18th Conference on the Theory of Quantum Computation, Communication and Cryptography

TQC 2023, July 24–28, 2023, Aveiro, Portugal

Edited by

Omar Fawzi

Michael Walter

Editors

Omar Fawzi
Univ Lyon, Inria, ENS Lyon, UCBL, LIP, Lyon, France
omar.fawzi@ens-lyon.fr

Michael Walter
Ruhr University Bochum, Germany
michael.walter@rub.de

ACM Classification 2012
Theory of computation → Quantum computation theory; Theory of computation → Quantum complexity theory; Theory of computation → Quantum communication complexity; Theory of computation → Quantum query complexity; Theory of computation → Quantum information theory; Hardware → Quantum communication and cryptography; Hardware → Quantum error correction and fault tolerance

ISBN 978-3-95977-283-9

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/978-3-95977-283-9.

Publication date
July, 2023

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

Digital Object Identifier: 10.4230/LIPIcs.TQC.2023.0

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.

ISSN 1868-8969

https://www.dagstuhl.de/lipics

Contents

Papers

18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023).
Editors: Omar Fawzi and Michael Walter
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

The Theory of Quantum Computation, Communication and Cryptography (TQC) conference is a leading annual international conference for students and researchers working in the theoretical aspects of quantum information science. The scientific objective of TQC is to bring together the theoretical quantum information science community to present and discuss the latest advances in the field. The 18th edition of TQC will be hosted by the University of Aveiro in Portugal and held from July 24 to July 28, 2023. A list of the previous editions of TQC follows:

- TQC 2022, University of Illinois at Urbana-Champaign, USA
- TQC 2021, University of Latvia, Latvia (virtual conference)
- TQC 2020, University of Latvia, Latvia (virtual conference)
- TQC 2019, University of Maryland, USA
- TQC 2018, University of Technology Sydney, Australia
- TQC 2017, Université Pierre et Marie Curie, France
- TQC 2016, Freie Universität Berlin, Germany
- TQC 2015, Université libre de Bruxelles, Brussels, Belgium
- TQC 2014, National University of Singapore, Singapore
- TQC 2013, University of Guelph, Canada
- TQC 2012, University of Tokyo, Japan
- TQC 2011, Universidad Complutense de Madrid, Spain
- TQC 2010, University of Leeds, UK
- TQC 2009, Institute for Quantum Computing, University of Waterloo, Canada
- TQC 2008, University of Tokyo, Japan
- TQC 2007, Nara Institute of Science and Technology, Nara, Japan
- TQC 2006, NTT R&D Center, Atsugi, Kanagawa, Japan

We wish to thank the members of the Program Committee and all subreviewers for their work towards composing the program of the conference. We would also like to thank the Local Organizing Committee for all their efforts in organizing the conference, as well as the Steering Committee for maintaining the conference's high standards. Last but not least, we thank the authors of all the TQC 2023 submissions.

May 2023
Omar Fawzi and Michael Walter

18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023).
Editors: Omar Fawzi and Michael Walter
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Conference organization

Local Organizing Committee

- Paulo Almeida, Universidade de Aveiro [host]
- Margarida Facão, Universidade de Aveiro
- Ricardo Guimarães Dias, Universidade de Aveiro
- Alexandre Madeira, Universidade de Aveiro
- Manuel António Martins, Universidade de Aveiro
- Nuriya Nurgalieva, ETH Zurich & Squids
- Armando Pinto, Universidade de Aveiro
- Raquel Pinto, Universidade de Aveiro
- Lídia del Rio, ETH Zurich & Squids [chair, contact person]

Program Committee

- Alvaro Alhambra, Max Planck Institute for Quantum Optics
- Simon Apers, CNRS IRIF
- Stephen Bartlett, The University of Sydney
- Daniel Brod, Fluminense Federal University
- Matthias Caro, California Institute of Technology
- Claude Crépeau, McGill University
- Omar Fawzi, INRIA/ENS de Lyon [chair]
- Sevag Gharibian, University of Paderborn
- David Gosset, University of Waterloo
- Daniel Grier, University of California, San Diego
- Michael Gullans, NIST/University of Maryland
- Yassine Hamoudi, Simons Institute for the Theory of Computing
- Hsin-Yuan Huang, California Institute of Technology
- Martin Kliesch, Hamburg University of Technology
- Tamara Kohle, Complutense University of Madrid
- Ludovico Lami, University of Amsterdam
- Cécilia Lancien, CNRS Institut Fourier Grenoble
- Xiongfeng Ma, Tsinghua University
- Giulio Malavolta, Max Planck Institute for Security and Privacy
- Ashley Montanaro, University of Bristol
- Markus Mueller, IQOQI Vienna
- Anand Natarajan, Massachusetts Institute of Technology
- Pavel Panteleev, Moscow State University
- Simon Perdrix, INRIA LORIA
- Daniel Ranard, Massachusetts Institute of Technology
- Patrick Rebentrost, National University of Singapore
- Joschka Roffe, Free University Berlin
- Jérémie Roland, Université Libre de Bruxelles
- Cambyse Rouzé, Technical University of Munich
- Daniel Stilck França, INRIA Lyon
- David Sutter, IBM Research Zurich

18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023).
Editors: Omar Fawzi and Michael Walter
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- Ryuji Takagi, Nanyang Technological University
- Yu Tong, California Institute of Technology
- Michael Walter, Ruhr University Bochum [co-chair]
- John van de Wetering, University of Oxford
- Takashi Yamakawa, NTT Tokyo
- Leo Zhou, California Institute of Technology

Steering Committee

- Gorjan Alagic, University of Maryland
- Andris Ambainis, University of Latvia
- Eric Chitambar, University of Illinois at Urbana-Champaign
- Steve Flammia, AWS Center for Quantum Computing
- François Le Gall, Nagoya University
- Min-Hsiu Hsieh, Hon Hai (Foxconn) [co-chair]
- Laura Mančinska, University of Copenhagen
- Lídia del Rio, ETH Zurich & Squids
- Marco Tomamichel, National University of Singapore [chair]

List of Authors

Scott Aaronson (12)
The University of Texas at Austin, TX, USA

Andris Ambainis (7)
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

Srinivasan Arunachalam (3)
IBM Quantum, Thomas J Watson Research Center, Yorktown Heights, NY, USA

Roozbeh Bassirian (11)
University of Chicago, IL, USA

Sergey Bravyi (3)
IBM Quantum, Thomas J Watson Research Center, Yorktown Heights, NY, USA

Mark Bun (1)
Department of Computer Science, Boston University, MA, USA

Alper Çakan (4)
Carnegie Mellon University, Pittsburgh, PA, USA

Nolan J. Coble (14)
Joint Center for Quantum Information and Computer Science (QuICS), Department of Computer Science, University of Maryland, College Park, MD, USA

Arjan Cornelissen (13)
QuSoft and University of Amsterdam, The Netherlands

Matthew Coudron (14)
Joint Center for Quantum Information and Computer Science (QuICS), Department of Computer Science, University of Maryland, College Park, MD, USA;
National Institute of Standards and Technology, Gaithersburg, MD, USA

Marcel Dall'Agnol (2)
University of Warwick, Coventry, UK

Vedran Dunjko (13)
applied Quantum algorithms (aQa), Leiden University, The Netherlands

Arkopal Dutt (3)
IBM Quantum, Thomas J Watson Research Center, Yorktown Heights, NY, USA;
MIT-IBM Watson AI Lab, Cambridge, MA, USA;
Department of Physics, Co-Design Center for Quantum Advantage, Massachusetts Institute of Technology, Cambridge, MA, USA

Bill Fefferman (11)
University of Chicago, IL, USA

Vipul Goyal (4)
NTT Research, Sunnyvale, CA, USA;
Carnegie Mellon University, Pittsburgh, PA, USA

Sabee Grewal (12)
The University of Texas at Austin, TX, USA

Ryo Hiromasa (9)
Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Japan

Daniel Hothem (6)
Quantum Algorithms and Applications Collaboratory, Sandial National Laboratories, Livermore, CA, USA

Stacey Jeffery (5)
QuSoft and CWI, Amsterdam, The Netherlands

Sofiene Jerbi (13)
Institute for Theoretical Physics, Universität Innsbruck, Austria

Shelby Kimmel (5)
Middlebury College, VT, USA

Martins Kokainis (7)
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

William Kretschmer (10)
University of Texas at Austin, TX, USA

Brad Lackey (8)
Microsoft Quantum, Redmond, WA, USA

Chen-Da Liu-Zhang (4)
NTT Research, Sunnyvale, CA, USA

Kunal Marwaha (11)
University of Chicago, IL, USA

18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023).
Editors: Omar Fawzi and Michael Walter
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Akihiro Mizutani (9)
Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Japan

Jon Nelson (14)
Joint Center for Quantum Information and Computer Science (QuICS), Department of Computer Science, University of Maryland, College Park, MD, USA

Seyed Sajjad Nezhadi (14)
Joint Center for Quantum Information and Computer Science (QuICS), Department of Computer Science, University of Maryland, College Park, MD, USA

Maris Ozols (13)
QuSoft and University of Amsterdam, The Netherlands

Ojas Parekh (6)
Quantum Algorithms and Applications Collaboratory, Sandial National Laboratories, Albuquerque, NM, USA

Alvaro Piedrafita (5)
QuSoft and CWI, Amsterdam, The Netherlands

João Ribeiro (4)
NOVA LINCS and NOVA School of Science and Technology, Caparica, Portugal

Nishant Rodrigues (8)
Department of Computer Science, University of Maryland, College Park, MD, USA;
Joint Center for Quantum Information and Computer Science, College Park, MD, USA

Nicholas Spooner (2)
University of Warwick, Coventry, UK

Yuki Takeuchi (9)
NTT Communication Science Laboratories, NTT Corporation, Atsugi, Japan

Seiichiro Tani (9)
NTT Communication Science Laboratories, NTT Corporation, Atsugi, Japan;
International Research Frontiers Initiative (IRFI), Tokyo Institute of Technology, Japan

Kevin Thompson (6)
Quantum Algorithms and Applications Collaboratory, Sandial National Laboratories, Albuquerque, NM, USA

Jevgēnijs Vihrovs (7)
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

Nadezhda Voronova (1)
Department of Computer Science, Boston University, MA, USA

Theodore J. Yoder (3)
IBM Quantum, Thomas J Watson Research Center, Yorktown Heights, NY, USA

Approximate Degree Lower Bounds for Oracle Identification Problems

Mark Bun
Department of Computer Science, Boston University, MA, USA

Nadezhda Voronova
Department of Computer Science, Boston University, MA, USA

Abstract

The approximate degree of a Boolean function is the minimum degree of real polynomial that approximates it pointwise. For any Boolean function, its approximate degree serves as a lower bound on its quantum query complexity, and generically lifts to a quantum communication lower bound for a related function.

We introduce a framework for proving approximate degree lower bounds for certain oracle identification problems, where the goal is to recover a hidden binary string $x \in \{0,1\}^n$ given possibly non-standard oracle access to it. Our lower bounds apply to decision versions of these problems, where the goal is to compute the parity of x. We apply our framework to the ordered search and hidden string problems, proving nearly tight approximate degree lower bounds of $\Omega(n/\log^2 n)$ for each. These lower bounds generalize to the weakly unbounded error setting, giving a new quantum query lower bound for the hidden string problem in this regime. Our lower bounds are driven by randomized communication *upper bounds* for the greater-than and equality functions.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic; Theory of computation → Communication complexity; Theory of computation → Quantum complexity theory

Keywords and phrases Approximate degree, quantum query complexity, communication complexity, ordered search, polynomial approximations, polynomial method

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.1

Related Version *Full Version*: `arXiv:2303.03921`

Funding Supported in part by NSF awards CCF-1947889 and CNS-2046425.
Mark Bun: Supported in part by a Sloan Research Fellowship.

Acknowledgements We thank Arkadev Chattopadhyay for suggesting the problem of determining the approximate degree of ordered search, and Arkadev and Justin Thaler for many helpful conversations about it. We also thank the anonymous TQC 2023 reviewers for helpful suggestions on the presentation.

1 Introduction

In an *oracle identification* problem, there is an unknown string $x \in \{0,1\}^n$. A query algorithm is given possibly non-standard oracle access to x, and its goal is to reconstruct x by making a minimal number of queries to this oracle. More specifically, an oracle identification problem is specified by a fixed family of Boolean functions $a_1, \ldots, a_N$. A query algorithm may inspect any value $a_i(x)$ of its choice at the cost of one query, and its goal is to determine x. Many influential problems in the study of quantum algorithms and complexity can be viewed as oracle identification problems, including van Dam's original oracle interrogation problem [44], the Bernstein-Vazirani problem [13], combinatorial group testing [6, 10], symmetric junta learning [10], and more [15, 4, 5, 32, 25, 34]. In this work, we study two such oracle identification problems:

Ordered Search. Consider the following abstraction of the problem of searching an ordered list of $N = 2^n$ elements. Given a list of N bits $a_i \in \{0,1\}$ under the promise that $a_0 \leq a_1 \leq \cdots \leq a_{N-1}$, find the (binary encoding of the) minimum index $x \in \{0,1\}^n$ such that $a_x = 1$. Binary search yields a deterministic algorithm making n queries, and it is not hard to see that this is optimal for randomized algorithms as well. As for quantum algorithms, it turns out that a constant-factor speedup is possible [27, 23, 12], but a lower bound of $\Omega(n)$ holds in this model as well [15, 26, 2, 30, 24]. Ordered search may be viewed as an oracle identification problem where the query algorithm is given oracle access to $a_0 = \mathsf{GT}_0(x), \ldots, a_{N-1} = \mathsf{GT}_{N-1}(x)$, where each "greater-than" function $\mathsf{GT}_i(x)$ evaluates to 1 if $i \geq x$ and to 0 otherwise.

Hidden String. In the hidden string problem, the goal is to reconstruct a hidden string $x \in \{0,1\}^n$ given information about the presence of absence of potential substrings of x. That is, the goal is to determine x given "substring oracle" access, i.e., oracle access to $a_s = \phi_s(x)$ for every binary string s of length at most n, where $\phi_s(x)$ evaluates to 1 iff s is a substring of x. Building on a classical query algorithm of Skiena and Sundaram [43], Cleve et al. [25] gave a $3n/4 + o(n)$ quantum query algorithm for this problem, and proved a nearly matching quantum query lower bound of $\Omega(n/\log^2 n)$.

The state-of-the-art quantum query lower bounds for both problems are proved via the quantum adversary method, which in its modern formulation [28], characterizes the bounded-error quantum query complexity of every function up to a constant factor [38]. The other major technique for proving quantum query lower bounds is the polynomial method [9], which lower bounds the quantum query complexity of a function by lower bounding its *approximate degree*. The approximate degree of a Boolean function is the least degree of a real polynomial that approximates it pointwise to error $1/3$. Since the acceptance probability of a T-query quantum algorithm is a polynomial of degree $2T$, the approximate degree of a function is always at most (half of its) quantum query complexity, but it can be much smaller [3, 1, 42, 18].

In this work, we prove lower bounds of $\Omega(n/\log^2 n)$ on the approximate degree of (decision variants) of the ordered search and hidden string problems. These lower bounds are nearly optimal, as the known quantum (indeed, even classical) query algorithms for these problems automatically yield $O(n)$ upper bounds on their approximate degree. For the ordered search problem, Childs and Lee [24] explicitly posed the question of investigating approximate degree lower bounds to circumvent limitations of the adversary method. Meanwhile, our lower bound on the approximate degree of the hidden string problem implies a quantum query lower bound matching the state-of-the-art [25].

Approximate degree is a fundamental measure of the complexity of Boolean functions that has been the subject of extensive study in its own right (see, e.g., [19] for a recent survey). And while nearly tight quantum query lower bounds for these problems were already known, we see two main quantum motivations for recovering these bounds via approximate degree. First, there are senses in which approximate degree is a more robust lower bound technique than the adversary method. For example, via Sherstov's pattern matrix method [39], any approximate degree lower bound for a Boolean function f can be "lifted" to give the same quantum communication lower bound for a related two-party function F. Such a generic lifting result is not known for any other general quantum query lower bound technique. Moreover, variants of the polynomial method are capable of proving lower bounds against zero-, small-, and unbounded-error quantum algorithms [9, 14], as well as time-space tradeoffs [33]. Indeed, using the polynomial method, we give weakly-unbounded-error quantum query lower bounds for the hidden string problem (see Corollary 2) that significantly improve over the lower bound implied by the adversary method [25].

Second, we believe that our approximate degree lower bounds shed additional light on what makes the ordered search and hidden string problems hard, and may be more transparent in this regard than existing adversary lower bounds. In particular, our lower bounds show that it is not only hard for quantum algorithms to reconstruct the hidden string x, but even to simply compute its parity (a decision problem). The other nearly tight lower bounds for the problems we consider appear to make essential use of the fact that the query algorithm needs to reconstruct all of x, and it isn't clear (at least to us) how to adapt them to hold for their decision variants. We believe that the technique we introduce, or at the very least the "indirect" method we use to prove our lower bounds, will be more broadly useful in understanding the approximate degree and quantum query complexity of other oracle identification problems.

1.1 Techniques

Here we give a brief summary of the ideas behind our lower bound for ordered search. A more detailed technical overview, including a discussion of how we apply our framework to the hidden string problem, appears in Section 2. Full proofs appear in Sections 3 and 4 in the full version of our paper [20].

The first lower bound for quantum ordered search was given by Buhrman and de Wolf [15], who actually showed an $\Omega(\sqrt{n})$ lower bound on its approximate degree. The starting point for the proof of our lower bound is their ingenious indirect argument, so let us review it here. Recall that the ability to solve ordered search on inputs $a_0 \le a_1 \le \cdots \le a_{N-1}$ enables recovering the string $x \in \{0,1\}^n$, where $N = 2^n$, for which every $a_i = \mathsf{GT}_i(x)$. This, in particular, enables the evaluation of any "hard" Boolean function of x, e.g., its parity. In light of this, define the partial Boolean function $\mathsf{OS}_N(a_0, \ldots, a_{N-1}) := \mathsf{parity}(x)$ whenever there exists an x for which $a_i = \mathsf{GT}_i(x)$ for every i. Let $p : \{0,1\}^N \to \mathbb{R}$ be a polynomial of degree d approximating OS_N. It is known that every polynomial approximating parity must have degree $\Omega(n)$, so the goal now is to use this fact to prove a lower bound on the degree of p. To do so, we use the additional fact that the functions GT_i can each be approximated by a degree $O(\sqrt{n})$ polynomial q_i arising from, say, a variant of Grover search. By making p "robust to noise" in its input without increasing its degree [17, 40], we get that the composed polynomial $p(q_0(x), \ldots, q_{N-1}(x)) \approx \mathsf{parity}(x)$ and has degree $O(d\sqrt{n})$. Now the fact that the approximate degree of parity is $\Omega(n)$ implies that $d = \Omega(\sqrt{n})$.

In summary, the lower bound for OS_N follows from the fact that we can express the function $\mathsf{parity}(x) = \mathsf{OS}_N(\mathsf{GT}_0(x), \ldots, \mathsf{GT}_{N-1}(x))$, where we have a lower bound on the approximate degree of parity and an *upper bound* on the approximate degree of GT. However, the lower bound gets stuck at degree $\Omega(\sqrt{n})$ because the functions GT_i themselves require nontrivial degree $O(\sqrt{n})$ to approximate, and this is tight.

To get an improved lower bound of $\widetilde{\Omega}(n)$ on the approximate degree of OS_N, we introduce the following idea to make GT behave as if it were easier to approximate by low degree polynomials, while preserving the hardness of parity. Given an input $x \in \{0,1\}^n$, we redundantly encode x as a longer string $\mathcal{Y}(x) \in \{0,1\}^m$ for some $m = \mathrm{poly}(n)$. This encoding is chosen so that

- Access to $\mathcal{Y}(x)$ instead of just x itself makes each function $\mathsf{GT}_i(x)$ approximable by a much lower degree polynomial. That is, for every i, there exists a polynomial q_i of degree $\mathrm{polylog}(n)$ such that $q_i(\mathcal{Y}(x)) \approx \mathsf{GT}_i(x)$ for every x.
- Even with access to $\mathcal{Y}(x)$, the function $\mathsf{parity}(x)$ remains hard to approximate. That is, for every polynomial p of degree at most $n/\,\mathrm{polylog}(n)$, we have that $p(\mathcal{Y}(x))$ fails to approximate $\mathsf{parity}(x)$.

We can now obtain our improved lower bound by applying Buhrman and de Wolf's argument to the redundantly encoded inputs. Specifically, given a robust polynomial $p : \{0,1\}^N \to \mathbb{R}$ of degree d approximating OS_N, we would have $p(q_0(\mathcal{Y}(x)), \ldots, q_{N-1}(\mathcal{Y}(x))) \approx \mathsf{OS}_N(\mathsf{GT}_0(x), \ldots, \mathsf{GT}_{N-1}(x)) = \mathsf{parity}(x)$ for every x. Our upper bound on the degrees of the q_i's, together with our lower bound on the degree needed to approximate parity, imply that $d \operatorname{polylog} n \geq n/\operatorname{polylog} n$, and hence $d \geq \widetilde{\Omega}(n)$.

All that remains is to construct the appropriate encoding $\mathcal{Y}$. Our approach is inspired by Nisan's classic randomized *communication* protocol for computing the two-party greater-than function. The most helpful way to think about this protocol for our purposes is as follows. Suppose Alice and Bob hold strings $a, b \in \{0,1\}^n$ and their goal is to determine whether the natural number represented by a is at least that represented by b. They may do so by performing binary search to identify the minimum index j for which $a_j \neq b_j$, at which point the answer is determined by which of a_j or b_j is 1. Each step of this binary search can be conducted by testing the equality of a substring of a with a substring of b. Each equality test, in turn, may be performed (with high success probability) by comparing the inner products of a and b with a shared random string. The protocol requires $\log n$ steps of binary search, and each equality test should be repeated $O(\log \log n)$ times to achieve high success probability, giving an overall communication cost of $\widetilde{O}(\log n)$.

Now let us see how to turn this communication protocol into a polynomial approximating $\mathsf{GT}_i(x)$. Think of x as Bob's input to the communication protocol, and of Bob's role as passively computing an encoding $\mathcal{Y}(x)$ that consists of many inner products of x with random strings. Now thinking of i as Alice's input, she can compute $\mathsf{GT}_i(x)$ (with high probability) by repeatedly querying $\mathcal{Y}(x)$ at the locations that correspond to the appropriate inner products from the protocol described above. This results in a $\widetilde{O}(\log n)$ randomized query algorithm for computing $\mathsf{GT}_i(x)$ from $\mathcal{Y}(x)$, the success probability of which is a degree-$\widetilde{O}(\log n)$ polynomial in $\mathcal{Y}(x)$.

The final step is to argue that even given $\mathcal{Y}(x)$, consisting of many inner products of random strings with x, the parity function $\mathsf{parity}(x)$ remains hard to compute. To see why this is true, note that a single inner product of x with a random bit string is itself a parity on a random subset of indices. That is, $\mathcal{Y}(x) = (\mathsf{parity}(x|_{S_1}), \ldots, \mathsf{parity}(x|_{S_m}))$ for random subsets $S_1, \ldots, S_m \subseteq [n]$. The key observation then, is that a degree-d polynomial of these random parities is able to approximate the full $\mathsf{parity}(x)$ if and only if some degree-d polynomial of these random parities *exactly* computes $\mathsf{parity}(x)$, which in turn happens if and only if a symmetric difference of at most d of the sets $S_1, \ldots, S_m$ yields the entire set of indices $[n]$. As a result, as long as neither the degree d nor the number of random inner products m is too large, we obtain that $\mathsf{parity}(x)$ cannot be approximated using $\mathcal{Y}(x)$.[1]

1.2 Our results in detail

Recall that we introduce a framework that allows us to prove lower bounds on approximate degree, and hence quantum query complexity. It most naturally applies to decision versions of oracle identification problems, and extends to the "weakly unbounded error" setting of error approaching 1/2. We summarize the results we prove using this framework in Table 1.

[1] In fact, this argument shows that it is impossible to approximate $\mathsf{parity}(x)$ to bounded error, but even to represent it in sign. This corresponds to a *threshold degree* lower bound.

Table 1 Summary of our results and prior work.

Problem	Model	Error	Previous work	This work
Ordered search	Approximate degree and quantum query complexity, decision version	Unbounded	$O(n - \log \frac{1}{\gamma})$, $\Omega(\sqrt{n} - \log \frac{1}{\gamma})$	$\Omega(\frac{n}{\log^2 n} - \log \frac{1}{\gamma})$
		Constant	$O(n)$, $\Omega(\sqrt{n})$	$\Omega(\frac{n}{\log^2 n})$
	Quantum query complexity, reconstruction version	Unbounded	$\Theta(n - \log \frac{1}{\gamma})$	$\Omega(\frac{n}{\log^2 n} - \log \frac{1}{\gamma})$
		Constant	$\Theta(n)$	$\Omega(\frac{n}{\log^2 n})$
Hidden string	Approximate degree and quantum query complexity, decision version	Unbounded	$O(n - \log \frac{1}{\gamma})$	$\Omega(\frac{n}{\log^2 n} - \log \frac{1}{\gamma})$
		Constant	$O(n)$	$\Omega(\frac{n}{\log^2 n})$
	Quantum query complexity, reconstruction version	Unbounded	$O(n - \log \frac{1}{\gamma})$, $\Omega(\gamma^2 \frac{n}{\log^2 n})$	$\Omega(\frac{n}{\log^2 n} - \log \frac{1}{\gamma})$
		Constant	$O(n)$	$\Omega(\frac{n}{\log^2 n})$

Ordered search. As mentioned, binary search yields a deterministic algorithm making n queries, which in turn yields a polynomial of degree n that exactly computes OS_{2^n}. To compute this function with error probability $\frac{1}{2} - \gamma$ for some parameter $\gamma > 0$, there is an easy way to modify binary search to obtain an $O(n - \log \frac{1}{\gamma})$-query randomized algorithm (see Appendix A in the full version of our paper [20] for details). This implies an upper bound of $O(n - \log \frac{1}{\gamma})$ on the approximate degree of OS_{2^n} with error parameter $1/2 - \gamma$.

Before this work, the best lower bound on approximate degree (for both bounded and unbounded error) was obtained by [15] and was $\Omega(\sqrt{n} - \log \frac{1}{\gamma})$ for approximation to error $\frac{1}{2} - \gamma$. We significantly improve their result and obtain the following lower bound.

▷ Claim. For every natural number n and $0 < \gamma < 1/2$, every polynomial that approximates OS_{2^n} pointwise to error $\frac{1}{2} - \gamma$ requires degree

$$\Omega\left(\frac{n}{\log^2 n} - \log \frac{1}{\gamma}\right).$$

This result is restated as Theorem 12. It shows that it is hard to approximate the decision version of the ordered search problem OS_{2^n} (with parity as the predicate converting from reconstruction to decision problem) not only to constant error, but even to small advantage γ over random guessing. For instance, approximating OS_{2^n} with advantage $\gamma = 2^{-n^{0.99}}$ still requires degree $\Omega(\frac{n}{\log^2 n})$. Our lower bound is nearly tight in both the bounded and unbounded error regimes.

Query complexity of ordered search. Most previous work on the quantum query complexity of ordered search addressed the bounded error regime and the reconstruction version of the problem, where the goal is to output the entire string x, rather than a specific Boolean predicate applied to x. To our knowledge, the best prior lower bound for the decision version of ordered search with unbounded error follows from [15] as described above and is $\Omega(\sqrt{n} - \log \frac{1}{\gamma})$. Note also that the $\Omega(n)$ lower bound of [2], stated there for constant error, also generalizes to a tight lower bound $\Omega(n - \log \frac{1}{\gamma})$ for unbounded error, but it appears to hold only for the reconstruction version of ordered search.

Our application of the polynomial method implies a nearly tight quantum query lower bound that applies to the decision version of the problem.

▶ **Corollary 1.** *Every quantum algorithm that computes* OS_{2^n} *(decision version with* parity*) with probability of error at most* $\frac{1}{2}-\gamma$ *requires* $\Omega(\frac{n}{\log^2 n}-\log\frac{1}{\gamma})$ *queries.*

Hidden string. The work of [43] yields a simple deterministic algorithm making $O(n)$ queries, which in turn yields a polynomial of degree $O(n)$ that exactly computes $\mathsf{HS}_{2^{n+1}-1}(\dots\phi_s(x)\dots) := \mathsf{parity}(x)$ where $x\in\{0,1\}^n$ is the hidden string in question. Again, this algorithm can be modified to get a $O(n-\log\frac{1}{\gamma})$-query algorithm with error $1/2-\gamma$ (see Appendix A in the full version of our paper [20] for details). This implies an upper bound $O(n-\log\frac{1}{\gamma})$ on the approximate degree of $\mathsf{HS}_{2^{n+1}-1}$.

We give the first lower bound on the approximate degree of the hidden string problem:

▷ Claim. For every natural number n and $0<\gamma<1/2$, every polynomial that approximates $\mathsf{HS}_{2^{n+1}-1}$ to error $\frac{1}{2}-\gamma$ requires degree

$$\Omega\left(\frac{n}{\log^2 n}-\log\frac{1}{\gamma}\right).$$

This result is restated as Corollary 21 in the full version of our paper [20] and gives a nearly tight lower bound for approximating the decision version of $\mathsf{HS}_{2^{n+1}-1}$ to both constant and weakly unbounded error.

Query complexity of hidden string. Complementing the $O(n)$-query deterministic algorithm of [43], it turns out that a constant-factor speedup is possible for quantum algorithms [25]. As for lower bounds, the latter work shows a lower bound $\Omega(\frac{n}{\log^2 n})$ on reconstruction by adversary method. This lower bounds holds for bounded error, but does not generalize well to unbounded error regime. (By [8, 29], the same proof implies a lower bound of $\Omega(\gamma^2\frac{n}{\log^2 n})$ for solving the reconstruction version of the hidden string problem with error $\frac{1}{2}-\gamma$.)

Our approximate degree lower bound recovers their lower bound for bounded error, and gives a significantly stronger lower bound for the weakly unbounded error regime, both for the decision version of the problem.

▶ **Corollary 2.** *Every quantum algorithm that computes* $\mathsf{HS}_{2^{n+1}-1}$ *(decision version with* parity*) with probability of error at most* $\frac{1}{2}-\gamma$ *requires* $\Omega(\frac{n}{\log^2 n}-\log\frac{1}{\gamma})$ *queries.*

1.3 Further discussion

One of our initial motivations for studying the approximate degree of ordered search came from the preliminary version of Chattopadhyay et al. [22]. They showed that $\mathsf{OS}_N\circ\mathsf{IP}_m^N$ has randomized communication complexity $\Omega(\log N\cdot m)$, where IP_m is a two-party inner product (mod 2) gadget on m-bit inputs. This was done via an involved simulation argument, showing how a communication protocol for $\mathsf{OS}_N\circ\mathsf{IP}_m^N$ could be used to construct a randomized decision tree for OS_N. The techniques were specialized to the both the outer function and the inner function. Subsequent work [21] recovered this result using a generic simulation theorem. A direct application of Sherstov's pattern matrix method [39] to our result yields a *quantum* communication lower bound of $\Omega(\log N/\log^2\log N)$ on $\mathsf{OS}_N\circ g^N$ even for a constant-sized gadget g.

Hoza [31] used ideas conceptually related to ours to nearly recover the known quantum query (but not approximate degree) lower bound for ordered search. Roughly, he used a Holevo-information argument to show that if an oracle identification problem specified by functions $a_1,\dots,a_N$ can be solved with T quantum queries, then $Q^*(A)\cdot T\gtrsim n$, where

$A(i,x) = a_i(x)$ and Q^* is the bounded-error two-party quantum communication complexity with shared entanglement. His quantum query lower bound for ordered search follows directly from the fact that the quantum communication complexity of the two-party greater-than function GT on n-bit inputs is $O(\log n)$. However, without opening up the communication protocol for GT as we do, it is not clear how to recover an approximate degree lower bound from his construction.

The idea of indirectly proving approximate degree lower bounds by combining a lower bound for one problem with an upper bound for another also appears in [11]. They gave a tight lower bound on the approximate degree of any function of the form $f \circ g^n$ where f is an n-input symmetric function by combining a known lower bound for $\mathsf{parity} \circ g^n$ [41] with a quantum query and approximate degree upper bound for the combinatorial group testing problem [10].

We believe it should be possible to extend our techniques to prove new lower bounds for other oracle identification problems. A family of special cases of oracle identification is captured by the symmetric junta learning problem [6]. Here, there is a symmetric function $h : \{0,1\}^k \to \{0,1\}$ and each f_S takes the form $f_S(x) = h(x|_S)$. An important instance of this problem is the combinatorial group testing problem, wherein one takes $h = \mathsf{OR}_k$. Belovs gave a tight upper bound of $O(\sqrt{k})$ [10] for this problem. He also determined the query complexity for $h = \mathsf{EXACT} - \mathsf{HALF}$ to be $\Theta(k^{1/4})$ and gave an upper bound of $O(k^{1/4})$ for $h = \mathsf{MAJ}$. These upper bounds were also (nearly) recovered algorithmically by Montanaro and Shao [35]. Despite its similarity to $\mathsf{EXACT} - \mathsf{HALF}$, no polynomial lower bound is known for the majority function MAJ.

In the counterfeit coin problem, there is a hidden string $x \in \{-1,1\}^n$ with Hamming weight at most k. A query is parameterized by a balanced (i.e., having an equal number of 1's and -1's) string $y \in \{-1,0,1\}^n$, and indicates whether $\langle x, y\rangle$ is zero or non-zero. Iwama et al. [32] gave a quantum algorithm making $O(k^{1/4})$ queries and conjectured this is tight, but no lower bound is known. Note that the oracle here is quite similar to the $\mathsf{EXACT} - \mathsf{HALF}$ oracle.

2 Technical ideas

2.1 Our lower bound framework

We begin with a somewhat more abstract description of our framework for proving approximate degree lower bounds for oracle identification problems. The main idea is to provide additional information about the hidden input to an oracle identification problem so as to selectively affect the ability of quantum query algorithms and approximating polynomials to compute the functions we wish to understand.

Recall that an oracle identification problem is specified by a family of functions $a_1, \ldots, a_N$. Given query access to the values $a_1(x), \ldots, a_N(x)$, the goal in our decision problems is to compute the function $\mathsf{parity}(x)$. Suppose that we may identify $\mathsf{parity}(x) = f(a_1(x), \ldots, a_N(x))$ for some function f. If we can construct a function $\mathcal{Y}$ such that:

- Given $\mathcal{Y}(x)$, every function $a_i(x)$ can be computed by a low-degree polynomial, but
- Given $\mathcal{Y}(x)$, computing the parity of x requires a high-degree polynomial,

Then by combining these two statements, we see that the function $f(a_1, \ldots, a_N)$ itself requires a high-degree polynomial. We apply this framework taking f to be either the OS function or for the "anchored hidden string" AHS function. The latter also implies a lower bound for the original (decisional) hidden string function HS described in the introduction.

In the following sections, we describe the main technical ideas that go into the proofs of our lower bounds. In order to provide more intuition about the structure of $\mathcal{Y}$, we describe the steps of constructing it for OS in detail before returning to the generalized framework.

2.2 Ordered search lower bound

First, notice that OS_N has the structure of an oracle identification problem since

$$\mathsf{OS}_N(\mathsf{GT}_{0^n}(x), \mathsf{GT}_{0^{n-1}1}(x), \ldots, \mathsf{GT}_{1^n}(x)) = \mathsf{parity}(x)$$

where $N = 2^n$ and $\mathsf{GT}_i(x) = 1$ if and only if $x \leq i$ where $i, x \in \{0,1\}^n$ if compared as numbers written in binary notation.

We want to show that there exists a function $\mathcal{Y}$ of x that we think of as revealing partial information about x such that:

- On one hand, for all $i \in \{0,1\}^n$ there is an algorithm that makes a small number of queries to $\mathcal{Y}$ and can identify the value of $\mathsf{GT}_i(x)$ with constant probability of success. Note that a query-efficient algorithm automatically gives rise to a low-degree approximating polynomial.
- On the other hand, approximating the value of $\mathsf{parity}(x)$ given $\mathcal{Y}$ with any probability of success requires a lot of queries to $\mathcal{Y}$. Let us denote this auxiliary problem by $\mathsf{PUR}(\mathcal{Y}) := \mathsf{parity}(x)$.

It is helpful to think of $\mathcal{Y}$ itself as an oracle, whose output is given to a polynomial or to a query algorithm, whose goal is then to compute some other function of x. We describe how we construct oracle $\mathcal{Y}$ through several attempts.

Let us first focus on constructing an oracle $\mathcal{Y}$ that meets the first condition. To do so, we can use the idea behind the $O(\log n \log\log n)$-bit communication protocol[2] for the two-party communication problem GT to obtain an efficient randomized query algorithm for every function GT_i. In the GT communication problem, Alice and Bob both get a string of n bits and the goal is to decide if the number represented by Alice's string is greater than the number represented by Bob's string.

In this randomized communication protocol for GT, Alice checks if the first halves of the inputs are equal and depending on the answer, she either recursively continues on the first halves of the inputs or the second halves. By doing so, she finds the most significant bit where the inputs differ. To perform each equality check, both Alice and Bob compute the inner products modulo 2 of each of the inputs with the same set of some α (publicly) random strings, Bob sends his values to Alice, and Alice compares these values to the values she obtained. If the original values were equal, then the inner products will be always equal, and otherwise, at least one pair of inner products will be unequal with high probability for sufficiently large α. This elementary operation (i.e., the ability to compute inner products with random strings) will be exactly what we want our oracle $\mathcal{Y}$ to be useful for.

First attempt. We will eventually give a randomized construction of the oracle $\mathcal{Y}$, and to this end, think of it as taking as input both the hidden string x and a random input r. Let $\mathcal{Y}(r,x)$ be a function that takes a collection of m n-bit strings $r \in \times_{i\in[m]}(\{0,1\}^n)$ and $x \in \{0,1\}^n$, and outputs m bits, each representing the inner product of r_i with x: $(\mathcal{Y}(r,x))_i = \langle r_i, x\rangle$.

[2] A more efficient $O(\log n)$-bit communication protocol is known and underlies our sharpest result for ordered search. We discuss it in Sections 2.4 and 3.

Our first attempt, however, will make no use of randomness at all. Let us consider $\mathcal{Y}(r, x)$ where r consists of all possible strings of length n. That is, the output of the oracle consists of $\langle x, r_i \rangle$ for every $r_i \in \{0,1\}^n$.

Let us now see how to construct a query algorithm C_i that, given oracle access to $\mathcal{Y}(r, x)$, computes $\mathsf{GT}_i(x)$ with high probability. This algorithm emulates Alice's side in the communication protocol, fixing her input to i. It samples random strings used in the communication protocol, computes the inner products of i with these random strings on its own, and asks the oracle (emulating Bob) for the inner products of x with the same random strings.

From the correctness of the communication protocol for GT we can conclude that for all $x, i \in \{0,1\}^n$

$$\Pr_{r_1 \dots, r_{\alpha \log n}} [C_i(\mathcal{Y}(r, x)) \neq \mathsf{GT}_i(x)] < \log n \cdot 2^{-\alpha}$$

where $r_1 \dots, r_{\alpha \log n}$ are the strings that C_i sampled during the run, and r is a collection of all n-bit strings. The number of queries is $\alpha \log n$.

Thus we see that this oracle satisfies the first condition: it helps to compute the GT_i efficiently for every i and x. But now there is a problem with the second condition: $\mathsf{parity}(x) = \mathsf{PUR}(\mathcal{Y})$ can be computed easily since $\mathsf{parity}(x) = \mathsf{PUR}(\mathcal{Y}(r, x)) = \langle x, 1^n \rangle$. So there is a 1-query algorithm (and hence a degree-1 polynomial) that exactly computes $\mathsf{PUR}(\mathcal{Y}(r, x))$, violating our second condition.

Second attempt. Our goal now is to reduce the efficacy of the oracle $\mathcal{Y}$ in terms of how well it can be used by low-degree polynomials to approximate PUR. To do this, we instead consider a distribution over the potential oracles defined by the collection of strings used in the protocol. Let r denote a sequence of the random strings that could appear in one run of GT protocol described earlier. Let $\hat{\mathcal{R}}$ denote the set of all such sequences. This allows us to define a distribution of oracles $\mathcal{Y}[\hat{\mathcal{R}}](r, x)$, where $r \leftarrow \hat{\mathcal{R}}$, and for us to consider a deterministic query algorithm. Let $B_{(r,i)}$ be a deterministic algorithm that is given access to the $\mathcal{Y}[\hat{\mathcal{R}}](r, x)$ where $r \leftarrow \hat{\mathcal{R}}$ is chosen uniformly at random, and which has the realization of r and i hardcoded into it. This algorithm is able to emulate the communication protocol (and the algorithm C_i), but now each time it needs a random string, it uses one provided in r.

From the correctness of the communication protocol for GT we again can conclude that for all $x, i \in \{0,1\}^n$

$$\Pr_{r \leftarrow \hat{\mathcal{R}}} [B_{(r,i)}(\mathcal{Y}[\hat{\mathcal{R}}](r, x)) \neq \mathsf{GT}_i(x)] < \log n \cdot 2^{-\alpha}.$$

So, with high probability, $B_{(r,i)}$ computes $\mathsf{GT}_i(x)$ over the choice of the oracle $\mathcal{Y}[\hat{\mathcal{R}}](r, x)$ for $r \leftarrow \hat{\mathcal{R}}$.

Does this new oracle satisfy the second condition? Now an approximation to $\mathsf{PUR}[\hat{\mathcal{R}}](\mathcal{Y}[\hat{\mathcal{R}}](r, x))$ needs to approximate $\mathsf{parity}(x)$ when given a set of random parities from $\hat{\mathcal{R}}$. Indeed, we show this requires high degree, as a consequence of the fact that high degree polynomial is necessary to construct the full parity of x from random parities.

However, we need to add one more improvement to our structure. For every fixed i, x, the algorithm $B_{(r,i)}$ when run on $\mathcal{Y}[\hat{\mathcal{R}}](r, x)$ computes $\mathsf{GT}_i(x)$ with high probability over $r \leftarrow \hat{\mathcal{R}}$. But we need to switch quantifiers: we want an oracle that is "good" for all possible inputs for GT simultaneously and, unfortunately, our current construction doesn't give an algorithm computing $\mathsf{GT}_i(x)$ for all $i, x \in \{0,1\}^n$ using the same $r \leftarrow \hat{\mathcal{R}}$.

Third (and final) attempt. So, is there a way to fix the source of randomness so it works for all possible inputs? Inspired by Newman's theorem [36] on simulating public randomness using private randomness in communication complexity, we show that there is. We show that by taking $t = O(\frac{n}{\delta^2})$ copies of $\hat{\mathcal{R}}$, denoted $\mathcal{R}_1, \mathcal{R}_2, \ldots \mathcal{R}_t$, we get a "good base" for the oracle. Consider a randomized algorithm $A_{(r,i)}$ that, given access to to $\mathcal{Y}[\mathcal{R}'](r,x)$ with $r \leftarrow \mathcal{R}' = \times_{j\in[t]}\mathcal{R}_j$, does the following:

- Sample $j \leftarrow [t]$ at random.
- Run $B_{(r,i)}$ using the set $\mathcal{R}_j$ as the source of randomness.

Following the argument underlying Newman's theorem, we show that this algorithm computes $\mathsf{GT}_i(x)$ with $\log n \cdot 2^{-\alpha} + \delta$ failure probability. It works for every i and x and it still makes only $\alpha \log n$ queries to the oracle. If we put $\delta = \frac{1}{12}$ and $\alpha = O(\log\log n)$ then the probability of this algorithm failing for some input pair is at most $\frac{1}{6}$ with only $\alpha \log n = O(\log n \log\log n)$ queries to the oracle, i.e.,

$$\Pr_{r\leftarrow\mathcal{R}'}[A_{(r,i)}(\mathcal{Y}[\mathcal{R}'](r,x)) \neq \mathsf{GT}_i(x)] < \frac{1}{6}.$$

This change also doesn't increase the "size" of the oracle (i.e., the number of queries it can answer) too much. This allows us to show that with high probability it is still impossible to combine the given partial parities to create the full parity using a low-degree polynomial, so the second condition is also satisfied. So there exists an oracle that allows computing the GT with low-degree polynomials but requires a high-degree polynomial to compute $\mathsf{parity}(x)$ which is exactly what allows us to prove the lower bound on the approximate degree of OS.

2.3 Technical ideas behind the parity lower bound

Our technique relies on a lower bound on the approximate degree of $\mathsf{parity}(x)$, or, more precisely, on the "Parity Under Randomness $\mathcal{R}$" function $\mathsf{PUR}[\mathcal{R}](\mathcal{Y}[\mathcal{R}](r,x))$ evaluates to $\mathsf{parity}(x)$ on input $\mathcal{Y}[\mathcal{R}](r,x)$. We, in fact, prove a more general statement lower bounding the approximate degree of $\mathsf{PUR}[\mathcal{R}]$ for a class of potential structures $\mathcal{R}$.

Specifically, we show that the parity function is hard, even to sign-represent, and even given access to $\mathcal{Y}[\mathcal{R}]$ consisting of inner products of x with random strings r_i where each bit of r_i is either fixed to zero or is an unbiased random bit. The only other restriction we need on $\mathcal{Y}[\mathcal{R}]$ is that its "size", i.e., the number of inner products it provides, is small. The bigger this number is, the worse our the lower bound becomes.

The proof idea is based on the hardness of sign-representing parity as described in [7], combined with the following combinatorial observation: given a set of n-bit strings (corresponding to samples from $\mathcal{R}$, and in turn to random inner products) where in every string each bit is either zero or is an unbiased random bit, with high probability no small subset of them adds up to the all-ones string (which corresponds to the parity function).

2.4 Improved ordered search and anchored hidden string lower bounds

Our generalized lower bound for approximating $\mathsf{PUR}[\mathcal{R}]$ allows us to obtain other lower bounds for oracle identification problems. For example, we give a slightly stronger lower bound for OS than what is implied by the discussion above. There is, in fact, a more efficient randomized communication protocol for GT that uses $O(\log n)$ bits of communication. It can be converted into randomized query algorithm and thus into a polynomial of degree $O(\log n)$. At the same time, this more efficient protocol is still based on computing equalities of substrings of inputs, and so the appropriate $\mathcal{Y}$ has a very similar structure to the one

described above while still satisfying the conditions of the generalized lower bound for PUR. Moreover, the necessary "size" of $\mathcal{Y}$ barely blows up at all. Putting everything together gives our improved lower bound of $\Omega\left(\frac{n}{\log^2 n}\right)$ on the approximate degree of OS.

Using the same framework, we can also obtain a nearly tight lower bound on the approximate degree of the anchored hidden string problem AHS. In the anchored hidden string problem, the goal is to determine the parity of x given oracle access to $y_{i,s} = \phi_{i,s}(x)$ for every index i and every binary string s of length at most n, where $\phi_{i,s}(x) = 1$ iff the substring of x starting at index i matches s. This oracle identification problem has the right form for our framework since

$$\mathsf{AHS}_N((\phi_{i,s}(x))_{i\in[n],s\in\{0,1\}^{\leq n-i+1}}) = \mathsf{parity}(x).$$

Moreover, each function $\phi_{i,s}(x)$ simply computes the equality function of s with a substring of x of length $|s|$ starting from position i. As we have already seen, we can compute the equality function very efficiently given an oracle $\mathcal{Y}$ of the right random structure, and such a $\mathcal{Y}$ meets the conditions of our generalized lower bound for $\mathsf{PUR}[\mathcal{Y}]$. This directly implies a lower bound of $\Omega\left(\frac{n}{\log n}\right)$ on the approximate degree of AHS.

Finally, the last lower bound described in this work is on the approximate degree of HS. This lower bound follows via a reduction from AHS. This reduction was first introduced in [25] in the quantum query model, but it holds for polynomial approximation as well.

3 Ordered search and generalized lower bound

In this section we give the formal proof of our lower bound on the approximate degree of ordered search. We show how our framework is used for this function and prove the generalized lower bound on parity that we later reuse for the hidden string problem.

3.1 Preliminaries

Our lower bounds on the approximate degree of (a decision version) of ordered search and the hidden string problem require the following definition of polynomial approximations for promise problems.

▶ **Definition 3.** *Let $f : D \to \{0,1\}$ where $D \subseteq \{0,1\}^n$ for some $n \in \mathbb{N}$ be a partial Boolean function. For $\frac{1}{2} > \varepsilon > 0$, a polynomial $p : \{0,1\}^n \to \mathbb{R}$ is an ε-approximation to f if $|p(x) - f(x)| \leq \varepsilon$ for every $x \in D$ and $-\varepsilon \leq p(x) \leq 1 + \varepsilon$ for all $x \in \{0,1\}^n$. The ε-approximate degree of f, denoted $\widetilde{\deg}_\varepsilon(f)$ is the the least degree of a polynomial p that ε-approximates f. We use the convention $\widetilde{\deg}(f) = \widetilde{\deg}_{1/3}(f)$ to refer to the "approximate degree of f" without qualification.*

That is, we require a polynomial approximation to a partial function defined on a domain D to approximate the function on D and remain bounded outside of D. Note that this is the type of approximation that arises from quantum query algorithms for promise problems.

We also formally define the ordered search function OS and the family of greater-than functions GT.

▶ **Definition 4.** *For all $i \in \{0,1\}^n$ define the function $\mathsf{GT}_i : \{0,1\}^n \to \{0,1\}$ to be the indicator of whether the value of the input is smaller than i: $\mathsf{GT}_i(x) = 1$ if and only if $x \leq i$ where i and x are compared as numbers written in binary notation.*

▶ **Definition 5.** *The ordered search function* $\mathsf{OS}_{2^n} : \{0^k 1^{2^n-k} \mid k \in [2^n]\} \to \{0,1\}$ *is a partial function defined the following way:* $\mathsf{OS}_{2^n}(0^k 1^{2^n-k}) = \mathsf{parity}(x)$ *where* $x \in \{0,1\}^n$ *is the binary representation of* k*.*

3.2 The notion of a *good base*

In order to formally define the oracle, i.e. the source of additional information about the input, we introduce the notion of a "*good base*" for the oracle. A set $\mathcal{R}$, consisting of tuples of strings, is a *good base* if it's constructed as follows.

Let $\mathcal{R}'$ be a Cartesian product of m' subsets of $\{0,1\}^n$ where each subset $\mathcal{R}^\tau$ is itself defined by an n-bit string-template $\tau = \tau_1\tau_2\ldots\tau_n \in \{0,1\}^n$

$$\mathcal{R}^\tau = S_{\tau_1} S_{\tau_2} S_{\tau_3} \ldots S_{\tau_n}$$

where $S_0 = \{0\}$ and $S_1 = \{0,1\}$.

For example, if $\tau = 00100010$ then $\mathcal{R}^\tau = S_{\tau_1} S_{\tau_2} S_{\tau_3} \ldots S_{\tau_n} = S_0 S_0 S_1 S_0 S_0 S_0 S_1 S_0 = \{0\}\{0\}\{0,1\}\{0\}\{0\}\{0\}\{0,1\}\{0\} = \{00000000, 00000010, 00100000, 00100010\}$.

Let $\mathcal{B} = \{\mathbf{1}_1\} \times \{\mathbf{1}_2\} \times \ldots \times \{\mathbf{1}_n\}$ where $\mathbf{1}_j = 0^{i-j}10^{n-j}$ is the string that has the value 1 in j-th position and has the value 0 everywhere else. Let $\mathcal{R} = \mathcal{B} \times \mathcal{R}'$, and thus $\mathcal{R}$ is a Cartesian product of $m = n + m'$ subsets of $\{0,1\}^n$. Note that every $r \in \mathcal{R}$ is a m-tuple of n-bit strings:

$$r = (r_1, r_2, \ldots r_m) = (\mathbf{1}_1, \mathbf{1}_2, \ldots, \mathbf{1}_{n-1}, \mathbf{1}_n, r_{n+1}, r_{n+2}, \ldots, r_{n+m'})$$

where each r_j is a string of length n, the first n strings are fixed for all $r \in \mathcal{R}$, and the last m' strings are from some sets $\mathcal{R}^\tau$ each for some template τ. If $r \leftarrow \mathcal{R}$ is chosen u.a.r. then each $r_j, n < j \le m'$ is chosen u.a.r. from some $\mathcal{R}^\tau$ and thus the subsequence of bits of r_j corresponding to ones in τ is a uniformly random string, and the subsequence of bits of r_j corresponding to zeros in τ is the all-zero string.

Any set $\mathcal{R}$ with the above structure will be called a *good base* of size m. Such an $\mathcal{R}$ is helpful for building our oracles as follows.

Let $\mathcal{Y}[\mathcal{R}] : \mathcal{R} \times \{0,1\}^n \to \{0,1\}^m$ be the following function: $(\mathcal{Y}[\mathcal{R}](r,x))_j = \langle r_j, x\rangle$ where r_j is an n-bit string from the collection $r \in \mathcal{R}$ and the inner product is taken modulo 2. Note that $\mathcal{Y}[\mathcal{R}]$ is parameterized by $\mathcal{R}$, so for each *good base* $\mathcal{R}$ the function $\mathcal{Y}[\mathcal{R}]$ will be different. We will omit the parameter $\mathcal{R}$ later in places where it is clear from context.

