition)} \\
&\gamma|F(T_0)| + \gamma|V''| + \gamma|V'| - |N(V')| = && (F(\tau) = V' \sqcup V'') \\
&\gamma|F(T)| + \gamma|V'| - |N(V')|. && (F(T) = F(T_0) \sqcup V'')
\end{aligned}
$$

Finally, $\gamma|F(T)| + \gamma|V'| - |N(V')| \leq \gamma|F(T)|$, since:

$$
\begin{aligned}
&|N(V')| \geq \\
&|V'| + |N(V') \setminus F(\tau)| \geq && (V' \subset N(V') \text{ and } V' \subset F(\tau)) \\
&|V'| + \frac{|V'|}{|F(\tau)|}|N(F(\tau)) \setminus F(\tau)| = && \text{(using Lemma 15 with } V_0 = F(\tau)) \\
&\frac{|N(F(\tau))|}{|F(\tau)|}|V'| = && (F(\tau) \subset N(F(\tau))) \\
&\gamma|V'|. && \text{(applying the second condition)}
\end{aligned}
$$

◀

4 Applications of the Main Theorem

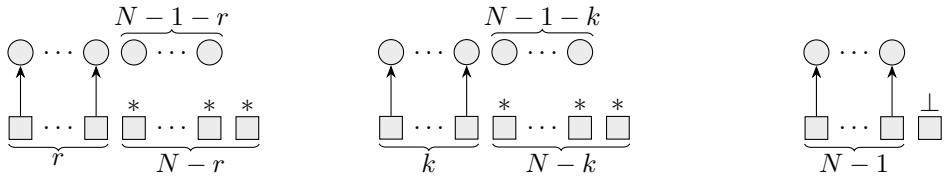
4.1 PPP

▶ **Definition 18.** PPP *is a class of* TFNP *problems that are reducible to the search problem* PIGEON.
$\textsc{Pigeon} = \{\textsc{Pigeon}_N \subset \Sigma_N^{l_N} \times O_N\}$, *where:*
$\Sigma_N = [N-1] \cup \{\perp\}$, $l_N = N$, $O_N = [N] \cup \{(i,j) \in [N]^2 \mid i \neq j\}$.
Positions $i \in [N]$ *are called pigeons, and values* $j \in [N-1]$ *are called holes,* $x_i = j$ *represents pigeon* i *being mapped to hole* j.
Correct solutions for input x*: any* i *such that* $x_i = \perp$ *(homeless pigeon) or* (i,j) *such that* $x_i = x_j$ *(collision).*

Our goal is to show that there is no randomized reduction from PIGEON to $S \in$ TFNP, given that there is no deterministic reduction from PIGEON to $S \in$ TFNP. To apply Theorem 13, we first need to come up with a set of inputs X_N for each N. We take X_N to be the set of all bijections from $[N]$ to $[N-1] \cup \{\perp\}$. Now we can apply Lemma 14 (with X' being all inputs) to show that (PIGEON, X) is also not reducible to S. The conditions of the lemma hold, since any non-witnessing partial assignment contains neither a homeless pigeon nor a collision, which means it can be extended to $x \in X_N$.

Now, given $p(N) = \text{poly}(\log N)$, we need to come up with $\mathcal{P}_1$ and $\mathcal{P}_2$ for every sufficiently large N. As $\mathcal{P}_1$ and $\mathcal{P}_2$ we take sets of all non-witnessing partial assignments of size r and k, respectively (see Figure 1). There $r = p(N)$, $k = N - \lfloor N^\varepsilon \rfloor$ for $\varepsilon = \frac{1}{4}$. Let us go through the sub-items of Theorem 13:

- 2a: For any non-witnessing partial assignment ρ with $|\rho| \le p(N)$ we can take $A_\rho = \{\rho' \in \mathcal{P}_1 \mid \rho' \sqsupset \rho\}$. It is clear that x extends ρ if and only if it extends some $\rho' \in A_\rho$.
- 2b: Note that for every N there is $\hat{N} = O(N^{\frac{1}{\varepsilon}})$ such that $\lfloor \hat{N}^\varepsilon \rfloor = N$. Also, for any $\kappa \in \mathcal{P}_2(\hat{N})$, the search problem $(\textsc{Pigeon}_{\hat{N}}, X_{\hat{N}})\restriction\kappa$ is equivalent to $(\textsc{Pigeon}_N, X_N)$ up to an alphabet renaming, which means there is a trivial always successful pseudo-reduction from $(\textsc{Pigeon}_N, X_N)$ to $(\textsc{Pigeon}_{\hat{N}}, X_{\hat{N}})\restriction\kappa$, with complexity $\log \hat{N} = O(\log N)$. From this follows the required reducibility.
- 2c: By the symmetry of $\mathcal{P}_2$ and X_N.
- 2d: For this one, we can use Lemma 16. Note that due to symmetry the value $\gamma := |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}|/|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|$ does not depend on the choice of $\tau \in \mathcal{P}_1$. Other conditions of the lemma also hold. Finally, simple combinatorial estimates (see Appendix A) show that $\gamma = (1 + O(N^{-\frac{1}{4}})) \Rightarrow \gamma^{-1} = (1 - o(N^{-\frac{1}{5}}))$, which concludes the proof.



(a) The structure of $\tau \in \mathcal{P}_1$. **(b)** The structure of $\kappa \in \mathcal{P}_2$. **(c)** The structure of $x \in X_N$.

Figure 1 The structure of elements in $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N for $\textsc{Pigeon}_N$ up to a permutation of pigeons and holes. Here squares represent pigeons, circles represent holes, and arrows show how pigeons are mapped to holes.

4.2 PPAD and PPADS

▶ **Definition 19.** PPAD *and* PPADS *are classes of* TFNP *problems that are reducible to search problems* InjPigeon *(Injective Pigeon) and* BijPigeon *(Bijective Pigeon), respectively.*
$\textsc{InjPigeon} = \{\textsc{InjPigeon}_N \subset \Sigma_N^{l_N} \times O_N\}$*, where:*
$\Sigma_N = [N] \cup \{\bot\}$, $l_N = 2N - 1$, $O_N = [N]$.
Any $x \in \Sigma_N^{l_N}$ *can be interpreted as a pair* $(s, p) \in \Sigma_N^N \times \Sigma_N^{N-1}$. *We count* $i \in [N]$ *as the correct solution for* $x = (s, p)$ *if either* $s_i \in \{\bot, N\}$ *or* $p_{s_i} \ne i$.
BijPigeon is defined in the same way, except that we also count $i \in [N]$ *as the correct solution for* $x = (s, p)$ *if* $i < N$ *and either* $p_i = \bot$ *or* $s_{p_i} \ne i$.

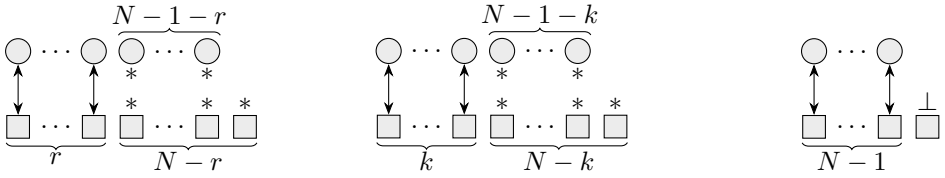
The search problems InjPigeon and BijPigeon are introduced and shown to be complete in the corresponding classes in [5]. For original definitions of PPAD and PPADS, see [15]. These problems are similar to Pigeon: there are N pigeons and $N - 1$ holes. The difference is that there is mapping not only from pigeons to holes but also from holes to pigeons. We say that pigeon i is homeless, if $s_i \in \{N, \bot\}$; hole j is empty if $p_j = \bot$, and pigeon-hole pair is a pair (i, j) such that $s_i = j$, $p_j = i$.

As with Pigeon, our goal is to show the equivalence of deterministic and randomized reducibility from these problems to any $S \in$ TFNP. First, we restrict our attention to consistent inputs: $X'_N = \{x = (s, p) \mid (\forall i \in [N]\ \forall j \in [N-1] : s_i = j \Leftrightarrow p_j = i) \wedge (\forall i \in [N] : s_i \ne N)\}$. It is easy to see that both of these problems are reducible to themselves, restricted to such inputs. From now on, we proceed with each problem separately.

4.2.1 PPAD

First, we can further refine X'_N to X_N so that there are no empty holes: $X_N = \{(s,p) \in X'_N \mid \forall j : p_j \neq \bot\}$. As with PIGEON, Lemma 14 can be applied here: any non-witnessing ρ, extendable to $x' \in X'$, consists only of pigeon-hole pairs, which means it is extendable to $x \in X_N$.