Notice the following properties of this function $\mathcal{Y}[\mathcal{R}](r,x)$ that hold whenever $\mathcal{R}$ is a *good base*:

- For every $r \in \mathcal{R}$, the values $\mathcal{Y}(r,x)$ completely determine x. Since the first n strings of r are $\mathbf{1}_1$, $\mathbf{1}_2$, …, $\mathbf{1}_{n-1}$, $\mathbf{1}_n$, the first n bits of $\mathcal{Y}(r,x)$ are exactly bits of x.
- Given $\mathcal{Y}(r,x)$ for $r \leftarrow \mathcal{R}$ and r itself, one can compute (with some probability of error) whether a subsequence of x specified by some pattern τ agrees with some fixed string s in those indices. To be more specific, if given $(\mathcal{Y}(r,x))_j = \langle r_j, x\rangle$ where r_j is sampled from $\mathcal{R}^\tau$ uniformly at random, and r_j itself, one can check whether the strings $x \wedge \tau$ (where $\wedge$ denotes bitwise AND) and $s \wedge \tau$ are equal for any $s \in \{0,1\}^n$ with one-sided error probability $\frac{1}{2}$.

So, $\mathcal{Y}[\mathcal{R}](r,x)$ could be used as an equality oracle for a fixed set of subsequences of x (predefined by $\mathcal{R}$) when r is chosen uniformly at random from $\mathcal{R}$. Thus, $\mathcal{Y}[\mathcal{R}](r,x)$ might give more information about x than x alone and might make some computations on x more efficient.

On the other hand, some functions of x remain "hard" even when given $\mathcal{Y}(r,x)$. We will later show that $\mathsf{parity}(x)$ remains hard to compute even with this additional information.

3.3 Approximating polynomials for GT_i

We start our proof by showing that for some *good base* $\mathcal{R}_{\mathsf{OS}}$ the oracle $\mathcal{Y}[\mathcal{R}_{\mathsf{OS}}]$ could be used to make the computation of GT functions more efficient.

▷ Claim 6. There exists a *good base* $\mathcal{R}_{\mathsf{OS}}$ of size $m = O(n^2 \log\log n)$ such that if $r \leftarrow \mathcal{R}_{\mathsf{OS}}$ is sampled uniformly at random, then with probability at least $\frac{2}{3}$ over the choice of r there exists a family of 2^n polynomials $\{q_{(r,i)} : \{0,1\}^m \to \{0,1\} \mid i \in \{0,1\}^n\}$, each of degree at most $2\log n \log\log n$, such that given $\mathcal{Y}[\mathcal{R}_{\mathsf{OS}}](r,x)$ as the input, each polynomial $q_{(r,i)}(\mathcal{Y}[\mathcal{R}_{\mathsf{OS}}](r,x))$ approximates the corresponding $\mathsf{GT}_i(x)$ with error at most $\frac{1}{6}$. That is,

$$\Pr_{r\leftarrow\mathcal{R}_{\mathsf{OS}}}\left[\exists i,x\in\{0,1\}^n : \left|q_{(r,i)}(\mathcal{Y}(r,x)) - \mathsf{GT}_i(x)\right| > \frac{1}{6}\right] < \frac{1}{3}.$$

Proof. This proof consists of two parts: constructing a *good base* $\mathcal{R}_{\mathsf{OS}}$ and showing that it actually helps to compute every GT_i.

Constructing the *good base* $\mathcal{R}_{\mathsf{OS}}$. We are going to construct $\mathcal{R}_{\mathsf{OS}}$ based on what random strings are useful in the communication protocol computing GT of two n-bit strings, x and i. Intuitively, in this protocol, we first need to check if the first half of i and x are equal using a randomized communication protocol for equality. To do that we need to compute and compare $\langle x,r\rangle$ and $\langle i,r\rangle$, for some number α of random strings r to be determined later, where each r is sampled from $\{0,1\}^{\frac{n}{2}}\{0\}^{\frac{n}{2}}$. If the computed values $\langle i,r\rangle = \langle x,r\rangle$ for all r we have considered, then we repeat this procedure on the second half of x and i, which corresponds to computing and comparing $\langle x,r\rangle$ and $\langle i,r\rangle$ for α random strings r sampled from $\{0\}^{\frac{n}{2}}\{0,1\}^{\frac{n}{4}}\{0\}^{\frac{n}{4}}$. If, on the other hand, the values were not equal then we repeat this procedure on the first half of x and i, which corresponds to computing and comparing $\langle x,r\rangle$ and $\langle i,r\rangle$ for α random strings r sampled from $\{0,1\}^{\frac{n}{4}}\{0\}^{\frac{3n}{4}}$. Since we want our oracle to be useful to emulate this procedure to compute $\mathsf{GT}_i(x)$, it should "contain" all the random strings used in this protocol.

Let $\hat{\mathcal{R}} = \mathcal{R}^{1^{n/2}0^{n/2}} \times \left(\mathcal{R}^{1^{n/4}0^{3n/4}} \times \mathcal{R}^{0^{n/2}1^{n/4}0^{n/4}}\right) \times \ldots \times \left(\bigtimes_{i=0}^{2^k-1} \mathcal{R}^{0^{2in/2^{k+1}}1^{n/2^{k+1}}0^{n-((2i+1)n/2^{k+1})}}\right) \times \ldots \times \left(\bigtimes_{i=0}^{n/2} \mathcal{R}^{0^{2i}1^{1}0^{n-(2i+1)}}\right)$. See Figure 1 for an illustration.

τ	$\mathcal{R}^\tau$	Structure of $\mathcal{R}^\tau$
$1^{\frac{n}{2}}0^{\frac{n}{2}}$	$\{0,1\}^{\frac{n}{2}}\{0\}^{\frac{n}{2}}$	★ ★ ★ ★ ★ ★ ★ ★ 0 0 0 0 0 0 0 0
$1^{\frac{n}{4}}0^{\frac{3n}{4}}$	$\{0,1\}^{\frac{n}{4}}\{0\}^{\frac{3n}{4}}$	★ ★ ★ ★ 0 0 0 0 0 0 0 0 0 0 0 0
$0^{\frac{n}{2}}1^{\frac{n}{4}}0^{\frac{n}{4}}$	$\{0\}^{\frac{n}{2}}\{0,1\}^{\frac{n}{4}}\{0\}^{\frac{n}{4}}$	0 0 0 0 0 0 0 0 ★ ★ ★ ★ 0 0 0 0
$1^{\frac{n}{8}}0^{\frac{7n}{8}}$	$\{0,1\}^{\frac{n}{8}}\{0\}^{\frac{7n}{8}}$	★ ★ 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$0^{\frac{n}{4}}1^{\frac{n}{8}}0^{\frac{5n}{8}}$	$\{0\}^{\frac{n}{4}}\{0,1\}^{\frac{n}{8}}\{0\}^{\frac{5n}{8}}$	0 0 0 0 ★ ★ 0 0 0 0 0 0 0 0 0 0
$0^{\frac{n}{2}}1^{\frac{n}{8}}0^{\frac{3n}{8}}$	$\{0\}^{\frac{n}{2}}\{0,1\}^{\frac{n}{4}}\{0\}^{\frac{3n}{8}}$	0 0 0 0 0 0 0 0 ★ ★ 0 0 0 0 0 0
$0^{\frac{3n}{4}}1^{\frac{n}{8}}0^{\frac{n}{8}}$	$\{0\}^{\frac{3n}{4}}\{0,1\}^{\frac{n}{8}}\{0\}^{\frac{n}{8}}$	0 0 0 0 0 0 0 0 0 0 0 0 ★ ★ 0 0

Figure 1 Structure of $\hat{\mathcal{R}}$. Blue cells with ⋆ represent indices in which either a 0 or a 1 could appear.

This $\hat{\mathcal{R}}$ describes all the strings used as the source of randomness in the $O(\log n \log\log n)$ communication protocol for GT, but each of the strings appears in the structure only once instead of α times. So, we need to duplicate this structure α times to properly simulate the protocol.

To finish the structure, we are going to add two other steps to the structure. First, we are going to have some number t of individual "prepackaged" copies to be determined later for the GT protocol. Let $\mathcal{R}_1 = \ldots = \mathcal{R}_t = \times_\alpha \hat{\mathcal{R}}$. Each of the copies has enough randomness and the right structure of that randomness to simulate one full run of the GT protocol. Let $\mathcal{R}' = \bigtimes_{j\in[t]} \mathcal{R}_j$ which allows us to handle t runs. Secondly, we want to be able to obtain the value of any specific index of x, so we add a set of "basis" strings to the structure: $\mathcal{B} = \{\mathbf{1}_1\} \times \{\mathbf{1}_2\} \times \ldots \times \{\mathbf{1}_n\} = \{10\ldots0\} \times \{010\ldots0\} \times \ldots \times \{00\ldots010\} \times \{00\ldots01\}$.

The final underlying structure of the oracle will be a Cartesian product of $\mathcal{R}'$ and $\mathcal{B}$: $\mathcal{R}_{\mathsf{OS}} = \mathcal{B} \times \mathcal{R}' = \mathcal{B} \times (\bigtimes_{j\in[t]} \mathcal{R}_j)$. See Figure 2 for an illustration.

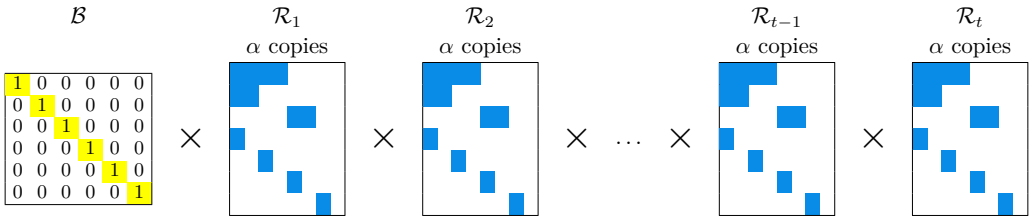


Figure 2 Structure of $\mathcal{R}_{\mathsf{OS}}$. Each $\mathcal{R}_j$ consist of α copies of $\hat{\mathcal{R}}$.

We also set the parameters to be $\alpha = 2\log(\log n), t = 250n\ln 2$. Notice that this set $\mathcal{R}_{\mathsf{OS}}$ is a *good base* by construction and has size $m = n + \alpha t n = n + cn^2 \log(\log n)$ for some constant c.

Constructing the family of approximating polynomials. In order to prove this claim, we first describe a randomized query algorithm that computes $\mathsf{GT}_i(x)$ correctly for all i and x with high probability given $\mathcal{Y}[\mathcal{R}_{\mathsf{OS}}](r,x)$ as input. We then explain how to convert this query algorithm into a polynomial. The algorithm construction itself consists of two parts. In the first part, for all $j \in [t]$ we show the existence of a deterministic algorithm $B_{(r,i,j)}$ that, given $\mathcal{Y}(r,x)$, can compute $\mathsf{GT}_i(x)$ for every specific $x, i \in \{0,1\}^n$ with good probability over the choice of $r \leftarrow \mathcal{R}_{\mathsf{OS}}$, and this algorithm is only going to use the parts of the input that correspond to $\mathcal{R}_j$ and $\mathcal{B}$. In the second part, we show that the algorithm $A_{(r,i)}$ that chooses a copy j to use randomly and runs $B_{(r,i,j)}$, computes $\mathsf{GT}_i(x)$ correctly for all i and x with high probability given $\mathcal{Y}(r,x)$ as input.

For all $i \in \{0,1\}^n, j \in [t], r \in \mathcal{R}_{\mathsf{OS}}$ let $B_{(r,i,j)}(\mathcal{Y}(r,x))$ be the following deterministic algorithm.

1. Set $\ell = 0, u = n/2$
2. While $\ell < u$:
3. Set $\tau = 0^\ell 1^{u-\ell} 0^{n-u}$
4. For all indices $v \in [m]$ corresponding to n-bit strings drawn from $\mathcal{R}^\tau$ within the j-th copy $\mathcal{R}_j$:
5. Compute $\langle i, r_v\rangle$ and compare it to $(\mathcal{Y}(r,x))_v = \langle x, r_v\rangle$.
6. If for all such v the inner products are equal, i.e., $\langle i, r_v\rangle = (\mathcal{Y}(r,x))_v$, then set $tmp = u, u = u + (u-\ell)/2, \ell = tmp$ and go step 2.
7. Otherwise, set $u = (u+\ell)/2$ and go step 2
8. Compare $i_\ell = \langle i, \mathbf{1}_\ell\rangle$ and $(\mathcal{Y}(r,x))_\ell = \langle x, \mathbf{1}_\ell\rangle = x_\ell$. If $x_\ell \le i_\ell$ then accept. Otherwise, reject.

The last step is possible specifically because of $\mathcal{B}$ in the structure of $\mathcal{R}_{\mathsf{OS}}$: $r_\ell = \mathbf{1}_\ell$ for all $\ell \le n$ and for all $r \in \mathcal{R}_{\mathsf{OS}}$. Notice that this algorithm emulates the randomized communication protocol for the GT communication problem.

In general, the algorithm emulates the randomized communication protocol for equality on the first half of the segment $[\ell, u + (u - \ell)]$ in x and i, and depending on the result it splits the inputs into smaller segments and continues recursively. In the end, if all the runs of equality protocols were correct, the algorithm finds and compares the most significant bit where x and i differ.

By [37] we know that this algorithm computes $\mathsf{GT}_i(x)$ with probability at least $1 - (\log n)2^{-\alpha} = 1 - (\log n)2^{-2\log(\log n)} = 1 - \frac{1}{\log n} \ge \frac{11}{12}$ for sufficiently large n independently of the choice of $j \in [t]$. That is, for all $j \in [t]$ and for all $i, x \in \{0,1\}^n$,

$$\Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}[B_{(r,i,j)}(\mathcal{Y}(r,x)) = \mathsf{GT}_i(x)] \ge \frac{11}{12}.$$

This algorithm makes at most $\alpha \log n = 2 \log n \log\log n$ queries to the oracle $\mathcal{Y}(r,x)$. Note that this algorithm needs access to the specific r needed to compute every $\langle i, r_v \rangle$ and we enable this by "hardcoding" this r into the algorithm and creating a separate algorithm for each possible r.

We have shown that for every fixed $i, x \in \{0,1\}^n$ there are many $r \in \mathcal{R}_{\mathsf{OS}}$ that if used as a first input for the oracle $\mathcal{Y}$ allow $B_{(r,i,j)}$ to compute $\mathsf{GT}_i(x)$. Unfortunately, this is not enough: our algorithm should be universal, i.e., we want a single algorithm that with high probability over r succeeds on all i and x. On the other hand, $B_{(r,i,j)}$ only uses one fixed "package" of random strings, namely the j-th package.

Let $W(i,x,r,j)$ be the indicator that the j-th package of random strings in r defines a set of "bad" random strings for (i,x): $W(i,x,r,j) = 1$ if and only if $B_{(r,i,j)}(\mathcal{Y}(r,x)) \ne \mathsf{GT}_i(x)$. We established that $B_{(r,i,j)}(\mathcal{Y}(r,x))$ works well if given a random $r \leftarrow \mathcal{R}_{\mathsf{OS}}$ for every $j \in [t]$ and the probability of this algorithm outputting an incorrect answer is at most $\frac{1}{12}$. So for all $i, x \in \{0,1\}^n, j \in [t]$, we have

$$\Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}[W(i,x,r,j) = 1] = \mathbb{E}_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}[W(i,x,r,j)] \le \frac{1}{12}.$$

We can't immediately get a useful upper bound on the probability of $r \leftarrow \mathcal{R}$ working out for all i and x at the same time. To achieve this, we'll design a new algorithm that uses all t packages of random strings. Its construction and analysis are inspired by Newman's classic argument used for simulating public randomness by private randomness in communication protocols.

For all $i \in \{0,1\}^n, r \in \mathcal{R}_{\mathsf{OS}}$ let $A_{(r,i)}(\mathcal{Y}(r,x))$ be the following randomized algorithm:

- Choose $j \leftarrow [t]$ uniformly at random.
- Run $B_{(r,i,j)}(\mathcal{Y}(r,x))$.

Let us now analyse $A_{(r,i)}$. The number of queries that $A_{(r,i)}$ makes to the oracle is the same as $B_{(r,i,j)}$ which is $\alpha \log n = 2 \log n \log\log n$. We fix a pair (i,x) and evaluate the following probability.

$$\Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}\left[\Pr_{j \leftarrow [t]}[B_{(r,i,j)} \ne \mathsf{GT}_i(x)] > \frac{1}{6}\right] = \Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}\left[\frac{1}{t}\sum_{j \in [t]} W(i,x,r,j) > \frac{1}{6}\right].$$

We established that $\mathbb{E}_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}[W(i,x,r,j)] \le \frac{1}{12}$ and so by Hoeffding's inequality,

$$\Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}\left[\frac{1}{t}\sum_{j \in [t]} W(i,x,r,j) > \frac{1}{12} + \frac{1}{12}\right] \le e^{-2\frac{t}{144}} \le 2^{-\frac{500n}{144}}.$$

By a union bound over all possible $i, x \in \{0,1\}^n$,

$$\Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}\left[\exists i, x \in \{0,1\}^n : \frac{1}{t}\sum_{j \in [t]} W(i,x,r,j) > \frac{1}{6}\right] \leq 2^{2n} 2^{-\frac{500n}{144}} \leq 2^{-n} < \frac{1}{3}.$$

Therefore, we have proven that

$$\Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}\left[\exists i, x \in \{0,1\}^n : \Pr_{j \leftarrow [t]}[A_{(r,i)}(\mathcal{Y}(r,x))) \neq \mathsf{GT}_i(x)] > \frac{1}{6}\right] < \frac{1}{3}.$$

The last step is to convert this family of query algorithms into a family of approximating polynomials. Let $q_{(r,i)}$ denote the acceptance probability of $A_{(r,i)}$. A standard argument (e.g., [16, Theorem 15]) implies that this is a polynomial of degree at most $2 \log n \log\log n$ such that

$$\Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}}}\left[\exists i, x \in \{0,1\}^n : \left|q_{(r,i)}(\mathcal{Y}(r,x)) - \mathsf{GT}_i(x)\right| > \frac{1}{6}\right] < \frac{1}{3},$$

which is exactly what we were looking for. ◁

We successfully converted the most well-known communication protocol for GT that requires $O(\log n \log\log n)$ bits of communication into a family of polynomials of degree $O(\log n \log\log n)$ that approximates GT_i. It's known that there is a better communication protocol for GT that requires only $O(\log n)$ bits of communication, as observed by Nisan [37]. The next claim establishes that this more efficient protocol can be converted into a family of polynomials as well.

▷ Claim 7. There exists a *good base* $\mathcal{R}_{\mathsf{OS}++}$ of size $m = O(n^3 \log n)$ such that if $r \leftarrow \mathcal{R}_{\mathsf{OS}++}$ is sampled uniformly at random, then with probability at least $\frac{2}{3}$ over the choice of r there exists a family of polynomials $\{q_{(r,i)} : \{0,1\}^m \to \{0,1\} \mid i \in \{0,1\}^n\}$, each of degree at most $O(\log n)$, such that given $\mathcal{Y}(r,x)$ as the input, each polynomial $q_{(r,i)}(\mathcal{Y}[\mathcal{R}_{\mathsf{OS}++}](r,x))$ approximates the corresponding $\mathsf{GT}_i(x)$ with error at most $\frac{1}{6}$. That is,

$$\Pr_{r \leftarrow \mathcal{R}_{\mathsf{OS}++}}\left[\exists i, x \in \{0,1\}^n : \left|q_{(r,i)}(\mathcal{Y}(r,x)) - \mathsf{GT}_i(x)\right| > \frac{1}{6}\right] < \frac{1}{3}.$$

The proof of Claim 7 is similar to the proof of Claim 6 and can be found in Appendix B in the full version of our paper [20].

3.4 General lower bound

To complete the framework and to obtain the lower bound for Ordered Search we need to show why computing the parity is hard even given $\mathcal{Y}[\mathcal{R}_{\mathsf{OS}}]$ or $\mathcal{Y}[\mathcal{R}_{\mathsf{OS}++}]$. We will show a stronger lower bound that would allow us to reuse this lower bound for other applications. Specifically, we will show that computing the parity of input x remains hard given $\mathcal{Y}[\mathcal{R}]$ for any *good base* $\mathcal{R}$ of small size.

3.4.1 Combinatorial claim

The hardness of parity in this model is based on the following statement. For every *good base* $\mathcal{R}$ of small size with high probability over the sample $r \leftarrow \mathcal{R}$ for every set of n-bit strings taken from the collection r of size at most $O(\frac{n}{\log n})$, the bitwise parity of these strings is not equal to the all-ones string.

▷ Claim 8. For every *good base* $\mathcal{R}$ of size m with probability at least $\frac{2}{3}$ over the choice of $r \leftarrow \mathcal{R}$ for every set of elements $T \subseteq [m]$ of size at most $d = \frac{n}{4 \log m} - 1$, the bitwise parity of n-bit strings r_i, $i \in T$ from the collection $r \leftarrow \mathcal{R}$ is not equal to the all-ones string:

$$\Pr_{r \leftarrow \mathcal{R}}\left[\forall T \subseteq [m], |T| \le d : \bigoplus_{i \in T} r_i \neq 1^n\right] \ge \frac{2}{3}.$$

Proof. Fix an arbitrary *good base* $\mathcal{R}$ of size m. Fix a set $T \subseteq [m]$ where $|T| \le d$. We want to bound the probability $\Pr_{r \leftarrow \mathcal{R}}[\bigoplus_{i \in T} r_i = 1^n]$ that for this r and for this T the strings corresponding to the indices in T sum up to the string of all ones. Fix a specific index $k \in [n]$. We compute the probability that index k is set to 1 in $\bigoplus_{i \in T} r_i$. To do this we need to understand how the candidate strings $r_i, i \in T$ can influence this value.

There are three possible scenarios for each index k:

- (Type I) There is at least one string $r_i \in \{0,1\}^n$ with $i \in T$ such that it is chosen from $\mathcal{R}^\tau$ where $\tau_k = 1$. Then in each such string, the bit at index k is sampled independently at random with probability $\frac{1}{2}$. Thus $\Pr_{r \leftarrow \mathcal{R}}[\langle \bigoplus_{i \in T} r_i, \mathbf{1}_k \rangle = 1] = \frac{1}{2}$.
- (Type II) There are no strings $r_i, i \in T$ such that r_i is chosen from $\mathcal{R}^\tau$ and $\tau_k = 1$, but there is $r_i, i \in T$ that is chosen from $\mathcal{B}$, such that $r_i = \mathbf{1}_k$. Then the value of $\langle \bigoplus_{i \in T} r_i, \mathbf{1}_k \rangle$ is one since there is exactly one string in this sum with the kth index value set to one. Thus $\Pr_{r \leftarrow \mathcal{R}}[\langle \bigoplus_{i \in T} r_i, \mathbf{1}_k \rangle = 1] = 1$.
- (Type III) There are no strings $r_i, i \in T$ such that r_i is chosen from $\mathcal{R}^\tau$ and $\tau_k = 1$, and there is no $r_i, i \in T$ that is chosen from $\mathcal{B}$, such that $r_i = \mathbf{1}_k$. Then for all strings r_i the index k is 0, so $\Pr_{r \leftarrow \mathcal{R}}[\langle \bigoplus_{i \in T} r_i, \mathbf{1}_k \rangle = 1] = 0$.

Index k	1	2	3	4	5	6	7	8
r_{i_1}	★	★	★	★	0	0	★	0
r_{i_2}	★	★	★	0	0	0	0	0
r_{i_3}	★	0	★	★	0	0	★	0
r_{i_4}	0	★	0	0	0	0	0	0
r_{i_5}	0	0	1	0	0	0	0	0
r_{i_6}	0	0	0	0	1	0	0	0
Type	I	I	I	I	II	III	I	III
$\Pr_{r \leftarrow \mathcal{R}}[\langle \bigoplus_{i \in T} r_i, \mathbf{1}_k \rangle]$	1/2	1/2	1/2	1/2	1	0	1/2	0

Figure 3 Example of index types, $T = \{i_1, i_2, i_3, i_4, i_5, i_6\}$.

Notice that T fully defines the types of all indices and thus the values of $\langle \bigoplus_{i \in T} r_i, \mathbf{1}_k \rangle$ for k of types II and III don't depend on the choice of $r \leftarrow \mathcal{R}$. On the other hand, the values of indices of type I do depend on the choice of $r \leftarrow \mathcal{R}$. Each of them is either a parity of independent random bits or the negation of a parity of independent random bits which is fixed by T too. Thus they behave as independent bits themselves and therefore the values $\langle \bigoplus_{i \in T} r_i, \mathbf{1}_k \rangle$ are mutually independent for all indices k.

Denote by $n_{\mathrm{I}}, n_{\mathrm{II}}, n_{\mathrm{III}}$ the numbers of indices of each type. Notice that $n_{\mathrm{I}} + n_{\mathrm{II}} + n_{\mathrm{III}} = n$ and $n_{\mathrm{II}} \leq d$. Then in this notation

$$\Pr_{r \leftarrow \mathcal{R}}\left[\bigoplus_{i \in T} r_i = 1^n\right] = \left(\frac{1}{2}\right)^{n_{\mathrm{I}}} 1^{n_{\mathrm{II}}} 0^{n_{\mathrm{III}}}.$$

If there exists $k \in [n]$ of the third type, the probability $\Pr_{r \leftarrow \mathcal{R}}[\bigoplus_{i \in T} r_i = 1^n]$ becomes 0, so to upper bound the probability we may assume all the indices have one of the first two types. And, since $n_{\mathrm{II}} \leq d$, to maximize the value we assume that $n_{\mathrm{II}} = d$. Thus we have

$$\Pr_{r \leftarrow \mathcal{R}}\left[\bigoplus_{i \in T} r_i = 1^n\right] = \left(\frac{1}{2}\right)^{n - n_{\mathrm{II}}} 1^{n_{\mathrm{II}}} \leq 1^d \left(\frac{1}{2}\right)^{n-d} = 2^{-(n-d)}.$$

Since $d = \frac{n}{4 \log m} - 1$ and $m \geq n$, we have $n - d > \frac{n}{2}$ for sufficiently large n. So for a fixed T,

$$\Pr_{r \leftarrow \mathcal{R}}\left[\bigoplus_{i \in T} r_i = 1^n\right] < 2^{-\frac{n}{2}}.$$

There are $\binom{m}{\leq d}$ ways to choose the set T, so by a union bound over the choice of T, the probability that for some set of size at most d the value $\bigoplus_{i \in T} r_i$ is equal to the string of all ones is

$$\Pr_{r \leftarrow \mathcal{R}}\left[\exists T \subseteq [m], |T| \leq d : \bigoplus_{i \in T} r_i = 1^n\right] \leq \binom{m}{\leq d} 2^{-\frac{n}{2}} = 2^{-\frac{n}{2}} \sum_{d'=0}^{d} \binom{m}{d'} \leq 2^{-\frac{n}{2}} \sum_{d'=0}^{d} m^{d'}$$

$$\leq 2^{-\frac{n}{2}} m^{d+1} = 2^{-\frac{n}{2}} m^{\frac{n}{4 \log m}} = 2^{-\frac{n}{2}} 2^{\frac{n}{4 \log m} \log m} = 2^{\frac{n}{4} - \frac{n}{2}} = 2^{-\frac{n}{4}} < \frac{1}{3}. \qquad \lhd$$

3.4.2 Lower bound on the degree of PUR$[\mathcal{R}]$

For every *good base* $\mathcal{R}$ and for every fixed $r \in \mathcal{R}$ define the function $\mathsf{PUR}[\mathcal{R}]_r : D[\mathcal{R}]_r \to \{0,1\}$ where $D[\mathcal{R}]_r = \{\mathcal{Y}[\mathcal{R}](r,x) \mid x \in \{0,1\}^n\}$ is the subset of $\{0,1\}^m$ where each domain point corresponds to one specific $x \in \{0,1\}^n$ and is consistent with the fixed r. This function outputs the parity of the string encoded by the input: $\mathsf{PUR}[\mathcal{R}]_r(\mathcal{Y}(r,x)) = \mathsf{parity}(x)$. It is well defined since $\mathsf{parity}(x) = \bigoplus_{r_i \in \mathcal{B}} \langle x, r_i \rangle = \bigoplus_{i \in [n]} (\mathcal{Y}(r,x))_i$. Note that both $D[\mathcal{R}]_r$ and $\mathsf{PUR}[\mathcal{R}]$ are parameterized by $\mathcal{R}$ and, as with $\mathcal{Y}[\mathcal{R}]$, we will omit the parameter later in places where the parameter is clear from the context.

Our goal is to show that $\mathsf{PUR}[\mathcal{R}]$ is hard to approximate if $\mathcal{R}$ is a *good base* of small size. We do this by showing that for every *good base* $\mathcal{R}$ of size m if $r \leftarrow \mathcal{R}$ u.a.r. then every polynomial p of degree at most $d = O(\frac{n}{\log m})$ is completely uncorrelated with $\mathsf{PUR}[\mathcal{R}]_r(\mathcal{Y}(r,x))$ with high probability over the choice of r.

▶ **Theorem 9.** *For every* good base $\mathcal{R}$ *of size* m *if* $r \leftarrow \mathcal{R}$ *u.a.r. then with probability at least* $\frac{2}{3}$ *over the choice of* r *every polynomial* $p : \{0,1\}^m \to \mathbb{R}$ *of degree at most* $d = \frac{n}{4 \log m} - 1$ *doesn't approximate* $\mathsf{PUR}[\mathcal{R}]_r$*:*

$$\Pr_{r \leftarrow \mathcal{R}}\left[\forall \varepsilon < \frac{1}{2},\ \forall p, \deg(p) \leq d, \exists y \in D[\mathcal{R}]_r : |p(y) - \mathsf{PUR}[\mathcal{R}]_r(y)| > \varepsilon\right] \geq \frac{2}{3}$$

Note that Theorem 9 rules out approximating polynomials that may be unbounded outside of the domain of $\mathsf{PUR}[\mathcal{R}]_r$. That is, it asserts that there is no low-degree approximating polynomial even when that polynomial is permitted to take values outside of $[0,1]$ on points outside of the domain of PUR_r. Note also that since the lower bound applies for all $\varepsilon < 1/2$, it actually entails a threshold degree lower bound on computing $\mathsf{PUR}[\mathcal{R}]$.

Proof. Fix an arbitrary *good base* $\mathcal{R}$ of size m.

For convenience in this proof, let us change notation to consider polynomials approximations over $\{-1,1\}$ instead of over $\{0,1\}$. Define $\mathcal{Y}' : \mathcal{R} \times \{-1,1\}^n \to \{-1,1\}^m$ to be $(\mathcal{Y}'(r,x'))_i = 1 - 2(\mathcal{Y}(r,(\frac{1-x'_1}{2}, \frac{1-x'_2}{2}, \dots, \frac{1-x'_n}{2})))_i = 1 - 2\langle x, r_i\rangle$ where r_i is the vector corresponding to ith component of $\mathcal{Y}(r,x)$ and $x \in \{0,1\}^n$ is the vector that corresponds to $x' \in \{-1,1\}^n$: $x_i = \frac{1-x'_i}{2}$ for all $i \in [n]$. Notice that this change of notation satisfies the following: if $a \in \{0,1\}$ and a' is the corresponding value in the new notation $a' \in \{-1,1\}$ then $a' = (-1)^a$.

Let's also rewrite PUR_r in this new notation. Let D'_r represent the domain of PUR'_r: $D'_r = \{\mathcal{Y}'(r,x') \mid x' \in \{-1,1\}^n\}$ and the function $\mathsf{PUR}'_r : D'_r \to \{-1,1\}$ be $\mathsf{PUR}'_r(\mathcal{Y}'_1, \mathcal{Y}'_2, \dots, \mathcal{Y}'_m) = 1 - 2\mathsf{PUR}_r(\frac{1-\mathcal{Y}'_1}{2}, \frac{1-\mathcal{Y}'_2}{2}, \dots, \frac{1-\mathcal{Y}'_m}{2})$.

Note that every polynomial $p' : \{-1,1\}^m \to \mathbb{R}$ that approximates PUR'_r to error ε can be converted by a linear transformation into a polynomial $p : \{0,1\}^m \to \mathbb{R}$ of the same degree that approximates PUR_r to error $\varepsilon/2$. So it suffices to prove that no polynomial p' of degree at most d approximates PUR'_r to error $\varepsilon < 1$.

Assume toward a contradiction that there is a polynomial p' of degree d that approximates PUR'_r. This means that there exists $\varepsilon < 1$ such that for all $y' \in D'_r$,

$$\left|p'(y') - \mathsf{PUR}'_r(y')\right| < \varepsilon.$$

Consider the following expression:

$$\begin{aligned}\frac{1}{2^n}\left|\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')(\mathsf{PUR}'_r(y') - p'(y'))\right| &\le \frac{1}{2^n}\left(\max_{y'\in D'_r}\left|p'(y') - \mathsf{PUR}'_r(y')\right|\right)\left(\sum_{y'\in D'_r}\left|\mathsf{PUR}'_r(y')\right|\right)\\ &< \frac{1}{2^n}\varepsilon|D'_r| = \varepsilon.\end{aligned} \tag{1}$$

The last equality holds because $\mathcal{Y}'(r,\cdot)$ is surjective, and hence $|D'_r| = 2^n$. On the other hand,

$$\begin{aligned}\frac{1}{2^n}\left|\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')(\mathsf{PUR}'_r(y') - p'(y'))\right| &= \frac{1}{2^n}\left|\left(\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')\mathsf{PUR}'_r(y')\right) - \left(\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')p'(y')\right)\right|\\ &= \frac{1}{2^n}\left||D'_r| - \left(\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')p(y')\right)\right|.\end{aligned} \tag{2}$$

We now show that with high probability the expression above is equal to $\frac{|D'_r|}{2^n}$.

▷ Claim 10. With probability at least $\frac{2}{3}$ over the choice of $r \leftarrow \mathcal{R}$, for every polynomial $p' : \{-1,1\}^m \to \mathbb{R}$ of degree at most $d = \frac{n}{4\log m} - 1$ we have

$$\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')p'(y') = 0.$$

Proof. Fix a polynomial p' of degree at most $d = \frac{n}{4\log m} - 1$. By linearity it suffices to consider the case where p' is a monomial, $p'(y') = \prod_{j\in T} y'_j$ for some $T \subseteq [m], |T| \le d$. So

$$\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')p'(y') = \sum_{x'\in\{-1,1\}^n}\left(\prod_{i\in[n]}(x'_i)\right)\left(\prod_{j\in T}(\mathcal{Y}'(r,x'))_j\right)$$

$$= \sum_{x\in\{0,1\}^n} \left((-1)^{\langle x,1^n\rangle}\right)\left(\prod_{j\in T}(-1)^{\langle x,r_j\rangle}\right) = \sum_{x\in\{0,1\}^n} (-1)^{\langle x,1^n\rangle}(-1)^{\sum_{j\in T}\langle x,r_j\rangle}$$
$$= \sum_{x\in\{0,1\}^n} (-1)^{\langle x,1^n\rangle}(-1)^{\langle x,\bigoplus_{j\in T} r_j\rangle} = \sum_{x\in\{0,1\}^n} (-1)^{\langle x,1^n\oplus(\bigoplus_{j\in T} r_j)\rangle}$$

This expression is not zero if and only if $\bigoplus_{j\in T} r_j = 1^n$. By Claim 8 the probability that such T exists is at most $\frac{1}{3}$. So the probability over the choice of r for some polynomial $p' : \{-1,1\}^m \to \mathbb{R}$ of degree at most $d = \frac{n}{4\log m} - 1$ to have

$$\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')p'(y') \neq 0$$

is at most $\frac{1}{3}$. ◁

Combining expressions (1) and (2) and Claim 10, we have that with probability at least $\frac{2}{3}$,

$$\varepsilon > \frac{1}{2^n}\left|\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')(\mathsf{PUR}'_r(y') - p'(y'))\right| = \frac{1}{2^n}\left||D'_r| - \left(\sum_{y'\in D'_r} \mathsf{PUR}'_r(y')p'(y')\right)\right| = \frac{|D'_r|}{2^n} = 1.$$

And so $\varepsilon > 1$ which contradicts our assumption. Thus $\mathsf{PUR}'_r(\mathcal{Y}(r,x))$ cannot be approximated by a polynomial of degree at most $\frac{n}{4\log m} - 1$ with probability at least $\frac{2}{3}$ over the choice $r \leftarrow \mathcal{R}$ sampled uniformly at random. And therefore $\mathsf{PUR}_r(\mathcal{Y}(r,x))$ cannot be ε-approximated for every constant $\varepsilon < \frac{1}{2}$ with a polynomial of degree less than $\frac{n}{4\log m}$ with probability at least $\frac{2}{3}$ over the choice $r \leftarrow \mathcal{R}$ sampled uniformly at random for any *good base* $\mathcal{R}$ of size m. ◀

3.5 Lower bound for ordered search

Finally, we combine our general lower bound on the approximate degree of PUR with the upper bound on approximating GT_i to conclude our lower bound on the approximate degree of ordered search. We will use the statement of Claim 7 with a lower degree of polynomials approximating GT_i since, even though its proof is more complicated than the proof of the weaker bound, as it allows us to obtain a better lower bound on ordered search.

First, we apply Theorem 9 to obtain a lower bound on the approximate degree for $\mathsf{PUR}[\mathcal{R}_{\mathsf{OS}++}]$.

▶ **Corollary 11.** *If $r \leftarrow \mathcal{R}_{\mathsf{OS}++}$ u.a.r. then with probability at least $\frac{2}{3}$ over the choice of r every polynomial $p : \{0,1\}^m \to \mathbb{R}$ of degree at most $d = \frac{n}{16\log n} - 1$ fails to approximate $\mathsf{PUR}[\mathcal{R}_{\mathsf{OS}++}]_r$:*

$$\Pr_{r\leftarrow\mathcal{R}_{\mathsf{OS}++}}\left[\forall\varepsilon < \frac{1}{2},\ \forall p, \deg(p) \le d, \exists y \in D[\mathcal{R}_{\mathsf{OS}++}]_r : |p(y) - \mathsf{PUR}[\mathcal{R}_{\mathsf{OS}++}]_r(y)| > \varepsilon\right] \ge \frac{2}{3}.$$

Proof. The set $\mathcal{R}_{\mathsf{OS}++}$ is a *good base* and has size $m = O(n^3\log n)$. By Theorem 9, with probability at least $\frac{2}{3}$ over the choice of r every polynomial $p : \{0,1\}^m \to \mathbb{R}$ of degree at most $\frac{n}{4\log m} - 1$ fails to approximate $\mathsf{PUR}[\mathcal{R}_{\mathsf{OS}++}]_r$. But since the size of $\mathcal{R}_{\mathsf{OS}++}$ is $m \le n^4$ for sufficiently large n then every polynomial of degree at most $d = \frac{n}{16\log n} - 1 = \frac{n}{4\log n^4} - 1 \le \frac{n}{4\log m} - 1$ fails to approximate $\mathsf{PUR}[\mathcal{R}_{\mathsf{OS}++}]_r$. ◀

By combining Claim 7 and Corollary 11, we obtain the following.

▶ **Theorem 12.** *The approximate degree of ordered search is*

$$\widetilde{\deg}_{\frac{1}{2}-\gamma}(\mathsf{OS}_{2^n}) = \Omega\left(\frac{n}{\log^2 n} - \log\frac{1}{\gamma}\right)$$

where γ could depend on n, $0 < \gamma < \frac{1}{2}$.

Proof. Suppose OS_{2^n} can be $(\frac{1}{2}-\gamma)$-approximated by a bounded polynomial of degree d for some $\frac{1}{2} > \gamma > 0$. By [40, Theorem 1.1], for every $\delta > 0$, this polynomial can be converted to a polynomial p of degree $O(d + \log\frac{1}{\delta})$ that $(\frac{1}{2}-\gamma+\delta)$-approximates OS_{2^n} and is robust to noise in its inputs. That is,

$$|\mathsf{OS}_N(y) - p(y+\Delta)| < \frac{1}{2} - \gamma + \delta$$

for all $y \in \{0,1\}^N$, all $\Delta \in [-\frac{1}{6}, \frac{1}{6}]^N$, and $N = 2^n$. If we put $\delta = \frac{\gamma}{2}$, then p is a $(\frac{1}{2}-\frac{\gamma}{2})$-approximating polynomial for OS_{2^n} with degree $O\left(d + \log\left(\frac{1}{\gamma}\right)\right)$.

Note that $\mathsf{OS}_{2^n}(\mathsf{GT}_{0^n}(x), \mathsf{GT}_{0^{n-1}1}(x), \ldots, \mathsf{GT}_{1^n}(x)) = \mathsf{PUR}[\mathcal{R}_{\mathsf{OS}++}]_r(\mathcal{Y}(r,x))$ for every $r \in \mathcal{R}_{\mathsf{OS}++}$. So by Claim 7, there exists a constant c such that the composed polynomial $p(q_{(r,0^n)}(\mathcal{Y}(r,x)), q_{(r,0^{n-1}1)}(\mathcal{Y}(r,x)), \ldots q_{(r,1^n)}(\mathcal{Y}(r,x)))$ has degree at most $\deg(p)\max_i(\deg(q_{(r,i)})) = c\left(d + \log\left(\frac{1}{\gamma}\right)\right)\log n$ and approximates $\mathsf{PUR}[\mathcal{R}_{\mathsf{OS}++}]_r(\mathcal{Y}(r,x))$ to error $(\frac{1}{2}-\frac{\gamma}{2})$ with probability at least $\frac{2}{3}$ over the choice of $r \leftarrow \mathcal{R}_{\mathsf{OS}++}$. This holds because although the polynomials $q_{(r,i)}$ do not compute the functions GT_i exactly, but only approximate them with small error, the outer polynomial p is robust to this small error in the inputs. Note also that while the composed polynomial is bounded on the domain of PUR_r, it may be arbitrarily unbounded on points outside its domain.

On the other hand, by Claim 11, with probability at least $\frac{2}{3}$ over the choice of r, the function $\mathsf{PUR}[\mathcal{R}_{\mathsf{OS}++}]_r$ cannot be approximated to any error $(\frac{1}{2}-\frac{\gamma}{2}) \in (0, \frac{1}{2})$ by a polynomial in $\mathcal{Y}$ of degree less than $\frac{n}{16\log n}$. By a union bound, with probability at least $1-(1-\frac{2}{3})-(1-\frac{2}{3}) = \frac{1}{3}$ both conditions on r hold simultaneously. Thus there exists $r \in \mathcal{R}_{\mathsf{OS}++}$ such that $p(q_{(r,0^n)}(\mathcal{Y}(r,x)), q_{(r,0^{n-1}1)}(\mathcal{Y}(r,x)), \ldots q_{(r,1^n)}(\mathcal{Y}(r,x)))$ approximates $\mathsf{PUR}_r(\mathcal{Y}(r,x))$ and $\mathsf{PUR}_r(\mathcal{Y}(r,x))$ cannot be approximated by a polynomial of degree less than $\frac{n}{16\log n}$. So

$$c\left(d + \log\left(\frac{1}{\gamma}\right)\right)\log n \geq \frac{n}{16\log n}.$$

And thus

$$d + \log\left(\frac{1}{\gamma}\right) \geq \frac{n}{16c\log^2 n},$$

so we conclude that

$$d = \Omega\left(\frac{n}{\log^2 n} - \log\left(\frac{1}{\gamma}\right)\right).$$

◀

References

1 Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. 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 863–876. ACM, 2016.

2 A. Ambainis. A better lower bound for quantum algorithms searching an ordered list. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 352–357, 1999. doi:10.1109/SFFCS.1999.814606.

3 Andris Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. Syst. Sci.*, 72(2):220–238, 2006. doi:10.1016/j.jcss.2005.06.006.

4 Andris Ambainis, Kazuo Iwama, Akinori Kawachi, Hiroyuki Masuda, Raymond H. Putra, and Shigeru Yamashita. Quantum identification of boolean oracles. In Volker Diekert and Michel Habib, editors, *STACS 2004, 21st Annual Symposium on Theoretical Aspects of Computer Science, Montpellier, France, March 25-27, 2004, Proceedings*, volume 2996 of *Lecture Notes in Computer Science*, pages 105–116. Springer, 2004.

5 Andris Ambainis, Kazuo Iwama, Akinori Kawachi, Rudy Raymond, and Shigeru Yamashita. Improved algorithms for quantum identification of boolean oracles. *Theor. Comput. Sci.*, 378(1):41–53, 2007.

6 Andris Ambainis and Ashley Montanaro. Quantum algorithms for search with wildcards and combinatorial group testing. *Quantum Inf. Comput.*, 14(5-6):439–453, 2014. doi:10.26421/QIC14.5-6-4.

7 James Aspnes, Richard Beigel, Merrick Furst, and Steven Rudich. The expressive power of voting polynomials. In *Proceedings of the twenty-third annual ACM symposium on Theory of Computing*, pages 402–409, 1991.

8 Howard Barnum, Michael Saks, and Mario Szegedy. Quantum decision trees and semidefinite programming. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2001.

9 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM (JACM)*, 48(4):778–797, 2001.

10 Aleksandrs Belovs. Quantum algorithms for learning symmetric juntas via the adversary bound. *Comput. Complex.*, 24(2):255–293, 2015. doi:10.1007/s00037-015-0099-2.

11 Shalev Ben-David, Adam Bouland, Ankit Garg, and Robin Kothari. Classical lower bounds from quantum upper bounds. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 339–349. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00040.

12 Michael Ben-Or and Avinatan Hassidim. The bayesian learner is optimal for noisy binary search (and pretty good for quantum as well). In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 221–230, 2008. doi:10.1109/FOCS.2008.58.

13 Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 11–20. ACM, 1993. doi:10.1145/167088.167097.

14 Harry Buhrman, Richard Cleve, Ronald de Wolf, and Christof Zalka. Bounds for small-error and zero-error quantum algorithms. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 358–368. IEEE Computer Society, 1999. doi:10.1109/SFFCS.1999.814607.

15 Harry Buhrman and Ronald de Wolf. A lower bound for quantum search of an ordered list. *Information Processing Letters*, 70(5):205–209, 1999. doi:10.1016/S0020-0190(99)00069-1.

16 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002.

17 Harry Buhrman, Ilan Newman, Hein Röhrig, and Ronald de Wolf. Robust polynomials and quantum algorithms. *Theory Comput. Syst.*, 40(4):379–395, 2007.

18 Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. *Theory Comput.*, 16:1–71, 2020. doi:10.4086/toc.2020.v016a010.

19 Mark Bun and Justin Thaler. Approximate degree in classical and quantum computing. *Found. Trends Theor. Comput. Sci.*, 15(3-4):229–423, 2022.

20 Mark Bun and Nadezhda Voronova. Approximate degree lower bounds for oracle identification problems, 2023. arXiv:2303.03921.

21 Arkadev Chattopadhyay, Yuval Filmus, Sajin Koroth, Or Meir, and Toniann Pitassi. Query-to-communication lifting using low-discrepancy gadgets. *SIAM J. Comput.*, 50(1):171–210, 2021. doi:10.1137/19M1310153.

22 Arkadev Chattopadhyay, Michal Koucký, Bruno Loff, and Sagnik Mukhopadhyay. Composition and simulation theorems via pseudo-random properties. *Electron. Colloquium Comput. Complex.*, page 14, 2017. URL: https://eccc.weizmann.ac.il/report/2017/014, arXiv:TR17-014.

23 Andrew M. Childs, Andrew J. Landahl, and Pablo A. Parrilo. Quantum algorithms for the ordered search problem via semidefinite programming. *Phys. Rev. A*, 75:032335, March 2007. doi:10.1103/PhysRevA.75.032335.

24 Andrew M. Childs and Troy Lee. Optimal quantum adversary lower bounds for ordered search. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, volume 5125 of *Lecture Notes in Computer Science*, pages 869–880. Springer, 2008.

25 Richard Cleve, Kazuo Iwama, François Le Gall, Harumichi Nishimura, Seiichiro Tani, Junichi Teruyama, and Shigeru Yamashita. Reconstructing strings from substrings with quantum queries. In *Scandinavian Workshop on Algorithm Theory*, pages 388–397. Springer, 2012.

26 E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation for insertion into an ordered list, 1998. doi:10.48550/ARXIV.QUANT-PH/9812057.

27 Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Invariant quantum algorithms for insertion into an ordered list, 1999. doi:10.48550/ARXIV.QUANT-PH/9901059.

28 Peter Høyer, Troy Lee, and Robert Spalek. Negative weights make adversaries stronger. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 526–535. ACM, 2007.

29 Peter Hoyer, Troy Lee, and Robert Spalek. Negative weights make adversaries stronger. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 526–535, 2007.

30 Peter Høyer, Jan Neerbek, and Yaoyun Shi. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica*, 34(4):429–448, 2002.

31 William M. Hoza. Quantum communication-query tradeoffs. *CoRR*, abs/1703.07768, 2017. arXiv:1703.07768.

32 Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Junichi Teruyama. Quantum counterfeit coin problems. *Theor. Comput. Sci.*, 456:51–64, 2012. doi:10.1016/j.tcs.2012.05.039.

33 Hartmut Klauck, Robert Spalek, and Ronald de Wolf. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. *SIAM J. Comput.*, 36(5):1472–1493, 2007. doi:10.1137/05063235X.

34 Robin Kothari. An optimal quantum algorithm for the oracle identification problem. In Ernst W. Mayr and Natacha Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, volume 25 of *LIPIcs*, pages 482–493. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014.

35 Ashley Montanaro and Changpeng Shao. Quantum algorithms for learning graphs and beyond. *CoRR*, abs/2011.08611, 2020. arXiv:2011.08611.

36 Ilan Newman. Private vs. common random bits in communication complexity. *Inf. Process. Lett.*, 39(2):67–71, 1991. doi:10.1016/0020-0190(91)90157-D.

37 Noam Nisan. The communication complexity of threshold gates. In *Combinatorics, Paul Erdös is Eighty, number 1 in Bolyai Society Mathematical Studies*, pages 301–315, 1993.

38 Ben Reichardt. Reflections for quantum query algorithms. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 560–569. SIAM, 2011. doi:10.1137/1.9781611973082.44.

39 Alexander A. Sherstov. The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011. doi:10.1137/080733644.

40 Alexander A Sherstov. Making polynomials robust to noise. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 747–758, 2012.

41 Alexander A. Sherstov. Strong direct product theorems for quantum communication and query complexity. *SIAM J. Comput.*, 41(5):1122–1165, 2012.

42 Alexander A. Sherstov. Algorithmic polynomials. *SIAM J. Comput.*, 49(6):1173–1231, 2020. doi:10.1137/19M1278831.

43 Steven Skiena and Gopalakrishnan Sundaram. Reconstructing strings from substrings. *J. Comput. Biol.*, 2(2):333–353, 1995. doi:10.1089/cmb.1995.2.333.

44 Wim van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 362–367. IEEE Computer Society, 1998.

On the Necessity of Collapsing for Post-Quantum and Quantum Commitments

Marcel Dall'Agnol
University of Warwick, Coventry, UK

Nicholas Spooner
University of Warwick, Coventry, UK

Abstract

Collapse binding and collapsing were proposed by Unruh (Eurocrypt '16) as post-quantum strengthenings of computational binding and collision resistance, respectively. These notions have been very successful in facilitating the "lifting" of classical security proofs to the quantum setting. A basic and natural question remains unanswered, however: are they the *weakest* notions that suffice for such lifting?

In this work we answer this question in the affirmative by giving a classical commit-and-open protocol which is post-quantum secure if and only if the commitment scheme (resp. hash function) used is collapse binding (resp. collapsing). We also generalise the definition of collapse binding to *quantum commitment schemes*, and prove that the equivalence carries over when the sender in this commit-and-open protocol communicates quantum information.

As a consequence, we establish that a variety of "weak" binding notions (sum binding, CDMS binding and unequivocality) are in fact *equivalent* to collapse binding, both for post-quantum and quantum commitments.

Finally, we prove a "win-win" result, showing that a post-quantum computationally binding commitment scheme that is not collapse binding can be used to build an equivocal commitment scheme (which can, in turn, be used to build one-shot signatures and other useful quantum primitives). This strengthens a result due to Zhandry (Eurocrypt '19) showing that the same object yields quantum lightning.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography; Theory of computation → Quantum computation theory

Keywords and phrases Quantum cryptography, Commitment schemes, Hash functions, Quantum rewinding

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.2

Related Version *Full Version*: `https://eprint.iacr.org/2022/786`

1 Introduction

The advent of quantum computing has led to a deep reevaluation of central ideas in cryptography. Most prominently, the hardness assumptions upon which many widely-used cryptographic schemes are based do not hold with respect to quantum computation. The past two decades have seen a great deal of progress in tackling this issue, by devising new schemes based on *post-quantum* assumptions.