Similarly to PIGEON, as $\mathcal{P}_1$ and $\mathcal{P}_2$ we take sets of all non-witnessing partial assignments of size $2r$ and $2k$, respectively, with the additional requirement on partial assignments: $s_i = j \Leftrightarrow p_j = i$. There $r = p(N)$, $k = N - \lfloor N^\varepsilon \rfloor$ for $\varepsilon = \frac{1}{4}$. Just like for PIGEON, the elements of these sets represent partially defined injections from $[N]$ to $[N-1]$ with domain sizes r and k, respectively. As it can be seen from Figure 1 and Figure 2, $\mathcal{P}_1, \mathcal{P}_2$ and X_N for PIGEON_N and for BIJPIGEON_N look very similar. From now on, the proof almost exactly repeats the proof for PIGEON, including combinatorial calculations.



(a) The structure of $\tau \in \mathcal{P}_1$. **(b)** The structure of $\kappa \in \mathcal{P}_2$. **(c)** The structure of $x \in X_N$.

Figure 2 The structure of elements in $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N for BIJPIGEON_N up to a permutation of pigeons and holes. Here squares represent pigeons, circles represent holes, and arrows show how pigeons and holes are mapped to each other.

4.2.2 Obstacles with PPADS

It may seem that INJPIGEON can be handled in the same way as BIJPIGEON. However, unlike BIJPIGEON, for INJPIGEON, partial assignments with empty holes do not witness an answer. From this it follows that if we want to apply Theorem 13, we require that elements of sets $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N contain empty holes. It seems that this difference is significant, since we were unable to come up with such $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N that all conditions of Theorem 13 hold.

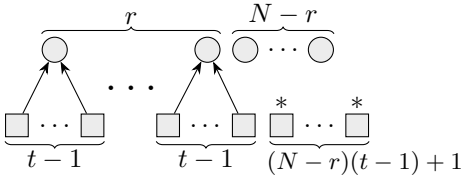
4.3 t-PPP

▶ **Definition 20.** *For nondecreasing $t = t(N)$, $2 \leq t \leq \text{poly}(\log N)$, t-*PPP *is a class of* TFNP *problems that are reducible to the search problem t-$\text{PIGEON} = \{t(N)\text{-PIGEON}_N\}_N$. t-PIGEON_N is defined as follows:*
$\Sigma_N = [N] \cup \{\bot\}$, $l_N = (t-1)N+1$, $O_N = [l_N] \cup \{(i_1, \ldots, i_t) \in [l_N]^t \mid i_s$ are pairwise distinct$\}$. Correct solution for input x: any i such that $x_i = \bot$ or $(i_1, \ldots, i_t)$ such that $x_{i_1} = \ldots = x_{i_t}$.

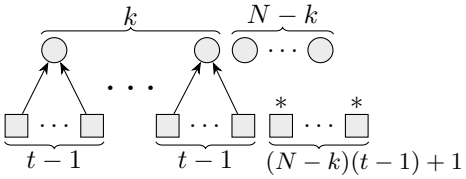
These problems were introduced in [8]. There, it was shown that for $n = \log N$, $t_1(n), t_2(n) = \text{poly}(n)$, if $t_1(n) \leq t_2(n^c)$ and $t_2(n) \leq t_1(n^c)$ for some c and all sufficiently large N, then t_1-PIGEON and t_2-PIGEON are reducible to each other. Using that, we may consider only $t(N) \leq \sqrt{\log N}$. Indeed, since $t(n) \leq n^c$ for some c and sufficiently large N, $\hat{t}(n) = t(\lfloor n^{\frac{1}{2c}} \rfloor) \leq \sqrt{\log N}$, and $\hat{t}(n) \leq t(n) \leq \hat{t}(n^{2c+1})$.

As a hard set of inputs for t-PIGEON_N we take $X_{N,t}$, which is a set of such inputs x that x equals $\bot$ in exactly one position, and for any $a \in [N]$, there are exactly $t-1$ positions of x with that value. For such inputs, the only correct solution is the position of $\bot$. Any

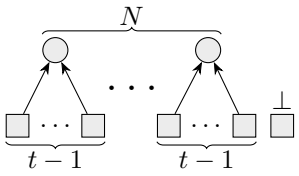
non-witnessing partial assignment can be extended to such x, so we can apply Lemma 14 (with X' being all inputs). Now we are ready to apply the main theorem with $X_N = X_{N,t(N)}$. Given $p(N) = \mathrm{poly}(\log N)$, as $\mathcal{P}_1$ and $\mathcal{P}_2$ we take sets of partial assignments of size $r(t-1)$ and $k(t-1)$, respectively, with the condition that these partial assignments do not contain $\perp$, and every other value occurs either 0 or $t-1$ times (see Figure 3). Here $r = p(N)$, $k = N - \lfloor N^{\varepsilon} \rfloor$ for $\varepsilon = \varepsilon(N) = \frac{1}{4t}$.



(a) The structure of $\tau \in \mathcal{P}_1$.



(b) The structure of $\kappa \in \mathcal{P}_2$.



(c) The structure of $x \in X_N$.

Figure 3 The structure of elements in $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N for t-PIGEON$_N$ up to a permutation of pigeons and holes. Here squares represent pigeons, circles represent holes, and arrows show how pigeons are mapped to holes.

Let us go through the sub-items of Theorem 13:

- 2a: For any non-witnessing partial assignment ρ with $|\rho| \le p(N)$ we can take $A_\rho = \{\rho' \in \mathcal{P}_1 \mid \rho' \sqsupset \rho\}$. It is clear that x extends ρ if and only if it extends some $\rho' \in A_\rho$.
- 2b: Note that for every N there is $\hat{N}$ such that $\lfloor \hat{N}^{\varepsilon(\hat{N})} \rfloor = N$. Moreover:

$$N \ge \hat{N}^{\frac{1}{8t(\hat{N})}} \ge \hat{N}^{\frac{1}{8\sqrt{\log \hat{N}}}} = 2^{\frac{\sqrt{\log \hat{N}}}{8}} \Rightarrow \hat{N} \le 2^{64(\log N)^2} = 2^{O((\log N)^2)}$$

  Also, for any $\kappa \in \mathcal{P}_2(\hat{N})$, the search problem $(t(\hat{N})\text{-PIGEON}_{\hat{N}}, X_{\hat{N},t(\hat{N})})\restriction\kappa$ is equivalent to $(t(\hat{N})\text{-PIGEON}_N, X_{N,t(\hat{N})})$ up to an alphabet renaming. Now, there is an always successful pseudo-reduction from $(t(N)\text{-PIGEON}_N, X_{N,t(N)})$ to $(t(\hat{N})\text{-PIGEON}_N, X_{N,t(\hat{N})})$, achieved by introducing for each of N holes $(t(\hat{N}) - t(N))$ new pigeons, pointing to it. Then, there is an always successful pseudo-reduction from $(t(N)\text{-PIGEON}_N, X_{N,t(N)})$ to $(t(\hat{N})\text{-PIGEON}_{\hat{N}}, X_{\hat{N},t(\hat{N})})\restriction\kappa$ with complexity $O(\log \hat{N}) = O((\log N)^2)$. From this follows the required reducibility.
- 2c: By the symmetry of $\mathcal{P}_2$ and X_N.
- 2d: For this one we can use Lemma 16. Note that due to symmetry the value $\gamma := |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\restriction\restriction} \tau\}|/|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|$ does not depend on choice of $\tau \in \mathcal{P}_1$. Other conditions of lemma also follow from symmetry. Finally, simple combinatorial estimates (see Appendix B) show that $\gamma = (1 + O(N^{-\frac{1}{2}})) \Rightarrow \gamma^{-1} = (1 - o(N^{-\frac{1}{3}}))$, which concludes the proof.

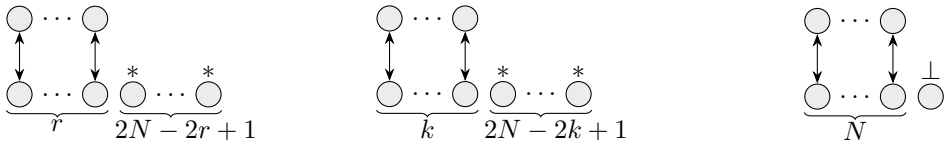
4.4 PPA

▶ **Definition 21.** PPA *is a class of* TFNP *problems that are reducible to the search problem* $\textsc{Lonely} = \{\textsc{Lonely}_N\}_N$. $\textsc{Lonely}_N$ *is defined as follows:*
$\Sigma_N = [2N+1] \cup \{\bot\}$, $l_N = 2N+1$, $O_N = [2N+1]$. *Correct solution for input* x*: any* i *such that* $x_i \in \{i, \bot\}$ *or* $x_{x_i} \neq i$.