This is, however, only part of the picture. Quantum computation is not simply more powerful than classical, it is *fundamentally different* in nature. Quantum information exhibits properties like superposition and unclonability that have no classical analogue. As such, we must also revisit another key ingredient in the study of cryptography: definitions. A number of works explore the implications of quantum information for security definitions; some examples include random oracles [6], message authentication codes [7, 17], as well as signatures and CCA-secure encryption [8].

This work studies the notion of *computational binding* (and the related notion of *collision resistance*) against quantum adversaries. While a natural quantum analogue of computational binding asserts that it is infeasible for a quantum computer to furnish valid openings of a commitment to more than one message, [1] demonstrated that this definition is not sufficient for many applications of commitment schemes. The key issue is that while binding rules out finding openings to distinct messages *simultaneously*, it does not rule out being able to "choose" the message that is opened. Note that this is an exclusively quantum problem: a classical algorithm able to make such a choice can break computational binding via rewinding.

Unruh [24] proposed post-quantum strengthenings of computational binding and collision resistance (for classical protocols) called *collapse binding* and *collapsing*, respectively. These have since become central in post-quantum cryptography: a sequence of works [24, 20, 3, 10, 11, 21] has demonstrated that this strengthening is sufficient to prove post-quantum security for various important schemes. Roughly speaking, these properties state that an adversary that has committed to a superposition of messages cannot tell whether or not that superposition has been measured.

Collapsing hash functions can be built from LWE [23]; additionally, any CRH that satisfies a certain regularity property is collapsing, which includes constructions from LPN and isogenies, and plausibly functions like SHA [30, 9]. Nonetheless, in general there remains a gap between collapsing and collision resistance. Zhandry [28, 29] showed that the existence of a hash function in this gap implies the existence of *quantum lightning*, which (among other things) yields public-key quantum money.

Quantum commitments

So far we have restricted our attention to the security of classical schemes against quantum adversaries (*post-quantum* security). Complicating matters further, however, quantum *communication* enables the construction of "intrinsically quantum" cryptographic constructions for which classical notions of security may not even apply. In *quantum commitment schemes*, where commitments and openings are (possibly entangled) quantum states, the basic notion of computational binding does not have a clear analogue; indeed, finding an appropriate definition of binding for quantum commitments has proved difficult [14, 27, 15, 4, 5], even in the statistical case, owing to an adversary's ability to commit to a superposition of messages.

2 Results

In this work we investigate collapse binding and related properties. We first propose a definition of collapse binding for quantum commitments (formalised in Definition 20). Then, using chosen-bit binding as a bridge, we show that collapse binding is equivalent to CBB (Theorems 2 and 4) and sum binding (Corollary 5), among others, both for post-quantum and quantum commitments.

Lastly, we use quantum rewinding techniques to show that, if computational and collapse binding are distinct, then a commitment scheme in this gap can be used to construct a *one-shot equivocal* scheme and, consequently, a variety of useful quantum cryptographic primitives (see Section 6).

▶ Remark 1 (Quantum vs. post-quantum results). For clarity, in this section we discuss the post-quantum versions of our experiments and results. We stress, however, that our proofs hold with respect to both quantum and classical (i.e., post-quantum) versions of the experiments.

(Note that, as the standard definition of quantum commitment schemes does not include post-quantum as a special case, this is not trivial; see Section 2.2 for a discussion.) ⌟

2.1 Chosen-bit binding commitments

We introduce a new notion of binding we call *chosen-bit binding*, which is defined in terms of an interactive game against a (potentially quantum) adversary Adv.

Let $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Commit})$ be a commitment scheme for the set of messages $M = \{0,1\}^{\ell(\lambda)}$. The chosen-bit binding experiment is as follows. (See Experiment 25 for the general version.)

1. Sample a commitment key $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$.
2. Obtain an index-commitment pair $(i, \mathsf{com}) \leftarrow \mathsf{Adv}(\mathsf{ck})$.
3. Sample $b \leftarrow \{0,1\}$ uniformly at random.
4. Obtain a message-opening pair $(m, \omega) \leftarrow \mathsf{Adv}(b)$.
5. Output 1 if $m_i = b$ and $\mathsf{Commit}(\mathsf{ck}, m, \omega) = \mathsf{com}$.

We say that COM is chosen-bit binding (CBB) if, for every efficient adversary Adv, the above experiment outputs 1 with probability at most $1/2 + \mathsf{negl}(\lambda)$. Note that the definition of CBB is agnostic to the actual form of the commitment, which is used only as an abstract functionality. It therefore readily applies to both classical and quantum commitments, as well as to schemes where the commit or reveal phases are interactive[1] (or even to "physical" commitments like a locked safe).

Note, also, that CBB is equivalent to requiring that COM be a *sum-binding bit commitment at every coordinate* $i \in [\ell]$ (which is distinct from Definition 18, the natural generalisation of sum binding to message spaces with size larger than 2); the CBB experiment concisely captures all ℓ sum binding experiments into one.

It is straightforward to show, via rewinding, that classical CBB is equivalent to computational binding. Our first result is an equivalence between CBB against quantum adversaries and collapsing.

▶ **Theorem 2.** *A classical commitment scheme is collapse binding if and only if it is post-quantum chosen-bit binding.*

Our results establish that collapsing is a "minimal" assumption which allows one to prove post-quantum security for the important class of *commit-and-open* sigma protocols (3-message protocols where the prover initiates, consisting of (1) commitments to s strings; (2) a challenge $C \subseteq [s]$; and (3) for each $i \in C$, an opening of the i^{th} string). Indeed, it was shown in [21] that any classically secure commit-and-open protocol is post-quantum secure when instantiated with a collapse binding commitment. Our result yields a converse:

▶ **Corollary 3.** *There exists a classical commit-and-open protocol which is insecure when instantiated with a commitment that is not collapse binding.*

We note, however, that Theorem 2 follows from a more general result: since Definition 20 captures collapse binding of commitments with either classical or quantum messages, we prove the equivalence between collapse and chosen-bit binding for a generalisation that captures both quantum and post-quantum schemes (Definition 14; see also Remark 16).

[1] In this work we restrict our attention to noninteractive commitments. All of our results easily generalise to the setting where the commit phase is interactive. However, the definition of collapse binding seems to crucially rely on the *reveal* phase being noninteractive.

▶ **Theorem 4.** *A quantum commitment scheme is collapse binding if and only if it is chosen-bit binding.*

Several works [15, 25, 5] aim to surmount the difficulties of basing cryptographic protocols on the binding guarantees of quantum commitments, especially for computational security. We hope that introducing a notion of collapse binding for quantum commitments will allow for some of the successes in the post-quantum case to be carried over to the quantum setting.

2.2 Connections to existing notions

▶ **Corollary 5.** *Sum binding is equivalent to collapse binding for quantum and post-quantum bit commitments.*

This corollary improves upon and generalises results from prior work. In the classical (post-quantum) setting, Unruh [23] proves that collapse binding implies sum binding; one of the main contributions of this paper is proving the converse.[2]

In the quantum setting, Yan [26, Appendix F] shows that for parallel repetitions of "canonical" quantum bit commitments (which capture the one-bit case of the schemes in Experiments 17 and 25), sum binding implies collapse binding – though that work does not give a definition of the latter.[3] Definition 20 is the natural extension of collapse binding to quantum commitments (which does not appear in prior work), and enables us to generalise Yan's result to arbitrary string commitments; note that these include *compressing* commitments, which implies an analogous equivalence for hash functions (see Section 2.2.2).

For general ℓ, (classical) chosen-bit binding is a special case of so-called "CDMS binding" [12, 24]. Informally, a commitment is CDMS binding with respect to a function class F if for every $f\colon X \to Y$ in F and every efficient adversary Adv,

$$\Pr_y[\mathsf{Adv}(y) \text{ opens } \mathsf{com} \text{ to } m \text{ s.t. } f(m) = y] \leq \frac{1}{|Y|} + \mathsf{negl}(\lambda) \ ,$$

where com is a fixed commitment previously output by Adv and y is chosen uniformly at random from Y. Unruh [23] showed that collapsing implies CDMS binding for all function classes where $|Y|$ is polynomial. CBB is easily seen to be equivalent to CDMS binding when F is the class of one-bit projection functions; we hence obtain the following corollary.

▶ **Corollary 6.** *CDMS binding against quantum adversaries is equivalent to collapse binding.*

It also follows that CDMS binding for one-bit projections implies CDMS binding for all function classes with polynomial range.

---

[2] We note that the following seemingly simpler strategy towards Theorem 2 does not suffice: (i) prove sum binding implies collapse binding for bit commitments; then (ii) use Unruh's parallel repetition theorem [23] to "lift" the equivalence to string commitments. This strategy only works for parallel repetitions of bit commitments, whereas Theorem 2 holds for any string commitment (and extends to hash functions).

[3] In fact, [26] shows that for canonical quantum commitments, (i) *honest* binding (a seemingly weaker notion) is equivalent to sum binding; and (ii) honest binding implies a "computational collapse" property that is equivalent to collapse binding. This result relies on the particular structure of canonical quantum bit commitments.

2.2.1 Somewhere statistical binding and parallel repetition

Unlike collapse binding, which is defined in terms of a quantum interaction, chosen-bit binding is defined in terms of a classical interaction with a (potentially quantum) adversary. This enables "fully classical" proofs that previously required quantum machinery, as we demonstrate next.

We use the chosen-bit binding definition to reprove two known results: the (folklore) fact that somewhere statistically binding (SSB) commitment schemes are collapse binding; and the preservation of the collapse-binding property under parallel repetitions [24].

▶ **Lemma 7.** *Any somewhere-statistically binding commitment scheme is chosen-bit binding; in particular, post-quantum SSB commitment schemes are collapsing.*

▶ **Lemma 8.** *If a commitment scheme* COM *is chosen-bit binding, then is* k*-fold parallel repetition* COM^k *is also chosen-bit binding.*

2.2.2 Hash functions

While we shall only discuss commitment schemes in the body of the paper, for our purposes collision-resistant hash functions are binding (but not hiding) *classical* commitment schemes where the length of the randomness is zero; therefore, many of our results extend to CRHs *mutatis mutandis*.

More precisely, consider the analogous (classical) chosen-bit binding experiment for a family $\mathcal{H}_\lambda \subseteq \{0,1\}^{m(\lambda)} \to \{0,1\}^{n(\lambda)}$ of hash functions defined next.

1. Sample $h \leftarrow \mathcal{H}_\lambda$.
2. Obtain $(y, i) \leftarrow \mathsf{Adv}(h)$, where $y \in \{0,1\}^{n(\lambda)}$ and $i \in [m(\lambda)]$.
3. Choose $b \leftarrow \{0,1\}$ uniformly at random.
4. Obtain $x \leftarrow \mathsf{Adv}(b)$.
5. Output 1 if $h(x) = y$ and $x_i = b$.

We say that $\mathcal{H}$ is classically (resp. post-quantum) chosen-bit binding (CBB) if for every efficient classical (resp. quantum) adversary Adv, the above experiment outputs 1 with probability at most $1/2 + \mathsf{negl}(\lambda)$.

Classical chosen-bit binding for hash functions is easily seen to be equivalent to collision resistance, and, by an essentially identical argument to Theorem 2, we can show that post-quantum CBB is equivalent to collapsing.

▶ **Corollary 9.** *A hash family* $\mathcal{H}$ *is collapsing if and only if it is post-quantum chosen-bit binding.*

Note that CBB also implies a method by which a quantum falsifier can convince a classical party that a hash function is *not* collapsing.

2.3 Equivocality

A (classical) commitment scheme is *one-shot equivocal* [2] if it has an additional functionality Eq, the equivocator, which produces a commitment string com and then, given a message m, outputs a valid opening ω to it (with probability close to 1).[4] In other words, Eq generates a commitment com it can *equivocate* to any message of its choice (but only once, if the scheme is computationally binding).

[4] While [2] defines equivocality for hash functions, it easily extends to commitment schemes. Indeed, the functionality they require is that of a commitment, which suffices to ensure security of the cryptographic objects constructed in that work.

We observe first that what [2] call "unequivocality" – roughly, that achieving the above with any nontrivial advantage is computationally infeasible – implies chosen-bit binding, and hence collapsing. This resolves an open question of [2].

However, we are able to show something much stronger, in the spirit of the "win-win" results of [28, 29]. In particular, we show that if a commitment scheme is (almost everywhere) *not* collapse binding, then it is one-shot equivocal. Note that the latter is a much stronger property than the negation of unequivocality, since Eq must succeed with probability close to 1. More formally, we obtain the following.[5]

▶ **Theorem 10** (Theorem 41, informally stated)**.** *If* COM *is a post-quantum computationally but not sum-binding commitment scheme, it can be transformed into a one-shot equivocal scheme.*

Our proof uses recent quantum rewinding techniques [11] to amplify success probability. We remark that while [21, 11] build upon "Unruh's lemma" [22] – which shows that if a pair of projective measurements succeed with sufficiently high probability, then so does their sequential application – it is insufficient for our purposes.

We instead use an early quantum rewinding lemma [13], which ensures one-shot equivocality for any inverse-polynomial advantage against COM in the collapse binding experiment (Unruh's lemma would only apply assuming constant advantage).

3 Preliminaries

We denote by $\lambda \in \mathbb{N}$ the security parameter, and when we refer to probabilistic/quantum polynomial-time (PPT/QPT) algorithms, the time complexity is a polynomial in λ. We denote by $\mathsf{negl}(\lambda)$ any function asymptotically smaller than every inverse polynomial, i.e, that is $o(\lambda^{-c})$ for every $c \in \mathbb{N}$.

For $n \in \mathbb{N}$, we write $[n]$ to denote the set $\{1, \ldots, n\}$. For a set S, we write $i \leftarrow S$ to denote that i is sampled uniformly from S. When D is a distribution, its support is denoted $\mathrm{supp}(D)$ and $i \leftarrow D$ denotes that i is chosen according to D.

We make use of the following simple fact, a consequence of Markov's inequality, and the Chernoff bound.

▶ **Proposition 11.** *Let* X *be a random variable supported on* $[0, 1]$*. Then for all* $\alpha \geq 0$*,* $\Pr[X \geq \alpha] \geq E[X] - \alpha$*.*

▶ **Proposition 12** (Chernoff bound)**.** *Let* $X_1, \ldots, X_k$ *be independent Bernoulli random variables distributed as* X*. Then, for every* $\delta \in [0, 1]$*,*

$$\Pr\left[\frac{1}{k}\sum_{i=1}^{k} X_i \geq (1+\delta)\mathbb{E}[X]\right] \leq e^{-\frac{\delta^2 k\mathbb{E}[X]}{3}} \text{ and}$$

$$\Pr\left[\frac{1}{k}\sum_{i=1}^{k} X_i \leq (1-\delta)\mathbb{E}[X]\right] \leq e^{-\frac{\delta^2 k\mathbb{E}[X]}{2}}.$$

We also make use of the Cauchy-Schwarz inequality with respect to the Hilbert-Schmidt inner product.

[5] It is claimed in [2] that if COM is not unequivocal, its parallel repetition COM^k is equivocal for large enough k. This is in fact true, but their argument is flawed; see Remark 34 for a discussion.

▶ **Lemma 13** (Cauchy-Schwarz). *For any complex matrices A, B such that $A^\dagger B$ is defined,*

$$\left|\mathrm{Tr}\left(A^\dagger B\right)\right|^2 \leq \mathrm{Tr}\left(A^\dagger A\right) \cdot \mathrm{Tr}\left(B^\dagger B\right) \ .$$

We say a commitment scheme is *classical* when all of its communication is classical (but an adversary may be quantum); that is, we use classical commitments as a shorthand for classical-*message* commitments.

By the *k-fold parallel repetition* of an experiment/interactive protocol, we denote that which results from repeating it independently k times *with the same first message* (in our case, a commitment key ck); the output of the experiment is the conjunction of the outputs of each execution.

3.1 Quantum information

We recall the basics of quantum information. (Most of the following is taken almost verbatim from [11].) A (pure) *quantum state* is a vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$ with $\||\psi\rangle\| = 1$; in this work, $\mathcal{H}$ is finite-dimensional, and we use $|\mathbf{0}\rangle$ to refer to a fixed ("zero") state in $\mathcal{H}$. We denote by $\mathbf{S}(\mathcal{H})$ the space of Hermitian operators on $\mathcal{H}$. A *density matrix* is a positive semi-definite operator $\rho \in \mathbf{S}(\mathcal{H})$ with $\mathrm{Tr}(\rho) = 1$. A density matrix represents a probabilistic mixture of pure states (a mixed state); the density matrix corresponding to the pure state $|\psi\rangle$ is $|\psi\rangle\langle\psi|$. Typically we divide a Hilbert space into *registers*, e.g. $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2$, and we sometimes write $\mathcal{H} \setminus \mathcal{H}_2$ to denote $\mathcal{H}_1$; we also write $\rho^{\mathcal{H}_1}$ to specify that $\rho \in \mathbf{S}(\mathcal{H}_1)$.

A unitary operation is a complex square matrix U such that $UU^\dagger = \mathbf{I}$. The operation U transforms the pure state $|\psi\rangle$ to the pure state $U|\psi\rangle$, and the density matrix ρ to the density matrix $U\rho U^\dagger$.

A *projector* Π is a Hermitian operator ($\Pi^\dagger = \Pi$) such that $\Pi^2 = \Pi$. If a (unitary U or) projector Π in a Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2$ acts trivially (as the identity $\mathbf{I}$) in $\mathcal{H}_2$, we may write Π or $\Pi^{\mathcal{H}_1}$ to denote $\Pi \otimes \mathbf{I}^{\mathcal{H}_2}$. A collection of projectors $\mathsf{M} = (\Pi_i)_{i \in S}$ is a *projective measurement* when $\sum_{i \in S} \Pi_i = \mathbf{I}$, and a *submeasurement* when there exists a projector Π such that $\sum_{i \in S} \Pi_i = \mathbf{I} - \Pi$.

The application of M to a pure state $|\psi\rangle$ yields outcome $i \in S$ with probability $p_i = \|\Pi_i|\psi\rangle\|^2$; we denote sampling from this distribution by $i \leftarrow \mathsf{M}(\rho)$, and in this case the post-measurement state is $|\psi_i\rangle = \Pi_i|\psi\rangle / \sqrt{p_i}$. We also use $\sigma \leftarrow \mathsf{M}(\rho)$ to denote the mixture of post-measurement states $\Pi_i|\psi\rangle / \sqrt{p_i}$ with probability p_i. A two-outcome projective measurement is called a *binary projective measurement*, and is written as $\mathsf{M} = (\Pi, \mathbf{I} - \Pi)$, where Π is associated with the outcome 1, and $\mathbf{I} - \Pi$ with the outcome 0.

General (non-unitary) evolution of a quantum state can be represented via a *completely-positive trace-preserving (CPTP)* map $T \colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H}')$. We omit the precise definition of these maps in this work; we only use the facts that they are trace-preserving (i.e., $\mathrm{Tr}(T(\rho)) = \mathrm{Tr}(\rho)$ for every $\rho \in \mathbf{S}(\mathcal{H})$) and linear. For every CPTP map $T \colon \mathbf{S}(\mathcal{H}) \to \mathbf{S}(\mathcal{H})$ there exists a *unitary dilation* U that operates on an expanded Hilbert space $\mathcal{H} \otimes \mathcal{K}$, so that, with $\mathrm{Tr}_{\mathcal{K}}$ the partial trace operator that traces out $\mathcal{K}$, we have $T(\rho) = \mathrm{Tr}_{\mathcal{K}}(U(\rho \otimes |0\rangle\langle 0|^{\mathcal{K}})U^\dagger)$. This is not necessarily unique; however, if T is described as a circuit then there is a dilation U_T represented by a circuit of size $O(|T|)$.

4 Commitment schemes

In this section, we define commitment schemes and the different notions of binding that we shall use (except for CBB, whose definition we defer to Section 5). While most of what follows is not novel, to the best of our knowledge the notion of collapse binding has as yet only been defined and studied for *classical* commitments. Our definition generalises that put forth by [24] (and coincides with it in the classical case).

▶ **Definition 14.** *A* quantum commitment scheme COM *consists of a PPT algorithm* Gen*, a unitary QPT algorithm* Commit *acting on a* 4*-tuple of registers* $\mathcal{K} \otimes \mathcal{M} \otimes \mathcal{C} \otimes \mathcal{O}$*, and a "check" subregister* $\mathcal{S} \subseteq \mathcal{C} \otimes \mathcal{O}$*.*

Commit *uses the key register* $\mathcal{K}$ *and message register* $\mathcal{M}$ *as classical controls. The dimension of* $\mathcal{K}$ *is* $\left|\mathrm{supp}\left(\mathsf{Gen}(1^\lambda)\right)\right|$ *and* $\mathcal{M}$ *has* $\ell(\lambda)$ *qubits; its computational basis is labeled by elements of the* message spaces $\{M_\lambda\}_{\lambda\in\mathbb{N}}$*, where* $M = \{0,1\}^{\ell(\lambda)}$*.*

In addition, $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Commit}, \mathcal{S})$ *is a* bit commitment *if* $\ell = 1$*, i.e., if* $M_\lambda = \{0,1\}$ *for all* $\lambda \in \mathbb{N}$*.*

As the register $\mathcal{S}$ will be clear from context, we use $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Commit})$ as shorthand for $(\mathsf{Gen}, \mathsf{Commit}, \mathcal{S})$. Moreover, we denote by $\mathsf{Commit}_{\mathsf{ck},m}$ the unitary acting on $\mathcal{C} \otimes \mathcal{O}$ as $\mathsf{Commit}_{\mathsf{ck},m} |\psi\rangle = \mathsf{Commit} |\mathsf{ck}\rangle |m\rangle |\psi\rangle$.

▶ **Definition 15.** *A* classical commitment scheme $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Commit})$ *is a quantum commitment scheme where* Commit *is a PPT algorithm and* $\mathcal{S} = \mathcal{C}$*.*

We use function notation for classical commitments, i.e., $\mathsf{Commit}(\mathsf{ck}, m, \omega)$ is the function computed and inserted (by a bitwise XOR) into the commitment register $\mathcal{C}$.

▶ Remark 16. Our definition of quantum commitment schemes deviates slightly from those in the literature in order to generalise classical commitments. In prior work it is typically assumed that quantum commitments are generated *deterministically*, which is without loss of generality since any randomness can be "purified out". Then the challenger may measure both $\mathcal{C}$ and $\mathcal{O}$ in the last step to check that $\mathsf{Commit}^\dagger_{\mathsf{ck},m}$ indeed inverts the adversary's computation (i.e., the challenger checks the register $\mathcal{S} = \mathcal{C} \otimes \mathcal{O}$).

However, in classical commitments randomness is inherent and *only the* $\mathcal{C}$ *register* is "uncomputed": the challenger reads ω from $\mathcal{O}$ and checks that the contents of $\mathcal{C}$ coincide with $\mathsf{Commit}(\mathsf{ck}, m, \omega)$. This corresponds to applying $\mathsf{Commit}^\dagger_{\mathsf{ck},m}(\mathcal{C}, \mathcal{O})$ and only measuring $\mathcal{S} = \mathcal{C}$.

(Given this discussion, it is natural to ask whether, for quantum commitments, it suffices to measure only $\mathcal{C}$. We leave this question to future work.) ⌟

We now define two notions of binding (sum and collapse) that apply to both quantum and classical commitments. Recall that, in order to be non-trivial, commitment schemes typically also satisfy a notion of hiding, which we omit since it is not relevant to the current work.

▶ **Experiment 17** (Sum binding)**.** *Given an adversary* Adv*, define the experiment* $\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{sum}}(\lambda)$*, parametrised by* $\lambda \in \mathbb{N}$*, as follows.*

1. *Generate* $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$*.*
2. *Obtain the commitment register* $\mathcal{C} \leftarrow \mathsf{Adv}(\mathsf{ck})$*.*
3. *Sample a (classical) message* $m \leftarrow M$*.*
4. *Obtain the opening register* $\mathcal{O} \leftarrow \mathsf{Adv}(m)$*, apply* $\mathsf{Commit}^\dagger_{\mathsf{ck},m}(\mathcal{C}, \mathcal{O})$ *and measure* $\mathcal{S}$ *in the computational basis.*
5. *Output* 1 *if the measurement yields* $|\mathbf{0}\rangle$*.*

▶ **Definition 18.** *A quantum commitment scheme* COM *is sum binding if, for all non-uniform QPT adversaries* Adv *in Experiment 17,*

$$\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{sum}}(\lambda) = 1\right] \leq \frac{1}{|M|} \cdot \left(1 + \mathsf{negl}(\lambda)\right) \ .$$

When COM *is classical and* Adv *is PPT (resp. QPT), we say it is classically (resp. post-quantum) sum binding.*

Note that the definition of sum binding given by [24] refers only to bit commitments; the above is a natural generalisation to quantum commitments and larger message spaces (which seems, however, to be of limited use when M is of superpolynomial size).

We proceed to the definition of collapse binding for quantum commitments.

▶ **Experiment 19** (Collapse binding). *For an adversary* Adv*, define the experiment* $\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{coll}}(\lambda)$ *as follows.*

1. *Generate* $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$.
2. *Obtain the registers* $\mathcal{C} \otimes \mathcal{M} \otimes \mathcal{O} \leftarrow \mathsf{Adv}(\mathsf{ck})$.
3. *Sample* $b \leftarrow \{0,1\}$*. If* $b = 1$*, measure* $\mathcal{M}$ *in the computational basis.*
4. *Obtain* $b' \leftarrow \mathsf{Adv}(\mathcal{M} \otimes \mathcal{O})$.
5. *Output* 1 *if* $b = b'$.

We say that Adv is valid if, for all $\mathsf{ck} \in \operatorname{supp}\big(\mathsf{Gen}(1^\lambda)\big)$, the state ρ in $\mathcal{C}\otimes\mathcal{M}\otimes\mathcal{O} \leftarrow \mathsf{Adv}(\mathsf{ck})$ is a mixture of superpositions of valid commitments; that is, $\rho = \sum_i p_i \ket{\psi_i}\bra{\psi_i}$ where $\ket{\psi_i}$ has nonzero amplitude only on computational basis states $\ket{\mathsf{com}, m, \omega}$ in the image of the projector $\mathsf{Commit}_{\mathsf{ck},m}\ket{\mathbf{0}}\bra{\mathbf{0}}^{\mathcal{S}}\mathsf{Commit}^\dagger_{\mathsf{ck},m}$. (In the post-quantum case, this simplifies to $\ket{m,\omega}$ satisfying $\mathsf{Commit}(\mathsf{ck}, m, \omega) = \mathsf{com}$.)

▶ **Definition 20.** *A quantum commitment scheme* COM *is collapse binding if, for all* valid *non-uniform QPT adversaries* Adv *in Experiment 19,*[6]

$$\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{coll}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \ .$$

Note that the challenger does not return the register $\mathcal{C}$ to the adversary in Step 4 for the purpose of distinguishing; this is crucially used in the proof of Theorem 4, and would otherwise lead to an unsatisfiable generalisation of classical commitments: an adversary that sends $\sum_{m\in M}\ket{\mathsf{Commit}(\mathsf{ck}, m, \omega)}\ket{m}\ket{\omega}$ (normalised) and receives all three registers can detect a measurement with high probability by uncomputing Commit and using the binary measurement with projector $\ket{\psi}\bra{\psi}^{\mathcal{M}}$ where $\ket{\psi} = \sum_{m\in M}\ket{m}$.

4.1 Classical binding

We conclude this section with a discussion of notions of binding that we only apply to *classical* commitments (with possibly quantum adversaries).

▶ **Experiment 21** (Computational binding). *Given an adversary* Adv*, define* $\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{bind}}(\lambda)$ *as follows.*

1. *Generate* $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$.
2. *Obtain* $(m_0, \omega_0, m_1, \omega_1) \leftarrow \mathsf{Adv}(\mathsf{ck})$.
3. *Output* 1 *if* $m_0 \neq m_1$ *and* $\mathsf{Commit}(\mathsf{ck}, m_0, \omega_0) = \mathsf{Commit}(\mathsf{ck}, m_1, \omega_1)$.

▶ **Definition 22.** *A commitment scheme* COM *is classically (resp. post-quantum) computationally binding if for all PPT (resp. QPT) adversaries* Adv *in Experiment 21,*

$$\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{bind}}(\lambda) = 1\right] = \mathsf{negl}(\lambda) \ .$$

[6] Equivalently, we could drop the validity constraint by measuring the state obtained in Step 2 with the appropriate binary projective measurement and aborting unless the outcome is 1.

Somewhere statistical binding (SSB)

Finally, we recall the notion of somewhere statistical binding, introduced by [19] in the context of hash functions. Here we present the equivalent notion for commitments; note that this is different to the more sophisticated notion of SSB commitments given by [16].

▶ **Definition 23** (Somewhere statistical binding)**.** *Let ℓ be a polynomial in λ. A commitment scheme* $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Commit})$ *is said to be somewhere statistically binding (SSB) if:*

- *For all $i, j \in [\ell(\lambda)]$, the distributions* $\mathsf{Gen}(1^\lambda, i)$ *and* $\mathsf{Gen}(1^\lambda, j)$ *are computationally indistinguishable.*
- *For all $i \in [\ell(\lambda)]$ and all* $\mathsf{ck} \in \operatorname{supp}\big(\mathsf{Gen}(1^\lambda, i)\big)$*, if* $\mathsf{Commit}(\mathsf{ck}, m, \omega) = \mathsf{Commit}(\mathsf{ck}, m', \omega')$ *for some (m, ω, m', ω'), then $m_i = m'_i$.*

More precisely, computational indistinguishability of $\mathsf{Gen}(\cdot, i)$ and $\mathsf{Gen}(\cdot, j)$ is defined by the experiment defined next.

▶ **Experiment 24.** *Given a commitment scheme* COM*, define* $\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{ssb}}(\lambda)$ *as follows.*

1. *Sample $j \leftarrow [\ell(\lambda)]$ and generate* $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda, j)$*.*
2. *Obtain* $i \leftarrow \mathsf{Adv}(\mathsf{ck})$*.*
3. *Output* 1 *if* $i = j$*.*

Therefore, $(\mathsf{Gen}, \mathsf{Commit})$ is classically (resp. post-quantum) somewhere-statistically binding if for all non-uniform PPT (resp. QPT) adversaries Adv,

$$\Pr\Big[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{ssb}}(\lambda) = 1\Big] \leq \frac{1}{\ell} + \mathsf{negl}(\lambda) \ .$$

(And, in addition, commitment keys ck determine the i^{th} coordinate of messages that map to the same commitment string.)

5 Chosen-bit binding

We begin this section with the definition of our main conceptual tool: the notion of chosen-bit binding. We define this notion in generality, for quantum schemes (and, owing to Definition 15, for classical schemes as a special case). Recall that $\mathcal{S} \subseteq \mathcal{C} \otimes \mathcal{O}$ is the subregister checked in a quantum (de)commitment.

▶ **Experiment 25** (Chosen-bit binding)**.** *Given a commitment scheme* COM*, define* $\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)$ *as follows.*

1. *Sample* $\mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda)$*.*
2. *Obtain the index and commitment register pair* $(i, \mathcal{C}) \leftarrow \mathsf{Adv}(\mathsf{ck})$*.*[7]
3. *Sample* $b \leftarrow \{0, 1\}$*.*
4. *Obtain the message and opening register pair* $(m, \mathcal{O}) \leftarrow \mathsf{Adv}(b)$*.*
5. *Apply* $\mathsf{Commit}^\dagger_{\mathsf{ck}, m}$ *to $\mathcal{C} \otimes \mathcal{O}$ and measure $\mathcal{S}$ in the computational basis.*
6. *Output* 1 *if $m_i = b$ and the measurement outcome is* $|\mathbf{0}\rangle$*.*

▶ **Definition 26.** *A quantum commitment scheme is chosen-bit binding if, for all non-uniform QPT adversaries* Adv *in Experiment 25,*

$$\Pr\Big[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda) = 1\Big] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \ .$$

[7] Alternatively, $\mathsf{Adv}(\mathsf{ck})$ may output two quantum registers $(\mathcal{I}, \mathcal{C})$; then i is obtained by a computational basis measurement of $\mathcal{I}$. (An analogous observation holds for Step 4, with $(\mathcal{M}, \mathcal{O}) \leftarrow \mathsf{Adv}(b)$ and a measurement of $\mathcal{M}$.)

Note that, in the case of bit commitments (i.e., when $M = \{0,1\}$), this notion coincides with sum binding. Recall that, in the case of classical adversaries, we have:

▶ **Lemma 27.** *A (classical) commitment scheme is chosen-bit binding against classical adversaries if and only if it is computationally binding.*

(The proof of this lemma is straightforward and hence omitted.)

We now prove the first of our main results: an equivalence between chosen-bit binding and collapse binding. We will make extensive use of the following binary projective measurements associated with a quantum commitment scheme COM. With $\big(|m\rangle\big)_{m\in M}$ and $\big(|\omega\rangle\big)_{\omega}$ as bases for the registers $\mathcal{M}$ and $\mathcal{O}$, respectively, we define:
$\mathsf{M}_{\mathsf{ck},m} \coloneqq (\Pi_{\mathsf{ck},m}, \mathbf{I} - \Pi_{\mathsf{ck},m})$ by

$$\Pi_{\mathsf{ck},m} \coloneqq \mathsf{Commit}_{\mathsf{ck},m}\left(|\mathbf{0}\rangle\langle\mathbf{0}|^{\mathcal{S}} \otimes \mathbf{I}^{(\mathcal{C}\otimes\mathcal{O})\setminus\mathcal{S}}\right)\mathsf{Commit}^{\dagger}_{\mathsf{ck},m} \ ; \tag{1}$$

$\mathsf{M}_{\mathsf{ck}} \coloneqq (\Pi_{\mathsf{ck}}, \mathbf{I} - \Pi_{\mathsf{ck}})$ by

$$\Pi_{\mathsf{ck}} \coloneqq \sum_{m\in M} |m\rangle\langle m|^{\mathcal{M}} \otimes \Pi_{\mathsf{ck},m} \ ; \tag{2}$$

$\mathsf{M}_{i,b} \coloneqq (\Pi_{i,b}, \mathbf{I} - \Pi_{i,b})$ by

$$\Pi_{i,b} \coloneqq \sum_{m,m_i=b} |m\rangle\langle m|^{\mathcal{M}} \otimes \Pi_{\mathsf{ck},m} \ ; \text{ and} \tag{3}$$

$\mathsf{M}_{i} \coloneqq (\Pi_{i}, \mathbf{I} - \Pi_{i})$ by

$$\Pi_{i} \coloneqq \sum_{b\in\{0,1\}} |b\rangle\langle b|^{\mathcal{B}} \otimes \Pi_{i,b} \ . \tag{4}$$

Note that $\Pi_{\mathsf{ck},m}$ (Equation 2) projects onto the subspace of valid commitment-opening register pairs, and the other measurements do so with additional restrictions: $\Pi_{\mathsf{ck},m}$ (Equation 1) projects onto valid messages; $\Pi_{i,b}$ (Equation 3) projects onto (valid) messages with $m_i = b$; and Π_i (Equation 4) onto messages whose i^{th} coordinate overlaps with the contents of $\mathcal{B}$.

▶ **Theorem 28** (Theorem 4, restated)**.** *A quantum commitment scheme* COM *is collapse binding if and only if it is chosen-bit binding.*

We first prove (via the contrapositive) that collapse binding implies chosen-bit binding, which extends [23, Theorem 32] to quantum commitments.

Proof (collapsing ⇒ CBB). Let Adv be an adversary that achieves advantage ε in Experiment 25 (the chosen-bit binding experiment). We may assume, without loss of generality, that the adversary's action in Step 4 consists of the application of a unitary U on $\mathcal{B}\otimes\mathcal{M}\otimes\mathcal{O}\otimes\mathcal{H}$ (where $\mathcal{B}$ contains the bit received from the challenger and $\mathcal{H}$ is an additional workspace register) followed by a computational basis measurement of $\mathcal{M}$. We construct an adversary Adv' for the collapse binding experiment as follows.

- Upon receipt of ck:
  1. Run $\mathsf{Adv}(\mathsf{ck})$ to obtain $i \in [\ell]$ and state ρ on $\mathcal{C}\otimes\mathcal{M}\otimes\mathcal{O}\otimes\mathcal{H}$.
  2. Apply $U \otimes \mathbf{I}^{\mathcal{C}}$ to $\sigma = |+\rangle\langle+|^{\mathcal{B}} \otimes \rho$ followed by the binary projective measurement M_i.
  3. If the measurement outcome is 0, overwrite $\mathcal{C}\otimes\mathcal{M}\otimes\mathcal{O}$ with a valid commitment (to, say, the all-zero string). Output $i \in [\ell]$ along with the registers $\mathcal{C}$, $\mathcal{M}$ and $\mathcal{O}$.

- Upon receipt of $\mathcal{M}$, $\mathcal{O}$:
  1. If the measurement outcome in the previous step was 0, stop and output a random bit.
  2. Apply $U^\dagger$ to $\mathcal{B} \otimes \mathcal{M} \otimes \mathcal{O} \otimes \mathcal{H}$ and measure $\mathcal{B}$ in the $\{|+\rangle, |-\rangle\}$ basis.
  3. If the outcome is $|+\rangle$, output 0; otherwise output 1.

Note that Adv' is valid, as $\Pi_{\mathsf{ck}} = \Pi_{i,0} + \Pi_{i,1}$ (by Equations 2 and 3) and Equation 4 implies $\mathrm{Tr}_{\mathcal{B},\mathcal{H}}(\Pi_i \sigma \Pi_i) \in \mathrm{Im}(\Pi_{i,0} + \Pi_{i,1})$. Moreover, if either (i) the challenger measures or (ii) the outcome of the first measurement by Adv' is 0, the experiment outputs a uniformly random bit.

For the case where the challenger does not measure, we use the following proposition:

▶ **Proposition 29.** *Let P, Q be projectors and ρ a density matrix such that $\rho Q = \rho$. Then*

$$\mathrm{Tr}(QP\rho P) \geq \mathrm{Tr}(P\rho)^2 \ .$$

Proof. $\mathrm{Tr}(P\rho) = \mathrm{Tr}(P\rho Q) \leq \sqrt{\mathrm{Tr}(\overline{QP\rho PQ})}$, by Cauchy-Schwarz (Lemma 13). ◀

Assume that $b = 0$ in Step 3 of Experiment 19, so $\mathcal{M}$ is not measured (we deal with the case $b = 1$ next). We lower bound the probability that the measurement outcomes of $\mathsf{Adv}'(\mathsf{ck})$ and $\mathsf{Adv}'(\mathcal{M}, \mathcal{O})$ are 1 and $|+\rangle$, respectively, whereupon the experiment outputs 1: since $\sigma \cdot |+\rangle\langle+| = \sigma$, by Proposition 29,

$$\begin{aligned}
\mathrm{Tr}\left(|+\rangle\langle+| \, \Pi_i \sigma \Pi_i\right) &\geq \mathrm{Tr}(\Pi_i \sigma)^2 \\
&= \left(\frac{1}{2}\mathrm{Tr}\left(\Pi_{i,0}\rho\right) + \frac{1}{2}\mathrm{Tr}\left(\Pi_{i,1}\rho\right)\right)^2 \\
&= \left(\frac{1}{2} + \varepsilon\right)^2 .
\end{aligned}$$

Now note that, if $b = 1$ in Step 3, the $\mathcal{M}$ register is measured and $\mathcal{B}$ collapses to a computational basis state, namely, $|m_i\rangle$ when the outcome is m; since the adversary measures $\mathcal{B}$ in the Hadamard basis, the experiment outputs 1 with (conditional) probability $1/2$ in this event. Moreover, if the adversary's first measurement outcome is 0 (an event with $1 - \mathrm{Tr}(\Pi_i \sigma)$ probability) it outputs a uniformly random bit; in this case, Experiment 19 also outputs 1 with probability $1/2$.

Overall, the probability that the experiment outputs 1 is thus

$$\begin{aligned}
&\frac{1}{4} + \frac{1}{2}\left(\mathrm{Tr}\left(|+\rangle\langle+| \, \Pi_i \sigma \Pi_i\right) + \frac{1}{2}\big(1 - \mathrm{Tr}(\Pi_i \sigma)\big)\right) \\
&\quad = \frac{1}{4} + \frac{1}{2}\left(\mathrm{Tr}\left(|+\rangle\langle+| \, \Pi_i \sigma \Pi_i\right) + \frac{1}{2}\left(\frac{1}{2} - \varepsilon\right)\right) \\
&\quad \geq \frac{1}{4} + \frac{1}{2}\left(\left(\frac{1}{2} + \varepsilon\right)^2 + \frac{1}{2}\left(\frac{1}{2} - \varepsilon\right)\right) \\
&\quad \geq \frac{1}{2} + \frac{\varepsilon}{2} \ .
\end{aligned}$$

◀

Before proving the reverse implication, we show a basic fact about non-commuting projective measurements. Let M be a projective measurement and $\mathsf{B} = (D, \mathbf{I} - D)$ a binary projective measurement. Consider the following experiment applied to a state ρ:

1. Measure $i \leftarrow \mathsf{M}$.
2. Apply B (and ignore the result).
3. Measure $j \leftarrow \mathsf{M}$.

The following claim gives a lower bound on the probability that $i \neq j$ in terms of how well B distinguishes ρ from $\mathsf{M}(\rho)$ (which is a measure of how "non-commuting" B and M are). Variants of this claim have appeared independently and concurrently in [30, 9].

▷ Claim 30. Let D be a projector, $\mathsf{M} = (\Pi_i)_{i\in[N]}$ be a projective submeasurement and ρ be a Hermitian matrix such that $\sum_i \mathrm{Tr}(\Pi_i\rho) = \mathrm{Tr}(\rho)$. Then

$$\sum_j \sum_{i\neq j} \mathrm{Tr}(\Pi_i D \Pi_j \rho \Pi_j D) \geq \frac{\mathrm{Tr}\left(D(\rho - \mathsf{M}(\rho))\right)^2}{N \cdot \mathrm{Tr}(\rho)} \ .$$

Proof. Inserting resolutions of the identity, and since $(\mathbf{I} - \sum_i \Pi_i)\rho = 0$,

$$\begin{aligned}\mathrm{Tr}(D\rho) &= \sum_i \mathrm{Tr}(D\Pi_i\rho\Pi_i) + \sum_{i\neq j} \mathrm{Tr}(\Pi_i D \Pi_j \rho) \\ &= \mathrm{Tr}(D\mathsf{M}(\rho)) + \sum_j \mathrm{Tr}(\Pi_{\neq j} D \Pi_j \rho) \ ,\end{aligned}$$

where $\Pi_{\neq j} \coloneqq \sum_{i\neq j} \Pi_i$. Applying Cauchy-Schwarz (Lemma 13, with $A = \sqrt{\rho}\cdot\Pi_j D \Pi_{\neq j}$ and $B = \sqrt{\rho}$) yields $|\mathrm{Tr}(\Pi_{\neq j} D \Pi_j \rho)| \leq \sqrt{\mathrm{Tr}(\Pi_{\neq j} D \Pi_j \rho \Pi_j D)}\sqrt{\mathrm{Tr}(\rho)}$. Substituting into the above equation and squaring we have

$$\frac{\mathrm{Tr}\left(D(\rho - \mathsf{M}(\rho))\right)^2}{\mathrm{Tr}(\rho)} \leq \left(\sum_j \sqrt{\mathrm{Tr}(\Pi_{\neq j} D \Pi_j \rho \Pi_j D)}\right)^2 \ ,$$

and applying Cauchy-Schwarz again (with respect to Euclidean norm and the N-dimensional pair of vectors with 1 and $\sqrt{\mathrm{Tr}(\Pi_{\neq j} D \Pi_j \rho \Pi_j D)}$ in the j^{th} coordinate, respectively) yields the claim. ◁

We now prove the reverse implication.

Proof (CBB ⇒ collapsing). Let Adv be an adversary that achieves ε collapsing advantage. We design an adversary Adv' for the chosen-bit binding experiment as follows.

- Upon receipt of ck:
  1. Run $\mathsf{Adv}(\mathsf{ck})$ obtain a quantum state ρ in $\mathcal{C}\otimes\mathcal{M}\otimes\mathcal{O}\otimes\mathcal{H}$.
  2. Output a random index $i \leftarrow [\ell]$ and $\mathcal{C}$.
- Upon receipt of b:
  1. Measure the first i bits of $\mathcal{M}$, obtaining outcomes $b_1, \ldots, b_i$.
  2. If $b_i \neq b$, apply Adv's (projective) distinguishing measurement $(D, \mathbf{I}-D)$ to $\mathcal{M}\otimes\mathcal{O}\otimes\mathcal{H}$.[8]
  3. Measure $\mathcal{M}$ in the computational basis. Output the outcome m and the opening register $\mathcal{O}$.[9]

Let $\mathsf{M}_j(\rho)$ be the map corresponding to measuring the j^{th} qubit of $\mathcal{M}$, i.e.,

$$\mathsf{M}_j(\rho) = \Pi_i\rho\Pi_i + (\mathbf{I} - \Pi_i)\rho(\mathbf{I} - \Pi_i).$$

[8] Here we use that D acts trivially on $\mathcal{C}$.
[9] Note that in the case of classical commitments, $\mathcal{O}$ is a classical register containing an opening string ω; equivalently, we may assume $\mathcal{O}$ is implicitly measured.

Let $\mathsf{M}_{[j]} \coloneqq \mathsf{M}_1(\cdots \mathsf{M}_{j-1}(\mathsf{M}_j(\rho))\cdots)$ be the map corresponding to measuring the *first* j qubits of $\mathcal{M}$, where $\mathsf{M}_{[0]}$ is the identity map. We have that

$$\rho - \mathsf{M}_{[\ell]}(\rho) = \sum_{j=0}^{\ell-1} \mathsf{M}_{[j]}(\rho) - \mathsf{M}_{[j+1]}(\rho) = \sum_{j=0}^{n-1} \rho_j - \mathsf{M}_{j+1}(\rho_j)$$

where $\rho_j \coloneqq \mathsf{M}_{[j]}(\rho)$.

The adversary's success probability γ in Experiment 25 can be written as

$$\frac{1}{2\ell} \sum_{i\in[\ell]} \sum_{b\in\{0,1\}} \operatorname{Tr}(\Pi_{i,b}\rho_{i-1}) + \operatorname{Tr}(\Pi_{i,b} D \Pi_{i,1-b}\rho_{i-1}\Pi_{i,1-b} D).$$

Note that the validity of Adv ensures ρ_{i-1} is in the span of Π_{ck}, which simplifies the first term of the sum: $\sum_{i\in[\ell]}\sum_{b\in\{0,1\}} \operatorname{Tr}(\Pi_{i,b}\rho_{i-1}) = \sum_{i\in[\ell]} \operatorname{Tr}(\rho_{i-1}) = \ell$. It also enables us to apply Claim 30 with respect to the submeasurement $(\Pi_{i,0}, \Pi_{i,1})$; using the claim and Cauchy-Schwarz (Lemma 13), we obtain that

$$\begin{aligned}
\gamma &\geq \frac{1}{2} + \frac{1}{4\ell} \sum_{i\in[\ell]} \operatorname{Tr}(D(\rho_i - \mathsf{M}_{i+1}(\rho_i)))^2 \\
&\geq \frac{1}{2} + \frac{1}{4\ell^2} \left(\sum_{i\in[\ell]} \operatorname{Tr}(D(\rho_i - \mathsf{M}_{i+1}(\rho_i))) \right)^2 \\
&= \frac{1}{2} + \frac{1}{4\ell^2} \big(\operatorname{Tr}\big(D(\rho - \mathsf{M}_{[\ell]}(\rho))\big)\big)^2 \\
&= \frac{1}{2} + \Big(\frac{\varepsilon}{2\ell}\Big)^2
\end{aligned}$$

where the final equality follows by assumption on Adv. This completes the proof. ◀

5.1 Somewhere statistical binding and parallel repetitions

Using chosen-bit binding, we give "fully classical" proofs that somewhere-statistical binding commitments are collapse binding, and that the parallel repetition of collapse binding commitments are collapse binding.

▶ **Lemma 31.** *Post-quantum somewhere statistically binding commitment schemes are chosen-bit binding against quantum adversaries, and therefore collapse binding.*

Proof. Let Adv be an adversary satisfying $\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda) = 1\right] = 1/2 + \varepsilon$.

We construct an adversary $\mathsf{Adv}'(\mathsf{ck})$ for Experiment 24 (SSB) as follows: simulate Experiment 25 (CBB) with the key ck, obtaining $(\mathsf{com}, i, b, m, \omega)$. (Recall that Experiment 24 is classical, so Adv outputs strings com and ω.) If $m_i \neq b$ or $\mathsf{Commit}(\mathsf{ck}, m, \omega) \neq \mathsf{com}$ (i.e., if the adversary loses), output $k \leftarrow [\ell]$; otherwise, output $k \leftarrow [\ell] \setminus \{i\}$. We denote by j the uniformly sampled binding index (which determines $\mathsf{Gen}(1^\lambda, j)$ as the generator in the experiment).

The success probability of this adversary is

$$\Pr[k = j] = \frac{1}{\ell} \cdot \Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda) = 0\right] + \frac{1}{\ell - 1} \cdot \Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda) = 1 \wedge j \neq i\right] \quad . \tag{5}$$

Observe that the experiment outputs 1 with probability at most $1/2$ when conditioned on $j = i$ (since, by Definition 23, one of the choices for $b \in \{0, 1\}$ is such that no message-opening pair

(m,ω) with $\mathsf{Commit}(\mathsf{ck},m,\omega)=\mathsf{com}$ and $m_i=b$ exists); that is, $\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)=1 \;\middle|\; j=i\right]\leq 1/2$. Hence

$$\begin{aligned}\frac{1}{2}+\varepsilon &= \Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)=1 \;\middle|\; j=i\right]\Pr[j=i]+\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)=1\wedge j\neq i\right]\\ &\leq \frac{1}{2}\cdot\Pr[j=i]+\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)=1\wedge j\neq i\right]\quad .\end{aligned}$$

Note that, if $\Pr[j=i]\geq(1+\varepsilon)/\ell$ (infinitely often), the adversary that always outputs i has inverse polynomial advantage. We therefore assume otherwise; then

$$\frac{1}{2}+\varepsilon\leq\frac{1+\varepsilon}{2\ell}+\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)=1\wedge j\neq i\right]\quad,$$

and so $\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)=1\wedge j\neq i\right]\geq\frac{1}{2}(1-\frac{1}{\ell})+\varepsilon\cdot(1-\frac{1}{2\ell})$.

Substituting into (5) and using $\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)=0\right]=1/2-\varepsilon$ (by hypothesis) yields

$$\Pr[k=j]\geq\frac{1}{2\ell}+\frac{1-\frac{1}{\ell}}{2(\ell-1)}+\varepsilon\cdot\left(\frac{1-\frac{1}{2\ell}}{\ell-1}-\frac{1}{\ell}\right)=\frac{1}{\ell}+\frac{\varepsilon}{2\ell(\ell-1)}\quad,$$

which completes the proof. ◀

Observe that Theorem 28 implies that parallel repetitions preserve collapse binding if and only if they preserve chosen-bit binding. Then,

▶ **Proposition 32.** *If a quantum commitment scheme* $\mathsf{COM}=(\mathsf{Gen},\mathsf{Commit})$ *is chosen-bit binding, then its* k*-fold parallel repetition is also chosen-bit binding.*

Proof. Let Adv be an adversary satisfying $\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{cbb}}(\lambda)=1\right]=1/2+\varepsilon$ in the k-wise parallel repetition of Experiment 25. (Recall that the same key ck is used in each repetition; we index message bits by pairs $(i,j)\in[k]\times[\ell]$, so that m_{ij} is the j^{th} bit of the i^{th} message.)

Then an adversary $\mathsf{Adv}'(\mathsf{ck})$ for the original commitment scheme, with the same advantage, simply executes $\mathsf{Adv}(\mathsf{ck})$ to obtain an index (i,j) along with commit registers $\mathcal{C}_1\otimes\ldots\otimes\mathcal{C}_k$, and outputs $(j,\mathcal{C}_i)$; upon receipt of b, it obtains $(m_1,\ldots,m_k,\mathcal{O}_1\otimes\cdots\otimes\mathcal{O}_k)\leftarrow\mathsf{Adv}(b)$ and returns $(m_i,\mathcal{O}_i)$ in the last step.

Since $m_{ij}=(m_i)_j=b$ and applying $\mathsf{Commit}^\dagger_{\mathsf{ck},m_i}(\mathcal{C}_i,\mathcal{O}_i)$ followed by a measurement of $\mathcal{S}_i$ yields $|\mathbf{0}\rangle$ with probability at least $1/2+\varepsilon$ (because applying $\mathsf{Commit}^\dagger_{\mathsf{ck},m_1}\otimes\cdots\otimes\mathsf{Commit}^\dagger_{\mathsf{ck},m_k}$ to $(\mathcal{C}_1\otimes\mathcal{O}_1)\otimes\cdots\otimes(\mathcal{C}_k\otimes\mathcal{O}_k)$ and measuring $\mathcal{S}_1\otimes\cdots\otimes\mathcal{S}_k$ yields $|\mathbf{0}\rangle$ with probability $1/2+\varepsilon$), the result follows. ◀

6 Equivocality

Amos, Georgiou, Kiayias and Zhandry [2] define two closely related notions they call *equivocal* and *one-shot chameleon* collision-resistant hash functions, and show how they can be used to obtain a variety of interesting quantum cryptographic constructions. Here we consider a slight variant, which we call a *one-shot equivocal commitment scheme*. We note that an equivocal CRHF associated to a predicate p is a one-shot equivocal commitment to the bit $p(x)$ where x is the hash preimage.[10]

[10] While [2] distinguish between the notions of equivocal and one-shot chameleon hash functions (roughly speaking, equivocal hashes allow equivocation to *some* string under a predicate constraint, while one-shot chameleon hashes equivocate to *any* string), they also prove how to construct one from the other. We choose to only define the (syntactically) stronger property, which we call *one-shot equivocality* – both to distinguish it from classical notions of equivocality and to evince the connection to one-shot chameleon hashes.

▶ **Definition 33.** *A commitment scheme* $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Commit})$ *is* one-shot equivocal *if there exists a stateful QPT algorithm* Eq *such that for all messages* $m \in M$*,*

$$\Pr\left[\mathsf{Commit}(\mathsf{ck}, m, \omega) = \mathsf{com} \;\middle|\; \begin{array}{c} \mathsf{ck} \leftarrow \mathsf{Gen}(1^\lambda) \\ \mathsf{com} \leftarrow \mathsf{Eq}(\mathsf{ck}) \\ \omega \leftarrow \mathsf{Eq}(m) \end{array}\right] = 1 - \mathsf{negl}(\lambda) \ .$$

While this definition allows arbitrary message spaces, hereafter we focus on the case $M = \{0,1\}$. We also note that Definition 18 (sum binding) is identical to a "converse" notion to the above, which [2] define informally and call *unequivocality*.

▶ Remark 34. Despite what the terminology may suggest, we stress that (one-shot) equivocality and unequivocality (i.e., sum binding) are *not* the logical negation of one another: aside from the usual technical issues of infinitely-often vs. almost-everywhere, equivocality is syntactically much stronger than "non-unequivocality", as it requires a correct opening with all but negligible probability.

It is claimed in [2] that an adversary breaking unequivocality yields a one-shot equivocal commitment scheme as follows (we adapt their argument to our definitions). The new commitment is a parallel repetition of the original, where the committed bit is taken to be the *majority* of the underlying commitments. To equivocate, we ask the adversary to open each underlying commitment to the same bit b. The idea is that taking the majority amplifies the small bias that an adversary achieves. However, this argument has a significant flaw: what do we do when the adversary fails to equivocate on a particular commitment? In this case it may either produce an invalid opening, preventing us from opening the commitment altogether, or even consistently provide openings for $1 - b$, leading to a valid opening to the wrong bit!

Regardless, we show in Theorem 41 that the implication still holds: sum binding can be "boosted" to one-shot equivocality via quantum rewinding. ⌟

One-shot equivocal commitments only differ from equivocal hashes in their mildly weaker "collision-resistance", which does not prevent an adversary from efficiently finding distinct valid openings for the same message. However, we remark that the construction of one-shot signatures of [2] can be based on one-shot equivocal commitments rather than hashes without harm to their security: while an adversary may find distinct signatures for the same message, the resulting scheme still ensures it *cannot sign distinct messages*. (As a result, subsequent constructions that rely on one-shot signatures – quantum money and proofs of quantumness, among others – also satisfy this weakened but sufficient security guarantee.)

Nontrivial (i.e., computationally binding) one-shot equivocal string commitments can be obtained from one-shot equivocal bit commitments by the usual composition, which we prove next for completeness.

▶ **Proposition 35.** *If a bit commitment scheme* $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Commit})$ *is computationally binding and one-shot equivocal, then its* k*-fold parallel repetition is also computationally binding and one-shot equivocal when* $k = \mathrm{poly}(\lambda)$*.*

Proof. Computational binding follows from the fact that an adversary Adv in the parallel repetition of Experiment 21 achieving $\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{bind}}(\lambda)\right] = \varepsilon$ with message space $M = \{0,1\}^k$ immediately yields Adv' with advantage ε/k when $M = \{0,1\}$: Adv' samples $i \leftarrow [k]$, runs the (bit) experiment with the challenger on this coordinate and simulates the interaction for coordinates $j \neq i$. When $\varepsilon = \mathrm{poly}\left(\lambda^{-1}\right)$, the resulting advantage ε/k is also inverse polynomial.

If Eq with quantum auxiliary input ρ is the equivocator for COM, we define Eq' as the natural equivocator for the parallel repetition: $\mathsf{Eq}'(\mathsf{ck})$, with auxiliary input $\rho^{\otimes k}$, obtains from each copy of ρ a commitment string $\mathsf{com}_i \leftarrow \mathsf{Eq}(\mathsf{ck})$ and a post-measurement state ρ_i, then returns $(\mathsf{com}_1, \ldots, \mathsf{com}_k)$. Upon receipt of a message, $\mathsf{Eq}'(m)$ runs each $\mathsf{Eq}(m_i)$ on the state ρ_i, obtains ω_i and returns $(\omega_1, \ldots, \omega_k)$. Since $\mathsf{Commit}(\mathsf{ck}, m_i, \omega_i) = \mathsf{com}_i$ with probability $1 - \mathsf{negl}(\lambda)$ for each i, all k openings succeed except with probability $k \cdot \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)$. ◀

We will show via quantum rewinding techniques that a commitment scheme that is computationally but not sum binding is indeed one-shot equivocal. To this end, we first recall an early "basic quantum rewinding" lemma, first used in [13], which shows that when two different computations (on the same state) yield prescribed outcomes with sufficiently high probability, performing the computations sequentially obtains both outcomes with non-negligible probability. We state a slightly more general statement than [13] and prove it for completeness.

▶ **Lemma 36.** *For any projectors P, Q and quantum state ρ it holds that*

$$\operatorname{Tr}(PQP\rho) \geq \frac{1}{4}\big(\operatorname{Tr}(P\rho) + \operatorname{Tr}(Q\rho) - 1\big)^2 \ .$$

Proof. Let $\varepsilon \coloneqq \operatorname{Tr}(P\rho) + \operatorname{Tr}(Q\rho) - 1$. Then $\operatorname{Tr}((P+Q)\rho) = 1 + \varepsilon$ by assumption and linearity, and, by Cauchy-Schwarz,

$$\begin{aligned}(1+\varepsilon)^2 = \operatorname{Tr}\big((P+Q)\rho\big)^2 &\leq \operatorname{Tr}\big((P+Q)\rho(P+Q)\big) \\ &= \operatorname{Tr}(P\rho) + \operatorname{Tr}(Q\rho) + 2\operatorname{Re}\operatorname{Tr}(QP\rho) \ .\end{aligned}$$

It follows that $\operatorname{Re}\operatorname{Tr}(QP\rho) \geq \varepsilon/2$. Then, again by Cauchy-Schwarz (Lemma 13),

$$\varepsilon/2 \leq \operatorname{Re}\operatorname{Tr}(QP\rho) \leq |\operatorname{Tr}(QP\rho)| \leq \sqrt{\operatorname{Tr}(QP\rho PQ)} \ ,$$

which completes the proof. ◀

Next, we recall Jordan decompositions and two singular vector algorithms that we shall use in our construction.

▶ **Lemma 37** (Jordan decomposition)**.** *Any pair of projectors Π_A and Π_B induces a decomposition of the Hilbert space they act upon into $\oplus_i \mathcal{S}_i$ where each $\mathcal{S}_i$ has dimension* 1 *or* 2*.*

The projectors can be written as $\Pi_A = \sum_i |v_i\rangle\langle v_i|$ and $\Pi_B = \sum_i |w_i\rangle\langle w_i|$ for $\mathcal{S}_i$-bases $\big\{|v_i\rangle, |v_i^\perp\rangle\big\}$ and $\big\{|v_i\rangle, |v_i^\perp\rangle\big\}$; the sums range over all $\mathcal{S}_i$ except the one-dimensional ones where the projector acts trivially (as the zero projector).

We call the $\mathcal{S}_i$ *Jordan subspaces*, and define $p_i \coloneqq \big|\langle v_i|w_i|\rangle\big|^2 = \big|\langle v_i^\perp|w_i^\perp\rangle\big|^2$. We also define the *Jordan measurement* $\mathsf{M}^{\mathsf{Jor}} = \big(\Pi_i^{\mathsf{Jor}}\big)$ by

$$\Pi_i^{\mathsf{Jor}} \coloneqq |v_i\rangle\langle v_i| + |v_i^\perp\rangle\langle v_i^\perp| = |w_i\rangle\langle w_i| + |w_i^\perp\rangle\langle w_i^\perp|;$$

that is, $\mathsf{M}^{\mathsf{Jor}}$ projects onto a subspace $\mathcal{S}_i$ and outputs its index i.

The singular vector algorithms, due to [21, 18], allow us to effectively "filter out" components of a quantum state below a threshold of our choice and then "flip" the image of a projector to its complement if needed.

▶ **Lemma 38.** *Let Π_A, Π_B be projectors described by uniform* $\operatorname{poly}(\lambda)$*-size quantum circuits. Then there exists a (uniform) family $\{\mathsf{Threshold}_\theta\}_{\theta\in(0,1]}$ of algorithms described by* $\operatorname{poly}(\lambda)$*-size circuits that satisfy the following:*

- *if* $p_i \geq \theta$, $\mathsf{Threshold}_\theta(|v_i\rangle)$ *outputs* 1 *with probability* $1 - \mathsf{negl}(\lambda)$.
- *if* $p_i \leq \theta/2$, $\mathsf{Threshold}_\theta(|v_i\rangle)$ *outputs* 1 *with probability* $\mathsf{negl}(\lambda)$.

Moreover, $\mathcal{S}_i$ *is invariant under* $\mathsf{Threshold}_\theta$ *for all* i *and* θ*, and the post-measurement state is* $|v_i\rangle$ *when the measurement outputs* 1*.*

▶ **Lemma 39.** *Let* Π_A, Π_B *be projectors described by uniform* $\mathrm{poly}(\lambda)$*-size quantum circuits. Then there exists a (uniform) family of circuits* $\{\mathsf{Transform}_\gamma\}_{\gamma\in(0,1]}$ *of size* $\mathrm{poly}(\lambda)/\sqrt{\gamma}$ *such that, when* $p_i \geq \gamma$*, the output (i.e., post-measurement state) of* $\mathsf{Transform}(|v_i\rangle)$ *is* $|w_i\rangle$ *with probability* $1 - \mathsf{negl}(\lambda)$*.*

Moreover, $\mathcal{S}_i$ *is invariant under* $\mathsf{Transform}_\gamma$ *for all* i *and* γ*.*

We are now ready to show that (almost-everywhere) non-unequivocality implies one-shot equivocality. Our one-shot equivocal commitment scheme is constructed as follows.

▶ **Construction 40.** *Let* $\mathsf{COM} = (\mathsf{Gen}, \mathsf{Commit})$ *be a bit commitment scheme. For* $k \in \mathbb{N}$*, we construct* COM^k *by:*

- $\mathsf{Gen}^k(1^\lambda)$ *runs* $\mathsf{ck}_i \leftarrow \mathsf{Gen}(1^\lambda)$ *for each* $i \in [k]$ *and outputs* $\mathsf{ck} := (\mathsf{ck}_1, \ldots, \mathsf{ck}_k)$*.*
- $\mathsf{Commit}^k\big((\mathsf{ck}_1, \ldots, \mathsf{ck}_k), m, (i, \omega)\big) := \big(i, \mathsf{Commit}(\mathsf{ck}_i, m, \omega)\big)$*.*

Let Adv *be an adversary for* $\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{sum}}$ *with quantum auxiliary input* ρ*, which applies the projector* Π_b *and measures the opening register* $\mathcal{O}$ *when asked to open to bit* b*. We construct an equivocator* Eq*, whose auxiliary input consists of* k *copies of* ρ *on registers* $\mathcal{A}_1, \ldots, \mathcal{A}_k$*, as follows.*

- $\mathsf{Eq}^{\mathsf{Adv}}_\varepsilon(\mathsf{ck}_1, \ldots, \mathsf{ck}_k; \mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k)$*:*
  1. *For each* $j \in [k]$*:*
     a. *Run* $\mathsf{com}_j \leftarrow \mathsf{Adv}(\mathsf{ck}_j; \mathcal{A}_j)$*.*
     b. *Apply the measurement* $(\Pi_0, \mathbf{I} - \Pi_0)$ *followed by* $\mathsf{Threshold}_{\varepsilon^2/2}$ *to* $\mathcal{A}_j$*. If both outcomes are 1, set* $j^* := j$ *and skip to Step 3.*
  2. *If* j^* *is unset, output* $\perp$*.*
  3. *Output* $(j^*, \mathsf{com}_{j^*})$ *as the commitment. (At this point we can discard* $\mathcal{A}_j$ *for* $j \neq j^*$*.)*
- $\mathsf{Eq}^{\mathsf{Adv}}_\varepsilon(b; \mathcal{A}_{j^*})$*:*
  1. *If* $b = 1$*, apply* $\mathsf{Transform}_{\varepsilon^2/4}$ *followed by the measurement* $(\Pi_1, \mathbf{I} - \Pi_1)$ *to* $\mathcal{A}_{j^*}$*.*
  2. *Measure the opening register* $\mathcal{O} \subset \mathcal{A}_{j^*}$*, obtaining outcome* ω*, and output* (j^*, ω)*.*

Note that $\mathsf{COM}^k = (\mathsf{Gen}^k, \mathsf{Commit}^k)$ is not the k-wise parallel repetition of COM (as decommitting a single coordinate suffices).

▶ **Theorem 41.** *Let* $\varepsilon = \varepsilon(\lambda)$ *be an inverse polynomial, and let* COM *be a bit commitment scheme such that* $\Pr\left[\mathsf{Exp}^{\mathsf{Adv}}_{\mathsf{sum}}(\lambda) = 1\right] = 1/2 + \varepsilon$ *for some QPT adversary* Adv *and all sufficiently large* λ *(i.e., that violates sum binding almost everywhere). Then, with* $k = \lambda/\varepsilon^2$*, the commitment scheme* COM^k *of Construction 40 is one-shot equivocal.*

Proof. First, note that the running time of Eq is $\mathrm{poly}(\lambda)$, as it executes the QPT algorithm Adv (at most) $k = \mathrm{poly}(\lambda)$ times; $\mathsf{Threshold}$ (which is QPT regardless of the parameter) once; and $\mathsf{Transform}$ (with a $\mathrm{poly}(\lambda^{-1})$ parameter, in which case it is QPT) at most once.

For each j, denote by ρ_j the post-measurement state after Step 1a (where the mixture ρ_j includes the distribution over ck_j as well as the measurement that outputs com_j). By assumption, we have

$$\mathrm{Tr}\left((\Pi_0 + \Pi_1)\rho_j\right) \geq 1 + 2\varepsilon.$$

Hence, by Lemma 36,

$$\mathrm{Tr}\left(\Pi_0\Pi_1\Pi_0\rho_j\right) \geq \varepsilon^2 \ .$$

Now, consider the distribution obtained by applying $(\Pi_0, \mathbf{I}-\Pi_0)$ followed by the Jordan measurement $\mathsf{M}^{\mathsf{Jor}}$ (with respect to the pair of projectors Π_0, Π_1), obtaining outcomes (b,i) and outputting $b \cdot p_i$. Then

$$\begin{aligned}
\mathbb{E}\left[b \cdot p_i\right] &= \sum_i p_i \cdot \mathrm{Tr}\left(\Pi_i^{\mathsf{Jor}}\Pi_0\rho_j\Pi_0\right) \\
&= \mathrm{Tr}\left(\left(\sum_i p_i \Pi_0\Pi_i^{\mathsf{Jor}}\Pi_0\right)\rho_j\right) \\
&= \mathrm{Tr}\left(\left(\sum_i p_i \left|v_i\right\rangle\!\left\langle v_i\right|\right)\rho_j\right) \\
&= \mathrm{Tr}\left(\left(\sum_i \left|v_i\right\rangle\!\left\langle v_i\right|\right)\left(\sum_i \left|w_i\right\rangle\!\left\langle w_i\right|\right)\left(\sum_i \left|v_i\right\rangle\!\left\langle v_i\right|\right)\rho_j\right) \\
&= \mathrm{Tr}\left(\Pi_0\Pi_1\Pi_0\rho_j\right) \\
&\geq \varepsilon^2
\end{aligned}$$

where the second-to-last equality uses $p_i = \left|\langle v_i|w_i\rangle\right|^2$.

Therefore, the probability that Step 1b of $\mathsf{Eq}_\varepsilon^{\mathsf{Adv}}(\mathsf{ck}_1, \ldots, \mathsf{ck}_k)$ sets j^* to j (which is unchanged by the Jordan measurement, since $\mathsf{M}^{\mathsf{Jor}}$ commutes with $\mathsf{Threshold}$ and Π_0) is

$$\begin{aligned}
\Pr\left[b \cdot p_i \geq \frac{\varepsilon^2}{2} \text{ and } \mathsf{Threshold}_{\varepsilon^2/2}(|v_i\rangle) \text{ outputs } 1\right] &\geq \left(1-2^{-\lambda}\right) \cdot \Pr\left[b \cdot p_i \geq \frac{\varepsilon^2}{2}\right] \\
&\geq \left(1-2^{-\lambda}\right) \cdot \frac{\varepsilon^2}{2},
\end{aligned}$$

by Lemma 38 and Proposition 11.

By the Chernoff bound (Proposition 12), the probability j^* is left unset in all $j \in [k]$ (causing $\mathsf{Eq} = \mathsf{Eq}_\varepsilon^{\mathsf{Adv}}$ on input $(\mathsf{ck}_1, \ldots, \mathsf{ck}_k)$ to abort in Step 2) is at most $e^{-\Omega(\lambda)} = \mathsf{negl}(\lambda)$.

We now move on to the analysis of $\mathsf{Eq}(b)$. Set $\mathsf{ck} = \mathsf{ck}_{i^*}$, $\mathsf{com} = \mathsf{com}_{i^*}$, $\mathcal{A} = \mathcal{A}_{i^*}$ and recall that $(\Pi_b, \mathbf{I}-\Pi_b)$ is the projective measurement corresponding to the whether Adv wins the sum binding experiment when the challenge is b (that is, Π_b projects onto the subspace spanned by $|\mathsf{ck}, b, \omega\rangle$ such that $\mathsf{Commit}(\mathsf{ck}, b, \omega) = \mathsf{com}$). Then, if $b = 0$, the output of Step 2 of $\mathsf{Eq}(0)$ is a correct opening (with probability 1), since the post-measurement state of Step 3 of $\mathsf{Eq}(\mathsf{ck}_1, \ldots, \mathsf{ck}_k)$ is contained in $\mathrm{Im}(\Pi_0)$; we thus only need to argue that the measurement $(\Pi_1, \mathbf{I}-\Pi_1)$ in Step 1 of $\mathsf{Eq}(1)$ outputs 1 except with probability $\mathsf{negl}(\lambda)$.

For a fixed $j \in [k]$, consider the distribution of (binary) outcomes that arises from applying the measurements $\mathsf{Threshold}_{\varepsilon^2/2}$, $\mathsf{Transform}_{\varepsilon^2/4}$ and $(\Pi_1, \mathbf{I}-\Pi_1)$ in this order to an arbitrary quantum state in $\mathrm{Im}(\Pi_0)$. Note that it suffices to show that the first output is 1 and the last is 0 with probability $\mathsf{negl}(\lambda)$, as this ensures (by a union bound over j) that the probability $\mathsf{Eq}(1)$ fails to return a valid opening remains negligible.

By commutativity of the Jordan measurement with $\mathsf{Threshold}$ and $\mathsf{Transform}$ (and Π_1; recall that every $\mathcal{S}_i$ is invariant under all three), the distribution is identical to that which arises by applying $\mathsf{M}^{\mathsf{Jor}}$ before $\mathsf{Threshold}_{\varepsilon^2/2}$. We now analyse two cases: (i) when $\mathsf{M}^{\mathsf{Jor}}$ outputs i such that $p_i \leq \varepsilon^2/4$, and (ii) when $p_i > \varepsilon^2/4$. (Note that the post-measurement outcome is $|v_i\rangle$ in both cases, as the sequence of measurements is applied to a state in $\mathrm{Im}(\Pi_0)$.)

In case (i), Lemma 38 immediately implies that the outcome of $\mathsf{Threshold}_{\varepsilon^2/2}$ is 1 with probability $\mathsf{negl}(\lambda)$. In case (ii), while Lemma 38 does not allow us to analyse the distribution of $\mathsf{Threshold}_{\varepsilon^2/2}$ (when $\varepsilon^2/4 < p_i < \varepsilon^2/2$), it ensures that *conditioned on outcome* 1 the post-measurement state remains unchanged; then Lemma 39 implies the output of $\mathsf{Transform}_{\varepsilon^2/4}(|v_i\rangle)$ is $|w_i\rangle$ with probability $1 - \mathsf{negl}(\lambda)$, in which case the $(\Pi_1, \mathbf{I} - \Pi_1)$ measurement always outputs 1.

The probability $\mathsf{Threshold}_{\varepsilon^2/2}$ outputs 1 *and* $(\Pi_1, \mathbf{I} - \Pi_1)$ outputs 0 is thus $\mathsf{negl}(\lambda)$ in either case, which concludes the proof. ◀

References

1 Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 474–483. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.57. 2

2 Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 255–268. ACM, 2020. doi:10.1145/3357713.3384304. 5, 6, 15, 16

3 Prabhanjan Ananth, Kai-Min Chung, and Rolando L. La Placa. On the concurrent composition of quantum zero-knowledge. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 346–374. Springer, 2021. doi:10.1007/978-3-030-84242-0_13. 2

4 Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. 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 I*, volume 13507 of *Lecture Notes in Computer Science*, pages 208–236. Springer, 2022. doi:10.1007/978-3-031-15802-5_8. 2

5 Nir Bitansky and Zvika Brakerski. Classical binding for quantum commitments. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part I*, volume 13042 of *Lecture Notes in Computer Science*, pages 273–298. Springer, 2021. doi:10.1007/978-3-030-90459-3_10. 2, 4

6 Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011. doi:10.1007/978-3-642-25385-0_3. 1

7 Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 592–608. Springer, 2013. doi:10.1007/978-3-642-38348-9_35. 1

8 Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 361–379. Springer, 2013. doi:10.1007/978-3-642-40084-1_21. 1

9 Shujiao Cao and Rui Xue. The gap is sensitive to size of preimages: Collapsing property doesn't go beyond quantum collision-resistance for preimages bounded hash functions. 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 III*, volume 13509 of *Lecture Notes in Computer Science*, pages 564–595. Springer, 2022. doi:10.1007/978-3-031-15982-4_19. 2, 13

10 Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. A black-box approach to post-quantum zero-knowledge in constant rounds. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 315–345. Springer, 2021. doi:10.1007/978-3-030-84242-0_12. 2

11 Alessandro Chiesa, Fermi Ma, Nicholas Spooner, and Mark Zhandry. Post-quantum succinct arguments: Breaking the quantum rewinding barrier. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 49–58. IEEE, 2021. doi:10.1109/FOCS52979.2021.00014. 2, 6, 7

12 Claude Crépeau, Paul Dumais, Dominic Mayers, and Louis Salvail. Computational collapse of quantum state with application to oblivious transfer. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 374–393. Springer, 2004. doi:10.1007/978-3-540-24638-1_21. 4

13 Claude Crépeau, Louis Salvail, Jean-Raymond Simard, and Alain Tapp. Two provers in isolation. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 407–430. Springer, 2011. doi:10.1007/978-3-642-25385-0_22. 6, 17

14 Paul Dumais, Dominic Mayers, and Louis Salvail. Perfectly concealing quantum bit commitment from any quantum one-way permutation. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2000. doi:10.1007/3-540-45539-6_21. 2

15 Junbin Fang, Dominique Unruh, Jun Yan, and Dehua Zhou. How to base security on the perfect/statistical binding property of quantum bit commitment? In Sang Won Bae and Heejin Park, editors, *33rd International Symposium on Algorithms and Computation, ISAAC 2022, December 19-21, 2022, Seoul, Korea*, volume 248 of *LIPIcs*, pages 26:1–26:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ISAAC.2022.26. 2, 4

16 Prastudy Fauzi, Helger Lipmaa, Zaira Pindado, and Janno Siim. Somewhere statistically binding commitment schemes with applications. In Nikita Borisov and Claudia Díaz, editors, *Financial Cryptography and Data Security - 25th International Conference, FC 2021, Virtual Event, March 1-5, 2021, Revised Selected Papers, Part I*, volume 12674 of *Lecture Notes in Computer Science*, pages 436–456. Springer, 2021. doi:10.1007/978-3-662-64322-8_21. 10

17 Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 342–371. Springer, 2017. doi:10.1007/978-3-319-63715-0_12. 1

18 András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT*

Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019, pages 193–204. ACM, 2019. doi:10.1145/3313276.3316366. 17

19 Pavel Hubácek and Daniel Wichs. On the communication complexity of secure function evaluation with long output. In Tim Roughgarden, editor, *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 163–172. ACM, 2015. doi:10.1145/2688073.2688105. 10

20 Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 326–355. Springer, 2019. doi:10.1007/978-3-030-26951-7_12. 2

21 Alex Lombardi, Fermi Ma, and Nicholas Spooner. Post-quantum zero knowledge, revisited or: How to do quantum rewinding undetectably. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 851–859. IEEE, 2022. doi:10.1109/FOCS54457.2022.00086. 2, 3, 6, 17

22 Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152. Springer, 2012. doi:10.1007/978-3-642-29011-4_10. 6

23 Dominique Unruh. Collapse-binding quantum commitments without random oracles. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, volume 10032 of *Lecture Notes in Computer Science*, pages 166–195, 2016. doi:10.1007/978-3-662-53890-6_6. 2, 4, 11

24 Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527. Springer, 2016. doi:10.1007/978-3-662-49896-5_18. 2, 4, 5, 7, 9

25 Jun Yan. Quantum computationally predicate-binding commitments with application in quantum zero-knowledge arguments for NP. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 575–605. Springer, 2021. doi:10.1007/978-3-030-92062-3_20. 4

26 Jun Yan. General properties of quantum bit commitments (extended abstract). In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV*, volume 13794 of *Lecture Notes in Computer Science*, pages 628–657. Springer, 2022. doi:10.1007/978-3-031-22972-5_22. 4

27 Jun Yan, Jian Weng, Dongdai Lin, and Yujuan Quan. Quantum bit commitment with application in quantum zero-knowledge proof (extended abstract). In Khaled M. Elbassioni and Kazuhisa Makino, editors, *Algorithms and Computation - 26th International Symposium, ISAAC 2015, Nagoya, Japan, December 9-11, 2015, Proceedings*, volume 9472 of *Lecture Notes in Computer Science*, pages 555–565. Springer, 2015. doi:10.1007/978-3-662-48971-0_47. 2

28 Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany,*

May 19-23, 2019, Proceedings, Part III, volume 11478 of *Lecture Notes in Computer Science*, pages 408–438. Springer, 2019. doi:10.1007/978-3-030-17659-4_14. 2, 6

29 Mark Zhandry. Quantum lightning never strikes the same state twice. Or: Quantum money from cryptographic assumptions. *J. Cryptol.*, 34(1):6, 2021. doi:10.1007/s00145-020-09372-x. 2, 6

30 Mark Zhandry. New constructions of collapsing hashes. 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 III*, volume 13509 of *Lecture Notes in Computer Science*, pages 596–624. Springer, 2022. doi:10.1007/978-3-031-15982-4_20. 2, 13

Optimal Algorithms for Learning Quantum Phase States

Srinivasan Arunachalam
IBM Quantum, Thomas J Watson Research Center, Yorktown Heights, NY, USA

Sergey Bravyi
IBM Quantum, Thomas J Watson Research Center, Yorktown Heights, NY, USA

Arkopal Dutt
IBM Quantum, Thomas J Watson Research Center, Yorktown Heights, NY, USA
MIT-IBM Watson AI Lab, Cambridge, MA, USA
Department of Physics, Co-Design Center for Quantum Advantage, Massachusetts Institute of Technology, Cambridge, MA, USA

Theodore J. Yoder
IBM Quantum, Thomas J Watson Research Center, Yorktown Heights, NY, USA

Abstract

We analyze the complexity of learning n-qubit quantum phase states. A degree-d phase state is defined as a superposition of all 2^n basis vectors x with amplitudes proportional to $(-1)^{f(x)}$, where f is a degree-d Boolean polynomial over n variables. We show that the sample complexity of learning an unknown degree-d phase state is $\Theta(n^d)$ if we allow separable measurements and $\Theta(n^{d-1})$ if we allow entangled measurements. Our learning algorithm based on separable measurements has runtime $\mathsf{poly}(n)$ (for constant d) and is well-suited for near-term demonstrations as it requires only single-qubit measurements in the Pauli X and Z bases. We show similar bounds on the sample complexity for learning generalized phase states with complex-valued amplitudes. We further consider learning phase states when f has sparsity-s, degree-d in its $\mathbb{F}_2$ representation (with sample complexity $O(2^d sn)$), f has Fourier-degree-t (with sample complexity $O(2^{2t})$), and learning quadratic phase states with ε-global depolarizing noise (with sample complexity $O(n^{1+\varepsilon})$). These learning algorithms give us a procedure to learn the diagonal unitaries of the Clifford hierarchy and IQP circuits.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Tomography, binary phase states, generalized phase states, IQP circuits

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.3

Related Version *Full Version*: `https://arxiv.org/abs/2208.07851` [6]

Funding SA, SB, and TY were supported in part by the Army Research Office under Grant Number W911NF-20-1-0014.
Arkopal Dutt: AD was supported in part by the MIT-IBM Watson AI Lab, and in part by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-Design Center for Quantum Advantage under contract DE-SC0012704.

Acknowledgements AD thanks Isaac L Chuang for suggesting applications of the learning algorithms presented here and for useful comments on the manuscript. SA thanks Giacomo Nannicini and Chinmay Nirkhe for useful discussions.

1 Introduction

Quantum state tomography is the problem of learning an unknown quantum state ρ drawn from a specified class of states by performing measurements on multiple copies of ρ. The preeminence of this problem in verification of quantum experiments has motivated an in-depth study of state tomography protocols and their limitations for various classes of quantum states [23, 40, 5, 46]. The main figure of merit characterizing a state tomography protocol is its *sample complexity* defined as the number of copies of ρ consumed by the protocol in order to learn ρ. Of particular interest are classes of n-qubit quantum states that can be learned efficiently, such that the sample complexity grows only polynomially with n. Known examples of efficiently learnable states include Matrix Product States describing weakly entangled quantum spin chains [17], output states of Clifford circuits [36], output states of Clifford circuits with a single layer of T gates [30], and high-temperature Gibbs states of local Hamiltonians [4, 24]. Apart from their potential use in experiments, efficiently learnable quantum states are of great importance for quantum algorithm design. For example, a quantum algorithm for solving the dihedral hidden subgroup problem [7] can be viewed as a tomography protocol for learning so-called hidden subgroup states (although this protocol is efficient in term of its sample complexity, its runtime is believed to be super-polynomial [7]).

A natural question to then ask is: What are other classes of n-qubit quantum states that are ubiquitous in quantum computing, which can be learned efficiently? In this work, we consider the problem of state tomography for *phase states* associated with (generalized) Boolean functions. Phase states are encountered in quantum information theory [26], quantum algorithm design [7], quantum cryptography [29, 11], and quantum-advantage experiments [13, 15].

By definition, an n-qubit, degree-d binary phase state has the form

$$|\psi_f\rangle = 2^{-n/2} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle, \tag{1}$$

where $f : \{0,1\}^n \to \{0,1\}$ is a degree-d polynomial, that is,

$$f(x) = \sum_{J\subseteq[n],\,|J|\le d} \alpha_J \prod_{j\in J} x_j \pmod 2, \tag{2}$$

for some coefficients $\alpha_J \in \{0,1\}$. Phase states associated with homogeneous degree-2 polynomials $f(x)$ coincide with graph states that play a prominent role in quantum information theory [26]. Such states can be alternatively represented as

$$|\psi_f\rangle = \prod_{(i,j)\in E} \mathsf{CZ}_{i,j}|+\rangle^{\otimes n},$$

where n qubits live at vertices of a graph, E is the set of graph edges, $\mathsf{CZ}_{i,j}$ is the controlled-Z gate applied to qubits i, j, and $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. It is known that the output state of any Clifford circuit is locally equivalent to a graph state for a suitable graph [44]. Our results imply that graph states can be learned efficiently using only single-qubit gates and measurements. The best previously known protocol for learning graph states [36] requires entangled measurements across two copies of $|\psi_f\rangle$. Other examples of circuits producing phase states include measurement-based quantum computing [42] and a subclass of IQP circuits (Instantaneous Quantum Polynomial-time), which correspond to degree-3 phase states [37]. IQP circuits are prevalent in quantum-advantage experiments [13, 15] and are believed to be hard to simulate classically.

We also consider generalized degree-d phase states

$$|\psi_f\rangle = 2^{-n/2} \sum_{x\in\{0,1\}^n} \omega_q^{f(x)} |x\rangle, \qquad \omega_q = e^{2\pi i/q} \tag{3}$$

where $q \geq 2$ is an even integer and $f : \{0,1\}^n \to \mathbb{Z}_q$ is a degree-d polynomial, that is,

$$f(x) = \sum_{J\subseteq[n],\, |J|\leq d} \alpha_J \prod_{j\in J} x_j \pmod q. \tag{4}$$

for coefficients $\alpha_J \in \mathbb{Z}_q = \{0, 1, \ldots, q-1\}$. It is also known that generalized degree-d phase states with $q = 2^d$ can be prepared from diagonal unitary operators [18] in the d-th level of the Clifford hierarchy [22]. Additionally, it is known that the output state of a random n-qubit Clifford circuit is a generalized $q = 4$, degree-2 phase state with a constant probability [12, Appendix D]. Binary and generalized phase states have also found applications in cryptography [29, 11], and complexity theory [28] (we discuss this in the next section).

In this work, we consider learning phase states through two types of tomography protocols based on *separable* and *entangled* measurements. The former can be realized as a sequence of M independent measurements, each performed on a separate copy of $|\psi_f\rangle$ (furthermore our learning algorithms only require single *qubit* measurements). The latter performs a joint measurement on the state $|\psi_f\rangle^{\otimes M}$. Our goal is to then derive upper and lower bounds on the sample complexity M of learning f, as a function of n and d. In the next section, we state our main results. Interestingly, our protocols based on separable measurements require only single-qubit gates and single-qubit measurements making them well suited for near-term demonstrations.

1.1 Summary of contributions and applications

We first introduce some notation before giving an overview of our contributions. For every n and $d \leq n/2$, let $\mathcal{P}(n,d)$ be the set of all degree-d polynomials of the form Eq. (2). Let $\mathcal{P}_q(n,d)$ be the set of all degree-d $\mathbb{Z}_q$-valued polynomials of the form Eq. (3). By definition, $\mathcal{P}_2(n,d) \equiv \mathcal{P}(n,d)$. To avoid confusion, we shall refer to states defined in Eq. (1) as binary phase states and in Eq. (3) as generalized phase states. Our learning protocol takes as input integers n, d and M copies of a degree-d phase state $|\psi_f\rangle$ with unknown $f \in \mathcal{P}(n,d)$ (or $f \in \mathcal{P}_q(n,d)$). The protocol outputs a classical description of a polynomial $g \in \mathcal{P}(n,d)$ (or $g \in \mathcal{P}_q(n,d)$) such that $f = g$ with high probability.

The main result in this work are optimal algorithms for learning phase states if the algorithm is allowed to make separable or entangled measurements. Prior to our work, we are aware of only two works in this direction (i) algorithms for efficiently learning degree-1 and degree-2 phase states; (ii) Montanaro [35] considered learning multilinear polynomials f, assuming we have *query access* to f, which is a stronger learning model than the sample access model that we assume for our learning algorithm. In this work, we show that if allowed separable measurements, the *sample* complexity of learning binary phase states and generalized phase states is $O(n^d)$. If allowed entangled measurements, we obtain a sample complexity of $O(dn^{d-1})$ for learning binary phase states. We further consider settings where the unknown function f we are trying to learn is known to be sparse, has a small Fourier-degree and the setting when given noisy copies of the quantum phase state. In Table 1, we summarize all our main results (except the first two rows, which include the main prior work in this direction).

Table 1 Upper and lower bounds of sample complexity for exact learning of n-qubit phase states with degree-d. For precise statements of the bounds, we refer the reader to the theorem statements in this work and in the full version of the paper [6].

	Sample complexity	Time complexity	Measurements
Binary phase state $\mathbb{F}_2$-degree-1 [10]	$\Theta(1)$	$O(n^3)$	Separable
Binary phase state $\mathbb{F}_2$-degree-2 [36, 43]	$O(n)$	$O(n^3)$	Entangled
Binary phase state $\mathbb{F}_2$-degree-d	$\Theta(n^d)$ Theorem 7, 10	$O(n^{3d-2})$	Separable
Binary phase state $\mathbb{F}_2$-degree-d	$\Theta(n^{d-1})$ Theorem 9	$O(\exp(n^d \log 2))$	Entangled
Generalized phase states degree-d	$\Theta(n^d)$ Theorem 11	$O(\exp(n^d \log q))$	Separable
Sparse Binary phase state $\mathbb{F}_2$-degree-d, $\mathbb{F}_2$-sparsity s	$O(2^d sn)$ [6, Theorem 6]	$O(2^{3d} s^3 n)$	Separable
Binary phase state $\mathbb{F}_2$-degree-2 with global depolarizing noise ε	$n^{1+O(\varepsilon)}$ [6, Theorem 9]	$O(2^{n/\log n})$	Entangled
Binary phase state $\mathbb{F}_2$-degree-2 with local depolarizing noise ε	$\Theta((1-\varepsilon)^n)$ [6, Theorem 11]	$O(2^{n/\log n})$	Entangled
Binary phase state Fourier-degree-d	$O(2^{2d})$ [6, Theorem 7]	$O(\exp(n^2))$	Entangled

Before we give a proof sketch of these results, we first discuss a couple of motivations for considering the task of learning phase states and corresponding applications.

Quantum complexity. Recently, there has been a few results in quantum cryptography [29, 3, 11] and complexity theory [28] which used the notion of phase states.

Ji et al. [29] introduced the notion of *pseudorandom quantum states* as states of the form $|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} \omega_N^{F(x)} |x\rangle$ where F is a pseudorandom function.[1] Ji et al. showed that states of the form $|\phi\rangle$ are efficiently preparable and statistically indistinguishable from a Haar random state, which given as input to a polynomial-time quantum algorithm. A subsequent work of Brakerski [11] showed that it suffices to consider $|\phi'\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{F(x)} |x\rangle$ (where F again is a pseudorandom function) and such states are also efficiently preparable and statistically indistinguishable from Haar random states. Subsequently, these states have found applications in proposing many cryptosystems [3]. Although none of these works discuss the degree of the phase function F, our result shows implicitly that when F is low-degree, then $|\phi\rangle$ is exactly learnable and hence distinguishable from Haar random states, implying that they cannot be quantum pseudorandom states. In another recent work, Irani et al. [28] considered the power of quantum witnesses in proof systems. In particular, they showed that in order to construct the witness to a QMA complete problem, say the ground state

[1] We do not discuss the details of pseudorandom functions here, we refer the interested reader to [29].

$|\phi\rangle$ to a local-Hamiltonian problem, it suffices to consider a phase state $\frac{1}{\sqrt{2^n}}\sum_x(-1)^{F(x)}|x\rangle$ which has a good overlap to $|\phi\rangle$. To this end, they show a strong property that, for *every* state $|\tau\rangle$ and a random Clifford operator U (or, more generally, an element of some unitary 2-design), the state $U|\tau\rangle$ has constant overlap with a phase state [28, Lemma A.5]. Our learning result implicitly shows that, assuming QMA $\neq$ QCMA, then the phase state that has constant overlap with the ground space energy of the local Hamiltonian problem, cannot be of low degree.

Learning quantum circuits. Given access to a quantum circuit U, the goal of this learning task is to learn a circuit representation of U. The sample complexity for learning a general n-qubit quantum circuit is known to be $2^{\Theta(n)}$ [16, 34], which is usually impractical.

If we restrict ourselves to particular classes of quantum circuits, there are some known results for efficient learnability. Low [31] showed that an n-qubit Clifford circuit can be learned using $O(n)$ samples. However, this result was only an existential proof and requires access to the conjugate of the circuit. Constructive algorithms were given in Low [31], and Lai and Cheng [30], both of which showed that Clifford circuits can be learned using $O(n^2)$ samples. Both these algorithms require entangled measurements with the former algorithm using pretty-good measurement [25], and the latter using Bell sampling. In this work, we show that Clifford circuits producing degree-2 binary phase states, can be learned in $O(n^2)$ samples, matching their result but only using separable measurements. Moreover, Low [31] also gave an existential proof of algorithms for learning circuits in the d-th level of the Clifford hierarchy, using $O(n^{d-1})$ samples. In this work, we give constructive algorithms for learning the diagonal elements of the Clifford hierarchy in $O(n^d)$ samples using separable measurements. A direct result of this is that a subset of IQP circuits, which are also believed to be hard to simulate classically [13, 14], are shown to be efficiently learnable. Our learning result thus gives an efficient method for verifying IQP circuits that may be part of quantum-advantage experiments [15, 39].

Learning hypergraph states. We finally observe that degree-3 (and higher-degree) phase states have appeared in works [42, 45] on measurement-based quantum computing (MBQC), wherein they refer to these states as *hypergraph states*. These works show that single-qubit measurements in the Pauli X or Z basis performed on a suitable degree-3 hypergraph state are sufficient for universal MBQC. Our learning algorithm gives a procedure for learning these states in polynomial-time and could potentially be used as a subroutine for verifying MBQC.

1.2 Proof sketch

In this section we briefly sketch the proofs of our main results.

1.2.1 Binary phase states

As we mentioned earlier, Montanaro [36] and Roettler [43] showed how to learn degree-2 phase states using $O(n)$ copies of the state. Crucial to both their learning algorithms was the following so-called Bell-sampling procedure: given two copies of $|\psi_f\rangle = \frac{1}{\sqrt{2^n}}\sum_x(-1)^{f(x)}|x\rangle$ where $f(x) = x^\top Ax$ (where $A \in \mathbb{F}_2^{n\times n}$), perform n CNOTs from the first copy to the second, and measure the second copy. One obtains a uniformly random $y \in \mathbb{F}_2^n$ and the state

$$\frac{1}{\sqrt{2^n}}\sum_x(-1)^{f(x)+f(x+y)}|x\rangle = \frac{(-1)^{y^\top Ay}}{\sqrt{2^n}}\sum_x(-1)^{x^\top(A+A^\top)\cdot y}|x\rangle.$$

Using Bernstein-Vazirani [10] one can apply n-qubit Hadamard transform to obtain the bit string $(A + A^\top) \cdot y$. Repeating this process $O(n \log n)$ many times, one can learn n linearly independent constraints about A, and along with Gaussian elimination, allows to learn $A + A^\top$. Diagonal elements of A can be learned with one additional copy of $|\psi_f\rangle$. Applying a controlled-Z gate between all pairs of qubits $i > j$ for which $(A + A^\top)_{ij} = 1$ results in the state $\sum_x (-1)^{\sum_i x_i A_{ii}} |x\rangle$, which can be learned using Bernstein-Vazarani.

Applying this same Bell-sampling procedure to degree-3 phase states does not easily learn the phase function. In this direction, from two copies of the degree-3 phase state $|\psi_f\rangle$ one obtains a uniformly random $y \in \mathbb{F}_2^n$ and the state $|\psi_{g_y}\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{g_y(x)} |x\rangle$ for a degree-2 polynomial $g_y(x) = f(x) + f(x + y)$. One might now hope to apply the degree-2 learning algorithm from above, but since the single copy of $|\psi_{g_y}\rangle$ was randomly generated, it takes $\Omega(\sqrt{2^n})$ copies of $|\psi_f\rangle$ to obtain enough copies of $|\psi_{g_y}\rangle$. Our main idea is to circumvent this Bell-sampling approach and instead propose two techniques that allow us to learn binary phase states using separable and entangled measurements which we discuss further below.

Separable measurements, upper bound. Our first result is that we are able to learn binary phase states using separable measurements with sample complexity $O(n^d)$. In order to prove our upper bounds of sample complexity for learning with separable measurements, we make a simple observation. Given one copy of $|\psi_f\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} |x\rangle$, measure qubits $2, 3, \ldots, n$ in the computational basis. Suppose the resulting string is $y \in \{0,1\}^{n-1}$. The post-measurement state of qubit 1 is then given by

$$|\psi_{f,y}\rangle = \frac{1}{\sqrt{2}} \left[(-1)^{f(0y)} |0\rangle + (-1)^{f(1y)} |1\rangle \right].$$

By applying a Hadamard transform to $|\psi_{f,y}\rangle$ and measuring, the algorithm obtains $p_1(y) = f(0y) + f(1y) \mod 2$, which can be viewed as the derivative of f in the first direction at point y. Furthermore observe that p_1 is a degree $\leq d-1$ polynomial over $(n-1)$ variables. Hence, the learning algorithm repeatedly measures the last $(n-1)$ qubits and obtains $y^{(1)}, \ldots, y^{(M)}$ for $M = n^{d-1}$ and obtains $(y^{(k)}, p_1(y^{(k)}))$ for all $k = 1, 2, \ldots, M$ using the procedure above, which suffices to learn p_1 completely. Then the algorithm repeats the same procedure by measuring all the qubits except the second qubit in the computational basis and learns the derivative of f in the second direction. This is repeated over all the n qubits. Through this procedure, a learning algorithm learns the partial derivatives of f in the n directions and a simple argument shows that this is sufficient to learn f completely. This gives an overall sample complexity of $O(n^d)$. The procedure above only uses single qubit measurements in the $\{X, Z\}$ basis.

Separable measurements, lower bound. Given the algorithm for learning binary phase states using separable measurements, a natural question is: Is the upper bound on sample complexity we presented above tight? Furthermore, suppose the learning algorithm was allowed to make arbitrary n-qubit measurements on a single copy of $|\psi_f\rangle$, instead of *single qubit* measurements (which are weaker than single *copy* measurements), then could we potentially learn f using fewer than $O(n^d)$ copies?

Here we show that if we allowed *arbitrary* single copy measurements, then a learning algorithm needs $\Omega(n^d)$ many copies of $|\psi_f\rangle$ to learn f. In order to prove this lower bound, our main technical idea is the following. Let f be a degree-d polynomial with n variables

sampled uniformly at random. Suppose a learning algorithm measures the phase state $|\psi_f\rangle$ in an arbitrary orthonormal basis $\{U|x\rangle\}_x$. We show that the distribution describing the measurement outcome x is "fairly" uniform. In particular,

$$\mathop{\mathbb{E}}_{f}[H(x|f)] \geq n - O(1), \tag{5}$$

where $H(x|f)$ is the Shannon entropy of a distribution $P(x|f) = |\langle x|U^*|\psi_f\rangle|^2$. Thus, for a typical f, measuring one copy of the phase state $|\psi_f\rangle$ provides at most $O(1)$ bits of information about f. Since a random uniform degree-d polynomial f with n variables has entropy $\Omega(n^d)$, one has to measure $\Omega(n^d)$ copies of ψ_f in order to learn f. To prove Eq. (5), we first lower bound the Shannon entropy by Renyi-two entropy and bound the latter by deriving an explicit formula for $\mathbb{E}_f[|\psi_f\rangle\langle\psi_f|^{\otimes 2}]$.

Entangled measurements. After settling the sample complexity of learning binary phase states using separable measurements, one final question question remains: Do entangled measurements help in reducing the sample complexity? For the case of quadratic polynomials, we know that Bell measurements (which are entangled measurements) can be used to learn these states in sample complexity $O(n)$. However, as mentioned earlier, it is unclear how to extend the Bell measurement procedure for learning larger degree polynomials.

Here, we give a learning algorithm based on the so-called pretty-good measurements (PGM) that learns $|\psi_f\rangle$ for a degree-d polynomial f using $O(n^{d-1})$ copies of $|\psi_f\rangle$. In order to prove this bound, we follow the following three step approach: (a) we first observe that in order to learn degree-d binary phase states, the *optimal* measurement is the pretty good measurement since the ensemble $\mathcal{S} = \{|\psi_f\rangle\}_f$ is geometrically uniform. By geometrically uniform, we mean that $\mathcal{S}$ can be written as $\mathcal{S} = \{U_f|\phi\rangle\}_f$ where $\{U_f\}_f$ is an Abelian group. (b) We next observe a property about the geometrically uniform state identification problem (which is new as far as we are aware): suppose $\mathcal{S}$ is a geometrically uniform ensemble, then the success probability of the PGM in correctly identifying f, given copies of $|\psi_f\rangle$, is *independent* of f, i.e., every element of the ensemble has the same probability of being identified correctly when measured using the PGM. (c) Finally, we need one powerful tool regarding the the weight distribution of Boolean polynomials: it was shown in [1] that for any degree-d polynomial f, the following relation on $\mathsf{wt}(f)$ or the fraction of strings in $\{0,1\}^n$ for which f is one holds:

$$|\{f \in \mathcal{P}(n,d) : \mathsf{wt}(f) \leq (1-\varepsilon)2^{-\ell}\}| \leq (1/\varepsilon)^{C\ell^4 \cdot \binom{n-\ell}{\leq d-\ell}},$$

for every $\varepsilon \in (0, 1/2)$ and $\ell \in \{1, \ldots, d-1\}$. Using this statement, we can comment on the average inner product of $|\langle\psi_f|\psi_g\rangle|$ over all ensemble members with $f \neq g \in \mathcal{P}(n,d)$. Combining this with a well-known result of PGMs, we are able to show that, given $M = O(n^{d-1})$ copies of $|\psi_f\rangle$ for $f \in \mathcal{S}$, the PGM identifies f with probability ≥ 0.99. Combining observations (a) and (b), the PGM also has the same probability of acceptance given an arbitrary $f \in \mathcal{S}$. Hence, we get an overall upper bound of $O(n^{d-1})$ for sample complexity of learning binary phase states using entangled measurements.

The lower bound for entangled measurement setting is straightforward: each quantum sample $\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}|x\rangle$ provides n bits of information and the goal is to learn f which contains $O(n^d)$ bits of information, hence by Holevo's bound, we need at least n^{d-1} quantum samples in order to learn f with high probability.

1.2.2 Generalized phase states

As far as we are aware, ours is the first work that considers the learnability of generalized phase states (using either entangled or separable measurements). The sample complexity upper bounds follow the same high-level idea as that in the binary phase state setting. However, we need a few more technical tools for this setting which we discuss below.

Separable bounds. At a high-level, the learning procedure for generalized phase states is similar to the procedure for learning binary phase states with the exception of a couple of subtleties that we need to handle here. Suppose we perform the same procedure as in binary phase states by measuring the last $(n-1)$ qubits in the computational basis. We then obtain a uniformly random $y \in \mathbb{F}_2^{n-1}$, and the post-measurement state for a generalized phase state is given by

$$|\psi_{f,y}\rangle = \frac{1}{\sqrt{2}}(\omega_q^{f(0y)}|0\rangle + \omega_q^{f(1y)}|1\rangle).$$

This state is proportional to $(|0\rangle + \omega_q^c|1\rangle)/\sqrt{2}$, where $c = f(1y) - f(0y) \pmod q$. In the binary case, $q = 2$, the states associated with $c = 0$ and $c = 1$ are orthogonal, so that the value of c can be learned with certainty by measuring $|\psi_{f,y}\rangle$ in the Pauli X basis. However, in the generalized case, $q > 2$, the states $(|0\rangle + \omega_q^c|1\rangle)/\sqrt{2}$ with $c \in \mathbb{Z}_q$ are not pairwise orthogonal. It is then unclear how to learn c given a single copy of $|\psi_{f,y}\rangle$. However, we observe that it is still possible to obtain a value $b \in \mathbb{Z}_q$ such that $b \neq c$ with certainty. To this end, consider a POVM whose elements are given by $\mathcal{M} = \{|\phi_b\rangle\langle\phi_b|\}_{b\in\mathbb{Z}_q}$, where $|\phi_b\rangle = \frac{1}{\sqrt{2}}(|0\rangle - \omega_q^b|1\rangle)$. Applying this POVM $\mathcal{M}$ onto an unknown state $(|0\rangle + \omega_q^c|1\rangle)/\sqrt{2}$ we observe that c is the outcome with probability 0 and furthermore *every* other outcome $b \neq c$ appears with non-negligible probability $\Omega(q^{-3})$.

Hence with one copy of $\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}\omega_q^{f(x)}|x\rangle$, we obtain uniformly random $y \in \{0,1\}^{n-1}$ and $b \in \mathbb{Z}_q$ such that $f(1y) - f(0y) \neq b$. We now repeat this process $m = O(n^{d-1})$ many times and obtain $(y^{(k)}, b^{(k)})$ for $k = 1, 2, \ldots, M$ such that $f(1y^{(k)}) - f(0y^{(k)}) \neq b^{(k)}$ for all $k \in [M]$. We next show a variant of the Schwartz-Zippel lemma in the following sense: that for every $f \in \mathcal{P}_q(n, d)$ and $c \in \mathbb{Z}_q$, then either f is a constant function or the fraction of $x \in \mathbb{F}_2^n$ for which $f(x) \neq c$ is at least 2^{-d}. Using this, we show that after obtaining $O(2^d n^{d-1})$ samples, we can find a polynomial $g \in \mathcal{P}_q(n-1, d-1)$ for which $f(1y) - f(0y) = g(y)$. We now repeat this protocol for n different directions (by measuring each of the n qubits in every iteration) and we learn all the n directional derivatives of f, which suffices to learn f completely.