First, we restrict our attention to consistent inputs: $X'_N = \{x \mid (\forall i, j \in [2N+1] : x_i = j \Leftrightarrow x_j = i) \wedge (\forall i \in [2N+1] : x_i \neq i)\}$. It is easy to see that PPA is reducible to itself, restricted to such inputs. Now all inputs represent graph, consisting of pairs and lonely nodes. We can further refine X' to $X \subset X'$ such that there is exactly one lonely node. Lemma 14 shows that X is sufficient for checking reducibility from PPA.

Now we move to the second part of Theorem 13. We need to specify $\mathcal{P}_1$ and $\mathcal{P}_2$, given $p(N) = \text{poly}(\log N)$. We take sets of partial assignments, extendable to inputs in X_N, that reveal exactly r and k pairs, respectively (see Figure 4). There $r = p(N)$, $k = N - \lfloor N^\varepsilon \rfloor$ for $\varepsilon = \frac{1}{4}$. Now we are ready to apply Theorem 13. Let us go through the sub-items:

- 2a: For any non-witnessing partial assignment ρ with $|\rho| \leq p(N)$ and which is extendable to $x \in X_N$, we can take $A_\rho = \{\rho' \in \mathcal{P}_1 \mid \rho' \sqsupset \rho\}$. It is clear that x extends ρ if and only if it extends some $\rho' \in A_\rho$.
- 2b: Note that for every N there is $\hat{N} = O(N^{\frac{1}{\varepsilon}})$ such that $\lfloor \hat{N}^\varepsilon \rfloor = N$. Also, for any $\kappa \in \mathcal{P}_2(\hat{N})$, search problem $(\textsc{Lonely}_{\hat{N}}, X_{\hat{N}})\restriction\kappa$ is equivalent to $(\textsc{Lonely}_N, X_N)$ up to renaming the alphabet, which means there is a trivial always successful pseudo-reduction from $(\textsc{Lonely}_N, X_N)$ to $(\textsc{Lonely}_{\hat{N}}, X_{\hat{N}})\restriction\kappa$ with complexity $\log \hat{N} = O(\log N)$. From this follows the required reducibility.
- 2c: By the symmetry of $\mathcal{P}_2$ and X_N.
- 2d: Here we can use Lemma 16. Note that due to symmetry the value $\gamma := |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}|/|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|$ does not depend on choice of $\tau \in \mathcal{P}_1$. Other conditions of lemma also follow from symmetry. Finally, simple combinatorial estimates (see Appendix C) show that $\gamma = (1 + O(N^{-\frac{1}{4}})) \Rightarrow \gamma^{-1} = (1 - o(N^{-\frac{1}{5}}))$, which concludes the proof.



(a) The structure of $\tau \in \mathcal{P}_1$. **(b)** The structure of $\kappa \in \mathcal{P}_2$. **(c)** The structure of $x \in X_N$.

Figure 4 The structure of elements in $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N for $\textsc{Lonely}_N$ up to a permutation of nodes. Here circles represent nodes, and arrows show how they are matched.

5 Future Directions

In addition to all these classes, we also attempted to apply this method to complete problems in classes PLS, PPADS, and PWPP. In most cases, our method fails in combinatorial estimates (condition 2d of Theorem 13). We hope that this method can be generalized to also work for these classes. But it is quite possible that for some of these classes, similarly to Lossy, randomized reducibility from them does not imply deterministic reducibility.

References

1 Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of np search problems. *Journal of Computer and System Sciences*, 57(1):3–19, 1998. doi:10.1006/jcss.1998.1575.

2 J. Buresh-Oppenheim and T. Morioka. Relativized np search problems and propositional proof systems. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 54–67, 2004. doi:10.1109/CCC.2004.1313795.

3 Sam Buss, Noah Fleming, and Russell Impagliazzo. TFNP Characterizations of Proof Systems and Monotone Circuits. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:40, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2023.30.

4 Noah Fleming, Stefan Grosser, Siddhartha Jain, Jiawei Li, Hanlin Ren, Morgan Shirley, and Weiqiang Yuan. Total Search Problems in ZPP. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:26, Dagstuhl, Germany, 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2026.60.