Entangled bounds. We do not give a result on learning generalized phase states with entangled measurements. We expect the proof of the sample complexity upper bound for learning generalized phase states using entangled measurements should proceed similarly to our earlier analysis of learning binary phase states using entangled measurements. However, we need a new technical tool that generalizes the earlier work on the weight distribution [2] of Boolean functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ to those of form $f : \mathbb{F}_2^n \to \mathbb{Z}_q$ with $q = 2^d$.

1.2.3 Learning with further constraints

Learning sparse and low-Fourier degree states. A natural constraint to put on top of having low $\mathbb{F}_2$-degree in the polynomial is the sparsity, i.e., number of monomials in the $\mathbb{F}_2$ decomposition of f. Sparse low-degree phase states appear naturally when learning circuits

with few gates. In particular, suppose we are learning a quantum circuit U with s gates from $\{\mathsf{Z}, \mathsf{CZ}, \ldots, \mathsf{C}^{d-1}\mathsf{Z}\}$ (where $\mathsf{C}^m\mathsf{Z}$ is the controlled-Z gate with m controls), then the output of $U|+\rangle^{\otimes n}$ is a phase state with sparsity-s and degree-d.

One naive approach to learn sparse $\mathbb{F}_2$ polynomials is to directly apply our earlier learning algorithm for binary phase states but this ignores the $\mathbb{F}_2$-sparsity information, and doesn't improve the sample complexity. Instead, here we use ideas from compressed sensing [20] to propose a linear program that allows us to improve the sample complexity to $O(2^d sn)$. Finally we make an observation that, if the function has *Fourier*-degree d, then one can learn f, given only $O(2^d \log n)$ many copies of $|\psi_f\rangle$, basically using the fact that there are only 2^{2^d} many such functions, each having at least a 2^{-d} distance between them.

Learning with depolarizing noise. One motivation for learning stabilizer states was potential experimental demonstrations of the learning algorithm [41]. Here, we consider a theoretical framework in order to understand the sample complexity of learning degree-2 phase states under global and local depolarizing noise. In this direction, we present two results. Under global depolarizing noise, i.e., when we are given $\rho_f = (1-\varepsilon)|\psi_f\rangle\langle\psi_f| + \varepsilon \cdot \mathbb{I}$, then it suffices to take $O(n^{1+\varepsilon})$ many copies ρ_f in order to learn f. The crucial observation is that one can use Bell sampling to reduce learning ρ_f to learning parities with noise, which we can accomplish using $O(n^{1+\varepsilon})$ samples and in time $2^{n/(\log\log n)}$ [32]. Additionally, however, a simple argument reveals that under local depolarizing noise, the sample complexity of learning stabilizer states is exponential in n.

1.3 Organization

In Section 2, we introduce phase states, discuss separable and entangled measurements. In Section 3, we prove our upper and lower bounds for learning binary phase states with separable and entangled measurements. We omit our results on learning sparse and low-Fourier-degree phase states, and binary phase states under depolarizing noise from this version of the paper (see [6]). In Section 4, we prove our upper bound for learning generalized phase states using separable and entangled measurements. Our algorithms for learning quantum phase states can be used to learn the corresponding circuits that produce them. We explicitly discuss the connection between phase states, and the diagonal unitaries in the d-th level of the Clifford hierarchy and IQP circuits in [6].

2 Preliminaries

2.1 Notation

Let $[n] = \{1, \ldots, n\}$. Let e_i be an n-dimensional vector with 1 in the ith coordinate and 0s elsewhere. We denote the finite field with the elements $\{0, 1\}$ as $\mathbb{F}_2$ and the ring of integers modulo q as $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$ with q usually being a power of 2 in this work. For a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, the bias of f is defined as

$$\text{bias}(f) = \mathop{\mathbb{E}}_x[(-1)^{f(x)}],$$

where the expectation is over a uniformly random $x \in \{0,1\}^n$. For $g : \mathbb{F}_2^n \to \mathbb{Z}_{2^d}$, the bias of g in the coordinate $j \in \mathbb{F}_{2^d}^\star$ is defined as $\text{bias}_j(g) = \mathbb{E}_x[(\omega_{2^d})^{j \cdot g(x)}]$. For a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, $y \in \mathbb{F}_2^{n-1}$ and $k \in [n]$, we denote $(D_k f)(y) = f(y^{k=1}) + f(y^{k=0})$, where $y^{i=1}, y^{i=0} \in \mathbb{F}_2^n$ is defined as: the ith bit of $y^{i=1}$ equals 1 and $y^{i=0}$ equals 0 and otherwise equals y.

2.2 Boolean Functions

A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be uniquely represented by a polynomial over $\mathbb{F}_2$ as follows (which we call its $\mathbb{F}_2$ representation):

$$f(x) = \sum_{J\subseteq[n]} \alpha_J \prod_{i\in J} x_i \pmod 2, \tag{6}$$

where $\alpha_J \in \{0,1\}$. Similar to Eq. (6), we can write Boolean functions $f : \mathbb{F}_2^n \to \mathbb{Z}_q$ as

$$f(x) = \sum_{J\subseteq[n]} \alpha_J \prod_{i\in J} x_i \pmod q \tag{7}$$

for some integer coefficients $\alpha_J \in \{0,1,\ldots,q-1\}$. Throughout this paper, unless explicitly mentioned, we will be concerned with writing Boolean functions as a decomposition over $\mathbb{F}_2$ or $\mathbb{Z}_q$ with $q = 2^d$. The $\mathbb{F}_2$ degree of f is defined as

$$\deg(f) = \max\{|J|\colon \alpha_J \neq 0\}.$$

Similarly for polynomials over $\mathbb{Z}_{2^d}$, we can define the degree as the size of the largest monomial whose coefficient α_J is non-negative.

We will call $g : \mathbb{F}_2^n \to \mathbb{F}_2$ with $g = \prod_{i\in J} x_i$ as monic monomials over n variables of at most degree-d, characterized by set $J \subseteq [n]$, $|J| \leq d$. We will denote the set of these monic monomials by $\mathcal{M}(n,d)$. Note that $|\mathcal{M}(n,d)| = \sum_{j=0}^{d} \binom{n}{j} = O(n^d)$. We will denote the set of polynomials over n variables of $\mathbb{F}_2$-degree d as $\mathcal{P}(n,d)$. Note that these polynomials are just linear combinations of monomials in $\mathcal{M}(n,d)$. We will denote the set of polynomials over n variables of $\mathbb{F}_2$-degree d with sparsity s as $\mathcal{P}(n,d,s)$. Similarly, we will denote $\mathcal{P}_q(n,d)$ as the set of all degree-d Boolean polynomials $f : \mathbb{F}_2^n \to \mathbb{Z}_q$ with n variables. In particular, one can specify any polynomial $f \in \mathcal{P}_q(n,d)$ by $O(dn^d)$ bits and $|\mathcal{P}_q(n,d)| \leq 2^{O(dn^d)}$.

Consider a fixed d, and any $x \in \mathbb{F}_2^n$. Let the d-evaluation of x, denoted by $\mathrm{eval}_d(x)$, be a column vector in $\mathbb{F}_2^{|\mathcal{M}(n,d)|}$ with its elements being the evaluations of x under different monomials $g \in \mathcal{M}(n,d)$. This can be expressed as follows:

$$\mathrm{eval}_d(x) = \left(\prod_{i\in J\subseteq[n], |J|\leq d} x_i \right)^{\top} \tag{8}$$

For a set of points $\mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)}) \in (\mathbb{F}_2^n)^m$, we will call the matrix in $\mathbb{F}_2^{|\mathcal{M}(n,d)|\times m}$ with its kth column corresponding to d-evaluations of $x^{(k)}$, as the d-evaluation matrix of $\mathbf{x}$, and denote it by $Q_{\mathbf{x}}$.

2.3 Useful Lemmas

Let $e_i \in \mathbb{F}_2^n$ denote the vector of all zeros except for a 1 in the i^{th} coordinate.

▶ **Fact 1.** *Let $d \in [n]$, $s \leq |\mathcal{M}(n,d)| = \sum_{k=1}^{d} \binom{n}{k}$, and $f \in \mathcal{P}(n,d,s)$. There exists $g_i \in \mathcal{P}(n,d-1,s)$ such that $g_i(x) = f(x+e_i) + f(x) \pmod 2$ for all $x \in \{0,1\}^n$.*

The proof of this fact is straightforward. Without loss of generality, consider $i = 1$. For every $f(x) = \sum_S \alpha_S \prod_{i\in S} x_i$, we can express it as

$$f(x) = x_1 p_1(x_2,\ldots,x_n) + p_2(x_2,\ldots,x_n),$$

where p_1 has degree $\leq d-1$ and p_2 has degree $\leq d$. Observe that $f(x+e_1)-f(x)$ is either $p_1(x_2,\ldots,x_n)$ or $-p_1(x_2,\ldots,x_n)$ which has degree $d-1$ and corresponds to the polynomial g_1 in the fact statements. This applies for every coordinate i.

Note that the polynomial g_i above is also often called the *directional* derivative of f in direction w and is denoted as $D_i f$.

▶ **Fact 2.** *Let $N, s \geq 1$ such that $\gamma = s/N \leq 1/2$. Then we have*

$$\sum_{\ell=1}^{s} \binom{N}{\ell} \leq 2^{H_b(\gamma)N} \leq 2^{2\gamma \log(1/\gamma)}.$$

where we used above that $H_b(\gamma) = \gamma \log \frac{1}{\gamma} + (1-\gamma)\log \frac{1}{1-\gamma} \leq 2\gamma \log \frac{1}{\gamma}$ (for $\gamma \leq 1/2$).

▶ **Lemma 1** (The Schwartz-Zippel Lemma). *Let $p(y_1,\ldots,y_n)$ be a nonzero polynomial on n variables with degree d. Let S be a finite subset of $\mathbb{R}$, with at least d elements in it. If we assign $y_1,\ldots,y_n$ values from S independently and uniformly at random, then*

$$\Pr[p(y_1,\ldots,y_n)=0] \leq \frac{d}{|S|}. \tag{9}$$

▶ **Lemma 2** ([38]). *Let $p(x_1,\ldots,x_n)$ be a non-zero multilinear polynomial of degree d. Then*

$$\Pr_{x\in\{0,1\}^n}[p(x)=0] \leq 1-2^{-d},$$

where the probability is over a uniformly random distribution on $\{0,1\}^n$.

We will also need the following structural theorem about Reed-Muller codes which comments on the weight distribution of Boolean functions $f:\mathbb{F}_2^n \to \mathbb{F}_2$.

▶ **Theorem 3** ([2, Theorem 3]). *Let $n \geq 1$ and $d \leq n/2$. Define $|f| = \sum_{x\in\{0,1\}^n}[f(x)=1]$ and $\mathsf{wt}(f) = |f|/2^n$. Then, for every $\varepsilon \in (0,1/2)$ and $\ell \in \{1,\ldots,d-1\}$, we have that*

$$|\{f \in P(n,d) : \mathsf{wt}(f) \leq (1-\varepsilon)2^{-\ell}\}| \leq (1/\varepsilon)^{C\ell^4 \cdot \binom{n-\ell}{\leq d-\ell}}.$$

Fix $w = (1-\varepsilon)2^{n-\ell}$ and we get

$$|\{f \in P(n,d) : |f| \leq w\}| \leq (1-w/2^{n-\ell})^{-C\ell^4 \cdot \binom{n-\ell}{\leq d-\ell}}.$$

▶ **Lemma 4** (Fano's inequality). *Let A and B be classical random variables taking values in $\mathcal{X}$ (with $|\mathcal{X}| = r$) and let $q = \Pr[\mathsf{A} \neq \mathsf{B}]$. Then,*

$$H(\mathsf{A}|\mathsf{B}) \leq H_b(q) + q\log(r-1),$$

where $H(\mathsf{A}|\mathsf{B})$ is the conditional entropy and $H_b(q)$ is the standard binary entropy.

2.4 Measurements

Throughout this paper we will be concerned with learning algorithms that use either separable or entangled measurements. Given $|\psi_f\rangle^{\otimes k}$, a learning algorithm for f is said to use separable measurements if it only measure each copy of $|\psi_f\rangle$ separately in order to learn f. Similarly, a learning algorithm for f is said to use entangled measurements if it makes an entangled measurement on the k-fold tensor product $|\psi_f\rangle^{\otimes k}$. In this direction, we will often use two techniques which we discuss in more detail below: sampling random partial derivatives in order to learn from separable measurements and Pretty Good Measurements in order to learn from entangled measurements.

2.4.1 Separable Measurements

Below we discuss a subroutine that we will use often to learn properties about $f : \mathbb{F}_2^n \to \mathbb{F}_2$: given a single copy of $|\psi_f\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle$, the subroutine produces a uniformly random $y \in \mathbb{F}_2^{n-1}$ and $f(1y) + f(0y) \pmod 2$. To this end, suppose we measure qubits $2, 3, \ldots, n$ of $|\psi_f\rangle$ in the usual Z basis. We denote the resulting string as $y \in \{0,1\}^{n-1}$. The post-measurement state of qubit 1 is then given by

$$|\psi_{f,y}\rangle = \frac{1}{\sqrt{2}} \left[(-1)^{f(0y)}|0\rangle + (-1)^{f(1y)}|1\rangle\right]. \tag{10}$$

We note that $|\psi_{f,y}\rangle$ is then an X-basis state ($|+\rangle$ or $|-\rangle$) depending on the values of $f(1y)$ and $f(0y)$. If $f(1y) = f(0y)$, then $|\psi_{f,y}\rangle = |+\rangle$ and if $f(1y) = f(0y) + 1 \pmod 2$, then $|\psi_{f,y}\rangle = |-\rangle$. Measuring qubit 1 in the X-basis and qubits $2, 3, \ldots, n$ in the Z-basis thus produces examples of the form (y, b) where $y \in \{0,1\}^{n-1}$ is uniformly random and $b = f(0y) + f(1y) \pmod 2$. Considering Fact 1 with the basis of e_1, we note that theses examples are of the form $(y, D_1 f(y))$, where $D_1 f(y) = f(1y) + f(0y) \pmod 2$ is the partial derivative of f along direction e_1. Changing the measurement basis chosen above to $ZZ \cdots X_k \cdots Z$ such that we measure all the qubits in the Z basis except for the kth qubit which is measured in the X basis, will allow us to obtain random samples of the form $(y, D_k f(y))$. Accordingly, we introduce a new subroutine.

▶ **Definition 1** (Random Partial Derivative Sampling (RPDS) along e_k)**.** *For every $k \in [n]$, measuring every qubit of $|\psi_f\rangle$ in the Z basis, except the kth qubit which is measured in the X basis, we obtain a uniformly random $y \in \mathbb{F}_2^{n-1}$ and $(D_k f)(y)$.*

2.4.2 Entangled Measurements

In general one could also consider a joint measurement applied to multiple copies of $|\psi_f\rangle$, which we refer to as entangled measurements. In this work, we consider two types of entangled measurements, Bell sampling and the pretty-good measurement. We omit a detailed discussion on Bell sampling as we do not include the corresponding results for learning binary phase states under depolarizing noise (see the full version [6] for more).

Pretty Good Measurements. Consider an ensemble of states, $\mathcal{E} = \{(p_i, |\psi_i\rangle)\}_{i\in[m]}$, where $p = \{p_1, \ldots, p_m\}$ is a probability distribution. In the quantum state identification problem, a learning algorithm is given an unknown quantum state $|\psi_i\rangle \in \mathcal{E}$ sampled according to the distribution p and the learning algorithm needs to identity i with probability $\geq 2/3$. In this direction, we are interested in maximizing the average probability of success to identify i. For a POVM specified by positive semidefinite matrices $\mathcal{M} = \{M_i\}_{i\in[m]}$, the probability of obtaining outcome j equals $\langle\psi_i|M_j|\psi_i\rangle$ and the average success probability is given by

$$P_{\mathcal{M}}(\mathcal{E}) = \sum_{i=1}^{m} p_i \langle\psi_i|M_i|\psi_i\rangle.$$

Let $P^{opt}(\mathcal{E}) = \max_{\mathcal{M}} P_{\mathcal{M}}(\mathcal{E})$ denote the optimal average success probability of $\mathcal{E}$, where the maximization is over the set of valid m-outcome POVMs. For every ensemble $\mathcal{E}$, the so-called *Pretty Good Measurement* (PGM) is a specific POVM (depending on the ensemble $\mathcal{E}$) that does *reasonably* well against $\mathcal{E}$. In particular, it is well-known that

$$P^{opt}(\mathcal{E})^2 \leq P^{PGM}(\mathcal{E}) \leq P^{opt}(\mathcal{E}).$$

We now define the POVM elements of the pretty-good measurement. Let $|\psi_i'\rangle = \sqrt{p_i}|\psi_i\rangle$, and $\mathcal{E}' = \{|\psi_i'\rangle : i \in [m]\}$ be the set of states in $\mathcal{E}$, renormalized to reflect their probabilities. Define $\rho = \sum_{i\in[m]} |\psi_i'\rangle\langle\psi_i'|$. The PGM is defined as the set of measurement operators $\{|\nu_i\rangle\langle\nu_i|\}_{i\in[m]}$ where $|\nu_i\rangle = \rho^{-1/2}|\psi_i'\rangle$ (the inverse square root of ρ is taken over its non-zero eigenvalues). We will use the properties of these POVM elements later on and will also need the following theorems about PGMs.

▶ **Theorem 5** ([25]). *Let $\mathcal{S} = \{\rho_1, \ldots, \rho_m\}$. Suppose $\rho \in \mathcal{S}$ is an unknown quantum state picked from $\mathcal{S}$. Let $\max_{i\neq j} \|\sqrt{\rho_i}\sqrt{\rho_j}\|_1 \leq F$. Then, given*

$$M = O((\log(m/\delta))/\log(1/F))$$

copies of ρ, the Pretty good measurement identifies ρ with probability at least $1-\delta$.

The above theorem in fact implies the following stronger statement immediately (also stated in [8]) that we use here.

▶ **Lemma 6.** *Let $\mathcal{S} = \{\rho_1, \ldots, \rho_m\}$. Suppose $\rho \in \mathcal{S}$ is an unknown quantum state picked uniformly from $\mathcal{S}$. Suppose there exists k such that*

$$\frac{1}{m}\sum_{i\neq j} \|\sqrt{\rho_i^{\otimes k}}\sqrt{\rho_j^{\otimes k}}\|_1 \leq \delta,$$

then given k copies of ρ, the Pretty Good Measurement identifies ρ with probability at least $1-\delta$.

3 Learning Binary Phase States

In this section, we consider the problem of learning binary phase states as given by Eq. (1), assuming that f is a Boolean polynomial of $\mathbb{F}_2$-degree d.

3.1 Learning algorithm using separable measurements

We now describe our learning algorithm for learning binary phase states $|\psi_f\rangle$ when f has $\mathbb{F}_2$-degree d, using separable measurements. We carry out our algorithm in n rounds, which we index by t. In the t-th round, we perform RPDS along e_t (Def. 1) in order to obtain samples of the form $(y, D_t f(y))$ where $y \in \{0,1\}^{n-1}$. For an $m \geq 1$ to be fixed later, we use RPDS on m copies of $|\psi_f\rangle$ to obtain $\{(y^{(k)}, D_t f(y^{(k)}))\}_{k\in[m]}$ where $y^{(k)} \in \{0,1\}^{n-1}$ is uniformly random. We now describe how to learn $D_t f$ using these m samples.

Using Fact 1, we know that $D_t f \in \mathcal{P}(n-1, d-1)$. Thus, there are at most $N = |\mathcal{M}(n-1, d-1)| = \sum_{k=1}^{d-1} \binom{n}{k} = n^{O(d)}$ monomials in the $\mathbb{F}_2$ representation of $D_t f$. Let $A_t \in \mathbb{F}_2^{m\times N}$ be the transpose of the $(d-1)$-evaluation matrix (defined in Eq. (8)), such that the kth row of A_t corresponds to the evaluations of $y^{(k)}$ under all monomials in $\mathcal{M}(n-1, d-1)$, i.e., $(y_S^{(k)})_{|S|\leq d-1}$, where $y_S^{(k)} = \prod_{j\in S} y_j^{(k)}$, and let $\beta_t = (\alpha_S)_{|S|\leq d-1}$ be the vector of unknown coefficients. Obtaining $\{(y^{(k)}, D_t f(y^{(k)}))\}_{k\in[m]}$, allows one to solve $A_t\beta_t = D_t f(\mathbf{y})$ for β_t (where $\mathbf{y} = (y^{(1)}, \ldots, y^{(m)})$ and $(D_t f(\mathbf{y}))_k = D_t f(y^{(k)})$) and learn the $\mathbb{F}_2$-representation of $D_t f$ completely. Over n rounds, one then learns $D_1 f, D_2 f, \ldots, D_n f$. The $\mathbb{F}_2$-representations of these partial derivatives can then be used to learn f completely, as show in Fact 3. This procedure is shown in Algorithm 1.

▶ **Fact 3.** *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be such that $f \in \mathcal{P}(n, d)$. Learning $D_1 f, \ldots, D_n f$ suffices to learn f.*

Proof. Let the $\mathbb{F}_2$-representation of the unknown f be

$$f(x) = \sum_{J\subseteq[n],|J|\leq d} \alpha_J \prod_{i\in J} x_i. \tag{11}$$

The $\mathbb{F}_2$-representation of $D_t f$ for any $t \in \{1, 2, \ldots, n\}$ is then given by

$$D_t f(x) = \sum_{\substack{J\subseteq[n]:\\ t\in J,|J|\leq d}} \alpha_J \prod_{i\in J\setminus t} x_i, \tag{12}$$

where we notice that $D_t f$ only contains those monomials that correspond to sets J containing the component x_t. Let the $\mathbb{F}_2$-representation of $D_t f$ with the coefficient vector β_t be given by

$$D_t f(x) = \sum_{S\subset[n],|S|\leq d-1} (\beta_t)_S \prod_{i\in S} x_i. \tag{13}$$

Suppose an algorithm learns $D_1 f, \ldots, D_n f$. In order to learn f, we must retrieve the coefficients α_J from the learned coefficients $\{\beta_t\}_{t\in\{1,2,\ldots,n\}}$. We accomplish this by noting that $(\beta_t)_S = \alpha_{S\cup t}$ or in other words, $\alpha_J = \{\beta_t\}_{J\setminus t}$, $t \in J$. However, there may be multiple values of t that will allow us retrieve α_J. For example, suppose f contains the monomial term $x_1x_2x_3$ (i.e., $J = \{1, 2, 3\}$) then $\alpha_{\{1,2,3\}}$ could be retrieved from $(\beta_1)_{\{2,3\}}$, $(\beta_2)_{\{1,3\}}$, or $(\beta_3)_{\{1,2\}}$. When $D_t f$ (or β_t) for all t is learned with zero error, all these values coincide and it doesn't matter which learned coefficient is used. When there may be error in learning $D_t f$ (or β_t), we can carry out a majority vote: $\alpha_J = \text{Majority}(\{(\beta_t)_{J\setminus t} | t \in J\})$ for all $J \subseteq [n], |J| \leq d$. The majority vote is guaranteed to succeed as long as there is no error in at least half of the contributing β_t (which is the case in our learning algorithm). ◀

Algorithm 1 Learning binary phase states through separable measurements.

Input: Given $M = O((2n)^d)$ copies of $|\psi_f\rangle$ where $f \in \mathcal{P}(n, d)$

1: **for** qubit $t = 1, \ldots, n$ **do**
2: Set $m = M/n$
3: Perform RPDS along e_t to obtain $\{(y^{(k)}, D_t f(y^{(k)})\}_{k\in[m]}$ by measuring m copies of $|\psi_f\rangle$.
4: Solve the linear system of equations $A_t \cdot \beta_t = D_t f(\mathbf{y})$ to learn $D_t f$ explicitly.
5: **end for**
6: Use Fact 3 to learn f using $D_1 f, \ldots, D_n f$ (let $\tilde{f}$ be the output).

Output: Output $\tilde{f}$

We now prove the correctness of this algorithm.

▶ **Theorem 7.** *Let $n \geq 2, d \leq n/2$. Algorithm 1 uses $M = O(2^d n^d)$ copies of an unknown $|\psi_f\rangle$ for $f \in \mathcal{P}(n, d)$ and with high probability identifies f using single qubit X, Z measurements.*

Proof. Algorithm 1 learns f by learning $D_1 f, \ldots, D_n f$ and thereby learns f completely. Here we prove that each $D_t f$ can be learned with $m = O(2^d n^{d-1})$ copies of $|\psi_f\rangle$ and an exponentially small probability of error. This results in an overall sample complexity of $O(2^d n^d)$ for learning f and hence $|\psi_f\rangle$. Let us consider round t in Algorithm 1. We generate m constraints $\{(y^k, (D_t f)(y^{(k)}))\}_{k\in[m]}$ where $y^{(k)} \in \mathbb{F}_2^{n-1}$ by carrying out RPDS along e_t on m copies of $|\psi_f\rangle$.

We learn the $\mathbb{F}_2$-representation of $D_t f$ by setting up a linear system of equations using these m samples: $A_t \beta_t = D_t f(\mathbf{y})$, where A_t is the transposed $(d-1)$-evaluation matrix in round t, evaluated over $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots y^{(m)})$, and $\beta_t \in \mathbb{F}_2^{|\mathcal{M}(n-1,d-1)|}$ is the collective vector of coefficients corresponding to the monomials in $\mathcal{M}(n-1, d-1)$. By construction, this system has at least one solution. If there is exactly one solution, then we are done. Otherwise, the corresponding system has a non-zero solution, that is, there exists a non-zero degree-$(d-1)$ polynomial $g : \mathbb{F}_2^{n-1} \to \mathbb{F}_2$ such that $g(y^{(j)}) = 0$ for all $j = 1, 2, \dots, m$.

Below we prove that the probability of this bad event can be bounded through the Schwartz-Zippel lemma. Applying Lemma 2 and by noting that $y^j \in \mathbb{F}_2^{(n-1)}$ are independent and uniformly distributed, we have that

$$\Pr[g(y^{(1)}) = g(y^{(2)}) = \cdots = g(y^{(m)}) = 0] \leq (1 - 2^{-d})^m \leq e^{-m2^{-d}} \tag{14}$$

Let $\mathcal{P}_{\text{nnz}}(n, d)$ be the set of all degree-d polynomials $g : \mathbb{F}_2^n \to \mathbb{F}_2$ which are not identically zero. Define event

$$\mathsf{BAD}(y^1, \dots, y^m) = [\exists g \in \mathcal{P}_{\text{nnz}}(n-1, d-1) \,:\, g(y^1) = \dots = g(y^m) = 0 \pmod 2]. \tag{15}$$

We note that $|\mathcal{P}_{\text{nnz}}(n-1, d-1)| \leq 2^N$ where $N = O(n^{d-1})$. By union bound and Eq. (14), we have

$$\Pr[\mathsf{BAD}(y^{(1)}, \dots, y^{(m)})] \leq |\mathcal{P}_{\text{nnz}}(n-1, d-1)| \cdot (1 - 2^{-d})^m \leq 2^{n^{d-1} - m2^{-d}(\ln 2)}. \tag{16}$$

Thus choosing $m = O((2n)^{d-1})$ is enough to learn all coefficients $\{\alpha_J\}_{t \in J}$ (through β_t) in the $\mathbb{F}_2$ representation of f with an exponentially small probability of error. We need to repeat this over all the n qubits in order to learn $D_1 f, \dots, D_n f$ and then use Fact 3 to learn f completely. This gives an overall sample complexity of $O((2n)^d)$ for learning binary phase states. Observe that the only measurements that we needed in this algorithm were single qubit $\{X, Z\}$ measurements. ◀

▶ **Corollary 8.** *An n-qubit state $|\psi_f\rangle$ with the unknown Boolean function f of given Fourier-sparsity s can be learned with Algorithm 1 that consumes M copies of $|\psi_f\rangle$ with probability $1 - 2^{-\Omega(n)}$ provided that $M \geq O(sn^{\log s})$.*

The proof of this corollary simply follows from the following: for a Boolean function, the Fourier sparsity s of f is related to the $\mathbb{F}_2$-degree d of f [9] as $d \leq \log s$. Along with Theorem 7 we obtain the corollary.

3.2 Learning using entangled measurements

We now consider the problem of learning binary phase states using entangled measurements. We have the following result.

▶ **Theorem 9.** *Let $n \geq 2, d \leq n/2$. There exists an algorithm that uses $M = O((2n)^{d-1})$ copies of an unknown $|\psi_f\rangle$ for $f \in \mathcal{P}(n, d)$ and identifies f using entangled measurements with probability $\geq 2/3$. There is also a lower bound of $\Omega(n^{d-1})$ for learning these states.*

Proof. In order to prove this theorem, we follow the following steps. We first observe that the optimal measurement for our state distinguishing problem is the pretty good measurement (PGM). Second we observe that the success probability of the PGM is the same for *every* concept in the ensemble. We bound the success probability of the PGM using Corollary 6 we get our upper bound.

For $f \in \mathcal{P}(n,d)$, let U_f be the unitary defined as $U_f = \mathsf{diag}(\{(-1)^{f(x)}\}_x)$, that satisfies $U_f|+\rangle^n = |\psi_f\rangle$. Observe that the set $\{U_f\}_{f\in\mathcal{P}(n,d)}$ is an Abelian group. The ensemble we are interested in is $\mathcal{S} = \{U_f|+\rangle^n\}_{f\in\mathcal{P}(n,d)}$ and such an ensemble is called *geometrically uniform* if the $\{U_f\}$ is an Abelian group. A well-known result of Eldar and Forney [21] showed that the optimal measurement for state distinguishing a geometrically uniform (in particular $\mathcal{S}$) is the pretty-good measurement. We now show that the success probability of the PGM is the same for every state in the ensemble. In this direction, for $M \geq 1$, let $\sigma_f = |\psi_f\rangle\langle\psi_f|^{\otimes M}$. The POVM elements of the pretty good measurement $\{E_f : f \in \mathcal{P}(n,d)\}$ is given by the POVM elements $E_f = S^{-1/2}\sigma_f S^{-1/2}$ where $S = \sum_{f\in\mathcal{P}(n,d)} \sigma_f$. The probability that the PGM identifies the unknown σ_f is given by

$$\Pr(f) = \mathrm{Tr}(\sigma_f E_f) = \langle\psi_f^{\otimes M}|S^{-1/2}|\psi_f^{\otimes M}\rangle^2.$$

Our claim is that $\Pr(f)$ is the same for every $f \in \mathcal{P}(n,d)$. Using the Abelian property of the unitaries $\{U_f\}_f$, observe that $U_f|\psi_g\rangle = |\psi_{f\oplus g}\rangle$ for every $f,g \in \mathcal{P}(n,d)$. Thus, we have that $(U_f^{\otimes M})^\dagger S U_f^{\otimes M} = S$, which implies that$(U_f^{\otimes M})^\dagger S^{-1/2} U_f^{\otimes M} = S^{-1/2}$. Hence it follows that

$$\Pr(f) = (\langle+|^{\otimes M}(U_f^{\otimes M})^\dagger S^{-1/2} U_f^{\otimes M}|+\rangle^{\otimes M})^2 = (\langle+|^{\otimes M} S^{-1/2}|+\rangle^{\otimes M})^2 = \Pr(0),$$

for every $f \in \mathcal{P}(n,d)$. Finally, observe that $\langle\psi_f|\psi_g\rangle = \mathbb{E}_x\left[(-1)^{f(x)+g(x)}\right] = 1 - 2\Pr_x[f(x) \neq g(x)]$. Let $\mathcal{P}^*(n,d)$ be the set of non-constant polynomials in $\mathcal{P}(n,d)$. We now have the following

$$\frac{1}{2^{\binom{n}{\leq d}}} \sum_{\substack{f\neq g:\\ f,g\in P(n,d)}} \|\sqrt{\rho_f^{\otimes k}}\sqrt{\rho_g^{\otimes k}}\|_1 = \sum_{g\in P^*(n,d)} (1 - 2\Pr_x[g(x)=1])^{2k} = \sum_{g\in P^*(n,d)} (1-2\mathsf{wt}(g))^{2k}$$

which we can further upper bound as follows

$$\begin{aligned}
&\sum_{\ell=1}^{d-1} \sum_{g\in P^*(n,d)} (1-2|g|/2^n)^{2k} \cdot \Big[|g| \in [2^{n-\ell-1}, 2^{n-\ell}-1]\Big] \\
&= \sum_{g\in P^*(n,d)} (1-2|g|/2^n)^{2k} \cdot \Big[|g| \in [2^{n-2}, 2^{n-1}-1]\Big] \\
&\qquad + \sum_{\ell=2}^{d-1} \sum_{g\in P^*(n,d)} (1-2|g|/2^n)^{2k} \cdot \Big[|g| \in [2^{n-\ell-1}, 2^{n-\ell}-1]\Big] \\
&\leq 2^{n-1} 2^{-2k + C\binom{n-1}{\leq d-1}} + \sum_{\ell=2}^{d-1} (1-\frac{1}{2^\ell})^{2k} \sum_{g\in P^*(n,d)} \Big[|g| \leq 2^{n-\ell}\Big],
\end{aligned}$$

where the first equality used that the PGM has the same success probability for every $f,g \in \mathcal{P}(n,d)$, third equality used that $|g| \geq 2^{n-d}$ for any non-zero polynomial $g \in P(n,d)$ [33] and last inequality used Theorem 3. For $k = O(n^{d-1})$ (by picking a sufficiently large constant in $O(\cdot)$), the first term is at most $\leq 1/100$. To bound the second term, using Theorem 3 we have

$$\sum_{\ell=2}^{d-1} (1-\frac{1}{2^\ell})^{2k} \sum_{g\in P^*(n,d)} \Big[|g| \leq 2^{n-\ell}\Big] \leq \sum_{\ell=2}^{d-1} 2^{n-\ell} \exp(-2k/2^\ell + (n-\ell)\ell^4 \binom{n-\ell}{\leq d-\ell}).$$

Each term is $\exp(-n^{d-1})$ for $k = O(n^{d-1})$, so the overall sum is $\leq 1/100$. Corollary 6 implies our desired upper bound.

In order to see the lower bound, observe that each state $|\psi_f\rangle$ contains n bits of information and the goal of the learning algorithm is to learn an unknown f, i.e., obtain $O(n^d)$ bits of information. Hence by Holevo's theorem [27], one requires $\Omega(n^{d-1})$ copies of the unknown state for state identification.[2] ◀

3.3 Lower bounds

In the last section we saw that $\Theta(n^{d-1})$ many copies of $|\psi_f\rangle$ with degree-d are necessary and sufficient to learn f if we allowed only entangled measurements. Earlier we saw that $O(n^d)$ many copies of $|\psi_f\rangle$ sufficed to learn f using separable measurements. A natural question is: Can we learn f using fewer copies if we are restricted to using only separable measurements? In the theorem below, we provide a lower bound that complements our upper bound, thereby showing $\Theta(n^d)$ copies are necessary and sufficient to learn f using separable measurements.

▶ **Theorem 10.** *Let $2 \leq d \leq n/2$. Suppose there exists an algorithm that with probability $\geq 1/10$, learns an n-variate polynomial $f \in \mathcal{P}(n,d)$, given M copies of the phase state $|\psi_f\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle$, measuring each copy in an arbitrary orthonormal basis, and performing an arbitrary classical processing. Then $M = \Omega(\log|\mathcal{P}(n,d)|) = \Omega(n^d)$.*

Proof. The proof is given in the full version of this paper [6]. ◀

4 Learning generalized phase states

In this section, we consider the problem of learning generalized phase states $|\psi_f\rangle$ as given by Eq. (3), assuming that f is a degree-d $\mathbb{Z}_q$-valued polynomial, $f \in \mathcal{P}_q(n,d)$. Note that since our goal is to learn $|\psi_f\rangle$ up to an overall phase, we shall identify polynomials which differ only by a constant shift.

▶ **Definition 2.** *Polynomials $f, g \in \mathcal{P}_q(n,d)$ are equivalent if $f(x) - g(x)$ is a constant.*

To simplify notation, here and below we omit modulo operations keeping in mind that degree-d polynomials take values in the ring $\mathbb{Z}_q$. Thus all equal or not-equal constraints that involve a polynomial's value are modulo q.

4.1 Learning using separable measurements

Let $q \geq 2$ and $d \geq 1$ be integers. For technical reasons, we shall assume that q is even. Let $\omega_q = e^{2\pi i/q}$. Our main result is as follows.

▶ **Theorem 11.** *Let $d \leq n/2$. There exists an algorithm that uses $M = O(2^d q^3 n^d \log q) = O(n^d)$ copies of a generalized phase state $|\psi_f\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} \omega_q^{f(x)}|x\rangle$ with an unknown polynomial $f \in \mathcal{P}_q(n,d)$ and outputs a polynomial $g \in \mathcal{P}_q(n,d)$ such that g is equivalent to f with the probability at least $1 - 2^{-\Omega(n)}$. The quantum part of the algorithm requires only single-qubit unitary gates and measurements in the standard basis.*

Moreover, suppose there exists an algorithm that with probability $\geq 1/10$, learns an n-variate polynomial $f \in \mathcal{P}_q(n,d)$, given k copies of $|\psi_f\rangle$, measuring each copy in an arbitrary orthonormal basis, and performing an arbitrary classical processing. Then $M = \Omega(n^d)$.

[2] We refer the reader to Montanaro [35, Proposition 1] for a detailed exposition of this lower bound proof.

Before stating our learning algorithm and sample complexity, we need the following lemmas.

▶ **Lemma 12.** *Choose any $f \in \mathcal{P}_q(n,d)$ and $c \in \mathbb{Z}_q$. Then either $f(x)$ is a constant function or the fraction of inputs $x \in \{0,1\}^n$ such that $f(x) \neq c$ is at least $1/2^d$.*

Proof. We shall use the following simple fact which is proved in [6].

▶ **Proposition 1.** *Consider a function $f : \{0,1\}^n \to \mathbb{Z}_q$ specified as a polynomial*

$$f(x) = \sum_{J \subseteq [n]} \alpha_J \prod_{j \in J} x_j \pmod q. \tag{17}$$

Here $\alpha_J \in \mathbb{Z}_q$ are coefficients. The function f is constant if and only if $\alpha_J = 0 \pmod q$ for all non-empty subsets $J \subseteq [n]$.

We shall prove Lemma 12 by induction in n. The base case of induction is $n = d$. Clearly, a non-constant function $f : \{0,1\}^d \to \mathbb{Z}_q$ takes a value different from c at least one time, that is, the fraction of inputs $x \in \{0,1\}^d$ such that $f(x) \neq c$ is at least $1/2^d$.

Suppose $n > d$ and $f \in \mathcal{P}_q(n,d)$ is not a constant function. Let d' be the maximum degree of non-zero monomials in f. Clearly $1 \leq d' \leq d$. Suppose f contains a monomial $\alpha_S \prod_{j \in S} x_j$ where $\alpha_S \in \mathbb{Z}_q \setminus \{0\}$ and $|S| = d'$. Since $|S| < n$, one can choose a variable x_i with $i \in [n] \setminus S$. Let $g_a : \{0,1\}^{n-1} \to \mathbb{Z}_q$ be a function obtained from f by setting the variable x_i to a constant value $a \in \{0,1\}$. Clearly, $g_a \in \mathcal{P}_q(n-1,d)$. The coefficients of the monomial $\prod_{j \in S} x_j$ in g_0 and g_1 are α_S and $\alpha_S + \alpha_{S \cup \{i\}} \pmod q$ respectively. However, $\alpha_{S \cup \{i\}} = 0 \pmod q$ since otherwise f would contain a monomial $x_i \prod_{j \in S} x_j$ of degree larger than d'. We conclude that both g_0 and g_1 contain a non-zero monomial $\alpha_S \prod_{j \in S} x_j$. By Proposition 1, g_0 and g_1 are not constant functions. Since g_0 and g_1 are degree-d polynomials in $n-1$ variables, the induction hypothesis gives

$$\Pr_y[g_a(y) \neq c] \geq \frac{1}{2^d}. \tag{18}$$

Here $y \in \{0,1\}^{n-1}$ is picked uniformly at random. Thus

$$\Pr_x[f(x) \neq c] = \frac{1}{2}\left[\Pr_y[g_0(y) \neq c] + \Pr_y[g_1(y) \neq c]\right] \geq \frac{1}{2^d}. \tag{19}$$

Here $x \in \{0,1\}^n$ is picked uniformly at random. This proves the induction step. ◀

With this lemma, we are now ready to prove Theorem 11. In the section below we first describe our learning algorithm and in the next section we prove the theorem by proving the sample complexity upper bound.

4.1.1 Learning Algorithm in Theorem 11

We are now ready to state our learning algorithm. As in Section 3.1 for learning binary phase states with separable measurements, we learn generalized phase states through examples containing information about the derivatives of $f(x)$. The crucial difference between the binary phase state learning algorithm and the generalized setting is, in the binary case, we obtained a measurement outcome b_y that corresponded to $b_y = f(0y) - f(1y)$, however in the generalized scenario, we obtain a measurement outcome b'_y that satisfies $f(0y) - f(1y) \neq b'_y$. Nevertheless, we are able to still learn f using such measurement outcomes which we describe in the rest of the section.

We now describe the learning algorithm. We carry out the algorithm in n rounds, which we index by t. For simplicity, we describe the procedure for the first round. Suppose we measure qubits $2, 3, \ldots, n$ of the state $|\psi_f\rangle$ in the Z-basis. Let $y \in \{0,1\}^{n-1}$ be the measured bit string. Note that the probability distribution of y is uniform. The post-measurement state of qubit 1 is

$$|\psi_{f,y}\rangle = \frac{1}{\sqrt{2}}(\omega_q^{f(0y)}|0\rangle + \omega_q^{f(1y)}|1\rangle) \tag{20}$$

For each $b \in \mathbb{Z}_q$ define a single-qubit state

$$|\phi_b\rangle = \frac{1}{\sqrt{2}}(|0\rangle - \omega_q^b|1\rangle) \tag{21}$$

Using the identity $\sum_{b\in\mathbb{Z}_q} \omega_q^b = 0$ one gets

$$I = \frac{2}{q}\sum_{b\in\mathbb{Z}_q} |\phi_b\rangle\langle\phi_b| \tag{22}$$

One can view Eq. (22) as a single-qubit POVM with q elements $(2/q)|\phi_b\rangle\langle\phi_b|$. Let $\mathcal{M}$ be the single-qubit measurement described by this POVM. Applying $\mathcal{M}$ to the state $|\psi_{f,y}\rangle$ returns an outcome $b \in \mathbb{Z}_q$ with the probability

$$\Pr(b|y) := \frac{2}{q}|\langle\phi_b|\psi_{f,y}\rangle|^2 = \frac{1}{2q}\left|1 - \omega_q^{f(1y)-f(0y)-b}\right|^2. \tag{23}$$

Clearly, $\Pr(b|y)$ is a normalized probability distribution, $\sum_{b\in\mathbb{Z}_q}\Pr(b|y) = 1$. Furthermore,

$$f(1y) - f(0y) = b \quad \text{implies } \Pr(b|y) = 0, \tag{24}$$

$$f(1y) - f(0y) \neq b \quad \text{implies } \Pr(b|y) \geq \frac{2}{q}\sin^2(\pi/q) = \Omega(1/q^3). \tag{25}$$

To conclude, the combined n-qubit measurement consumes one copy of the state $|\psi_f\rangle$ and returns a pair $(y, b) \in \{0,1\}^{n-1} \times \mathbb{Z}_q$ such that

$$f(1y) - f(0y) \neq b \tag{26}$$

with certainty and all outcomes b satisfying Eq. (26) appear with a non-negligible probability. Define a function $g : \{0,1\}^{n-1} \to \mathbb{Z}_q$ such that

$$g(y) = f(1y) - f(0y). \tag{27}$$

We claim that g is a degree-$(d-1)$ polynomial, that is, $g \in \mathcal{P}_q(n-1, d-1)$. Indeed, it is clear that $g(y)$ is a degree-d polynomial. Moreover, all degree-d monomials in $f(x)$ that do not contain the variable x_1 appear in $f(1y)$ and $f(0y)$ with the same coefficient. Such monomials do not contribute to $g(y)$. A degree-d monomial in $f(x)$ that contains the variable x_1 contributes a degree-$(d-1)$ monomial to $g(y)$. Thus $g \in \mathcal{P}_q(n-1, d-1)$, as claimed.

From Eq. (26) one infers a constraint $g(y) \neq b$whenever the combined n-qubit measurement of $|\psi_f\rangle$ returns an outcome (y, b). Suppose we repeat the above process m times obtaining constraints

$$g(y^{(k)}) \neq b^{(k)}, \qquad k = 1, 2, \ldots, m. \tag{28}$$

This consumes m copies of $|\psi_f\rangle$. We claim that the probability of having more than one polynomial $g \in \mathcal{P}_q(n-1, d-1)$ satisfying the constraints Eq. (28) is exponentially small if we choose

$$m = O(q^3 \log(q) 2^d n^{d-1}). \tag{29}$$

4.1.2 Sample Complexity bound in Theorem 11

Define a probability distribution $\pi(\vec{y},\vec{b})$ where

$$\vec{z}=(y^{(1)},\dots,y^{(m)})\in\{0,1\}^{(n-1)m}\quad\text{and}\qquad \vec{b}=(b^{(1)},\dots,b^{(m)})\in(\mathbb{Z}_q)^{\times m} \tag{30}$$

such that $y^{(j)}$ are picked uniformly at random and $b^{(k)}$ are sampled from the distribution $\Pr(b^{(k)}|y^{(k)})$ defined in Eq. (23). For each polynomial $h\in\mathcal{P}_q(n-1,d-1)$ define an event

$$\mathsf{BAD}(h)=\{(\vec{y},\vec{b})\,:\,h(y^{(k)})\neq b^{(k)}\quad\text{for all}\quad k\in[m]\}. \tag{31}$$

We claim that

$$\Pr[\mathsf{BAD}(h)]:=\sum_{(\vec{y},\vec{b})\in\mathsf{BAD}(h)}\pi(\vec{y},\vec{b})\leq\left[1-\Omega(2^{-d}q^{-3})\right]^m \tag{32}$$

for any $h\neq g$. Indeed, consider some fixed $k\in[m]$. The event $b^{(k)}\neq h(y^{(k)})$ occurs automatically if $h(y^{(k)})=g(y^{(k)})$. Otherwise, if $h(y^{(k)})\neq g(y^{(k)})$, the event $b^{(k)}\neq h(y^{(k)})$ occurs with the probability at most $1-\Omega(1/q^3)$ since $b^{(k)}=h(y^{(k)})$ with the probability at least $\Omega(1/q^3)$ due to Eq. (25). It follows that

$$\Pr_{y^{(k)},b^{(k)}}[h(y^{(k)})\neq b^{(k)}]\leq\Pr_{y^{(k)}}[h(y^{(k)})=g(y^{(k)})]+\Pr_{y^{(k)}}[h(y^{(k)})\neq g(y^{(k)})]\left(1-\Omega(1/q^3)\right) \tag{33}$$

$$=1-\Pr_{y^{(k)}}[h(y^{(k)})\neq g(y^{(k)})]\cdot\Omega(1/q^3). \tag{34}$$

If h and g are equivalent then $h(y)=g(y)+c$ for some constant $c\in\mathbb{Z}_q$. Note that $c\neq 0$ since we assumed $h\neq g$. In this case

$$\Pr_{y^{(k)}}[h(y^{(k)})\neq g(y^{(k)})]=1. \tag{35}$$

If h and g are non-equivalent, apply Lemma 12 to a non-constant degree-$(d-1)$ polynomial $h-g$. It gives

$$\Pr_{y^{(k)}}[h(y^{(k)})\neq g(y^{(k)})]\geq\frac{1}{2^{d-1}}. \tag{36}$$

In both cases we get

$$\Pr_{y^{(k)},b^{(k)}}[h(y^{(k)})\neq b^{(k)}]\leq 1-\Omega(2^{-d}q^{-3}), \tag{37}$$

which proves Eq. (32) since the pairs $(y^{(k)},b^{(k)})$ are i.i.d. random variables.

As noted earlier in the preliminaries, observe that $|\mathcal{P}_q(n-1,d-1)|\leq q^{O(n^{d-1})}=2^{O(\log(q)n^{d-1})}$. By the union bound, one can choose $m=O(2^dq^3\log(q)n^{d-1})$ such that

$$\Pr\left[\bigcup_{h\in\mathcal{P}_q(n-1,d-1)\setminus g}\mathsf{BAD}(h)\right]\leq 2^{O(\log(q)n^{d-1})}\left[1-\Omega(2^{-d}q^{-3})\right]^m\leq 2^{-\Omega(n)}. \tag{38}$$

In other words, the probability that g is the unique element of $\mathcal{P}_q(n-1,d-1)$ satisfying all the constraints Eq. (28) is at least $1-2^{-\Omega(n)}$. One can identify such polynomial g by checking the constraints Eq. (28) for every $g\in\mathcal{P}_q(n-1,d-1)$. If the constraints are satisfied for more than one polynomial, declare a failure.

At this point we have learned a polynomial $g \in \mathcal{P}_q(n-1, d-1)$ such that $f(1y) - f(0y) = g(y)$ for all $y \in \{0,1\}^{n-1}$. For simplicity, we ignore the exponentially small failure probability. Applying the same protocol n times to copies of the quantum state $|\psi_f\rangle$ by a cyclic shift of qubits, one can learn polynomials $g_0, g_1, \ldots, g_{n-1} \in \mathcal{P}_q(n-1, d-1)$ such that

$$f(C^i(1y)) - f(C^i(0y)) = g_i(y) \quad \text{for all} \quad i \in [n] \quad \text{and} \quad y \in \{0,1\}^{n-1}, \tag{39}$$

where C is the cyclic shift of n bits. This consumes $M = O(nm) = O(2^d q^3 \log(q) n^d)$ copies of the state $|\psi_f\rangle$. We can assume wlog that $f(0^n) = 0$ since our goal is to learn $f(x)$ modulo a constant shift. Suppose we have already learned values of $f(x)$ for all bit strings x with the Hamming weight $|x| \leq w$ (initially $w = 0$). Any bit string x with $|x| = w+1$ can be represented as $x = C^i(1y)$ for some $y \in \{0,1\}^{n-1}$ such that $|y| = w$. Now Eq. (39) determines $f(x)$ since $|C^i(0y)| = |y| = w$ so that $f(C^i(0y))$ is already known and the polynomial $g_i(y)$ has been learned. Proceeding inductively one can learn $f(x)$ for all x.

It remains to note that the POVM Eq. (22) is a probabilistic mixture of projective single-qubit measurements whenever q is even. Indeed, in this case the states $|\phi_b\rangle$ and $|\phi_{b+q/2}\rangle = Z|\phi_b\rangle$ form an orthonormal basis of a qubit, see Eq. (21). Thus the POVM defined in Eq. (22) can be implemented by picking a random uniform $b \in \mathbb{Z}_q$ and measuring a qubit in the basis $\{|\phi_b\rangle, Z|\phi_b\rangle\}$. Thus the learning protocol only requires single-qubit unitary gates and measurements in the standard basis.

The lower bound in the proof of Theorem 11 follows in a straightforward manner from the lower bound for binary phase states. Indeed, suppose

$$f'(x) = \sum_{J \in [n]} \alpha_J \prod_{j \in J} x_j \pmod 2$$

is an $\mathbb{F}_2$-valued degree-d polynomial, $f' \in \mathcal{P}(n,d)$. Suppose $q = 2r$ for some integer r. Define a polynomial $f(x) = rf'(x) \pmod q$. Clearly $f \in \mathcal{P}_q(n,d)$ and $\omega_q^{f(x)} = (-1)^{f'(x)}$ for all x, that is the binary phase state corresponding to f' coincides with the generalized phase state corresponding to f. Using Theorem 10, we obtain a lower bound of $M = \log|\mathcal{P}(n,d)| = \Omega(n^d)$ for learning ψ_f. This concludes the proof of Theorem 11.

4.2 Learning stabilizer states

We now describe how the algorithm stated in Theorem 11 could be used to learn any n-qubit stabilizer state (produced by a Clifford circuit applied to $|0^n\rangle$ state) using separable measurements. Note that we can learn a subclass of stabilizer states called graph states (which are simply binary phase states with $d = 2$) using Algorithm 1 with the sample complexity of $O(n^2)$ (as shown in Theorem 7).

From a result in [19], we know that a stabilizer state can be represented as follows

$$|\psi\rangle = \frac{1}{\sqrt{|A|}} \sum_{x \in A} i^{\ell(x)} (-1)^{q(x)} |x\rangle, \tag{40}$$

where A is an affine subspace of $\mathbb{F}_2^n$, $\ell : \mathbb{F}_2^n \to \mathbb{F}_2$ is a linear function and $q : \mathbb{F}_2^n \to \mathbb{F}_2$ is quadratic function. Clearly, an alternate form is a generalized phase state with degree-2

$$|\psi_f\rangle = \frac{1}{\sqrt{|A|}} \sum_{x \in A} i^{f(x)} |x\rangle \tag{41}$$

where the summation is over A instead of the entire $\mathbb{F}_2^n$, and the function $f : \mathbb{F}_2^n \to \mathbb{Z}_4$ has its coefficients corresponding to the quadratic monomials take values in $\{0, 2\}$. We can now learn this using separable measurements as stated in the following statement as opposed to entangled measurements as required by Bell sampling [36].

▶ **Corollary 13.** *There exists an algorithm that uses $M = O(n^2)$ copies of a stabilizer state $|\psi_f\rangle = \frac{1}{\sqrt{|A|}} \sum_{x \in A} i^{f(x)}|x\rangle$ with an unknown polynomial $f \in \mathcal{P}_4(n, 2)$ and outputs a polynomial $g \in \mathcal{P}_4(n, 2)$ such that g is equivalent to f with the probability at least $1 - 2^{-\Omega(n)}$. The quantum part of the algorithm requires only single-qubit unitary gates and measurements in the standard basis.*

Proof. The subspace A of an unknown stabilizer state can be denoted as $a + S_A$ where $a \in \mathbb{F}_2^n$ is a translation vector and S_A is a linear subspace of $\mathbb{F}_2^n$. To learn a and a basis of the subspace S_A, it is enough to measure $O(n \log n)$ copies of $|\psi_f\rangle$ in the computational basis. This in turn defines a subset of the n directions $\{e_i\}$ along which we need to search for non-zero monomials in the partial derivatives of f. We can now use the learning algorithm in Theorem 11 to learn the unknown stabilizer state using $O(n^2)$ copies with the desired probability. ◀

References

1 Emmanuel Abbe, Amir Shpilka, and Avi Wigderson. Reed-Muller Codes for Random Erasures and Errors. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 297–306, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2746539.2746575.

2 Emmanuel Abbe, Amir Shpilka, and Min Ye. Reed–Muller Codes: Theory and Algorithms. *IEEE Transactions on Information Theory*, 67(6):3251–3277, 2020. doi:10.1109/TIT.2020.3004749.

3 Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. *arXiv preprint arXiv:2112.10020*, 2021.

4 Anurag Anshu, Srinivasan Arunachalam, Tomotaka Kuwahara, and Mehdi Soleimanifar. Sample-efficient learning of interacting quantum systems. *Nature Physics*, 17(8):931–935, 2021.

5 Apeldoorn van Joran, Arjan Cornelissen, Andras Gilyén, and Giacomo Nannicini. Quantum tomography using state-preparation unitaries, 2022. arXiv:2207.08800.

6 Srinivasan Arunachalam, Sergey Bravyi, Arkopal Dutt, and Theodore J Yoder. Optimal algorithms for learning quantum phase states. *arXiv preprint arXiv:2208.07851*, 2022.

7 Dave Bacon, Andrew M Childs, and Wim van Dam. Optimal measurements for the dihedral hidden subgroup problem. *arXiv preprint quant-ph/0501044*, 2005.

8 Howard Barnum and Emanuel Knill. Reversing quantum dynamics with near-optimal quantum and classical fidelity. *Journal of Mathematical Physics*, 43(5):2097–2106, 2002.

9 A. Bernasconi and B. Codenotti. Spectral analysis of Boolean functions as a graph eigenvalue problem. *IEEE Transactions on Computers*, 48(3):345–351, 1999. doi:10.1109/12.755000.

10 Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

11 Zvika Brakerski and Omri Shmueli. (Pseudo) Random Quantum States with Binary Phase. In *Theory of Cryptography Conference*, pages 229–250. Springer, 2019.

12 Sergey Bravyi and David Gosset. Improved classical simulation of quantum circuits dominated by Clifford gates. *Physical Review Letters*, 116(25):250501, 2016.

13 Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):459–472, 2011.

14 Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Physical Review Letters*, 117(8):080501, 2016.

15 Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. Achieving quantum supremacy with sparse and noisy commuting quantum computations. *Quantum*, 1:8, 2017.

16 Isaac L Chuang and Michael A Nielsen. Prescription for experimental determination of the dynamics of a quantum black box. *Journal of Modern Optics*, 44(11-12):2455–2467, 1997. doi:10.1080/09500349708231894.

17 Marcus Cramer, Martin B Plenio, Steven T Flammia, Rolando Somma, David Gross, Stephen D Bartlett, Olivier Landon-Cardinal, David Poulin, and Yi-Kai Liu. Efficient quantum state tomography. *Nature Communications*, 1(1):1–7, 2010.

18 Shawn X Cui, Daniel Gottesman, and Anirudh Krishna. Diagonal gates in the Clifford hierarchy. *Physical Review A*, 95(1):012329, 2017.

19 Jeroen Dehaene and Bart De Moor. Clifford group, stabilizer states, and linear and quadratic operations over GF(2). *Physical Review A*, 68(4):042318, 2003.

20 Stark C. Draper and Sheida Malekpour. Compressed sensing over finite fields. In *2009 IEEE International Symposium on Information Theory*, pages 669–673. IEEE, 2009. doi:10.1109/ISIT.2009.5205666.

21 Yonina C Eldar and G David Forney. On quantum detection and the square-root measurement. *IEEE Transactions on Information Theory*, 47(3):858–872, 2001.

22 Daniel Gottesman and Isaac L Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390–393, 1999.

23 Jeongwan Haah, Aram W Harrow, Zhengfeng Ji, Xiaodi Wu, and Nengkun Yu. Sample-optimal tomography of quantum states. *IEEE Transactions on Information Theory*, 63(9):5628–5641, 2017.

24 Jeongwan Haah, Robin Kothari, and Ewin Tang. Optimal learning of quantum Hamiltonians from high-temperature Gibbs states. *arXiv preprint arXiv:2108.04842*, 2021.

25 Aram W Harrow and Andreas Winter. How many copies are needed for state discrimination? *IEEE Transactions on Information Theory*, 58(1):1–2, 2012.

26 Marc Hein, Jens Eisert, and Hans J Briegel. Multiparty entanglement in graph states. *Physical Review A*, 69(6):062311, 2004.

27 Alexander Semenovich Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973.

28 Sandy Irani, Anand Natarajan, Chinmay Nirkhe, Sujit Rao, and Henry Yuen. Quantum search-to-decision reductions and the state synthesis problem. *arXiv preprint arXiv:2111.02999*, 2021.

29 Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In *Annual International Cryptology Conference*, pages 126–152. Springer, 2018.

30 Ching-Yi Lai and Hao-Chung Cheng. Learning quantum circuits of some T gates. *IEEE Transactions on Information Theory*, 68(6):3951–3964, 2022.

31 Richard A Low. Learning and testing algorithms for the Clifford group. *Physical Review A*, 80(5):052314, 2009.

32 Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *Approximation, randomization and combinatorial optimization. Algorithms and techniques*, pages 378–389. Springer, 2005.

33 Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error correcting codes*, volume 16. Elsevier, 1977.

34 Masoud Mohseni, Ali T Rezakhani, and Daniel A Lidar. Quantum-process tomography: Resource analysis of different strategies. *Physical Review A*, 77(3):032322, March 2008. doi:10.1103/PhysRevA.77.032322.

35 Ashley Montanaro. The quantum query complexity of learning multilinear polynomials. *Information Processing Letters*, 112(11):438–442, 2012.

36 Ashley Montanaro. Learning stabilizer states by Bell sampling. *arXiv preprint arXiv:1707.04012*, 2017.

37 Ashley Montanaro. Quantum circuits and low-degree polynomials over $\mathbb{F}_2$. *Journal of Physics A: Mathematical and Theoretical*, 50(8):084002, January 2017. doi:10.1088/1751-8121/aa565f.

38 Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994.

39 Leonardo Novo, Juani Bermejo-Vega, and Raúl García-Patrón. Quantum advantage from energy measurements of many-body quantum systems. *Quantum*, 5:465, 2021.

40 Ryan O'Donnell and John Wright. Efficient quantum tomography. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 899–912, 2016.

41 Andrea Rocchetto, Scott Aaronson, Simone Severini, Gonzalo Carvacho, Davide Poderini, Iris Agresti, Marco Bentivegna, and Fabio Sciarrino. Experimental learning of quantum states. *Science Advances*, 5(3):eaau1946, 2019.

42 Matteo Rossi, Marcus Huber, Dagmar Bruß, and Chiara Macchiavello. Quantum hypergraph states. *New Journal of Physics*, 15(11):113022, 2013.

43 Martin Rötteler. Quantum algorithms to solve the hidden shift problem for quadratics and for functions of large Gowers norm. In *International Symposium on Mathematical Foundations of Computer Science*, pages 663–674. Springer, 2009.

44 Dirk Schlingemann. Stabilizer codes can be realized as graph codes. *Quantum Info. Comput.*, 2(4):307–323, June 2002.

45 Yuki Takeuchi, Tomoyuki Morimae, and Masahito Hayashi. Quantum computational universality of hypergraph states with Pauli-X and Z basis measurements. *Scientific Reports*, 9(1):1–14, 2019.

46 Henry Yuen. An improved sample complexity lower bound for quantum state tomography. *arXiv preprint arXiv:2206.11185*, 2022.

Computational Quantum Secret Sharing

Alper Çakan
Carnegie Mellon University, Pittsburgh, PA, USA

Vipul Goyal
NTT Research, Sunnyvale, CA, USA
Carnegie Mellon University, Pittsburgh, PA, USA

Chen-Da Liu-Zhang
NTT Research, Sunnyvale, CA, USA

João Ribeiro
NOVA LINCS and NOVA School of Science and Technology, Caparica, Portugal

Abstract

Quantum secret sharing (QSS) allows a dealer to distribute a secret *quantum state* among a set of parties in such a way that certain authorized subsets can reconstruct the secret, while unauthorized subsets obtain no information about it. Previous works on QSS for general access structures focused solely on the *existence* of perfectly secure schemes, and the share size of the known schemes is necessarily exponential even in cases where the access structure is computed by polynomial size monotone circuits. This stands in stark contrast to the classical setting, where *polynomial-time* computationally-secure secret sharing schemes have been long known for all access structures computed by polynomial-size monotone circuits under standard hardness assumptions, and one can even obtain shares which are much shorter than the secret (which is impossible with perfect security).

While QSS was introduced over twenty years ago, previous works only considered information-theoretic privacy. In this work, we initiate the study of *computationally-secure* QSS and show that computational assumptions help significantly in building QSS schemes, just as in the classical case. We present a simple compiler and use it to obtain a large variety results: We construct *polynomial-time* computationally-secure QSS schemes under standard hardness assumptions for a rich class of access structures. This includes many access structures for which previous results in QSS necessarily required exponential share size. In fact, we can go even further: We construct QSS schemes for which the size of the quantum shares is significantly smaller than the size of the secret. As in the classical setting, this is impossible with perfect security.

We also apply our compiler to obtain results beyond computational QSS. In the information-theoretic setting, we improve the share size of perfect QSS schemes for a large class of n-party access structures to $1.5^{n+o(n)}$, improving upon best known schemes and matching the best known result for general access structures in the classical setting. Finally, among other things, we study the class of access structures which can be efficiently implemented when the quantum secret sharing scheme has access to a given number of copies of the secret, including all such functions in P and NP.

2012 ACM Subject Classification Theory of computation → Cryptographic primitives; Theory of computation → Quantum computation theory

Keywords and phrases Quantum secret sharing, quantum cryptography

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.4

Related Version *Full Version*: `https://eprint.iacr.org/2023/613` [14]

Funding Research supported by the following grants of Vipul Goyal: NSF award 1916939, DARPA SIEVE program, a gift from Ripple, a DoE NETL award, a JP Morgan Faculty Fellowship, a PNC center for financial services innovation award, and a Cylab seed funding award.
Chen-Da Liu-Zhang: Part of the work was done while at Carnegie Mellon University.
João Ribeiro: Part of the work was done while at Carnegie Mellon University. Research also supported by NOVA LINCS (UIDB/04516/2020) with the financial support of FCT.IP.

1 Introduction

Classical secret sharing [12, 38] is a fundamental cryptographic primitive which allows one to share a classical bit-string (the secret) among n parties so that (i) only authorized subsets of parties can reconstruct the secret, and (ii) unauthorized subsets have essentially no information about the secret. The associated class of authorized sets, called the *access structure*, is defined by a monotone function $f : \{0,1\}^n \to \{0,1\}$ with a set $P \subseteq [n]$ being authorized if and only if $f(v^P) = 1$, where $v^P \in \{0,1\}^n$ is the characteristic vector of P satisfying $v^P_i = 1$ exactly when $i \in P$. For the sake of convenience, we will often write $f(P)$ for $f(v^P)$ when the context is clear.

Secret sharing has found several applications in cryptography, see the extensive survey of Beimel [9] for a discussion of such applications. Motivated by these applications, it is important to design secret sharing schemes realizing a given monotone function which are as efficient as possible, be it in terms of requiring polynomial-time sharing and reconstruction procedures or, more modestly, requiring that the resulting shares be as short as possible. Blakley [12] and Shamir [38] originally described efficient secret sharing schemes for *threshold* monotone functions (where a set P is authorized if and only if $|P| \geq t$ for a given threshold t). A long line of research over the past 40 years has significantly extended and complemented these results.

Of particular importance to us, it is known how to exploit different computational hardness assumptions to obtain efficient *computational* secret sharing schemes realizing a broad class of monotone functions, where by *efficient* we mean that the scheme enjoys polynomial-time sharing and reconstruction. In particular, such schemes also have polynomial sized shares. Early work of Yao [44, 41] described families of efficient *computational* secret sharing schemes realizing all functions in monotone P, i.e., all sequences of monotone functions $(f_n)_{n\in\mathbb{Z}^+}$ with $f_n : \{0,1\}^n \to \{0,1\}$ that are computed by a sequence of poly(n)-size monotone circuits,[1] based on the existence of one-way functions. A more recent work by Komargodski, Naor, and Yogev [28] succeeded in designing such efficient computational schemes realizing all functions in[2] mNP assuming witness encryption for NP and one-way functions.[3] Following this, Bartusek and Malavolta [8] designed efficient computational schemes for *classical* secrets realizing all monotone functions in QMA assuming witness encryption for this class.

Given the advent of quantum computing, it is natural to consider the construction of schemes for sharing a *quantum state*, as opposed to a classical bitstring. This problem was first considered in [24, 26, 17] for some specific monotone functions. Follow-up works by Gottesman [21] and Smith [39] showed how to design quantum secret sharing schemes for all allowable monotone functions.[4] Of particular relevance, Smith [39] constructed quantum secret sharing schemes realizing monotone functions f whose total share size is equal to the size of the smallest *monotone span program* computing f, thus generalizing a seminal classical result of Karchmer and Wigderson [25] to the quantum setting.

Remarkably, the result of Smith [39] still remains the state-of-the-art for quantum secret sharing realizing broad classes of monotone functions more than 20 years later, and it leaves significant loose ends. In fact, it is now known that there are functions in monotone P

[1] Throughout this paper, we make the parametrization of the sequence of monotone functions in terms of n implicit when clear from context.
[2] We denote by mNP the class of monotone functions in NP.
[3] The notion of secret sharing for functions in mNP requires that the reconstructor receive not only an authorized subset of shares, but also a polynomial-size witness certifying that this subset is authorized.
[4] Not all monotone functions allow a quantum secret sharing scheme. We discuss this in more detail later.

which only have exponentially large associated monotone span programs [37, 35]. This means that with the current methods the best quantum secret sharing scheme realizing such monotone functions requires exponential share size. In contrast, as mentioned before, we have efficient computational *classical* secret sharing schemes realizing all such functions under standard hardness assumptions [44, 41]. Moreover, if we are willing to upgrade our hardness assumptions, then we know such efficient classical schemes for the much broader class of monotone functions in mNP [28].

1.1 A summary of our contributions

Given the state of affairs above, the following questions arise naturally:

> *Can we use computational hardness assumptions to significantly expand the class of monotone functions that can be realized by* efficient *quantum secret sharing schemes? Furthermore, can we improve the share size of schemes for an even broader class of monotone functions?*

We make significant progress in this direction via a new and streamlined approach. While the concept of quantum secret sharing has been around for over twenty years, the work so far has only considered the notion of information-theoretic privacy for such schemes, in contrast to the classical setting. In this work, we initiate the study of *computationally-secure quantum secret sharing*. By leveraging standard hardness assumptions, we show how a conceptually simple compiler utilizing the idea of hybrid encoding allows us to obtain schemes which are far more efficient than those constructed using the current methods. Our simple compiler allows us to obtain a quantum secret sharing scheme from a classical secret sharing scheme and a quantum erasure correcting code. Using this compiler, we design quantum secret sharing schemes with various desirable properties by lifting well-known classical results to the quantum setting. In particular, we are able to lift the results of Yao [44, 41] and Komargodski-Naor-Yogev [28] to the quantum setting for a broad class of monotone functions which we show inherits many relevant properties from general monotone functions. Moreover, we are also able to use our general approach to lift many other results from classical secret sharing to the quantum setting, such as computational secret sharing of long messages with short shares and general perfect secret sharing with share size breaking the circuit barrier. Finally, we are able to obtain efficient schemes for any function in monotone P, given sufficiently many (at most n) copies of the secret.

The key difficulty

One major difficulty that separates quantum from classical secret sharing is the *no-cloning theorem* [43], which precludes copying unknown quantum states. This means that basic techniques, such as giving copies of the same component to several parties, cannot be exploited. In other words, there are no quantum secret sharing schemes realizing the OR function. Consequently, the approaches behind many fundamental classical results cannot be directly extended to the quantum setting, and so this lifting requires different ideas. We proceed to describe our contributions and formal results in more detail. Our results follow from a simple generic compiler, using the general paradigm of hybrid encoding, from classical to quantum secret sharing that we design and analyze.

Heavy monotone functions

Due to the no-cloning theorem, quantum secret sharing schemes are only able to realize what we call *no-cloning* monotone functions [17]. These are monotone functions f with the property that $f(P) = 1$ implies $f(\overline{P}) = 0$, i.e., the complement of an authorized set is unauthorized. As discussed above, we know how to construct quantum secret sharing schemes realizing all no-cloning monotone functions [21, 39], but the state-of-the-art share size for all such functions f corresponds to the size of the smallest monotone span program computing f, which may be extremely large even for "simple" no-cloning monotone functions f in monotone P.

We first focus on a natural subclass of no-cloning monotone functions which we call *heavy* monotone functions.

▶ **Definition 1** (Heavy function). *A monotone function $f : \{0,1\}^n \to \{0,1\}$ is said to be* t-heavy *if for any set $P \subseteq [n]$ with $f(P) = 1$, we have $|P| \geq t$. When $t \geq \lfloor \frac{n}{2} \rfloor + 1$, we simply say that f is* heavy*.*

Equivalently, t-heavy monotone functions correspond exactly to the class of monotone functions with *minimal authorized sets*[5] of size at least t. Note also that a t-out-of-n threshold function is a special case of a t-heavy function.

A t-heavy function with $t > n/2$ satisfies the no-cloning property, and thus can be realized by a quantum secret sharing scheme. Naturally, one may wonder whether the class of heavy monotone functions is interesting. For example, it could be the case at first sight that all heavy monotone functions are computed by polynomial-size monotone span programs, in which case we would already know efficient quantum secret sharing schemes for all such functions via Smith's construction [39]. We show that this is, in fact, not the case: Heavy monotone functions inherit the complexity of *arbitrary* monotone functions, as made precise in the following result (see the full version [14] for the proof).

▶ **Proposition 2.** *Let $\mathsf{mSP}(f)$ and $\mathsf{mC}(f)$ denote the size of the smallest monotone span program and monotone circuit computing f, respectively. Then, for every monotone function $f : \{0,1\}^n \to \{0,1\}$ there exists a heavy monotone function $f' : \{0,1\}^{2n} \to \{0,1\}$ such that $\mathsf{mSP}(f') \geq \frac{\mathsf{mSP}(f)}{2n}$ and $\mathsf{mC}(f') \leq \mathsf{mC}(f) + n$. Moreover, if f is in mNP, then so is f'.*

In words, Proposition 2 states that for every monotone function f there is a corresponding heavy monotone function f' with essentially the same complexity in terms of monotone span programs and monotone circuits. Combining Proposition 2 with recent results from [37, 35] leads to the following corollary.

▶ **Corollary 3.** *There exist heavy monotone functions in* monotone P *which require monotone span programs of size* $\exp\big(n^{\Omega(1)}\big)$.

This corollary is very relevant in the context of secret sharing. It shows that there are *heavy* monotone functions for which we have efficient computational *classical* secret sharing schemes via Yao's construction, but for which the best known method [39] for constructing *quantum* secret sharing schemes requires exponential share size.

[5] We say that P is a *minimal authorized set for* f if $f(P) = 1$ but $f(S) = 0$ for all strict subsets $S \subsetneq P$.

Computational quantum secret sharing of long messages with short shares

As our first contribution, we consider the task of sharing secrets consisting of multiple qubits at once. Naively, this can be accomplished by sharing each qubit in parallel. However, we would like to do considerably better.

In the classical setting, Krawczyk [29] showed that we can share a sufficiently long secret bitstring *using shares that are much shorter than the secret* under standard hardness assumptions via a basic technique. Moreover, it is easy to see that this is impossible to achieve in the case of perfect secret sharing [27]. Using our general approach, we obtain a quantum analogue of Krawczyk's result, thus showing that we can share a high-dimensional secret quantum state using quantum shares of much lower dimension than the secret under standard post-quantum hardness assumptions. As in the classical setting, this is impossible to achieve with perfect security. Gottesman [21] showed that, in this case, shares of important parties (i.e., parties whose removal from some authorized set make it unauthorized) must be at least as large as the secret. Therefore, in particular, the above cannot be achieved by perfectly secure quantum secret sharing schemes even for the simplest case of threshold monotone functions. Furthermore, the currently known quantum secret sharing schemes can only achieve exponential share size per message size ratio for some functions whereas below we show that we can achieve a share size per message size ratio below 1 for sufficiently long messages.

We now provide some more details. Suppose we wish to share a secret composed by m qubits. Then, our quantity of interest is the *information ratio* of a quantum secret sharing scheme [9], which is given by $\frac{\max_{i\in[n]}|S_i|}{m}$, where $|S_i|$ denotes the number of qubits used to describe the i-th share of the scheme. Then, our goal is to design efficient quantum secret sharing schemes whose information ratio is as small as possible as a function of the secret size m and the number of parties n. As mentioned above, information-theoretic schemes always have information ratio at least 1 [21], and so we must use computational assumptions to break this barrier. The following theorem is a notable special case of our general approach.

▶ **Theorem 4.** *If f is a t-heavy monotone function, with $t > n/2$, computed by monotone circuits of size $O(n^d)$, then there is an efficient computational quantum secret sharing scheme realizing f with asymptotic information ratio at most $\frac{32}{2t-n}$ for secrets composed of at least $m = \Omega(n^{cd})$ qubits for a universal constant $c > 0$ based on the existence of post-quantum secure one-way functions.*

In particular, observe that when $t > (\frac{1}{2} + \delta)n$ for an arbitrary constant $\delta > 0$, Theorem 4 guarantees that the information ratio is not only well below 1 but actually behaves as $O(1/n)$ when n is large enough. This is optimal up to the constant factor, since the sum of the sizes of all shares must be at least $m \geq 1$.

Efficient computational quantum secret sharing

As our second result, we show that we can leverage computational hardness assumptions to obtain significantly more efficient quantum secret sharing schemes for heavy monotone functions, even when sharing a single qubit.

▶ **Theorem 5.** *If f is a heavy monotone function in* monotone P*, then there is an efficient computational quantum secret sharing scheme realizing f based on the existence of post-quantum secure one-way functions.*

▶ **Theorem 6.** *If f is a heavy monotone function in* mNP*, then there is an efficient computational quantum secret sharing scheme realizing f based on the existence of post-quantum secure witness encryption for* NP *and one-way functions.*

Note that, by Corollary 3, it follows that both Theorems 5 and 6 provide an *exponential* improvement on the efficiency of known quantum secret sharing schemes for heavy monotone functions.

Interestingly, computational quantum secret sharing had not been studied until our work. Furthermore, there is a curious phenomenon with respect to computational privacy in the quantum setting: Because of the no-cloning theorem, there are monotone functions for which correctness of the quantum secret sharing scheme gives perfect privacy for free, and so computational assumptions cannot be used to obtain improved schemes realizing such monotone functions. More precisely, if f is *self-dual*, meaning that $f(P) = 1$ if and only if $f(\overline{P}) = 0$, then correcting erasures on $\overline{P}$ implies (via the no-cloning theorem) that the shares corresponding to the subset of parties $\overline{P}$ yield no information about the secret qubit [17]. Therefore, perfect reconstruction implies perfect privacy in this case. On the other hand, our results above show that computational assumptions *can* be helpful in obtaining efficient quantum secret sharing schemes realizing a broader class of monotone functions.

We remark that our results extend beyond the class of heavy monotone functions. We discuss these extensions in detail later in this section.

Beyond computational secret sharing: Perfect quantum secret sharing with share size breaking the circuit size barrier

We also extend our approach to obtain new results beyond computational quantum secret sharing. Until recently, the state-of-the-art classical *perfect* secret sharing schemes for arbitrary n-party monotone functions required share size $\Omega(2^n/\sqrt{n})$ [11] – the so-called *circuit-size barrier*. However, a recent groundbreaking line of research [32, 31, 4, 5, 6] has succeeded in constructing classical perfect secret sharing schemes for arbitrary monotone functions with share size $1.5^{n+o(n)}$, well below the circuit size barrier. In this work, we obtain a quantum analogue of this result.

▶ **Theorem 7.** *If $f : \{0,1\}^n \to \{0,1\}$ is a heavy monotone function, then there is a* perfect *quantum secret sharing scheme realizing f with a total share size of $1.5^{n+o(n)}$ classical bits and $O(n \log n)$ qubits.*

Observe that Theorem 7 is a significant improvement over the previous known results for information-theoretic quantum secret sharing [21, 39]. In fact, if all heavy monotone functions were computed by monotone span programs of size less than $1.5^{n+o(n)}$, then Proposition 2 implies that the same would hold for *all monotone functions* – a major improvement over currently known results. Therefore, the previous constructions from [21, 39] require much larger total share size for some heavy monotone functions than Theorem 7.

Bypassing the no-cloning theorem: Quantum secret sharing with multiple copies

As we have seen above, not all monotone functions can be realized by standard quantum secret sharing schemes due to the no-cloning theorem. Therefore, it is natural to wonder what kind of additional assumptions are necessary to bypass this barrier and design quantum secret sharing schemes for a wider range of monotone functions. Arguably, one of the most reasonable directions is to assume we have access to several copies of the quantum state to

be shared. For example, this makes sense in the setting of multiparty computation, where each party has a classical description of their quantum input and thus may create as many copies as it wants. The following question arises naturally from this discussion:

> *How many copies are required to design quantum secret sharing schemes realizing* all *monotone functions in* monotone P*?*

Chien [16] considered this question for the special case of threshold monotone functions. More precisely, he gave a scheme showing (without proof of security) that $\max(1, n-2t+2)$ copies of the quantum secret are sufficient to obtain a t-out-of-n quantum secret sharing scheme. Yet, the efficiency of such schemes was not considered. We exploit our approach to show that the t-out-of-n threshold function is the most demanding among all t-heavy monotone functions, and then construct efficient quantum secret sharing schemes for threshold functions given multiple copies, which allows us to settle (a more general version of) the question above in both the computational and information-theoretic settings.

▶ **Theorem 8.** *A total of* $\max(1, n-2t+2)$ *copies of the quantum secret are sufficient to obtain efficient computational quantum secret sharing schemes realizing all t-heavy monotone functions f in* monotone P *assuming the existence of post-quantum secure one-way functions.*

Observe that every monotone function f is t-heavy when t is the minimum size of its authorized sets. Therefore, given sufficiently many copies, we are able to obtain efficient computational quantum secret sharing schemes for any function $f \in$ monotone P.

▶ **Corollary 9.** *For any monotone function $f : \{0,1\}^n \to \{0,1\}$ in* monotone P *there is an efficient computational quantum secret sharing scheme using at most n copies of the secret realizing f based on the existence of post-quantum secure one-way functions.*

If we are willing to settle for larger share size, we can obtain an analogous result for all t-heavy functions, including those that are not in monotone P.

▶ **Theorem 10.** *A total of* $\max(1, n-2t+2)$ *copies of the quantum secret are sufficient to obtain perfect quantum secret sharing schemes realizing all t-heavy monotone functions f over n parties.*

Beyond heavy monotone functions

Our techniques can be applied to classes of monotone functions greatly generalizing the class of heavy monotone functions. Naturally, such classes of functions also inherit the hardness properties of arbitrary monotone functions detailed in Proposition 2. We give two interesting examples.

Weighted-heavy monotone functions. As a natural generalization of heavy functions, we consider *weighted-heavy* functions. Intuitively, in a weighted-heavy monotone function each party is assigned a weight – the function must evaluate to 0 for a set of parties if the sum of their weights is below some threshold t, and is otherwise unconstrained. Such functions can also be seen as natural extensions of weighted threshold functions.

▶ **Definition 11** (Weighted heavy function)**.** *Let $w : [n] \to \mathbb{N}$ be an integer weight function. A monotone function $f : \{0,1\}^n \to \{0,1\}$ is said to be* (w,t)-weighted-heavy *if for any set $P \subseteq [n]$ with $f(P) = 1$ we have $\sum_{i \in P} w(i) \geq t$. Moreover, we call $W = \sum_{i=1}^{n} w(i)$ the* total weight *of f. If f is (w,t)-weighted-heavy for some w and $t \geq \lfloor \frac{W}{2} \rfloor + 1$, we simply call it* w-weighted-heavy *or just* weighted-heavy *if w is clear from context.*

Note that all (w,t)-weighted heavy monotone functions with $t > W/2$ satisfy the no-cloning property, and we may see t-heavy functions as (w,t)-weighted heavy functions with $w(i) = 1$ for all $i \in [n]$. One of the reasons why this is interesting is the following proposition, which is related to Proposition 2 connecting general and heavy monotone functions and shows that weighted heavy functions strictly generalize heavy and weighted threshold functions (see the full version [14] for a proof).

▶ **Proposition 12.** *There are families of w-weighted heavy monotone functions with total weight $W = \mathrm{poly}(n)$ which are neither heavy nor weighted threshold functions with $\mathrm{poly}(n)$ weights. Moreover, there exist such functions which are also in* monotone P *but require monotone span programs of size $\exp\left(n^{\Omega(1)}\right)$.*

In general, we show that if the sum of the weights W is polynomial in the number of parties n and $t > W/2$, as is the case for the family of functions in Proposition 12, then we can construct efficient computational quantum secret sharing schemes for weighted-heavy monotone functions in monotone P, generalizing Theorem 5.

▶ **Theorem 13.** *If $f : \{0,1\}^n \to \{0,1\}$ is a (w,t)-weighted-heavy monotone function in* monotone P *with total weight $W = \mathrm{poly}(n)$ and threshold $t > W/2$, then there is an efficient computational quantum secret sharing scheme realizing f based on the existence of post-quantum secure one-way functions.*

Trees of weighted-heavy monotone functions. Our techniques can be further applied to constant depth trees that are composed of gates computing weighted heavy functions. Concretely, consider the family $\mathcal{F}$ of weighted heavy functions in monotone P with $\mathrm{poly}(n)$ total weight that satisfy the no-cloning property. Then, we design efficient computational quantum secret sharing schemes realizing any monotone function computed by constant depth trees consisting of gates in $\mathcal{F}$ with fan-in at most n.

▶ **Theorem 14.** *Let $\mathcal{F}$ be the family of weighted heavy functions in* monotone P *with $\mathrm{poly}(n)$ total weight and fan-in at most n. If $f : \{0,1\}^n \to \{0,1\}$ is computed by a constant depth polynomial size tree composed of gates in $\mathcal{F}$, then there is an efficient computational quantum secret sharing scheme realizing f based on the existence of post-quantum secure one-way functions.*

Making a parallel with Proposition 12, we are able to show that trees of weighted-heavy monotone functions strictly generalize weighted-heavy monotone functions – in fact, depth 2 is already sufficient for this (see the full version [14] for a proof). Recall that we had already seen in Proposition 12 that the latter strictly generalize heavy monotone functions and weighted threshold functions.

▶ **Proposition 15.** *There are families of functions $f : \{0,1\}^n \to \{0,1\}$ which satisfy all of the following at once:*

- *f is not weighted-heavy;*
- *$f \in$* monotone P*;*
- *f requires monotone span programs of size $\exp\left(n^{\Omega(1)}\right)$;*
- *f can be represented as a depth-2 polynomial size tree of weighted-heavy monotone functions where each gate is in* monotone P *and has polynomially bounded total weight.*

1.2 Technical overview

We now discuss our techniques and approach in more detail.

The general compiler

The starting point in our approach is a simple and versatile compiler which exploits the fact that we can perfectly encrypt a quantum state using a classical key. This compiler shares some ideas with Krawczyk's classical scheme [29], and similar techniques have been exploited in an orthogonal direction to reduce the number of quantum shares in perfect quantum secret sharing schemes realizing threshold monotone functions [33, 20]. It combines a classical secret sharing scheme with the quantum one-time pad (QOTP) [3] and a quantum erasure-correcting code.

On a high level, our compiler works by perfectly encrypting a state using QOTP. Then, since the keys are classical, we establish the security according to the desired access structure by simply secret sharing the keys using an efficient classical scheme. Then, to allow any authorized set of parties to reconstruct the secret, we *distribute* the encrypted state using a quantum erasure correcting code. We note that the "hybrid encoding" approach we undertake here is prevalent in quantum computing. Other examples of this approach have appeared in the literature (see Section 1.3 for details).

Quantum erasure-correcting codes (QECCs) are a quantum analogue of classical erasure-correcting codes. Intuitively, a length-n QECC of dimension k maps an input quantum state ρ over k qubits into a higher-dimensional quantum state E_ρ over n qubits with the property that even if some qubits at known positions of E_ρ are subjected to any error, then it is still possible to perfectly recover ρ from the corrupted quantum codeword. General constructions with good parameters have been known since at least the seminal work of Calderbank and Shor [15] and Steane [40].

More precisely, for a given monotone function $f : \{0,1\}^n \to \{0,1\}$, suppose we have access to a *classical* secret sharing scheme $\mathsf{SS} = (\mathsf{SS.share}, (\mathsf{SS.rec}_P)_{P\subseteq[n]})$ realizing f and a QECC $\mathsf{QC} = (\mathsf{QC.Enc}, \mathsf{QC.Rec})$ with n *components* "realizing" some monotone function f' satisfying $f'(x) \geq f(x)$ for all $x \in \{0,1\}^n$. That is, QC corrects erasures in the complement of all sets P such that $f'(P) = 1$. Then, our general compiler proceeds as follows on input an arbitrary qubit ρ:

1. Sample a classical key $k = (k_1, k_2) \leftarrow \{0,1\}^2$;
2. Encrypt ρ with QOTP using key k, yielding the perfectly encrypted qubit

$$E_\rho = \mathsf{OTPEnc}(\rho, k) = X^{k_1} Z^{k_2} \rho (X^\dagger)^{k_1} (Z^\dagger)^{k_2},$$

   where X and Z are Pauli gates;
3. Share k using $\mathsf{SS.share}$, yielding classical shares $(S_1, \ldots, S_n)$;
4. Encode E_ρ using $\mathsf{QC.Enc}$, yielding entangled quantum systems $(E_1, \ldots, E_n)$;
5. Set (S_i, E_i) as the final share of the i-th party.

Note that this compiler can be easily generalized to states of arbitrary dimension. Naturally, we need to assume f' (and hence f) satisfies the *no-cloning* property ($f(P) = 1 \implies f(\overline{P}) = 0$). This must be so that we can employ an appropriate QECC. Nevertheless, there is no loss of generality, since, as we have discussed before, quantum secret sharing is impossible when f does not satisfy this property, due to the no-cloning theorem [17].

We claim that the resulting scheme is a quantum secret sharing scheme realizing f. Let $(S_P, E_{\rho,P})$ denote the set of shares of ρ that belong to a subset of parties P. It is straightforward to show that we can reconstruct ρ from $(S_P, E_{\rho,P})$ when $f(P) = 1$. Intuitively,

privacy when $f(P) = 0$ follows from the fact that S_P reveals almost no information about the key k, and so the tuple $(S_P, E_{\rho,P})$ also reveals essentially no information about ρ by the perfect security of the QOTP [3].

The compiler above opens an avenue towards porting results in classical secret sharing to the realm of quantum secret sharing by combining classical secret sharing schemes satisfying relevant properties (such as efficient sharing/reconstruction and small share size) with an appropriate QECC. The apparent bottleneck in this approach is the selection of the QECC, since designing efficient coding schemes for quantum states is more challenging than for classical strings. However, remarkably, we may take any code QC realizing *any* monotone function f' satisfying $f' \geq f$. This means that even if we do not know efficient QECCs for f, we may instead hope to use an efficient QECC for some $f' \geq f$. Moreover, observe that we do not require any privacy properties from the *QECC* QC. In fact, the privacy of our quantum scheme follows directly from the privacy of the *classical* scheme combined with the perfect security of the QOTP, and we only need the QECC to distribute the encrypted state and overcome the no-cloning theorem.

We discuss the compiler in more detail in Section 3.

Heavy monotone functions

After setting up our abstract approach, we would like to instantiate it in concrete settings. Therefore, we turn our attention to the rich class of heavy monotone functions we discussed in Section 1.1 (see Definition 1). Taking into account Proposition 2 and the adjacent discussion, we know that there are heavy monotone functions f computed by poly(n)-size monotone circuits but which require exponentially large monotone span programs. This means that known methods [39] for quantum secret sharing schemes realizing f require exponential share size.

We use our compiler to obtain the first quantum secret sharing schemes with information ratio below 1 and also the first polynomial-time quantum secret sharing schemes for all heavy monotone functions. We discuss this in more detail in the remainder of this overview. The crucial observation behind this is that t-heavy monotone functions f satisfy the following property: If Th_n^t denotes the "t-out-of-n" threshold function such that $\mathsf{Th}_n^t(P) = 1$ if and only if $|P| \geq t$, then $\mathsf{Th}_n^t \geq f$. This is useful because we know simple and highly efficient QECCs realizing Th_n^t. For example, in this case we may take $f' = \mathsf{Th}_n^t$ and the efficient quantum Shamir threshold secret sharing scheme [17] as our QECC QC in the compiler described above.

Computational quantum secret sharing of long messages with short shares

We exploit our compiler above to obtain Theorem 4, i.e., efficient computational quantum secret sharing schemes with constant information ratio from standard hardness assumptions.

Note that when sharing a secret composed of m qubits the QOTP encryption in the compiler requires a key k of length $2m$ and outputs an encrypted state composed of m qubits as well. First, we may instantiate the classical computational secret sharing scheme SS as follows: Instead of sharing the key k directly, replace it by the output of a post-quantum secure pseudorandom generator [7] with a much shorter uniformly random seed s, and share s using an appropriate classical secret sharing scheme realizing f. Then, it remains to instantiate the QECC appropriately so as to encode the m qubits into shares as short as possible. To this end, we choose an appropriate Calderbank-Shor-Steane (CSS) code [15, 40] as the QECC.

CSS codes provide a general framework for designing QECCs from *classical* linear codes[6]. Such codes enjoy efficient encoding and decoding procedures [34, Section 10.4.2], and they may be seen as being analogous to packed secret sharing. We obtain our QECC by combining Reed-Solomon codes with appropriate parameters via this framework. More details can be found in Section 4.

Efficient computational quantum secret sharing

We now discuss how to obtain efficient computational quantum secret sharing schemes for all heavy functions in monotone P or in mNP. When f is heavy and is computed by poly(n)-size monotone circuits, we can set our classical scheme SS to be Yao's scheme [44, 41], whose privacy is based on the existence of one-way functions, or the Komargodski-Naor-Yogev scheme [28], whose privacy is based on the existence of witness encryption for NP and one-way functions. Note that both schemes are efficient. As our quantum erasure correcting code QC, we plug in quantum Shamir's scheme. This leads to Theorems 5 and 6, which state that there exist efficient computational quantum secret sharing schemes for all heavy functions in monotone P or mNP, respectively, under the hardness assumptions detailed above.

For more details, see Section 3 and the full version [14].

Perfect quantum secret sharing with share size breaking the circuit size barrier

We can also exploit the compiler to lift state-of-the-art results beyond computational secret sharing. In particular, we focus on a recent line of work improving general perfect classical secret sharing, and lift it to the quantum setting. This leads to Theorem 7, which states that there exist perfect quantum secret sharing schemes with a total share size of $1.5^{n+o(n)}$ classical bits and $O(n \log n)$ qubits realizing all heavy monotone functions.

To do this, we take any heavy monotone function f and set SS to be the classical secret sharing scheme realizing f constructed by Applebaum and Nir [6] with total share size at most $1.5^{n+o(n)}$, and QC to be the quantum Shamir secret sharing scheme for threshold functions with shares of size $O(n \log n)$ qubits [17]. This leads to total share size of $1.5^{n+o(n)}$ classical bits and $O(n \log n)$ qubits, as desired. The security of the compiler in the information-theoretic setting immediately yields the desired result.

For more details, see the full version [14].

Quantum secret sharing with multiple copies

We also use our compiler to upper bound the number of copies of the quantum secret required to design quantum secret sharing schemes realizing *all* t-heavy monotone functions. More precisely, we exploit the aforementioned fact that $\mathsf{Th}_n^t \geq f$ for any t-heavy monotone function f. Using this, we instantiate our compiler with an appropriate classical secret sharing scheme SS realizing f (which is not bound by the no-cloning theorem), and instantiate our QECC QC with an efficient perfect quantum secret sharing scheme realizing the much simpler threshold function Th_n^t using $\max(1, n - 2t + 2)$ copies of the quantum secret. As a result, for any t-heavy monotone function f, we get a quantum secret sharing scheme realizing it using $\max(1, n - 2t + 2)$ copies of the secret.

[6] We say that $\mathsf{C} \subseteq \mathbb{F}_q^n$ is a *linear code* if C is a subspace of $\mathbb{F}_q^n$. For an extensive survey of linear codes, see [23].

We show an explicit construction of *efficient* perfect quantum secret sharing schemes realizing any Th_n^t. Using this, we obtain Theorem 8, stating that we can construct *efficient* computational quantum secret sharing schemes realizing all t-heavy monotone functions f in $\mathsf{monotone\ P}$ from standard hardness assumptions using $\max(1, n - 2t + 2)$ copies, by instantiating SS with Yao's scheme for f [44, 41] and instantiating QC with the scheme we constructed for Th_n^t. Since every monotone function is 1-heavy, setting $t = 1$ in the theorem above yields Corollary 9, which states that every monotone function in $\mathsf{monotone\ P}$ is realized by an efficient computational quantum secret sharing scheme using at most n copies of the secret.

To get Theorem 10, which states that we can construct perfect quantum secret sharing schemes realizing all t-heavy monotone functions f using $\max(1, n - 2t + 2)$ copies, we may, for example, instantiate SS with the perfect classical secret sharing scheme by Applebaum and Nir [6]. More details can be found in the full version [14].

Quantum secret sharing beyond heavy monotone functions

Finally, we discuss how we can use our compiler to extend the results on quantum secret sharing above well beyond heavy monotone functions. More details can be found in the full version [14].

Weighted heavy monotone functions

As we saw in Section 1.1, the family of weighted-heavy monotone functions (Definition 11) strictly generalizes the classes of heavy monotone functions and weighted threshold functions, as made precise in Proposition 12.

It turns out that we can apply our compiler to weighted heavy functions in a similar fashion to how we proceeded for heavy functions and generalize many of our results. The key observation that enables this, similarly to the case of heavy functions, is that if f is a (w, t)-weighted heavy monotone function, it holds that $f \leq f'$ with f' a weighted *threshold* function with the same weight function w and threshold t. Therefore, we may instantiate our compiler with an appropriate classical secret sharing scheme SS realizing f and a QECC QC for the weighted threshold function f'. If the sum of the weights $W = \sum_{i=1}^{n} w(i) = \mathrm{poly}(n)$ and $t > W/2$, as is already the case in Proposition 2, we can simply take the QECC to be quantum Shamir secret sharing over W parties [17], and then give $w(i)$ quantum shares to the i-th party.

In the particularly relevant case where f is computed by polynomial-size monotone circuits, we can combine the approach above with Yao's classical scheme realizing f to obtain Theorem 13: There exist efficient computational quantum secret sharing schemes realizing all such weighted heavy monotone functions f under standard hardness assumptions. This generalizes Theorem 5.

Trees of heavy functions

Similarly to the approach undertaken by Yao [44, 41], we can compose our quantum secret sharing schemes further to realize monotone functions computed by trees composed of gates computing weighted heavy functions. For the sake of exposition, we focus on the setting of computational privacy. Suppose that f is computed by a tree T whose gates compute weighted heavy functions, and we wish to share $|\psi\rangle$ according to f. Following [44], we can start by placing $|\psi\rangle$ on the output wire. Let g be the weighted heavy function on, say, a input bits computing the gate to which this output wire corresponds. Then, we share $|\psi\rangle$ using

a quantum secret sharing scheme realizing g, leading to a quantum shares $S_1, \ldots, S_a$. We place the i-th share S_i on the i-th in-wire of the gate computing g, and repeat this process with each share until we reach the leaves of the tree.

The process above yields an efficient computational quantum secret sharing scheme realizing f provided that (i) We have efficient quantum secret sharing schemes for each gate in the tree, and (ii) The dimension of the quantum state to be shared in each step is always $\text{poly}(n)$. Therefore, under standard hardness assumptions, we can consider all constant-depth trees whose gates are computed by weighted heavy functions in monotone P with $\text{poly}(n)$ total weight satisfying the no-cloning property by applying the efficient computational quantum secret sharing scheme from Theorem 13 iteratively to each gate. This corresponds to Theorem 14.

1.3 Related work

As already discussed above, previous works on quantum secret sharing has only considered perfectly secure schemes. Furthermore, the work on schemes for general no-cloning monotone functions have mostly focused on the *existence* of such schemes [21, 39]. The share size of Gottesman's scheme [21] for realizing a monotone function f corresponds essentially to the size of f when written as a monotone formula. Smith [39] constructed schemes realizing f whose share size corresponds to the size of the smallest monotone span program computing f. In both cases, as we show here, there are many no-cloning monotone functions which require exponentially long shares under the schemes above, including some which are computed by polynomial-size monotone circuits.

Our work is the first to introduce computational privacy for quantum secret sharing schemes and the first to study notions of *efficiency*, such as polynomial-time sharing and reconstruction or small share size, for quantum secret sharing schemes realizing a broad class of monotone functions. Exploiting standard computational hardness assumptions to obtain significantly more efficient quantum secret sharing schemes, sometimes beyond what is possible in the information-theoretic setting, had not been done prior to this work, although this has been standard in the classical setting since the early work of Yao [44, 41] and Krawczyk [29].

The study of the share size of classical secret sharing schemes already makes an appearance in the work of Benaloh and Leichter [11], which shows how to design general secret sharing schemes among n parties with shares of size $O(2^n/\sqrt{n})$. This remained the state-of-the-art result until a recent line of work [32, 31, 4, 5, 6] managed to reduce the share size over arbitrary monotone functions to $1.5^{n+o(n)}$. Remarkably, the existence of a monotone function requiring shares of size $\Omega(\frac{nm}{\log n})$, where n is the number of parties and m is the length of the secret, obtained by Csirmaz [19] remains the state of the art lower bound. In contrast, for the special case of classical linear secret sharing schemes we know that shares must be exponentially long even if we only wish to share 1 bit [37, 35]. Gottesman [21] proved the only known lower bound on the share size of quantum secret sharing schemes, which states that the dimension of each important share (i.e., a share which contributes to reconstruction) must be at least as large as the dimension of the secret state. However, if one only wishes to share a *classical* secret with a quantum secret sharing scheme, then sharing $2n$ bits requires quantum shares composed by at least n qubits, and this is tight [21].

Other important variants of secret sharing, such as weak and verifiable secret sharing, have also been extended to the quantum setting for threshold functions [18, 10]. Such variants have proved useful in the design of quantum multiparty computation protocols. In particular, note that [18] uses a similar compiler to ours in a different context and for unrelated goals.

There has also been prior work on optimizing aspects of quantum secret sharing incomparable to those considered here. Some works have attempted to minimize the number and size of the quantum shares at the expense of larger classical components [33, 20], while others have considered the case where a subset of the parties is restricted to be classical [30]. Finally, we note that the high level idea of hybrid encoding has also been utilized for different problems, such as for secure multiparty computation [10] and for fully homomorphic encryption [13].

2 Preliminaries

2.1 Notation

We denote sets by uppercase letters such as S and T, and denote $\{1, \ldots, n\}$ as $[n]$. For a vector v and a set S, we write v_S to denote $(v_i)_{i \in S}$. For a family of sets $\{A_i\}_{i \in [n]}$ and a set $S \subseteq [n]$, we let $A_S = \bigtimes_{j \in S} A_j$. For a set $\mathcal{R}$, we write $R \leftarrow \mathcal{R}$ to indicate that R is uniformly distributed on $\mathcal{R}$. We use $\dim(\mathcal{H})$ to denote the dimension of a Hilbert space $\mathcal{H}$. We denote the base-2 logarithm by log. With a slight overloading of notation, we will also use $\rho \in \mathcal{H}$ to denote a density matrix ρ acting on $\mathcal{H}$. Whether we mean a vector in $\mathcal{H}$ (representing a pure state) or a density matrix acting on $\mathcal{H}$ (representing a mixed state) will be clear from context and variable name (such as $|\psi\rangle$ versus ρ). We use monotone P to denote the set of functions that have a polynomial size *monotone circuit*. We use λ to denote a security parameter.

2.2 Quantum information theory

In this section, we give an overview of the quantum information theory concepts we will be using throughout the paper.

▶ **Definition 16** (Total variation distance). *The* total variation distance *between two random variables X, Y supported on the same set $\mathcal{R}$ is defined as*

$$\Delta(X, Y) = \max_{\mathcal{A} \subseteq \mathcal{R}} |\Pr[X \in \mathcal{A}] - \Pr[Y \in \mathcal{A}]| = \frac{1}{2} \sum_{a \in \mathcal{R}} |\Pr[X = a] - \Pr[Y = a]| .$$

The analogue of the total variation distance in the quantum setting is the *trace distance*.

▶ **Definition 17** (Trace distance [34]). *The* trace distance *between two density matrices ρ and σ with the same dimensions is defined as $D(\rho, \sigma) = \frac{1}{2} \operatorname{tr} |\rho - \sigma|$.*

We also define the computational analogue, advantage pseudometric A, which satisfies similar properties as the trace distance. We later use it to prove lifting of computational privacy.

▶ **Definition 18** (Advantage pseudometric). *For a family $\mathcal{F}$ of quantum circuits with single bit classical output and for any two density matrices ρ, σ of appropriate dimension, the* advantage *of $\mathcal{F}$ for distinguishing ρ versus σ is defined as $A_{\mathcal{F}}(\rho, \sigma) = \max_{C \in \mathcal{F}} |\Pr[C(\rho) = 1] - \Pr[C(\sigma) = 1]|$.*

See Appendix A for an overview of useful properties of trace distance and advantage.

Quantum one-time pad encryption

We recall quantum one-time pad encryption (QOTP) [3], which perfectly hides any quantum message using a random classical key. While we describe it for qubits, the generalization to qudits is straightforward.

The quantum one-time pad encryption scheme is defined by a pair of quantum encryption and decryption circuits $(\mathsf{OTPEnc}, \mathsf{OTPDec})$ with $\mathsf{OTPEnc} : (\mathbb{C}^2)^{\otimes n} \times \{0,1\}^{2n} \to (\mathbb{C}^2)^{\otimes n}$ and $\mathsf{OTPDec} : (\mathbb{C}^2)^{\otimes n} \times \{0,1\}^{2n} \to (\mathbb{C}^2)^{\otimes n}$ defined as

$$\mathsf{OTPEnc}(\rho, k) = (X_1^{k_1} Z_1^{k_2} \otimes \cdots \otimes X_i^{k_{2i-1}} Z_i^{k_{2i}} \otimes \cdots \otimes X_n^{k_{2n-1}} Z_n^{k_{2n}})(\rho),$$
$$\mathsf{OTPDec}(\rho, k) = (Z_1^{k_1} X_1^{k_2} \otimes \cdots \otimes Z_i^{k_{2i-1}} X_i^{k_{2i}} \otimes \cdots \otimes Z_n^{k_{2n-1}} X_n^{k_{2n}})(\rho),$$

for any message $\rho \in (\mathbb{C}^2)^{\otimes n}$ and key $k \in \{0,1\}^{\otimes 2n}$, where X_i, Z_i represent the quantum operation applying the standard Pauli gates X, Z respectively to the i-th qubit.