5 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Further collapses in tfnp, 2022. arXiv:2202.07761.

6 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Separations in proof complexity and tfnp. *Journal of the ACM*, 71(4):1–45, August 2024. doi:10.1145/3663758.

7 Pavel Hubáček, Erfan Khaniki, and Neil Thapen. TFNP Intersections Through the Lens of Feasible Disjunction. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:24, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.63.

8 Siddhartha Jain, Jiawei Li, Robert Robere, and Zhiyang Xun. On pigeonhole principles and ramsey in tfnp. *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 406–428, 2024. URL: https://api.semanticscholar.org/CorpusID:267094765.

9 Emil Jeřábek. Integer factoring and modular square roots. *Journal of Computer and System Sciences*, 82(2):380–394, 2016. doi:10.1016/j.jcss.2015.08.001.

10 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988. doi:10.1016/0022-0000(88)90046-3.

11 Oliver Korten. Derandomization from Time-Space Tradeoffs. In Shachar Lovett, editor, *37th Computational Complexity Conference (CCC 2022)*, volume 234 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:26, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2022.37.

12 Jiawei Li. Total NP Search Problems with Abundant Solutions. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 75:1–75:23, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.75.

13 Yuhao Li, William Pires, and Robert Robere. Intersection Classes in TFNP and Proof Complexity. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 74:1–74:22, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.74.

14 Tsuyoshi Morioka. Classification of search problems and their definability in bounded arithmetic. Master's thesis, University of Toronto, 2001. URL: https://www.collectionscanada.ca/obj/s4/f2/dsk3/ftp04/MQ58775.pdf.

15 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

16 Pavel Pudlák and Neil Thapen. Random Resolution Refutations. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:10, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2017.1.

17 Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 222–227, 1977. URL: https://api.semanticscholar.org/CorpusID:143169.

A Estimate on the number of consistent κ for Pigeon

For definitions of $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N refer to Figure 1. There $r = \text{poly}(\log N)$, $k = N - s$, $s = \lfloor N^{\varepsilon} \rfloor$, $\varepsilon = \frac{1}{4}$. For an arbitrary $\tau \in \mathcal{P}_1$, let $a_l = |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau$, κ and τ have $r - l$ common non-$*$ positions$\}$. Let us write an exact formula for this value. To choose such κ, we need to:

1. Choose $r - l$ pigeons that are common with τ.
2. Choose the remaining $k - (r - l)$ pigeons among $N - r$ not used in τ.
3. Choose the remaining $k - (r - l)$ holes among $N - 1 - r$ not used in τ.
4. Choose bijection from $k - (r - l)$ pigeons to $k - (r - l)$ holes.

To put it into combinatorial terms:

$$a_l = \binom{r}{r-l}\binom{N-r}{k-(r-l)}\binom{N-1-r}{k-(r-l)}(k-(r-l))! = \binom{r}{l}\binom{N-r}{s-l}\binom{N-1-r}{s-l-1}(N-s-(r-l))!$$

Now, let us show that a_l gets progressively smaller as l goes from 0 to r:

$$\frac{a_{l+1}}{a_l} = O(r) \cdot O\left(\frac{s}{N}\right) \cdot O\left(\frac{s}{N}\right) \cdot O(N) = o(N^{-\frac{1}{4}}).$$

Note that this estimate is uniform over $l \in \{0, \ldots, r-1\}$. Finally:

$$|\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| = \sum_{l=0}^{r} a_l = a_0 \left(1 + \sum_{l=1}^{r} \frac{a_l}{a_0}\right) = |\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}| \left(1 + o\left(\sum_{l=1}^{r} N^{-\frac{l}{4}}\right)\right) = |\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|(1 + o(N^{-\frac{1}{4}})).$$

B Estimate on the number of consistent κ for t-Pigeon

For definitions of $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N refer to Figure 3. There $r = \text{poly}(\log N)$, $k = N - s$, $s \coloneqq \lfloor N^{\varepsilon} \rfloor$, $\varepsilon = \frac{1}{4t}$, $t \leq \sqrt{\log N}$. For arbitrary $\tau \in \mathcal{P}_1$, let $a_l = |\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau$, κ and τ have exactly $r - l$ common non-$*$ symbols$\}|$. To choose such κ, we need to:

1. Choose $r-l$ "$t-1$ pigeons pointing to one hole" groups that are common with τ.
2. Choose the remaining $k-(r-l)$ holes among $N-r$ not used in τ.
3. Choose the remaining $(t-1)(k-(r-l))$ pigeons among $(t-1)(N-r)+1$ not used in τ.
4. Choose how to map $(t-1)(k-(r-l))$ pigeons to $k-(r-l)$ holes so that at each hole point exactly $t-1$ pigeons.