▶ **Lemma 19** ([3]). *The quantum one-time pad encryption scheme is correct and perfectly secure for a randomly chosen key. That is,* $\mathsf{OTPDec}(\mathsf{OTPEnc}(\rho, k), k) = \rho$ *for any key* $k \in \{0,1\}^{2n}$*, and* $\sum_{k \in \{0,1\}^{2n}} \frac{1}{2^{2n}} \mathsf{OTPEnc}(\rho, k) = \sum_{k \in \{0,1\}^{2n}} \frac{1}{2^{2n}} \mathsf{OTPEnc}(\sigma, k)$ *for any two quantum states* $\rho, \sigma \in (\mathbb{C}^2)^{\otimes n}$.

2.3 Quantum adversarial model

We now introduce our quantum adversarial model. By a *QPT adversary* or circuit C, we mean a non-uniform family of circuits $\{C_\lambda\}_{\lambda \in \mathbb{Z}^+}$ with 1-bit classical output where each circuit has size bounded by $\mathrm{poly}(\lambda)$ and is allowed to only use a fixed basis set of gates (e.g., $\{\mathsf{H}, \mathsf{S}, \mathsf{CNOT}, \mathsf{T}\}$), ancilla qubits each initialized to $|0\rangle$ and measurements only in standard computational basis. Furthermore, unless we explicitly state otherwise, we will assume that the adversary has access to quantum advice: that is, C_λ in addition to its input also gets a $\mathrm{poly}(\lambda)$ size quantum state ρ_λ that depends only (but non-uniformly) on λ. When we say that some cryptographic scheme is *post-quantum secure*, we will mean that it is secure against QPT adversaries.

Note that models both with and without quantum advice have been considered in the literature. For example, Watrous [42] and Bartusek, Coladangelo, Khurana, and Ma [7] consider the quantum advice model, while Adcock and Cleve [2] consider a model without advice. In the case of decision problems, it is not known if the quantum advice model is strictly stronger [1].

2.4 Classical secret sharing

We now introduce a definition of classical secret sharing, which allows a party to distribute a classical secret among n parties so that only certain subsets of parties are allowed to recover it. The definition takes into account a monotone function f, indicating which sets of parties are then authorized to recover the secret, and which sets do not obtain information about the secret.

▶ **Definition 20** (Classical secret sharing [9]). *Fix a number of parties* $n \in \mathbb{Z}^+$*, a randomness domain* $\mathcal{R}$*, a secret domain* S*, and share domains* $S_1, \ldots, S_n$*. A* classical secret sharing scheme with perfect privacy realizing the monotone function $f : \{0,1\}^n \to \{0,1\}$ *is a tuple of functions* $\mathsf{SS} = (\mathsf{share}, (\mathsf{rec}_P)_{P \subseteq [n]})$ *where* $\mathsf{share} : S \times \mathcal{R} \to S_{[n]}$ *and* $\mathsf{rec}_P : S_P \to S$ *are deterministic functions satisfying the following properties for all* $P \subseteq [n]$*:*

- **Correctness:** *If* $f(P) = 1$*, then for all* $s \in S$ *it holds that* $\Pr_{R \leftarrow \mathcal{R}}[\mathsf{rec}_P(\mathsf{share}(s; R)_P) = s] = 1$.
- **Perfect privacy:** *If* $f(P) = 0$*, then for all secrets* $a, b \in S$ *and share vectors* $v \in S_P$ *we have* $\Pr_{R \leftarrow \mathcal{R}}[\mathsf{share}(a; R)_P = v] = \Pr_{R \leftarrow \mathcal{R}}[\mathsf{share}(b; R)_P = v]$.

When constructing secret sharing schemes, there are several parameters and properties of interest. First and foremost, we would like to ensure that the sharing and reconstruction procedures run in time polynomial in the number of parties n. We call schemes with this property *efficient*. Another natural and well studied measure of complexity is the *size* of the shares in a secret sharing scheme.

▶ **Definition 21** (Share size)**.** *Given a secret sharing scheme* SS *over the share domains* $S_1, \ldots, S_n$*, we define its* share size*, denoted by* $\mathsf{size}(\mathsf{SS})$*, as* $\mathsf{size}(\mathsf{SS}) = \sum_{i=1}^{n} \lceil \log |S_i| \rceil$.

We will be interested in secret sharing schemes with several different privacy guarantees. We can replace the perfect privacy requirement in Definition 20 with weaker, but still natural, requirements to obtain statistical and computational secret sharing. For the latter, we introduce a security parameter that we pass to the scheme.

▶ **Definition 22** (Statistical privacy for classical secrets)**.** *We say that a secret sharing scheme* SS *realizing a monotone function* f *is* ε-statistically private *if for all* $P \subseteq [n]$ *such that* $f(P) = 0$ *and secrets* $a, b \in S$ *it holds that* $\Delta(\mathsf{share}(a; R_1)_P, \mathsf{share}(b; R_2)_P) \leq \varepsilon$*, where* $R_1 \leftarrow \mathcal{R}$ *and* $R_2 \leftarrow \mathcal{R}$ *are independent random variables.*

▶ **Definition 23** (Post-quantum computational privacy for classical secrets)**.** *We say that a secret sharing scheme* SS *realizing a monotone function* f *is* post-quantum computationally-private, or simply post-quantum computational, *if for all* $P \subseteq [n]$ *such that* $f(P) = 0$*, all secrets* $a, b \in S$*, and for any QPT adversary* $\{C_\lambda\}_\lambda$*, we have*

$$\left| \Pr_{R \leftarrow \mathcal{R}}[C_\lambda(\mathsf{share}(a; 1^\lambda, R)_P) = 1] - \Pr_{R \leftarrow \mathcal{R}}[C_\lambda(\mathsf{share}(b; 1^\lambda, R)_P) = 1] \right| \leq \mathsf{negl}(\lambda).$$

For brevity, we will hide the security parameter and the random coins R of the share functions when we do not need to use them explicitly.

2.5 Quantum erasure-correcting codes

We introduce the definition of a quantum erasure correcting code (QECC) [22], which allows to encode a quantum state into another quantum state of larger dimension, so that the original one can be retrieved perfectly even when there are erasures (arbitrary errors at known positions).

▶ **Definition 24** (Quantum erasure correcting code)**.** *We say a pair of trace-preserving quantum operations* $\mathsf{QC} = (\mathsf{QC.Enc}, \mathsf{QC.Dec})$ *is a* quantum erasure correcting code (QECC) over the input space $\mathcal{H}_{inp}$ and output space $\mathcal{H}_{out} = \bigotimes_{i \in [n]} \mathcal{H}_i$ for $P \subseteq [n]$ *if for any quantum operation* Λ *on* $\mathcal{H}_{out}$ *that acts as the identity on* $\mathcal{H}_i$ *for all* $i \in P$*, it holds for all states* ρ *on* $\mathcal{H}_{inp}$ *that* $(\mathsf{QC.Dec} \circ \Lambda \circ \mathsf{QC.Enc})(\rho) = \rho \otimes \sigma$ *for some state* σ*.*

If $(\mathsf{QC.Enc}, \mathsf{QC.Dec}_P)$ *is a QECC for all sets* $P \subseteq [n]$ *such that* $f(P) = 1$ *for a monotone function* $f : \{0,1\}^n \to \{0,1\}$*, then we say that the family of functions* $(\mathsf{QC.Enc}, (\mathsf{QC.Dec}_P)_{P \subseteq [n]})$ *is a QECC* realizing f*. As a shorthand, we define* $\mathsf{QC.Rec}_P(\tau) = \mathsf{QC.Dec}(\tau \otimes (|0\rangle\langle 0|)^{\otimes \overline{P}})$*. A quantum code that encodes* k *q-ary qudits into* n *q-ary qudits and can correct any* $d-1$ *erasures is said to be an* $[[n, k, d]]_q$ code.

2.6 Quantum secret sharing

A natural analogue of classical secret sharing is sharing quantum states, introduced by [24, 17, 26]. We start by formally defining quantum secret sharing with perfect privacy, and then introduce for the first time the alternative notion of computational privacy for quantum secret sharing.

▶ **Definition 25** (No-cloning function). *A monotone function $f : \{0,1\}^n \to \{0,1\}$ is called* no-cloning *if we have $f(\overline{P}) = 0$ for any $P \subseteq [n]$ with $f(P) = 1$.*

▶ **Definition 26** (Quantum secret sharing). *Fix a number of parties $n \in \mathbb{Z}^+$, a Hilbert space $\mathcal{S}$ for the secret, and Hilbert spaces $\mathcal{H}_1, \ldots, \mathcal{H}_n$ for the shares. Let $f : \{0,1\}^n \to \{0,1\}$ be a no-cloning monotone function. A* quantum secret sharing (QSS) scheme with perfect privacy realizing *f is a tuple of trace-preserving quantum operations $\mathsf{QSS} = (\mathsf{share}, (\mathsf{rec}_P)_{P \subseteq [n]})$ that satisfy the following properties for all $P \subseteq [n]$:*

- **Correctness:** *If $f(P) = 1$, then $(\mathsf{share}, \mathsf{rec}_P)$ is a QECC for P.*
- **Perfect Privacy:** *If $f(P) = 0$, then for any $|\psi_1\rangle, |\psi_2\rangle \in \mathcal{S}$ it holds that $\mathrm{tr}_{\overline{P}}(\mathsf{share}(|\psi_1\rangle\langle\psi_1|)) = \mathrm{tr}_{\overline{P}}(\mathsf{share}(|\psi_2\rangle\langle\psi_2|))$.*

We call a scheme QSS *efficient* if $\mathsf{QSS.share}, \mathsf{QSS.rec}$ are polynomial size circuits. Note that, in particular, efficient schemes have polynomial size shares. We can define weaker notions of privacy analogously to classical secret sharing in Section 2.4.

▶ **Definition 27** (Statistical privacy for quantum secrets). *We say that a quantum secret sharing scheme* QSS *realizing f is ε-*statistically private *if for all $P \subseteq [n]$ such that $f(P) = 0$ and any secrets $|\psi_1\rangle, |\psi_2\rangle \in \mathcal{S}$ it holds that $D(\mathrm{tr}_{\overline{P}}(\mathsf{share}(|\psi_1\rangle\langle\psi_1|)), \mathrm{tr}_{\overline{P}}(\mathsf{share}(|\psi_2\rangle\langle\psi_2|))) \leq \varepsilon$.*

Observe that perfect privacy corresponds to 0-statistical privacy.

As in the classical case, we have two different notions of computational privacy, namely, against quantum adversaries with no advice and quantum adversaries with quantum advice.

▶ **Definition 28** (Computational privacy for quantum secrets). *We say that a quantum secret sharing scheme* QSS *realizing f is* computationally-private, or simply computational, *if for all $P \subseteq [n]$ such that $f(P) = 0$, any secrets $|\psi_1\rangle, |\psi_2\rangle \in \mathcal{S}$, and any QPT adversary $\{C_\lambda\}_\lambda$ we have*

$$|\Pr[C_\lambda(\mathrm{tr}_{\overline{P}}(\mathsf{share}(|\psi_1\rangle\langle\psi_1|\,; 1^\lambda))) = 1] - \Pr[C_\lambda(\mathrm{tr}_{\overline{P}}(\mathsf{share}(|\psi_2\rangle\langle\psi_2|\,; 1^\lambda))) = 1]| \leq \mathsf{negl}(\lambda).$$

Note that requiring privacy (similarly, correctness) for pure states is sufficient, and privacy (correctness) for mixed states of any polynomial size follows by a simple diagonalization and triangle inequality argument. Also observe that any quantum secret sharing scheme for f is also a quantum erasure correcting code realizing f.

3 The compiler

In this section, we present a simple compiler that allows us to obtain several new results results by lifting a wide range of results on classical secret sharing to the quantum setting. As discussed in Section 1.2, these include:

- Efficient computational quantum secret sharing schemes for heavy functions in $\mathsf{monotone\ P}$ with information ratio *well below* 1 (i.e., with shares much shorter than the secret) from standard hardness assumptions. This is impossible in the information-theoretic setting.
- Efficient computationally-private quantum secret sharing for all heavy functions in $\mathsf{monotone\ P}$ and mNP. In contrast, known schemes [21, 39] require shares of exponential size for these functions (see Proposition 2 and adjacent discussion). The techniques can be further applied to classes of monotone functions that generalize heavy functions, including weighted heavy functions and trees of weighted heavy functions.
- Similar techniques can also be applied to obtain perfect quantum secret sharing schemes for heavy functions with share size $1.5^{n+o(n)}$, breaking the circuit size barrier.
- Polynomial size computational schemes for any function in $\mathsf{monotone\ P}$ given sufficiently many copies of the secret.

We now move on to the compiler. Unless otherwise specified, we assume any state space $\mathcal{H}$ is $(\mathbb{C}^2)^{\otimes \ell}$ for some $\ell \in \mathbb{Z}^+$, that is, we are working with qubits. The generalization to qudits is straightforward. We will also assume $\ell = 1$ unless otherwise stated.

Compiler Description

The compiler combines a classical secret sharing scheme SS realizing a no-cloning monotone function f, and a quantum error correcting code QC realizing an appropriate no-cloning monotone function $f' \geq f$, to create a quantum secret sharing scheme QSS realizing f. In order to secret share a quantum state ρ, the scheme $\mathsf{QSS.share}$ first samples a random classical key k and computes the encryption ρ' of ρ using the quantum one-time pad. We then distribute the classical key k using SS and the quantum state ρ' using QC. Intuitively, the privacy of the overall scheme follows directly from the privacy of the classical scheme SS and privacy of the quantum one-time pad.

The reconstruction procedure $\mathsf{QSS.rec}$ is straightforward. We simply let the set of parties P reconstruct the state ρ' using the decoding procedure for QC and the key k using the reconstruction procedure of SS. The quantum secret ρ is then reconstructed by decrypting ρ' with the obtained key k via the quantum one-time pad.

QSS Share for textittexorpdfstring$f$f: $\mathsf{QSS.share}(\rho)$

1. Sample key $\mathsf{k} \leftarrow \{0,1\}^{2 \log \dim \mathcal{S}}$.
2. Compute $\rho' = \mathsf{OTPEnc}(\rho, k)$, the encryption of ρ using QOTP with key k.
3. Let $(E_1, \ldots, E_n) = \mathsf{QC.Enc}(\rho')$ be the encoding of ρ'.
4. Let $(S_1, \ldots, S_n) = \mathsf{SS.share}(\mathsf{k})$ be a sharing of k.
5. Set (S_i, E_i) as the share for party P_i.

QSS Reconstruct for f: $\mathsf{QSS.rec}_P((S_i, E_i)_{i \in P})$

1. Compute $\rho' = \mathsf{QC.Rec}_P((E_i)_{i \in P})$.
2. Compute $\mathsf{k} = \mathsf{SS.rec}_P((S_i)_{i \in P})$.
3. Compute $\rho = \mathsf{OTPDec}(\rho', \mathsf{k})$.
4. Output ρ.

We now formally state the main theorem.

▶ **Theorem 29** (QSS Compiler). *Let $f, f' : \{0,1\}^n \to \{0,1\}$ be no-cloning monotone functions such that $f' \geq f$. Let $\mathsf{QC} = (\mathsf{QC.Enc}, (\mathsf{QSS.Rec}_P)_{P \subseteq [n]})$ be a QECC realizing f' and $\mathsf{SS} = (\mathsf{SS.share}, (\mathsf{SS.rec}_P)_{P \subseteq [n]})$ be a [post-quantum computational, statistical, perfect] classical secret sharing scheme realizing f. Then, QSS is a [computational, statistical, perfect] quantum secret sharing scheme for f with total share size*

$$\mathsf{size}(\mathsf{QC}) + 2 \log(\dim \mathcal{S}) \cdot \mathsf{size}(\mathsf{SS}).$$

Moreover, QSS has efficient sharing and reconstruction procedures whenever QC and SS do.

Proof. Let us denote by $\xi_{\rho,\mathsf{k}} = \mathsf{QC.Enc}(\mathsf{OTPEnc}(\rho,\mathsf{k}))$ the encoding of the state $\mathsf{OTPEnc}(\rho,\mathsf{k})$ using QC, and $\tau_{\mathsf{k},r} = |\mathsf{SS.share}(\mathsf{k},r)\rangle\langle\mathsf{SS.share}(\mathsf{k},r)|$ a sharing of the key k when the random input is r. Then, the scheme QSS can be formally described as

$$\mathsf{QSS.share}(\rho) = \sum_{k\in\{0,1\}^{2\log\dim\mathcal{S}}} \sum_{r\in\mathcal{R}} \frac{1}{2^{2\log\dim\mathcal{S}}}\frac{1}{|\mathcal{R}|}(\xi_{\rho,\mathsf{k}}\otimes\tau_{\mathsf{k},r}),$$

$$\mathsf{QSS.rec}_P(\sigma) = \mathsf{OTPDec}(\mathsf{QC.Rec}_P(\mathrm{tr}_{key}(\sigma))), \mathsf{SS.rec}_P(\mathrm{tr}_{state}(\sigma)),$$

where $\mathrm{tr}_{key}, \mathrm{tr}_{state}$ denotes tracing out the subsystem corresponding to the shares of the key and shares of the quantum secret respectively. For simplicity, we encode the shares of the classical keys as qubits in basis states, however, they can be kept as classical shares in practice without any change to the scheme.

Showing correctness is straightforward. Now, we show that if SS is ε-statistically private, then QSS is 2ε-statistically private. First, observe the following relation between trace distance and total variation distance. Consider any two keys, $k, k' \in \{0,1\}^{2\log\dim\mathcal{S}}$. For each $v \in V = S_P$, define $p_v = \Pr_{r\leftarrow R}[\mathsf{share}(k;r)_P = v]$ and define p'_v analogously for k'. Then, using the fact that $\mathrm{tr}_{\overline{P}}\left(\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\tau_{\mathsf{k},r}\right) = \sum_{v\in V} p_v\,|v\rangle\langle v|$, it is easy to see that

$$D\left(\mathrm{tr}_{\overline{P}}\left(\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\tau_{\mathsf{k},r}\right), \mathrm{tr}_{\overline{P}}\left(\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\tau_{\mathsf{k}',r}\right)\right) = D\left(\sum_{v\in V} p_v\,|v\rangle\langle v|, \sum_{v\in V} p'_v\,|v\rangle\langle v|\right)$$
$$\leq \Delta(p_v, p'_v). \tag{1}$$

We now study the privacy of QSS. Consider a set $P \subseteq [n]$ such that $f(P) = 0$ and any two secret states ρ, ρ'. We will use a hybrid argument. First, we will argue that when we replace the shares of the key with shares of an independent uniformly distributed key, the composite shares of $\overline{P}$ for the two secrets ρ, ρ' will be perfectly indistinguishable due to the perfect privacy of one-time pad from Lemma 19. That is, we will define the sharing of a random key

$$\kappa = \sum_{\substack{k'\in\{0,1\}^{2\log\dim\mathcal{S}}\\ r\in\mathcal{R}}} \frac{1}{2^{2\log\dim\mathcal{S}}|\mathcal{R}|}\tau_{\mathsf{k}',r}$$

and the hybrids

$$\zeta_1 = \mathrm{tr}_{\overline{P}}\left(\sum_{k\in\{0,1\}^{2\log\dim\mathcal{S}}}\left(\frac{1}{2^{2\log\dim\mathcal{S}}}\xi_{\rho,\mathsf{k}}\right)\otimes\kappa\right),$$

$$\zeta_2 = \mathrm{tr}_{\overline{P}}\left(\sum_{k\in\{0,1\}^{2\log\dim\mathcal{S}}}\left(\frac{1}{2^{2\log\dim\mathcal{S}}}\xi_{\rho',\mathsf{k}}\right)\otimes\kappa\right),$$

and will show that $D(\zeta_1,\zeta_2) = 0$. Then, we will show that composite shares of $\overline{P}$ for the same secret are close in trace distance when again the shares of the key are replaced with shares of a random key, versus when they are not replaced. That is, we will show that

$$D(\mathrm{tr}_{\overline{P}}(\mathsf{QSS.share}(\rho)), \zeta_1) \leq \varepsilon \quad \text{and} \quad D(\zeta_2, \mathrm{tr}_{\overline{P}}(\mathsf{QSS.share}(\rho'))) \leq \varepsilon. \tag{2}$$

Finally applying the triangle inequality will yield the desired result.

We start with $D(\zeta_1,\zeta_2) = 0$. By distributing the partial trace and using Lemma 33 (Item 5), we get

$$D\left(\zeta_1,\zeta_2\right) = D\left(\mathrm{tr}_{\overline{P}}\left(\sum_{k\in\{0,1\}^{2\log\dim\mathcal{S}}}\left(\frac{1}{2^{2\log\dim\mathcal{S}}}\xi_{\rho,\mathsf{k}}\right)\right),\mathrm{tr}_{\overline{P}}\left(\sum_{k\in\{0,1\}^{2\log\dim\mathcal{S}}}\left(\frac{1}{2^{2\log\dim\mathcal{S}}}\xi_{\rho',\mathsf{k}}\right)\right)\right).$$

Then, since the quantum one-time pad perfectly hides the input when the key is uniform (see Lemma 19), we get for some state ι that $D\left(\zeta_1,\zeta_2\right) = D(\mathrm{tr}_{\overline{P}}(\mathsf{QC.Enc}(\iota)),\mathrm{tr}_{\overline{P}}(\mathsf{QC.Enc}(\iota))) = 0$.

The inequalities in Equation (2) are proven using the privacy of SS and again properties of trace distance, along with Equation (1). By Lemma 33 (Items 3 and 5), we get

$$\begin{aligned}D(\zeta_1,\mathrm{tr}_{\overline{P}}\left(\mathsf{QSS.share}(\rho)\right)) = D\Bigg(&\sum_{k\in\{0,1\}^{2\log\dim\mathcal{S}}}\frac{1}{2^{2\log\dim\mathcal{S}}}\,\mathrm{tr}_{\overline{P}}\left(\xi_{\rho,\mathsf{k}}\right)\otimes\mathrm{tr}_{\overline{P}}\left(\kappa\right),\\ &\sum_{k\in\{0,1\}^{2\log\dim\mathcal{S}}}\frac{1}{2^{2\log\dim\mathcal{S}}}\,\mathrm{tr}_{\overline{P}}\left(\xi_{\rho,\mathsf{k}}\right)\otimes\mathrm{tr}_{\overline{P}}\left(\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\tau_{\mathsf{k},r}\right)\Bigg)\\ \leq&\sum_{k\in\{0,1\}^{2\log\dim\mathcal{S}}}\frac{1}{2^{2\log\dim\mathcal{S}}}D\left(\mathrm{tr}_{\overline{P}}\left(\kappa\right),\mathrm{tr}_{\overline{P}}\left(\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\tau_{\mathsf{k},r}\right)\right).\end{aligned}$$

Then, using Lemma 33 (Item 4), we get

$$\begin{aligned}&D(\zeta_1,\mathrm{tr}_{\overline{P}}\left(\mathsf{QSS.share}(\rho)\right))\\ &\qquad\leq\sum_{k,k'\in\{0,1\}^{2\log\dim\mathcal{S}}}\frac{1}{4^{2\log\dim\mathcal{S}}}D\left(\mathrm{tr}_{\overline{P}}\left(\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\tau_{\mathsf{k}',r}\right),\mathrm{tr}_{\overline{P}}\left(\sum_{r\in\mathcal{R}}\frac{1}{|\mathcal{R}|}\tau_{\mathsf{k},r}\right)\right).\end{aligned}$$

Observe that this is basically the statistical distance between classical sharings of keys k,k'. Therefore, invoking the ε-statistical privacy of SS and Equation (1), we conclude that $D(\zeta_1,\mathrm{tr}_{\overline{P}}\left(\mathsf{QSS.share}(\rho)\right)) \leq \varepsilon$. The same argument also shows that $D(\zeta_2,\mathrm{tr}_{\overline{P}}\left(\mathsf{QSS.share}(\rho')\right)) \leq \varepsilon$.

Finally, we combine these inequalities with the triangle inequality to obtain

$$\begin{aligned}&D(\mathrm{tr}_{\overline{P}}(\mathsf{QSS.share}(\rho)),\mathrm{tr}_{\overline{P}}(\mathsf{QSS.share}(\rho')))\\ &\leq D(\mathrm{tr}_{\overline{P}}(\mathsf{QSS.share}(\rho)),\zeta_1)+D(\zeta_1,\zeta_2)+D(\zeta_2,\mathrm{tr}_{\overline{P}}(\mathsf{QSS.share}(\rho')))\\ &\leq\varepsilon+0+\varepsilon=2\varepsilon.\end{aligned}$$

Plugging in $\varepsilon = 0$ yields the desired result for perfect privacy, since Δ and D are both metrics.

Lastly, we consider the setting of computational privacy. Here, we use the quantum advantage pseudometric A, which, as shown in Lemma 34, satisfies the same basic properties that we use above for trace distance. Hence, replacing trace distance with advantage in the proof above proves the lifting of computational privacy. More specifically, by Lemma 34 (Item 7), we get the following.[7]

- Suppose that SS is computationally-private with respect to QPT adversaries with no advice, that $\mathsf{QC.Enc}$ can be implemented by quantum circuits of size $\mathrm{poly}(\lambda)$, and that any pure secret in the space of secrets can be approximated by a quantum circuit of size $\mathrm{poly}(\lambda)$.[8] Then, QSS is computationally-private with respect to QPT adversaries with no advice.
- If SS is secure against QPT adversaries with quantum advice and the shares of QC are at most $\mathrm{poly}(\lambda)$ qubits, then QSS is also secure against the same family of adversaries. ◀

[7] Note that we need the extra assumptions below since otherwise one can construct pathological schemes and secrets so that the adversary obtains non-uniform quantum advice.

[8] This is readily true, for example, if $\log\dim(\mathcal{S}) = \Theta(1)$.

4 Computational quantum secret sharing of long messages with short shares

In this section we consider the problem of sharing a long secret consisting of multiple qubits and prove Theorem 4, which states that there exist efficient computational quantum secret sharing schemes for all heavy functions in monotone P with information ratio well below 1. Note that all previously known quantum secret sharing schemes are information-theoretic and require a share size that is as large as the secret, as proven by Gottesman [21], even for the very simple case of threshold monotone functions. We show that we can achieve a scheme with individual share sizes much shorter than the secret, with the help of computational hardness assumptions.

Our scheme follows the template of the general compiler presented in Section 3, with the following sub-protocols: (i) the key used for the one-time pad encryption is generated using a pseudo-random generator (PRG) and its seed is shared using a computational classical secret sharing scheme with short shares, and (ii) a CSS quantum erasure-correcting code with low share size.

We formally describe the share procedure $\mathsf{QSS.share}$ below. The reconstruction procedure $\mathsf{QSS.rec}_P$ is as in Section 3, except that the key is generated by reconstructing the seed and evaluating the PRG.

QSS Share for f: $\mathsf{QSS.share}(\rho)$

1. Sample a seed $\mathsf{x} \leftarrow \{0,1\}^{\ell(m)}$.
2. Compute $\mathsf{k} = \mathsf{PRG}(\mathsf{x})$.
3. Compute $\sigma = \mathsf{OTPEnc}(\rho, k)$, the encryption of ρ using QOTP with key k.
4. Let $(E_1, \ldots, E_n) = \mathsf{QC.Enc}(\sigma)$ be the encoding of σ.
5. Let $\mathsf{SS.share}(\mathsf{x})$ be a sharing of x.
6. Set (S_i, E_i) as the share for party P_i.

QSS Reconstruct for f: $\mathsf{QSS.rec}_P((S_i, E_i)_{i\in P})$

1. Compute $\sigma = \mathsf{QC.Rec}_P((E_i)_{i\in P})$.
2. Compute $\mathsf{x} = \mathsf{SS.rec}_P((S_i)_{i\in P})$.
3. Compute $\mathsf{k} = \mathsf{PRG}(\mathsf{x})$.
4. Compute $\rho = \mathsf{OTPDec}(\sigma, \mathsf{k})$.
5. Output ρ.

Correctness of the scheme is straightforward. Computational privacy easily follows from the privacy of the underlying PRG and the classical secret sharing scheme, by an argument analogous to the proof of Theorem 29.

Observe that by plugging in a pseudo-random generator PRG with polynomial stretch, it is possible to get the same ratio as that of the QECC for sufficiently long messages. As a concrete example, we show below a scheme with asymptotic information ratio $\frac{32}{2t-n}$ for any t-heavy monotone function in monotone P. Note that previous schemes [21, 39], or a naive application of Theorem 29, both yield information ratio above 1 (in fact, exponentially or polynomially larger, respectively).

Before we prove our main result, we need to construct a suitable QECC. To that end, we need the following general template for CSS codes (for a definition of linear codes, see Definition 35).

▶ **Lemma 30** (CSS Codes [22, Theorem 6] and [36]). *Let q be a prime power, C_1 an $[n, k_1, d_1]_q$ linear code, and C_2 an $[n, k_2, d_2]_q$ linear code with $\mathsf{C}_2^{\perp} \subseteq \mathsf{C}_1$. Then, there exists an $[[n, k_1 + k_2 - n, \min(d_1, d_2)]]_q$ quantum code.*

With the help of this lemma, we can obtain a QECC with the required parameter trade-offs to achieve QSS with small share size.

▶ **Lemma 31.** *For any m, n, t with $n \geq t > \frac{n}{2}$, there is a $[[N, 2K - N, N - K + 1]]_{2^r}$ CSS code QC where the parameters are defined as follows: Let N^* be such that $N^* \log_2(N^*) = \frac{2mn}{t - \frac{n}{2}}$, and set $c = \left\lceil \frac{N^*}{n} \right\rceil$ along with*

$$N = cn, \quad r = \lceil \log_2(N) \rceil \quad \text{and} \quad K = \left\lceil \frac{nc}{2} + \frac{1}{2} \left\lceil \frac{m}{r} \right\rceil \right\rceil.$$

Then, for large enough m we have

$$2K - N \geq \left\lceil \frac{m}{r} \right\rceil \quad \text{and} \quad N - K \geq c(n - t).$$

Proof. See the full version [14] for the proof. ◀

We are now ready to state the final theorem.

▶ **Theorem 32** (Theorem 4, restated). *If f is a t-heavy monotone function, with $t > n/2$, computed by monotone circuits of size $O(n^d)$, then there is an efficient computational quantum secret sharing scheme realizing f with asymptotic information ratio at most $\frac{32}{2t-n}$ for secrets composed of at least $m = \Omega(n^{cd})$ qubits for a universal constant $c > 0$ based on the existence of post-quantum secure one-way functions.*

Proof. See the full version [14] for the proof. ◀

References

1. Scott Aaronson. PDQP/qpoly=ALL. *Quantum Inf. Comput.*, 18(11&12):901–909, 2018. doi:10.26421/QIC18.11-12-1.
2. Mark Adcock and Richard Cleve. A quantum Goldreich-Levin theorem with cryptographic applications. In Helmut Alt and Afonso Ferreira, editors, *STACS 2002*, pages 323–334. Springer Berlin Heidelberg, 2002.
3. A. Ambainis, M. Mosca, A. Tapp, and R. De Wolf. Private quantum channels. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 547–553, 2000. doi:10.1109/SFCS.2000.892142.
4. Benny Applebaum, Amos Beimel, Oriol Farràs, Oded Nir, and Naty Peter. Secret-sharing schemes for general and uniform access structures. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 441–471, Cham, 2019. Springer International Publishing.
5. Benny Applebaum, Amos Beimel, Oded Nir, and Naty Peter. Better secret sharing via robust conditional disclosure of secrets. In *STOC 2020*, pages 280–293, 2020.
6. Benny Applebaum and Oded Nir. Upslices, downslices, and secret-sharing with complexity of 1.5^n. Cryptology ePrint Archive, Report 2021/470, 2021. URL: https://ia.cr/2021/470.
7. James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021*, pages 467–496. Springer, 2021. doi:10.1007/978-3-030-84242-0_17.

8 James Bartusek and Giulio Malavolta. Indistinguishability obfuscation of null quantum circuits and applications. Cryptology ePrint Archive, Report 2021/421, 2021. URL: https://ia.cr/2021/421.

9 Amos Beimel. Secret-sharing schemes: A survey. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology*, pages 11–46, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

10 Michael Ben-Or, Claude Crépeau, Daniel Gottesman, Avinatan Hassidim, and Adam Smith. Secure multiparty quantum computation with (only) a strict honest majority. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 249–260, 2006. doi:10.1109/FOCS.2006.68.

11 Josh Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer, 1988. doi:10.1007/0-387-34799-2_3.

12 G. R. Blakley. Safeguarding cryptographic keys. In *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pages 313–318, 1979. doi:10.1109/MARK.1979.8817296.

13 Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, pages 609–629, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

14 Alper Cakan, Vipul Goyal, Chen-Da Liu-Zhang, and João Ribeiro. Computational quantum secret sharing. Cryptology ePrint Archive, Paper 2023/613, 2023. doi:10.4230/LIPIcs.TQC.2023.4.

15 A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, August 1996. doi:10.1103/PhysRevA.54.1098.

16 Steven Chien. Augmented $((t, n))$-threshold quantum secret sharing schemes, 2020. Senior thesis, Princeton University, available at: http://arks.princeton.edu/ark:/88435/dsp01fb494c46v.

17 Richard Cleve, Daniel Gottesman, and Hoi-Kwong Lo. How to share a quantum secret. *Phys. Rev. Lett.*, 83:648–651, July 1999. doi:10.1103/PhysRevLett.83.648.

18 Claude Crépeau, Daniel Gottesman, and Adam Smith. Approximate quantum error-correcting codes and secret sharing schemes. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, pages 285–301, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

19 László Csirmaz. The size of a share must be large. *J. Cryptol.*, 10(4):223–231, 1997. doi:10.1007/s001459900029.

20 Ben Fortescue and Gilad Gour. Reducing the quantum communication cost of quantum secret sharing. *IEEE Transactions on Information Theory*, 58(10):6659–6666, 2012. doi:10.1109/TIT.2012.2205895.

21 Daniel Gottesman. Theory of quantum secret sharing. *Phys. Rev. A*, 61:042311, March 2000. doi:10.1103/PhysRevA.61.042311.

22 M. Grassl, Th. Beth, and T. Pellizzari. Codes for the quantum erasure channel. *Phys. Rev. A*, 56:33–38, July 1997. doi:10.1103/PhysRevA.56.33.

23 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book, 2022.

24 Mark Hillery, Vladimír Bužek, and André Berthiaume. Quantum secret sharing. *Phys. Rev. A*, 59:1829–1834, March 1999. doi:10.1103/PhysRevA.59.1829.

25 M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference*, pages 102–111, 1993. doi:10.1109/SCT.1993.336536.

26 Anders Karlsson, Masato Koashi, and Nobuyuki Imoto. Quantum entanglement for secret sharing and secret splitting. *Phys. Rev. A*, 59:162–168, January 1999. doi:10.1103/PhysRevA.59.162.

27 E. Karnin, J. Greene, and M. Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, 29(1):35–41, 1983. doi:10.1109/TIT.1983.1056621.

28 Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for NP. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014*, pages 254–273, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

29 Hugo Krawczyk. Secret sharing made short. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, pages 136–146, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

30 Lvzhou Li, Daowen Qiu, and Paulo Mateus. Quantum secret sharing with classical Bobs. *Journal of Physics A: Mathematical and Theoretical*, 46(4):045304, January 2013. doi:10.1088/1751-8113/46/4/045304.

31 Tianren Liu and Vinod Vaikuntanathan. Breaking the circuit-size barrier in secret sharing. In *STOC 2018*, pages 699–708, 2018. doi:10.1145/3188745.3188936.

32 Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Towards breaking the exponential barrier for general secret sharing. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 567–596, Cham, 2018. Springer International Publishing.

33 Anderson C. A. Nascimento, Joern Mueller-Quade, and Hideki Imai. Improving quantum secret-sharing schemes. *Phys. Rev. A*, 64:042311, September 2001. doi:10.1103/PhysRevA.64.042311.

34 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. doi:10.1017/CBO9780511976667.

35 Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 1246–1255, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3055399.3055478.

36 Eric M Rains. Nonbinary quantum codes. *IEEE Transactions on Information Theory*, 45(6):1827–1832, 1999.

37 Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A. Cook. Exponential lower bounds for monotone span programs. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 406–415, 2016. doi:10.1109/FOCS.2016.51.

38 Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979. doi:10.1145/359168.359176.

39 Adam D. Smith. Quantum secret sharing for general access structures. *arXiv e-prints*, pages quant–ph/0001087, January 2000. arXiv:quant-ph/0001087.

40 Andrew Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 452(1954):2551–2577, 1996.

41 V. Vinod, Arvind Narayanan, K. Srinathan, C. Pandu Rangan, and Kwangjo Kim. On the power of computational secret sharing. In Thomas Johansson and Subhamoy Maitra, editors, *Progress in Cryptology - INDOCRYPT 2003*, pages 162–176, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

42 John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009. doi:10.1137/060670997.

43 William K. Wootters and Wojciech H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

44 Andrew C. Yao. Unpublished manuscript, 1989. Presented at Oberwolfach and DIMACS workshops.

A Quantum information theory

In this section, we present useful properties of the distance measures for quantum information introduced in Section 2.2.

▶ **Lemma 33** (Trace distance). *For any two density matrices ρ, σ the following holds:*

1. *For any trace-preserving quantum operation Φ,*

$$D(\Phi(\rho), \Phi(\sigma)) \leq D(\rho, \sigma)$$

2. *Assuming the states are of a composite system AB,*

$$D(\rho^A, \sigma^A) \leq D(\rho^{AB}, \sigma^{AB})$$

3. *For any two probability distributions $\{p_i\}_{i\in I}, \{q_i\}_{i\in I}$, and ensembles of states $\{\rho_i\}_{i\in I}, \{\sigma_i\}_{i\in I}$,*

$$D\left(\sum_{i\in I} p_i\rho_i, \sum_{i\in I} q_i\sigma_i\right) \leq \Delta(p_i, q_i) + \sum_{i\in I} p_i D\left(\rho_i, \sigma_i\right)$$

4. *For a probability distribution $\{p_i\}_{i\in I}$ and an ensemble of states $\{\rho_i\}_{i\in I}$*

$$D\left(\sum_{i\in I} p_i\rho_i, \sigma\right) \leq \sum_{i\in I} p_i D\left(\rho_i, \sigma_i\right)$$

5. *For any state τ,*

$$D(\rho \otimes \tau, \sigma \otimes \tau) = D(\rho, \sigma)$$

6. *For any states τ, υ,*

$$D(\rho \otimes \tau, \sigma \otimes \upsilon) \leq D(\rho, \sigma) + D(\tau, \upsilon)$$

Proof. See [34, Section 9.2.1] for proofs of (i), (ii) and (iii). Inequality (iv) is a simple corollary of (iii) when q_i is set to p_i and σ_i to σ. Result (v) can be obtained from (i) by setting $\Phi(\gamma) = \gamma \otimes \tau$. Inequality (vi) can be obtained by applying triangle inequality in combination with (v) to $D(\rho \otimes \tau, \sigma \otimes \tau)$ and $D(\sigma \otimes \tau, \sigma \otimes \upsilon)$. ◀

The advantage satisfies similar properties as the trace distance.

▶ **Lemma 34** (Advantage). *For any circuit family $\mathcal{F}$ and two states ρ, σ, the following holds:*

1. $A_{\mathcal{F}}(\rho, \rho) = 0$;
2. $A_{\mathcal{F}}(\rho, \sigma) = A_{\mathcal{F}}(\sigma, \rho)$;
3. *For any state τ, $A_{\mathcal{F}}(\rho, \sigma) \leq A_{\mathcal{F}}(\rho, \tau) + A_{\mathcal{F}}(\tau, \sigma)$;*
4. *Assuming the states are of a composite system AB,*

$$A_{\mathcal{F}}(\rho^A \otimes |0\rangle\langle 0|, \sigma^A \otimes |0\rangle\langle 0|) \leq A_{\mathcal{F}}(\rho^{AB}, \sigma^{AB});$$

5. *For any two probability distributions $\{p_i\}_{i\in I}, \{q_i\}_{i\in I}$, and ensembles of states $\{\rho_i\}_{i\in I}, \{\sigma_i\}_{i\in I}$,*

$$A_{\mathcal{F}}\left(\sum_{i\in I} p_i\rho_i, \sum_{i\in I} q_i\sigma_i\right) \leq \Delta(p_i, q_i) + \sum_{i\in I} p_i A_{\mathcal{F}}\left(\rho_i, \sigma_i\right);$$

6. *For a probability distribution* $\{p_i\}_{i\in I}$ *and an ensemble of states* $\{\rho_i\}_{i\in I}$

$$A_{\mathcal{F}}\left(\sum_{i\in I} p_i\rho_i, \sigma\right) \leq \sum_{i\in I} p_i A_{\mathcal{F}}\left(\rho_i, \sigma_i\right);$$

7. *For any family* $\mathcal{F}'$ *and state* τ *such that there is* $C' \in \mathcal{F}'$ *satisfying* $C'(\rho) = C(\rho\otimes\tau)$ *and* $C'(\sigma) = C(\sigma\otimes\tau)$ *for any* $C \in \mathcal{F}$*,*

$$A_{\mathcal{F}}(\rho\otimes\tau, \sigma\otimes\tau) \leq A_{\mathcal{F}'}(\rho, \sigma).$$

Proof. See the full version [14] for the proof. ◀

B Coding theory

In this section, we state some basic concepts from coding theory.

▶ **Definition 35** (Linear code [23])**.** *An* $[n, k, d]_q$ code C *is a linear subspace of* $\mathbb{F}_q^n$ *of dimension* k *with* $\min_{c\in\mathsf{C}\setminus\{0\}} wt(c) \geq d$*, where* $wt(c) = |\{i \in [n] : c_i \neq 0\}|$ *denotes the* Hamming weight *of* c*.*

The following lemma is based on Reed-Solomon codes.

▶ **Lemma 36** (Reed-Solomon codes [23])**.** *For all integers* $k \leq n$ *and every prime power* $q \geq n$ *there exists an* $[n, k, d = n - k + 1]_q$ *linear code with efficient encoding and decoding procedures.*

Quantum Algorithm for Path-Edge Sampling

Stacey Jeffery
QuSoft and CWI, Amsterdam, The Netherlands

Shelby Kimmel
Middlebury College, VT, USA

Alvaro Piedrafita
QuSoft and CWI, Amsterdam, The Netherlands

Abstract

We present a quantum algorithm for sampling an edge on a path between two nodes s and t in an undirected graph given as an adjacency matrix, and show that this can be done in query complexity that is asymptotically the same, up to log factors, as the query complexity of detecting a path between s and t. We use this path sampling algorithm as a subroutine for st-path finding and st-cut-set finding algorithms in some specific cases. Our main technical contribution is an algorithm for generating a quantum state that is proportional to the positive witness vector of a span program.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Quantum query complexity; Theory of computation → Algorithm design techniques

Keywords and phrases Algorithm design and analysis, Query complexity, Graph algorithms, Span program algorithm, Path finding, Path detection

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.5

Related Version *Full Version*: `https://arxiv.org/pdf/2303.03319.pdf`

Funding *Shelby Kimmel and Stacey Jeffery:* sponsored by the U.S. Army Research Office and this work was accomplished under Grant Number W911NF-20-1-0327. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.
Stacey Jeffery: supported by NWO Klein project number OCENW.Klein.061, and the European Union (ERC, ASC-Q, 101040624). Views and opinions expressed are however those of the authors 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. SJ is a CIFAR Fellow in the Quantum Information Science Program.

Acknowledgements We thank Jana Sotáková and Mehrdad Tahmasbi for insightful discussions about path finding via edge sampling.

1 Introduction

Finding and detecting paths between two vertices in a graph are important related problems, both in and of themselves, and as subroutines in other applications, but there is still much to understand in this area. While classically these problems seem to be equivalent, an intriguing question is whether the same holds for quantum algorithms: there are cases where a quantum algorithm can *detect* a path between s and t in significantly less time than any known quantum algorithm takes to *find* such a path. In particular, path finding on a glued trees graph is one of Aaronson's top ten open problems in query complexity [12, 1], as the best known quantum algorithms that find an st-path in such graphs have exponentially worse running time than the best quantum algorithms for detecting one, and

understanding how these problems are related could improve our understanding of why quantum computers achieve dramatic speedups for certain problems. As an example of more immediate practical interest: path finding in supersingular isogeny graphs is one approach to attacking cryptosystems based on supersingular isogenies [11, 16], but currently the best known attack of this form still takes exponential time [36] (see also [19]).

In this paper, we consider the quantum query complexity of a somewhat intermediate problem: finding an edge on an st-path in an undirected graph.[1] In the classical case, it seems hard to imagine how one could find an edge on an st-path without first finding an st-path, but we show that in the quantum case, one can sample an st-path edge with similar resources to what is needed to detect the existence of an st-path. In some cases, this can be done with significantly fewer queries than the best previously known path-finding algorithms. We show this ability to sample an edge on a path has some useful applications, including to sabotaging networks (finding st-cut sets) and to finding paths in certain graphs faster than existing path finding algorithms.

Previously, Dürr, Heiligman, Høyer and Mhalla [18] described an algorithm for connectivity in the adjacency matrix model that uses $O(n^{3/2})$ queries for an n-vertex graph. Their algorithm works by keeping track of known connected components, and then uses a quantum search to look for any edge that connects any two components previously not known to be connected. While the authors use this algorithm to decide connectivity, we note that after $O(n^{3/2})$ queries, the algorithm will produce (with high probability) a list of the connected components of the graph, as well as a set of edges for each component that is a witness to that component's connectivity (a spanning tree). This data can then be used to find a path from s to t, if s and t are in the same component. This algorithm uses $O(\log n)$ qubits and $O(n \log n)$ classical bits, and applies to both directed and undirected graphs.

However, the algorithm of Dürr et al. does not take advantage of any structure in the graph. This is in contrast to an undirected path *detection* quantum algorithm of Belovs and Reichardt [7], further analyzed and refined in [22, 2], which, for example, can detect a path between vertices s and t with $\widetilde{O}(\sqrt{L}n)$ adjacency matrix queries when there is an st-path of length L, and even better in the case of multiple short paths, or in the case of certain promises when there is no path. In fact, there are even sufficiently structured promises on the input for which this algorithm performs superpolynomially better than the best possible classical algorithm [25]. While this path detection algorithm runs faster than $O(n^{3/2})$ in many cases, the algorithm does not output any information about the st-path – it simply determines whether a path exists.

Our contribution is an algorithm that reproduces the query complexity of the Belovs-Reichardt undirected path detection algorithm, even for structured inputs – for example, our algorithm uses $\widetilde{O}(\sqrt{L}n)$ queries when there is a path of length L – but now returns *some* information about edges on an st-path: namely, a path edge.[2] Specifically, our algorithm outputs an st-path edge sampled with probability that depends on the optimal st-flow between s and t. This is how electrons would flow in an electrical network if edges in the graph were replaced by wires with resistors and a battery were connected between s and t. For intuition, an edge is more likely to be sampled if it is on *more* or *shorter* paths. Thus,

[1] In this paper, we use *path* to refer to a self-avoiding path, meaning a path with no repeated vertices.

[2] As we hinted at with our statement of advantages for the Belovs-Reichardt algorithm in the case of shorter and/or multiple paths, the Belovs-Reichardt algorithm for st-path detection actually has a complexity that depends on the structure of the graph in a more subtle way, replacing L with an upper bound on the *effective resistance* between s and t, which is *at most* the length of the shortest path between s and t. This more subtle analysis also applies to our edge finding algorithm.

in the case of a single path between s and t, our algorithm samples each edge in the path with equal probability (up to some error in total variation distance). When there are disjoint paths of different lengths, our algorithm is more likely to sample an edge on a short path than a long path – the probability of sampling from a particular path of length ℓ is proportional to $1/\ell$. (This means, unfortunately, that if there are many long paths, we might still be more likely to sample an edge on some long path than an edge on a short path). We prove that finding an st-path edge classically requires $\Omega(n^2)$ queries in the worst case, even if promised that there is a path of length L, as long as $L \geq 3$.

With the ability to quickly find edges on short paths, we can create an improved algorithm for *finding* st-paths in undirected graphs with a unique, short st-path. Given an adjacency matrix for an n-vertex graph, if there is a unique st-path, whose (possibly unknown) length is L, we can find all of the edges in the path in $\widetilde{O}(L^{1+o(1)}n)$ expected queries. When $L = o(\sqrt{n})$, this is an improvement over the Dürr et al. algorithm. In the general case that there is more than one st-path, we prove that we can find all edges in a single path in $\widetilde{O}(L^{3/2}n)$ queries when L is the (possibly unknown) length of the *longest* path (although our approach in this case does not use the edge sampling algorithm as a subroutine). When $L = o(n^{1/3})$, this is an improvement over the Dürr et al. algorithm.

We additionally use our sampling algorithm to find st-cut sets, in the case that s and t are each part of a highly connected component, and there are only a few edges connecting those components. Because these few connecting edges are bottlenecks in the flow, there will be a lot of flow over those connecting edges, and so a high probability of sampling them, and hence finding an st-cut set. We describe a particular family of n-vertex graphs were we can find such a cut set in $\widetilde{O}(n)$ queries, where any classical algorithm would require $\Omega(n^2)$ queries.

Our edge sampling algorithm is a special case of a new span-program-based algorithm (Section 3) for generating quantum states called *span program witness states* (or simply *witness states*). One of the key elements of the analysis of span program algorithms for deciding Boolean functions [34] is the positive witness (see Definition 2), which is a vector that witnesses that the function evaluates a particular input to 1. While in the usual span program algorithm, the output on input x is $f(x)$, in our case, we output a quantum state proportional to the positive witness for input x. In the case of the Belovs-Reichardt span program for st-connectivity [7], a positive witness is a linear combination of edges that are on paths between s and t, where the amplitudes depend on the optimal st-flow (see Definition 4). Generating and then measuring such a state allows us to sample st-path edges.

Our results more generally hold for the case where the input x defines a subgraph $G(x)$ of some arbitrary graph G, that is not necessarily a complete graph. Although we do not attempt to analyze time complexity in this work, we suspect that our query algorithms on graphs are also time efficient when there is an efficient way to perform a quantum walk on the underlying graph G, as in [25]. For example, when G is the n-vertex complete graph (i.e. the oracle allows you to query elements of the full adjacency matrix for a n-vertex graph, as we have been assuming throughout this introduction), there is an efficient way to do this walk, and so in this case the time complexity of our algorithms is likely the same as the query complexity, up to log factors.

1.1 Future Directions

A natural future direction is to try to use our edge finding technique for path finding in more general settings than the ones we consider. One surprising aspect of our algorithm is that it does not necessarily find edges in the order in which they appear in the path, and instead

often finds edges in the middle of a path with high probability. The form of our algorithm thus seems to circumvent a recent lower bound on path-finding in glued trees graphs that applies to algorithms that always maintain a path from the starting node to any vertex in the algorithm's state [13]. However, one reason to be pessimistic for this particular application is that in the glued trees graph, *all* edges connected to the starting vertex are in some st-path. Still, we are hopeful that for some graphs, finding an edge in the middle of some st-path opens up the possibility of new divide-and-conquer approaches for path finding.

We are only able to take advantage of the fact that we sample edges according to the optimal st-flow for very specific graphs, like those with a single path, or with bottleneck flows, but we hope that this edge sampling distribution will prove useful in additional applications. In recent independent work, Apers and Piddock [3] develop a similar edge sampling algorithm in the adjacency list model, which they use to analyze connections between electric flows and quantum walks, and they prove that walks that proceed via their edge sampling algorithm need only logarithmically many rounds before they have a high probability of reaching a target vertex, on trees. We believe that such edge sampling methods will likely find further applications.

We have only applied our span program witness state generation algorithm to the span program for path detection. Span program algorithms exist for a wide range of graph problems, from bipartiteness [8] and cycle detection [8, 17], to triangle [9] and other subgraph detection [30], to other combinatorial search problems [6, 4]. Perhaps the span program witness states for these problems would be useful for certain applications. Beyond span program algorithms, dual adversary algorithms (which are equivalent to span programs for decision problems, but generalize to state conversion problems [31]) and multidimensional quantum walks [27, 23] all have a similar notion of witnesses in their design and analysis. Similar techniques might yield witness generation algorithms for these more general algorithm design paradigms.

We suspect our path finding algorithms are not optimal, as for graphs with longest paths of length $\Omega(n^{1/3})$, our algorithms do not outperform Dürr et al.'s algorithm. We wonder whether it is possible to find paths using $o(n^{3/2})$ queries whenever the longest path has length $o(n)$, or to prove that this is not possible, perhaps by expanding on techniques for lower bounding path-finding on welded trees [13].

Finally, all of our algorithms apply only to undirected graphs, while the algorithm of [18] applies equally well to directed or undirected graphs. While there are span program algorithms for problems on directed graphs (see e.g. [4]), they do not exhibit the same speedups with short or many paths that the undirected span program algorithms possess. It would be interesting to better understand whether there are ways to obtain similar improvements in query complexity for directed graphs.