To put it into combinatorial terms:

$$a_l = \binom{r}{r-l}\binom{N-r}{k-(r-l)}\binom{(t-1)(N-r)+1}{(t-1)(k-(r-l))}\frac{((t-1)(k-(r-l)))!}{((t-1)!)^{k-(r-l)}} = \binom{r}{l}\binom{N-r}{s-l}\binom{(t-1)(N-r)+1}{(t-1)(s-l)+1}\frac{((t-1)(N-s-(r-l)))!}{((t-1)!)^{N-s-(r-l)}}.$$

Now, let us show that a_l gets progressively smaller as l goes from 0 to r:

$$\frac{a_{l+1}}{a_l} = O(r)\cdot O\left(\frac{s}{N}\right)\cdot O\left(\frac{((t-1)s)^{t-1}}{((t-1)N)^{t-1}}\right)\cdot O\left(\frac{((t-1)N)^{t-1}}{(t-1)!}\right) = O\left(\frac{rs^t(t-1)^{t-1}}{N(t-1)!}\right) = O\left(\frac{rs^t}{N}\frac{(t-1)^{t-1}}{\left(\frac{t-1}{e}\right)^{t-1}}\right) = O\left(\frac{rs^te^{t-1}}{N}\right) = o(N^{-\frac{1}{2}}).$$

Note that this estimate is uniform over $l \in \{0,\dots,r-1\}$. Finally:

$$|\{\kappa\in\mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| = \sum_{l=0}^{r} a_l = a_0\left(1+\sum_{l=1}^{r}\frac{a_l}{a_0}\right) = |\{\kappa\in\mathcal{P}_2 \mid \kappa \sqsupset \tau\}|\left(1+o\left(\sum_{l=1}^{r}N^{-\frac{l}{2}}\right)\right) = |\{\kappa\in\mathcal{P}_2 \mid \kappa \sqsupset \tau\}|(1+o(N^{-\frac{1}{2}})).$$

C Estimate on the number of consistent κ for Lonely

For definitions of $\mathcal{P}_1$, $\mathcal{P}_2$ and X_N refer to Figure 4. There $r = \mathrm{poly}(\log N)$, $k = N-s$, $s \coloneqq \lfloor N^{\varepsilon}\rfloor$, $\varepsilon = \frac{1}{4}$. For arbitrary $\tau\in\mathcal{P}_1$, let $a_l = |\{\kappa\in\mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau$, κ and τ have $r-l$ common pairs$\}|$. To choose such κ, we need to:

1. Choose $r-l$ pairs that are common with τ.
2. Choose the remaining $2(k-(r-l))$ nodes among $2N-1-2r$ not used in τ.
3. Choose how to divide those $2(k-(r-l))$ nodes into pairs.

To put it into combinatorial terms:

$$a_l = \binom{r}{r-l}\binom{2N-1-2r}{2(k-(r-l))}(2(k-(r-l))-1)!! = \binom{r}{l}\binom{2N-1-2r}{2s-2l-1}(2(N-s-r+l)-1)!!$$

Now, let us show that a_l gets progressively smaller as l goes from 0 to r:

$$\frac{a_{l+1}}{a_l} = O(r)\cdot O\left(\frac{s^2}{N^2}\right)\cdot O(N) = O\left(\frac{rs^2}{N}\right) = o(N^{-\frac{1}{4}}).$$

Note that this estimate is uniform over $l \in \{0, \dots, r-1\}$. Finally:

$$|\{\kappa \in \mathcal{P}_2 \mid \kappa \overset{X_N}{\uparrow\uparrow} \tau\}| = \sum_{l=0}^{r} a_l = a_0 \left(1 + \sum_{l=1}^{r} \frac{a_l}{a_0}\right) =$$

$$|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}| \left(1 + o\left(\sum_{l=1}^{r} N^{-\frac{l}{4}}\right)\right) =$$

$$|\{\kappa \in \mathcal{P}_2 \mid \kappa \sqsupset \tau\}|(1 + o(N^{-\frac{1}{4}})).$$