Organization

In Section 3 we present our main technical result: an algorithm for generating a state proportional to a span program witness for x. In Section 4, we show how to apply this to finding a path edge (Section 4.1), and give an example of a particular family of graphs in which the classical complexity of finding a path edge is quadratically worse than our quantum algorithm (Theorem 20). In Section 4.2, we show how our edge finding algorithm can be applied to efficiently find an st-cut set in a particular family of graphs, and in Section 4.3 we show how it can be applied to find an st-path in $\widetilde{O}(nL^{1+o(1)})$ queries when there is a *unique* st-path of length L (Theorem 25); and also give an algorithm for finding an st-path in general graphs in $\widetilde{O}(nL^{3/2})$ queries when L is the length of the *longest* st-path (Theorem 26).

2 Preliminaries

We first introduce some basic notation. We let $\|\cdot\|$ denote the l_2 norm, $[m] \coloneqq \{1, 2, 3, \ldots, m\}$, and let $\mathcal{L}(H, V)$ denote the set of linear operators from the vector space H to the vector space V.

2.1 Span Programs

Span programs are a linear algebraic model of computation, introduced in [28], that have proven extremely useful for analyzing query [34, 35], space [24], and time complexity [7, 15, 5] in quantum algorithms. We follow Ref. [21] closely in our definitions.

▶ **Definition 1** (Span Program). *For a finite set R, a span program on R^m is a tuple $\mathcal{P} = (H, \mathcal{V}, |\tau\rangle, A)$ where*

1. *H is a direct sum of finite-dimensional inner product spaces: $H = H_1 \oplus H_2 \cdots H_m \oplus H_{true} \oplus H_{false}$, and for $j \in [m]$ and $a \in R$, we have $H_{j,a} \subseteq H_j$, such that $\sum_{a \in R} H_{j,a} = H_j$;*
2. *$\mathcal{V}$ is a vector space;*
3. *$|\tau\rangle \in \mathcal{V}$ is a target vector; and*
4. *$A \in \mathcal{L}(H, \mathcal{V})$.*

Given a string $x \in R^m$, we use $H(x)$ to denote the subspace $H_{1,x_1} \oplus \cdots \oplus H_{m,x_m} \oplus H_{true}$, and we denote by $\Pi_{H(x)}$ the orthogonal projector onto the space $H(x)$.

An important concept in the analysis of span programs and quantum query complexity is that of *witnesses*:

▶ **Definition 2** (Positive Witness). *Given a span program $\mathcal{P} = (H, \mathcal{V}, |\tau\rangle, A)$ on R^m and $x \in R^m$, $|w\rangle \in H(x)$ is a* positive witness *for x in $\mathcal{P}$ if $A|w\rangle = |\tau\rangle$. If a positive witness exists for x, we define the* witness size *of x in $\mathcal{P}$ as*

$$w_+(x) = w_+(\mathcal{P}, x) \coloneqq \min\left\{ \||w\rangle\|^2 : |w\rangle \in H(x) \text{ and } A|w\rangle = |\tau\rangle \right\}. \tag{1}$$

We say that $|w\rangle \in H(x)$ is the optimal positive witness for x if $\||w\rangle\|^2 = w_+(\mathcal{P}, x)$ and $A|w\rangle = |\tau\rangle$.

Our main algorithm produces a normalized version of this unique optimal positive witness, $|w\rangle/\||w\rangle\|$. (To see that the optimal positive witness is unique, for contradiction assume that the optimal positive witness is not unique – then a linear combination of two optimal positive witnesses produces a witness with smaller witness size than either.)

A span program $\mathcal{P}$ encodes a function $f : X \to \{0, 1\}$ in the following way. We say $f(x) = 1$ if x has a positive witness, and $f(x) = 0$ if x does not have a positive witness. We say such a $\mathcal{P}$ decides the function f.

We will also need the concept of an approximate negative witness.

▶ **Definition 3** (Negative Error, Approximate Negative Witness). *Given a span program $\mathcal{P} = (H, \mathcal{V}, |\tau\rangle, A)$ on R^m and $x \in R^m$, we define the negative error of x in $\mathcal{P}$ as*

$$e_-(x, \mathcal{P}) \coloneqq \min\left\{ \|\langle\widetilde{\omega}|A\Pi_{H(x)}\|^2 : \langle\widetilde{\omega}| \in \mathcal{L}(\mathcal{V}, \mathbb{R}), \langle\widetilde{\omega}|\tau\rangle = 1 \right\}. \tag{2}$$

Note that $e_-(x, \mathcal{P}) = 0$ if and only if $\mathcal{P}$ decides a function f with $f(x) = 0$. Any $\langle\widetilde{\omega}|$ such that $\|\langle\widetilde{\omega}|A\Pi_{H(x)}\|^2 = e_-(x, \mathcal{P})$ is called an approximate negative witness for x in P. We define the approximate negative witness size of x as:

$$\widetilde{w}_-(x, \mathcal{P}) \coloneqq \min\left\{ \|\langle\widetilde{\omega}|A\|^2 : \langle\widetilde{\omega}| \in \mathcal{L}(\mathcal{V}, \mathbb{R}), \langle\widetilde{\omega}|\tau\rangle = 1, \|\langle\widetilde{\omega}|A\Pi_{H(x)}\|^2 = e_-(x, \mathcal{P}) \right\}. \tag{3}$$

We call an approximate negative witness $\langle\widetilde{\omega}|$ that also minimizes $\|\langle\widetilde{\omega}|A\|^2$ an optimal approximate negative witness.

We use the following notation for maximum positive and approximate negative witness sizes:

$$W_+(\mathcal{P}, f) = W_+ \coloneqq \max_{x \in f^{-1}(1)} w_+(\mathcal{P}, x), \qquad \widetilde{W}_-(\mathcal{P}, f) = \widetilde{W}_- \coloneqq \max_{x \in f^{-1}(1)} \widetilde{w}_-(x, \mathcal{P}). \tag{4}$$

Note that we are restricting to 1-inputs of f. That is because our witness generation algorithm will assume that x is a 1-input, unlike previous span-program-based algorithms that *decide* f.

2.2 Quantum Query Algorithms

The algorithms we develop are query algorithms, where we can access a unitary oracle O_x for some $x \in X \subseteq R^m$ such that O_x acts on the space $\mathbb{C}^m \otimes \mathbb{C}^q$ as $O_x|i\rangle|a\rangle = |i\rangle|x_i + a \mod q\rangle$. where $q = |R|$, x_i is the value of the i^{th} element of x and $|i\rangle \in \mathbb{C}^m$ and $|a\rangle \in \mathbb{C}^q$ are standard basis states.

The query complexity of an algorithm is the number of times O_x must be used, in the worst case over $x \in X$. In our case, we will also consider the expected query complexity on input x, which is the average number of times O_x must be used when given a particular input x, where the randomness is due to random events in the course of the algorithm.

2.3 Graph Theory and Connection to Span Programs

Let $G = (V, E)$ be an undirected graph.[3] We will particularly consider graphs with specially labeled vertices $s, t \in V$, such that there is a path from s to t in G. Let $\overrightarrow{E} = \{(u, v) : \{u, v\} \in E\}$; that is $\overrightarrow{E}$ is the set of directed edges corresponding to the edges of G. Given a graph $G = (V, E)$, for $u \in V$, we denote by G_u^- the subgraph of G on the vertices $V \setminus \{u\}$, and with overloading of notation for $S \subseteq E$, we denote by G_S^- the subgraph of G with edges S removed. (It will be clear from context whether we are removing edges or vertices from the graph.)

On a graph G with s and t connected we will consider a *unit st-flow*, which is a linear combination of cycles and st-paths, formally defined as a function on $\overrightarrow{E}$ with the following properties.

▶ **Definition 4** (Unit *st*-flow). *Let $G = (V, E)$ be an undirected graph with $s, t \in V(G)$, and s and t connected. Then a* unit *st*-flow *on G is a function $\theta : \overrightarrow{E} \to \mathbb{R}$ such that:*

1. *For all $(u, v) \in \overrightarrow{E}$, $\theta(u, v) = -\theta(v, u)$;*
2. *$\sum_{v:(s,v)\in\overrightarrow{E}} \theta(s, v) = \sum_{v:(v,t)\in\overrightarrow{E}} \theta(v, t) = 1$; and*
3. *for all $u \in V \setminus \{s, t\}$, $\sum_{v:(u,v)\in\overrightarrow{E}} \theta(u, v) = 0$.*

▶ **Definition 5** (Unit Flow Energy). *Given a graph $G = (V, E)$ and a unit st-flow θ on G, the* unit flow energy *of θ is $J(\theta) = \frac{1}{2}\sum_{e\in\overrightarrow{E}} \theta(e)^2$.*

▶ **Definition 6** (Effective resistance). *Let $G = (V, E)$ be a graph with $s, t \in V$. If s and t are connected in G, the* effective resistance *of G between s and t is $R_{s,t}(G) = \min_\theta J(\theta)$, where θ runs over all unit st-unit flows of G. If s and t are not connected in G, $R_{s,t}(G) = \infty$.*

[3] Our results easily extend to multigraphs, see [22], but for simplicity, we will not consider multigraphs here.

Interpretation of the optimal flow

The st-flow with minimum energy is unique, and describes the electric current going through that edge if the graph represents a network of unit resistors and we put a potential difference between s and t. The minimum energy flow has several other interpretations and connections to other graph properties. For reference, and for those who would like to build their intuition for this object, we have collected some of these relationships in the full version of this work [26, Appendix A].

Graph access

We turn graph problems into oracle problems by letting a string $x \in \{0,1\}^m$ specify a subgraph $G(x)$ of G. In particular, we associate each edge $e \in E$ with a number in $[m]$. Then, given a string $x \in \{0,1\}^m$, let $G(x) = (V, E(x))$ be the subgraph of G that contains an edge $e \in E$ if e is associated with the integer $i \in [m]$ and $x_i = 1$, where x_i is the ith bit of x. In this oracle problem, one is given access to an oracle O_x for x (or classically, given the ability to query the values of the bits of x one at a time), and a description of the parent graph G along with the association between bits of x and edges of G, and the goal is to determine something about the graph $G(x)$ using as few queries as possible. Let $E_i \subset E$ be the set of edges associated with the ith bit of x. When not specified otherwise, one should assume that $m = |E|$, and then associate each edge of G uniquely with a bit of the input string. In this case, when G is the complete graph, O_x is equivalent to query access to the adjacency matrix of a graph. When we consider subgraphs of the original graph (like G_u^-), we assume that the edges are associated with the same indices as in the original graph, unless otherwise specified.

Most of the applications in this paper are related to the problem of detecting a path between s and t – more commonly called st-connectivity. We define $st\text{-CONN}_G(x) := 1$ if s and t are connected in $G(x)$, and 0 otherwise. The following span program, which we denote by $\mathcal{P}_{G_{st}}$, first introduced in Ref. [28] and used in the quantum setting in Ref. [7], decides $st\text{-CONN}_G(x)$: for a graph $G = (V, E)$, where $m = |E|$, define the span program $\mathcal{P}_{G_{st}}$ as:

$$\begin{aligned}
&\forall i \in [m], H_{i,1} = \text{span}\{|(u,v)\rangle : \{u,v\} \in E_i\}, H_{i,0} = \emptyset \\
&\mathcal{V} = \text{span}\{|v\rangle : v \in V(G)\} \\
&|\tau\rangle = |s\rangle - |t\rangle \\
&\forall (u,v) \in \overrightarrow{E} : \ A|u,v\rangle = |u\rangle - |v\rangle.
\end{aligned} \tag{5}$$

For $\mathcal{P}_{G_{st}}$, the negative approximate witness size is bounded by $\widetilde{W}_- = O(n^2)$ [21]. If s and t are connected in $G(x)$, the optimal positive witness of x in $\mathcal{P}_{G_{st}}$ is [7, 22]

$$|\theta^*\rangle = \frac{1}{2} \sum_{e \in \overrightarrow{E}} \theta^*(e)|e\rangle, \tag{6}$$

where θ^* is the st-unit flow with minimal energy, so by Definitions 2 and 6, $w_+(\mathcal{P}_{G_{st}}, x) = \frac{1}{2} R_{s,t}(G(x))$.

One of our main applications is to apply our witness state generation algorithm to the span program $\mathcal{P}_{G_{st}}$, in which case, we produce a quantum state close to $|\theta^*\rangle/\||\theta^*\rangle\|$ where θ^* is the optimal unit st-flow on $G(x)$. If we were to create $|\theta^*\rangle/\||\theta^*\rangle\|$ exactly, and then measure in the standard basis, the probability that we obtain the edge e is $\theta^*(e)^2/(2R_{s,t}(G(x)))$. Let $q_{G(x),s,t}$ denote the distribution such that for $\forall e \in \overrightarrow{E}$,

$$q_{G(x),s,t}(e) = \theta^*(e)^2/(2R_{s,t}(G(x))). \tag{7}$$

Additionally, this optimal flow θ^* is a convex combination of (self-avoiding) st-paths, as we prove in the full version of this work [26, Appendix A]:

▶ **Lemma 7.** *An* st-path *in* $G(x)$ *is a sequence of* distinct *vertices* $\vec{u} = (u_0, \ldots, u_\ell)$ *such that* $s = u_0$, $t = u_\ell$, *and for all* $i \in [\ell]$, $(u_{i-1}, u_i) \in \overrightarrow{E}(G(x))$. *From* $\vec{u}$, *we define*

$$|\rho_{\vec{u}}\rangle = \frac{1}{\sqrt{2}} \sum_{i=0}^{\ell-1} (|u_i, u_{i+1}\rangle - |u_{i+1}, u_i\rangle) \tag{8}$$

and refer to all such states as st-path states of $G(x)$. *Then if* $|\theta^*\rangle$ *is the optimal positive witness for* x *in* $\mathcal{P}_{G_{s,t}}$, *it is a linear combination of* st*-path states in* $G(x)$.

A final pair of tools we use are a quantum algorithm that decides st-$\text{CONN}_G(x)$ with fewer queries in the case of small effective resistance, without knowing the effective resistance ahead of time, and a quantum algorithm for estimating the effective resistance:

▶ **Lemma 8** ([2])**.** *Fix* $\delta > 0$ *and a family of* n*-vertex graphs* G *with vertices* s *and* t. *Then there is a quantum algorithm* $\texttt{PathDetection}(O_x, G, s, t, \delta)$ *such that,*

1. *The algorithm returns* st-$\text{CONN}_G(x)$ *with probability* $1 - O(\delta)$.
2. *On input* x, *the algorithm uses* $O\left(n\sqrt{R_{s,t}(G(x))} \log\left(\frac{n}{R_{s,t}(G(x))\delta}\right)\right)$ *expected queries if* st-$\text{CONN}_G(x) = 1$, *and* $O\left(n^{3/2} \log 1/\delta\right)$ *expected queries if* st-$\text{CONN}_G(x) = 0$.

▶ **Lemma 9** ([21])**.** *Fix* $\delta > 0$ *and a family of* n*-vertex graphs* G *with vertices* s *and* t. *Then there is a quantum algorithm* $\texttt{WitnessSizeEst}(O_x, G, s, t, \epsilon, \delta)$ *that, on input* x *such that* st-$\text{CONN}_G(x) = 1$, *with probability* $1 - \delta$, *outputs an estimate* $\hat{R}$ *for* $R_{s,t}(G(x))$ *such that*

$$\left|\hat{R} - R_{s,t}(G(x))\right| \leq \epsilon R_{s,t}(G(x)), \tag{9}$$

using $\widetilde{O}\left(\sqrt{\frac{R_{s,t}(G(x))n^2}{\epsilon^3}} \log(1/\delta)\right)$ *expected queries; and on input* x *such that* st-$\text{CONN}_G(x) = 0$, *uses at most* $\widetilde{O}\left((n/\epsilon)^{3/2} \log(1/\delta)\right)$.

Lemma 9 is a special case of [21, Theorem 3.8], which gives an algorithm for estimating the quantity $w_+(x)$ from *any* span program. If we apply this construction with the span program $\mathcal{P}_{G_{s,t}}$, we can estimate its positive witness sizes, which are precisely $\frac{1}{2}R_{s,t}(G(x))$. The algorithm described in [21, Theorem 3.8] assumes that the input is a 1-input to st-$\text{CONN}_G(x)$, but can easily be modified to always stop after at most $\widetilde{O}\left((n/\epsilon)^{3/2} \log(1/\delta)\right)$ steps, regardless of the input, since $R_{s,t}(G(x)) \leq n$. The algorithm as stated also only works with bounded error, but the success probability can be amplified to $1 - \delta$ by repeating $\log(1/\delta)$ times and taking the median estimate.

3 Witness Generation

Our main technical result, on generating span program witness states is the following:

▶ **Theorem 10.** *Given a span program* $\mathcal{P}$ *that decides a function* f, *and constants* ϵ, δ, *there is an algorithm (Algorithm 1) that, given as input an oracle* O_x *such that* $f(x) = 1$ *with optimal positive witness* $|w\rangle$, *outputs a state* $|\hat{w}\rangle/\||\hat{w}\rangle\|$ *such that* $\left\||w\rangle/\sqrt{w_+(x)} - |\hat{w}\rangle/\||\hat{w}\rangle\|\right\|^2 \leq O(\epsilon)$ *with probability* $1 - O(\delta)$, *and uses* $\widetilde{O}\left(\sqrt{\frac{w_+(x)\widetilde{W}_-}{\epsilon}} \log\left(\frac{1}{\delta}\right)\right)$ *expected queries to* O_x.

For comparison, a span program algorithm can *decide* f with bounded error in expected query complexity $\widetilde{O}\left(\sqrt{w_+(x)\widetilde{W}_-}\right)$, so Theorem 10 gives a matching complexity for generating a witness state. As we will see in Section 4.1, in the case of the span program $\mathcal{P}_{G_{s,t}}$ for st-connectivity on subgraphs of G, this implies that we can sample an st-path edge in the same complexity used by the span program algorithm to decide if an st-path exists.

A key subroutine for our witness state generation algorithm will be quantum phase estimation. In quantum phase estimation one implements a controlled version of a unitary U acting on a Hilbert space $\mathcal{H}_A$ on an input state $|\psi\rangle \in \mathcal{H}_A$. The state $|\psi\rangle$ can be decomposed into its eigenbasis with respect to U as $|\psi\rangle = \sum_i \alpha_i|\lambda_i\rangle$, where $U|\lambda_i\rangle = e^{i\phi_i\pi}$ and we say ϕ_i is the phase of the state $|\lambda_i\rangle$. Then when phase estimation is performed with precision Θ the probability that you measure a phase of 0 after the phase estimation procedure is approximately given by $\sum_{i:|\phi_i|\leq\Theta} |\alpha_i|^2$, and the non-normalized state that results after measuring a phase of 0 is approximately $\sum_{i:|\phi_i|\leq\Theta} \alpha_i|\lambda_i\rangle$. In other words, phase estimation can be used to project into the low phase space (with phase less than Θ) with probability that depends on the amount of amplitude the original state had on low-phase eigenstates. For an accuracy parameter ϵ, the number of uses of U in phase estimation scales as $O\left(\frac{1}{\Theta}\log\frac{1}{\epsilon}\right)$. A more rigorous description of the guarantees of phase estimation is given below in Lemma 11.

The basic idea of the algorithm that we use to prove Theorem 10 is to apply phase estimation with a unitary $U(\mathcal{P}, x, \alpha)$, (which can be implemented with access to an oracle O_x and depends on a span program $\mathcal{P}$, and a positive real parameter α), on a state $|\hat{0}\rangle$. We show that the eigenspectrum of $|\hat{0}\rangle$ relative to $U(\mathcal{P}, x, \alpha)$ decomposes into two states, $|\hat{0}\rangle \oplus \frac{1}{\alpha}|w\rangle$, which is a 0-phase eigenstate of $U(\mathcal{P}, x, \alpha)$, and $|\psi_{x,+}\rangle$, which has small overlap with the low-phase space of $U(\mathcal{P}, x, \alpha)$.

If we do phase estimation with $U(\mathcal{P}, x, \alpha)$ on $|\hat{0}\rangle$ with sufficiently small precision, and then if we measure a phase of 0, as discussed above, we will approximately project into the state $|\hat{0}\rangle \oplus \frac{1}{\alpha}|w\rangle$. From there, if we make the measurement $\{|\hat{0}\rangle\langle\hat{0}|, I - |\hat{0}\rangle\langle\hat{0}|\}$, and obtain outcome $I - |\hat{0}\rangle\langle\hat{0}|$ the state will project into $|w\rangle$, as desired.

Next, there comes a balancing act for our choice of α. When α is too small, $|\hat{0}\rangle$ has small overlap with the span of $|\hat{0}\rangle \oplus \frac{1}{\alpha}|w\rangle$, so we are not very likely to measure a phase of 0 when we do phase estimation with $U(\mathcal{P}, x, \alpha)$ on $|\hat{0}\rangle$. However, when α gets too large, while it becomes very likely to measure a phase of 0 and thus obtain the state $|\hat{0}\rangle \oplus \frac{1}{\alpha}|w\rangle$, we will be unlikely to subsequently measure outcome $I - |\hat{0}\rangle\langle\hat{0}|$.

The sweet spot is when $\alpha \approx \sqrt{w_+(x)}$, in which case both measurement outcomes we require have a reasonable probability of occurring. Since we don't know $w_+(x)$ ahead of time, we must first estimate an appropriate value of α to use, which we do by iteratively testing larger and larger values of α.[4] Our test involves estimating the probability of measuring a phase of 0 when phase estimation with $U(\mathcal{P}, x, \alpha)$ is performed on $|\hat{0}\rangle$, which we show provides an estimate of $\alpha/\sqrt{w_+(x)}$.

3.1 Proof of Theorem 10

Before introducing the algorithm we use to prove Theorem 10, we introduce some key concepts, lemmas, and theorems that will be used in the analysis.

Let $\tilde{H} = H \oplus \text{span}\{|\hat{0}\rangle\}$, and $\tilde{H}(x) = H(x) \oplus \text{span}\{|\hat{0}\rangle\}$, where $|\hat{0}\rangle$ is orthogonal to H. Then we define $\tilde{A}^\alpha \in \mathcal{L}(\tilde{H}, \mathcal{V})$ as

$$\tilde{A}^\alpha = \frac{1}{\alpha}|\tau\rangle\langle\hat{0}| - A. \tag{10}$$

[4] There is a similar algorithm in [21] that estimates $w_+(x)$, but it is more precise than we require.

Let $\Lambda^\alpha \in \mathcal{L}(\tilde{H}, \tilde{H})$ be the orthogonal projection onto the kernel of $\tilde{A}^\alpha$, and let $\Pi_x \in \mathcal{L}(\tilde{H}, \tilde{H})$ be the orthogonal projector onto $\tilde{H}(x)$. Finally, let $U(\mathcal{P}, x, \alpha) = (2\Pi_x - I)(2\Lambda^\alpha - I)$. Note that $2\Pi_x - I$ can be implemented with two applications of O_x [21, Lemma 3.1], and $2\Lambda^\alpha - I$ can be implemented without any applications of O_x.

We will use parallelized phase estimation, as described in Ref. [32], which provides improved error bounds over standard phase estimation. In particular, given a unitary U acting on a Hilbert Space $\mathcal{H}$, a precision $\Theta > 0$, and an accuracy $\epsilon > 0$, we can create a circuit $D(U)$ that implements $O(\log \frac{1}{\epsilon})$ parallel copies of the phase estimation circuit on U, each to precision $O(\Theta)$, that each estimate the phase of a single copy of a state $|\psi\rangle$. That is, $D(U)$ acts on the space $\mathcal{H}_A \otimes ((\mathbb{C}^2)^{\otimes b})_B$ where $b = O\left(\log \frac{1}{\Theta} \log \frac{1}{\epsilon}\right)$, and A labels the input state register, and B labels the registers that store the results of the parallel phase estimations.

We use the circuit $D(U)$ to check if an input state has high overlap with the low-valued eigenphase-space of U [29, 14, 32]. To characterize the low phase space of a unitary U, let $P_\Theta(U)$ (or just P_Θ when U is clear from context) be the projection onto $\text{span}\{|u\rangle : U|u\rangle = e^{i\theta}|u\rangle \text{ with } |\theta| \leq \Theta\}$ (the eigenspace of U with eigenphases less than Θ). Then the following lemma provides key properties of parallel phase estimation circuit $D(U)$:

▶ **Lemma 11** ([29, 14, 32]). *Let U be a unitary on a Hilbert Space $\mathcal{H}_A$, and let $\Theta, \epsilon > 0$. We call Θ the precision and ϵ the accuracy. Then there is a circuit $D(U)$ that acts on the space $\mathcal{H}_A \otimes ((\mathbb{C}^2)^{\otimes b})_B$ for $b = O\left(\log \frac{1}{\Theta} \log \frac{1}{\epsilon}\right)$, and that uses $O\left(\frac{1}{\Theta} \log \frac{1}{\epsilon}\right)$ controlled calls to U. Then for any state $|\psi\rangle \in \mathcal{H}_A$,*

1. $D(U)(P_0|\psi\rangle)_A|0\rangle_B = (P_0|\psi\rangle)_A|0\rangle_B$
2. $\|P_0|\psi\rangle\|^2 \leq \|(I_A \otimes |0\rangle\langle 0|_B)D(U)(|\psi\rangle_A|0\rangle_B)\|^2 \leq \|P_\Theta|\psi\rangle\|^2 + \epsilon$.

Iterative Quantum Amplitude Estimation is a robust version of amplitude estimation, which uses repeated applications of amplitude estimation to achieve improved error bounds:

▶ **Lemma 12** (Iterative Quantum Amplitude Estimation [20]). *Let $\delta > 0$ and $\mathcal{A}$ be a unitary quantum circuit such that on a state $|0\rangle$, $\mathcal{A}|\psi\rangle = \alpha_0|0\rangle|\psi_0\rangle + \alpha_1|1\rangle|\psi_1\rangle$. Then there is an algorithm that estimates $|\alpha_0|^2$ to additive error δ with success probability at least $1 - p$ using $O\left(\frac{1}{\delta} \log\left(\frac{1}{p} \log \frac{1}{\delta}\right)\right)$ calls to $\mathcal{A}$ and $\mathcal{A}^\dagger$.*

A key mathematical tool in analyzing span program algorithms is the Effective Spectral Gap Lemma:

▶ **Lemma 13** (Effective Spectral Gap Lemma, [31]). *Let Π and Λ be projections, and let $U = (2\Pi - I)(2\Lambda - I)$ be the unitary that is the product of their associated reflections. If $\Lambda|w\rangle = 0$, then $\|P_\Theta(U)\Pi|w\rangle\| \leq \frac{\Theta}{2}\||w\rangle\|$.*

We will need the following relationship between optimal positive witnesses and optimal negative approximate witnesses:

▶ **Theorem 14.** [21, Theorem 2.11] *Given a span program $\mathcal{P} = (H, \mathcal{V}, |\tau\rangle, A)$ on R^m and $x \in R^m$, if $|w\rangle$ is the optimal positive witness for x and $\langle\widetilde{\omega}|$ is an optimal negative approximate witness for x, then*

$$|w\rangle = w_+(x)\Pi_{H(x)}(\langle\widetilde{\omega}|A)^\dagger. \tag{11}$$

As discussed following Theorem 10, we decompose the state $|\hat{0}\rangle$ into a linear combination of two orthogonal states. They are

$$\begin{aligned}|\psi_{x,0}\rangle =&|\hat{0}\rangle + \frac{1}{\alpha}|w\rangle,\\ |\psi_{x,+}\rangle =&|\hat{0}\rangle - \frac{\alpha}{w_+(x)}|w\rangle,\end{aligned} \tag{12}$$

so we can write $|\hat{0}\rangle$ as

$$|\hat{0}\rangle = a_0|\psi_{x,0}\rangle + a_+|\psi_{x,+}\rangle, \quad \text{where} \quad a_0 = \frac{1}{1+\frac{w_+(x)}{\alpha^2}}, \qquad a_+ = \frac{1}{1+\frac{\alpha^2}{w_+(x)}}. \tag{13}$$

We first show that $|\psi_{x,0}\rangle$ is a 0-phase eigenvector of $U(\mathcal{P},x,\alpha)$. Note that $\tilde{A}^\alpha|\psi_{x,0}\rangle = \frac{1}{\alpha}(|\tau\rangle - |\tau\rangle) = 0$ (see Equation (10)), so recalling that Λ^α is the orthogonal projector onto the kernel of $\tilde{A}^\alpha$, we have $\Lambda^\alpha|\psi_{x,0}\rangle = |\psi_{x,0}\rangle$. Furthermore, since Π_x is the orthogonal projector onto $\tilde{H}(x) = H(x) \oplus \text{span}\{|\hat{0}\rangle\}$, it follows that $\Pi_x|\psi_{x,0}\rangle = |\psi_{x,0}\rangle$, where we use that $|w\rangle$ is a positive witness, so $|w\rangle \in H(x)$. Thus $U(\mathcal{P},x,\alpha)|\psi_{x,0}\rangle = |\psi_{x,0}\rangle$.

On the other hand $|\psi_{x,+}\rangle$ has low overlap with $P_\Theta(U(\mathcal{P},x,\alpha))$ for small enough Θ and α, as the following lemma shows.

▶ **Lemma 15.** *If $\alpha^2 \geq 1/\widetilde{W}_-$, then $\|P_\Theta(U(\mathcal{P},x,\alpha))|\psi_{x,+}\rangle\| \leq \Theta\alpha\sqrt{\widetilde{W}_-}$.*

Proof. Let $\langle\widetilde{\omega}|$ be an optimal negative approximate witness for x (see Definition 3), and let

$$|v\rangle = |\hat{0}\rangle - \alpha(\langle\widetilde{\omega}|A)^\dagger. \tag{14}$$

Using Theorem 14 and the fact that $\Pi_x|\hat{0}\rangle = |\hat{0}\rangle$, we have that

$$\Pi_x|v\rangle = |\hat{0}\rangle - \alpha\Pi_{H(x)}(\langle\widetilde{\omega}|A)^\dagger = |\hat{0}\rangle - \alpha\frac{|w\rangle}{w_+(x)} = |\psi_{x,+}\rangle. \tag{15}$$

Now we will show $\Lambda^\alpha|v\rangle = 0$. Let $|k\rangle$ be in the kernel of $\tilde{A}^\alpha$, so $\tilde{A}^\alpha|k\rangle = 0$. Using Equation (10) and rearranging,

$$A|k\rangle = \frac{1}{\alpha}|\tau\rangle\langle\hat{0}|k\rangle. \tag{16}$$

Then

$$\begin{aligned}\langle v|k\rangle &= \langle\hat{0}|k\rangle - \alpha\langle\widetilde{\omega}|A|k\rangle\\ &= \langle\hat{0}|k\rangle - \langle\hat{0}|k\rangle\langle\widetilde{\omega}|\tau\rangle\\ &= 0\end{aligned} \tag{17}$$

where we have used Equations (14) and (16) and the properties of optimal negative approximate witnesses. Thus $|v\rangle$ is orthogonal to any element of the kernel of $\tilde{A}^\alpha$, so $\Lambda^\alpha|v\rangle = 0$.

Now we can apply Lemma 13 to $|v\rangle$ to get:

$$\begin{aligned}\|P_\Theta(U(\mathcal{P},x,\alpha))|\psi_{x,+}\rangle\| &= \|P_\Theta(U(\mathcal{P},x,\alpha))\Pi_x|v\rangle\|\\ &\leq \frac{\Theta}{2}\||v\rangle\|\\ &= \frac{\Theta}{2}\sqrt{1+\alpha^2\widetilde{w}_-(x,\mathcal{P})}\\ &\leq \Theta\alpha\sqrt{\widetilde{W}_-},\end{aligned} \tag{18}$$

where in the first line we have used Equation (15), and in the last, our assumption that $\alpha^2\widetilde{W}_- \geq 1$. ◀

▶ **Corollary 16.** $\|P_0(U(\mathcal{P},x,\alpha))|\psi_{x,+}\rangle\| = 0$.

Proof. Apply Lemma 15 with Θ set to 0. ◀

To prove Theorem 10, we analyze the following algorithm:

Algorithm 1 `WitnessGeneration`$(\mathcal{P}, O_x, \delta, \epsilon)$.

Input : Error tolerance δ, accuracy ϵ, span program $\mathcal{P}$ that decides a function f, oracle O_x

Output : A quantum state $|\hat{w}\rangle/\||\hat{w}\rangle\|$ such that for the optimal positive witness $|w\rangle$ for x, $\||w\rangle/\sqrt{w_+(x)} - |\hat{w}\rangle/\||\hat{w}\rangle\|\|^2 \leq O(\epsilon)$ with probability $1 - O(\delta)$

$\epsilon' \leftarrow \min\{\epsilon, 1/96\}$; $\quad T \leftarrow \left\lceil \log\sqrt{W_+\widetilde{W}_-} \right\rceil$;

$p \leftarrow \min\left\{\delta/\log(W_+\widetilde{W}_-), 1/\sqrt{W_+\widetilde{W}_-}\right\}$

`// Probing Stage`

for $i = 0$ **to** T **do**

  $\alpha \leftarrow 2^i/\sqrt{\widetilde{W}_-}$

  $\hat{a} \leftarrow$ Iterative Amplitude Estimation (Lemma 12) estimate (with probability of failure p and additive error $1/48$) of the probability of outcome $|0\rangle_B$ in register B when $D(U(\mathcal{P},x,\alpha))$ (see Lemma 11) acts on $|\hat{0}\rangle_A|0\rangle_B$ with error ϵ', precision $\sqrt{\frac{\epsilon'}{\alpha^2\widetilde{W}_-}}$

  if $\frac{15}{48} \leq \hat{a} \leq \frac{35}{48}$ **then** Break

`// State Generation Stage`

for $j = 1$ **to** $\log(1/\delta)$ **do**

  Apply $D(U(\mathcal{P},x,\alpha))$ to $|\hat{0}\rangle_A|0\rangle_B$ with error ϵ', precision $\sqrt{\frac{\epsilon'}{\alpha^2\widetilde{W}_-}}$

  Make a measurement with outcome $M = \{(I - |\hat{0}\rangle\langle\hat{0}|)_A \otimes |0\rangle\langle 0|_B\}$ on the resultant state

  if *Measure outcome* M **then**

    Return the resultant state

Return "failure"

To analyze Algorithm 1, will need the following lemma and corollary. In Algorithm 1, we estimate the probability of measuring the outcome $|0\rangle$ in the B register after doing phase estimation. In the following lemma, we prove this probability is closely related to a_0 from Equation (13).

▶ **Lemma 17.** *Applying $D(U(\mathcal{P},x,\alpha)))$ with error ϵ and precision $\sqrt{\frac{\epsilon}{\alpha^2\widetilde{W}_-}}$ (see Lemma 11) to input state $|\hat{0}\rangle_A|0\rangle_B$ for $\alpha \geq 1/\sqrt{\widetilde{W}_-}$ results in the outcome $|0\rangle$ in the B register with probability in the range $[a_0, a_0 + 2\epsilon]$.*

Proof. Throughout the proof, let $U = U(\mathcal{P},x,\alpha)$. The probability that we measure $|0\rangle$ in register B after we apply $D(U)$ with error ϵ and precision Θ to $|\hat{0}\rangle_A|0\rangle_B$ is, by Lemma 11 Item 2, at most

$$\|P_\Theta(U)|\hat{0}\rangle\|^2 + \epsilon = \|a_0 P_\Theta(U)|\psi_{x,0}\rangle + a_+ P_\Theta(U)|\psi_{x,+}\rangle\|^2 + \epsilon, \tag{19}$$

by Equation (13). Now $P_\Theta(U)|\psi_{x,0}\rangle$ and $P_\Theta(U)|\psi_{x,+}\rangle$ are orthogonal, since

$$\langle\psi_{x,0}|P_\Theta(U)P_\Theta(U)|\psi_{x,+}\rangle = \langle\psi_{x,0}|\psi_{x,+}\rangle = 0, \tag{20}$$

where we've used that $P_\Theta(U)|\psi_{x,0}\rangle = |\psi_{x,0}\rangle$ and that $|\psi_{x,0}\rangle$ and $|\psi_{x,+}\rangle$ are orthogonal. Continuing from Equation (19) and using the orthogonality condition, we have, using $\Theta = \sqrt{\frac{\epsilon}{\alpha^2 \widetilde{W}_-}}$,

$$\begin{aligned}
\|P_\Theta(U)|\hat{0}\rangle\|^2 + \epsilon &= a_0^2 \|P_\Theta(U)|\psi_{x,0}\rangle\|^2 + a_+^2 \|P_\Theta(U)|\psi_{x,+}\rangle\|^2 + \epsilon \\
&\leq a_0^2 \||\psi_{x,0}\rangle\|^2 + a_+^2 \Theta^2 \alpha^2 \widetilde{W}_- + \epsilon && \text{by Lemma 15, since } \alpha^2 \widetilde{W}_- \geq 1 \\
&\leq a_0 + a_+^2 \epsilon + \epsilon \\
&\leq a_0 + 2\epsilon, && (21)
\end{aligned}$$

where we have used that $\||\psi_{x,0}\rangle\|^2 = 1/a_0$, and $a_+ \leq 1$ (see Equation (13)) .

By Lemma 11 Item 2, the probability that we measure $|0\rangle$ in register B after applying $D(U(\mathcal{P}, x, \alpha))$ on $|\hat{0}\rangle_A |0\rangle_B$ with error ϵ and any precision is at least

$$\|P_0(U)|\hat{0}\rangle\|^2 = \|a_0 P_0(U)|\psi_{x,0}\rangle + a_+ P_0(U)|\psi_{x,+}\rangle\|^2 = a_0^2 \||\psi_{x,0}\rangle\|^2 = a_0, \tag{22}$$

where we have used Corollary 16. ◀

▶ **Corollary 18.** *In Algorithm 1, if in an iteration of the Probing Stage, Iterative Amplitude Estimation does not fail at Line 4 and subsequently causes a break at Line 5, then*

$$a_0 \in \left[\frac{1}{4}, \frac{3}{4}\right], \qquad \frac{a_0^2 w_+(x)}{\alpha^2} \in \left[\frac{3}{16}, \frac{1}{4}\right]. \tag{23}$$

Proof. If Iterative Amplitude Estimation does not fail at Line 4 and causes a break at Line 5, then we have an estimate $\hat{a}$ that is in the range $[\frac{15}{48}, \frac{35}{48}]$. Thus, because of the additive error of $1/48$ in Iterative Amplitude Estimation, the probability of measuring outcome $|0\rangle_B$ is in the range $[\frac{14}{48}, \frac{36}{48}]$. By Lemma 17, this same probability is in the range $[a_0, a_0 + 2\epsilon']$, so in particular these two ranges overlap. Thus, since we choose $2\epsilon'$ to be at most $1/48$, we have that

$$a_0 \in \left[\frac{13}{48}, \frac{36}{48}\right] \subset \left[\frac{1}{4}, \frac{3}{4}\right]. \tag{24}$$

Using $a_0 = (1 + \frac{w_+(x)}{\alpha^2})^{-1}$ (see Equation (13)), this implies the stated ranges for $\frac{a_0^2 w_+(x)}{\alpha^2} = a_0(1 - a_0)$. ◀

Now we prove the main performance guarantees of Algorithm 1, bounding the success probability and the expected query complexity, thus proving Theorem 10.

Proof of Theorem 10. Letting $U = U(\mathcal{P}, x, \alpha)$, we analyze Algorithm 1. We first show that the algorithm will produce the desired state if both the Probing Stage and the State Generation stage are successful. Then we will analyze the probability of this occurring, in order to bound the success probability of the algorithm.

We say the Probing Stage is successful if in some iteration, Iterative Amplitude estimation, having not failed thus far, does not fail and then triggers a break at Line 6, in which case we can apply Corollary 18. Under these assumptions, we consider the outcome of a successful State Generation stage, when we achieve the measurement outcome $M = (I - |\hat{0}\rangle\langle\hat{0}|)_A \otimes |0\rangle\langle 0|_B$. The non-normalized state $|\hat{w}\rangle$ that is produced upon measurement outcome M is

$$
\begin{aligned}
|\hat{w}\rangle &= (I - |\hat{0}\rangle\langle\hat{0}|)_A \otimes |0\rangle\langle 0|_B D(U)|\hat{0}\rangle_A|0\rangle_B \\
&= a_0(I - |\hat{0}\rangle\langle\hat{0}|)_A D(U)|\psi_{x,0}\rangle_A|0\rangle_B + a_+(I - |\hat{0}\rangle\langle\hat{0}|)_A \otimes |0\rangle\langle 0|_B D(U)|\psi_{x,+}\rangle_A|0\rangle_B \\
&= a_0(I - |\hat{0}\rangle\langle\hat{0}|)_A|\psi_{x,0}\rangle_A|0\rangle_B + a_+(I - |\hat{0}\rangle\langle\hat{0}|)_A \otimes |0\rangle\langle 0|_B D(U)|\psi_{x,+}\rangle_A|0\rangle_B \\
&= \frac{a_0}{\alpha}|w\rangle_A|0\rangle_B + \underbrace{a_+(I - |\hat{0}\rangle\langle\hat{0}|)_A \otimes |0\rangle\langle 0|_B D(U)|\psi_{x,+}\rangle_A|0\rangle_B}_{=:|\xi\rangle}, \qquad (25)
\end{aligned}
$$

where in the final equality, we used Lemma 11 Item 1, since $P_0(U)|\psi_{x,0}\rangle = |\psi_{x,0}\rangle$.

We would like to bound Δ, where

$$
\begin{aligned}
\Delta := \left\| \frac{|\hat{w}\rangle}{\||\hat{w}\rangle\|} - \frac{|w\rangle_A|0\rangle_B}{\sqrt{w_+(x)}} \right\| &= \left\| \frac{\frac{a_0}{\alpha}|w\rangle_A|0\rangle_B + |\xi\rangle}{\||\hat{w}\rangle\|} - \frac{|w\rangle_A|0\rangle_B}{\sqrt{w_+(x)}} \right\| \\
&\leq \left| \frac{a_0}{\alpha\,\||\hat{w}\rangle\|} - \frac{1}{\sqrt{w_+(x)}} \right| \||w\rangle\| + \frac{\||\xi\rangle\|}{\||\hat{w}\rangle\|} \quad \text{by triangle ineq.} \\
&\leq \left| \frac{a_0\sqrt{w_+(x)}}{\alpha\,\||\hat{w}\rangle\|} - 1 \right| + \frac{\||\xi\rangle\|}{\||\hat{w}\rangle\|}. \qquad (26)
\end{aligned}
$$

To bound $\||\xi\rangle\|$, we have

$$
\begin{aligned}
\||\xi\rangle\|^2 = a_+^2 \left\|(I - |\hat{0}\rangle\langle\hat{0}|)_A \otimes |0\rangle\langle 0|_B D(U)|\psi_{x,+}\rangle_A|0\rangle_B\right\|^2 &\leq \|I_A \otimes |0\rangle\langle 0|_B D(U)|\psi_{x,+}\rangle_A|0\rangle_B\|^2 \\
&\leq \|P_\Theta|\psi_{x,+}\rangle\|^2 + \epsilon' \\
&\leq \Theta^2\alpha^2\widetilde{W}_- + \epsilon' \leq 2\epsilon', \qquad (27)
\end{aligned}
$$

where the first inequality is because a projection can only decrease the norm of a vector, and $a_+ \leq 1$; the second inequality is from by Lemma 11 Item 2, and the third inequality comes from Lemma 15 and our choice of Θ.

Next, to bound $\||\hat{w}\rangle\|$, we use the triangle inequality on the final line of Equation (25), and Equation (27) to get

$$
\frac{a_0\sqrt{w_+(x)}}{\alpha} - \sqrt{2\epsilon'} \leq \||\hat{w}\rangle\| \leq \frac{a_0\sqrt{w_+(x)}}{\alpha} + \sqrt{2\epsilon'}. \qquad (28)
$$

By our choice of ϵ', we have $2\epsilon' \leq 1/48$, and also applying Corollary 18 to Equation (28), we have

$$
\frac{1}{4} < \sqrt{3/16} - \sqrt{1/48} \leq \||\hat{w}\rangle\| \leq \sqrt{1/4} + \sqrt{1/48} < \frac{3}{4}. \qquad (29)
$$

Rearranging Equation (28) and applying Equation (29), we have

$$
\left| \frac{a_0\sqrt{w_+(x)}}{\alpha\,\||\hat{w}\rangle\|} - 1 \right| \leq \frac{\sqrt{2\epsilon'}}{\||\hat{w}\rangle\|}. \qquad (30)
$$

Then plugging Equations (27), (29), and (30) into Equation (26) we have:

$$
\Delta \leq \frac{2\sqrt{2\epsilon'}}{\||\hat{w}\rangle\|} < 8\sqrt{2\epsilon'} = O(\epsilon). \qquad (31)
$$

Now we analyze the probability that both the Probing Stage and State Generation Stage are successful, resulting in the state $|\hat{w}\rangle/\||\hat{w}\rangle\|$ as in Equation (26). First note that there is a value of α (if we iterate in the Probing Stage long enough), that will cause us to break out of the Probing Stage if Iterative Amplitude Estimation does not fail. In particular, when $w_+(x)/\alpha^2 \in [1/2, 2]$, then from Equation (13) $a_0 \in [1/3, 2/3]$. Thus by Lemma 17 and since $2\epsilon' \leq 1/48$, the probability of of outcome $|0\rangle_B$ is in $[16/48, 33/48]$, which in Line 5 causes us to leave the Probing Stage if Iterative Amplitude Estimation does not fail. This occurs for some value of α, as we are doubling α at each iteration of the Probing Stage, causing $w_+(x)/\alpha^2$ to decrease, and initially we have $w_+(x)/\alpha^2 = w_+(x)\widetilde{W}_- \geq 1$.[5]

Thus if no error occurs, the condition of Line 5 will be satisfied after some number L of rounds such that $L \in O(\log(w_+(x)\widetilde{W}_-)) = O(\log(W_+\widetilde{W}_-))$. As the probability of failing a single Iterative Amplitude Estimation round is $p \leq \delta/\log(W_+\widetilde{W}_-)$ (see Line 1), the probability of leaving the Probing Stage when Line 5 is satisfied (rather than before or after) is at least

$$(1-p)^L = 1 - O(\delta). \tag{32}$$

Assuming that we have successfully left the Probing Stage without failure, we next calculate the probability of getting a measurement outcome M during the at most $\log(1/\delta)$ iterations of the State Generation Stage. The probability of getting outcome M is lower bounded by (from Equation (29))

$$\||\hat{w}\rangle\|^2 \geq 1/16. \tag{33}$$

Thus the probability of success in the State Generation Stage is

$$1 - (15/16)^{\log(1/\delta)} = 1 - O(\delta). \tag{34}$$

Combining Equations (32) and (34), our probability of successfully producing a state $|\hat{w}\rangle/\||\hat{w}\rangle\|$ as in Equation (26) is

$$(1 - O(\delta))(1 - O(\delta)) = 1 - O(\delta). \tag{35}$$

To calculate the expected query complexity, we first note that if we terminate in round $t \in \{0, \ldots, \lceil \log \sqrt{W_+} \rceil\}$ of the Probing Stage, we use

$$\begin{aligned} &\sum_{i=0}^{t} O\left(\frac{2^i}{\sqrt{\epsilon}} \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{1}{p}\right)\right) + O\left(\log\left(\frac{1}{\delta}\right) \frac{2^t}{\sqrt{\epsilon}} \log\left(\frac{1}{\epsilon}\right)\right) \\ =&O\left(\frac{2^t}{\sqrt{\epsilon}} \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{1}{p\delta}\right)\right) \end{aligned} \tag{36}$$

queries, which comes from the cost of Iterative Amplitude Estimation (Lemma 12) applied to phase estimation (Lemma 11) in each round of the Probing Stage up to the t^{th} round, plus the cost of phase estimation in the State Conversion Stage.

The probability that we terminate in any round t when we have an estimate $\hat{a}$ that is not in the range $[\frac{15}{48}, \frac{35}{48}]$ is at most p. Using Equation (36) the the total contribution to the average query complexity from all such rounds is at most

$$\sum_{t=0}^{\lceil \log \sqrt{W_+\widetilde{W}_-} \rceil} O\left(p\frac{2^t}{\sqrt{\epsilon}} \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{1}{p\delta}\right)\right) = O\left(p\sqrt{\frac{W_+\widetilde{W}_-}{\epsilon}} \log\left(\frac{1}{\epsilon}\right) \log\left(\frac{1}{p\delta}\right)\right). \tag{37}$$

[5] To see that $w_+(x)\widetilde{W}_- \geq 1$, let $N_+ = \min\{\||w\rangle\|^2 : A|w\rangle = |\tau\rangle\}$, and $N_- = \min\{\|\langle\omega|A\|^2 : \langle\omega|\tau\rangle = 1\}$. Then $w_+(x) \geq N_+$, and $\widetilde{W}_- \geq N_-$, and by [21, Section 2.4], $N_+N_- = 1$.

where in the sum we have actually included all rounds, not just those that satisfy when $\hat{a}$ is not in the range $[\frac{15}{48}, \frac{35}{48}]$, which is acceptable since we are deriving an upper bound on the expected query complexity.

If we terminate at a round t^* when $\hat{a}$ is in the range $[\frac{15}{48}, \frac{35}{48}]$, which happens when Iterative Amplitude Estimation does not fail at Line 4 and then causes a break at Line 5, from Equation (23) we have $\frac{w_+(x)}{\alpha^2} \in \left[\frac{1}{3}, 4\right]$, and $2^{t^*} = \alpha\sqrt{\widetilde{W}_-}$ so $\sqrt{w_+(x)\widetilde{W}_-}/2 \leq 2^{t^*} \leq \sqrt{3w_+(x)\widetilde{W}_-}$. Because we double α at each iteration, there are only a constant number of rounds where we will find $\hat{a}$ in the appropriate range, and we trivially upper bound the probability of terminating at any such round by 1. Using Equation (36), these rounds add

$$O\left(\sqrt{\frac{w_+(x)\widetilde{W}_-}{\epsilon}}\log\left(\frac{1}{\epsilon}\right)\log\left(\frac{1}{p\delta}\right)\right) \tag{38}$$

to the total expected query complexity.

Combining Equations (37) and (38), and using that we set p to be $O\left(1/\sqrt{W_+\widetilde{W}_-}\right)$ (Line 1), we find the expected query complexity is

$$O\left(\sqrt{\frac{w_+(x)\widetilde{W}_-}{\epsilon}}\log\left(\frac{1}{\epsilon}\right)\log\left(\frac{1}{p\delta}\right)\right) = \widetilde{O}\left(\sqrt{\frac{w_+(x)\widetilde{W}_-}{\epsilon}}\log\left(\frac{1}{\delta}\right)\right). \tag{39}$$

4 Graph Applications

4.1 Finding an Edge on a Path

In this section, we consider the problem of finding an edge on an st-path in $G(x)$, which we denote st-EDGE$_G(x)$. That is, given query access to a string x that determines a subgraph $G(x) = (V, E(x))$ of an n-vertex graph G, as described in Section 2.3 (if G is a complete graph, x is just the adjacency matrix of $G(x)$), with $s, t \in V$ such that there is at least one path from s to t in $G(x)$, output an edge $e \in E(x)$ that is on a (self-avoiding) path from s to t.

Classically, it is hard to imagine that this problem is much easier than finding a path, and indeed, in our classical lower bound in Theorem 20 the set-up forces the algorithm to learn a complete path before it can find any edge on the path. However, we find that quantumly, when there are short or multiple paths, this problem is easier than any path finding algorithms known. This opens up the possibility of improved quantum algorithms for cases where it is not necessary to know the complete path, like the st-cut set algorithm of Section 4.2.

▶ **Theorem 19.** *Fix $p > 0$, and a family of n-vertex simple graphs G with vertices s and t. There is a quantum algorithm (Algorithm 2) that solves st-EDGE$_G(x)$ with probability $1-O(p)$ and uses $\widetilde{O}\left(\frac{n\sqrt{R_{s,t}(G(x))}}{p}\right)$ expected queries on input x. More precisely, with probability $1-O(p)$, the algorithm samples from a distribution $\hat{q}$ such the total variation distance between $\hat{q}$ and $q_{G(x),s,t}$ is $O(\sqrt{p})$, where $q_{G(x),s,t}(u,v)$ (defined in Equation (7)) is proportional to $\theta^*(u,v)^2$, where θ^* is the optimal unit st-flow on $G(x)$.*

To obtain this result, we run our witness state generation algorithm (Algorithm 1) using the span program for st-connectivity, $\mathcal{P}_{G_{st}}$ and an oracle O_x that defines a graph $G(x)$ with a path between s and t. When successful, the output will be a quantum state that is approximately proportional to the optimal flow state, Equation (6), which itself is a superposition of edges on paths by Lemma 7. Then from Equation (7), when we then measure in the standard basis, the probability of obtaining an edge e should be close to $q_{G(x),s,t}(e)$, and with high probability, we will measure some edge on a path.

Proof of Theorem 19: We analyze Algorithm 2.

Algorithm 2 `EdgeFinder`(O_x, p, G, s, t).

Input : Failure tolerance $p > 0$, oracle O_x for the graph $G(x) = (V, E(x))$, $s, t \in V$ such that there is a path from s to t.

Output : An output e, or "Failure", such that with probability $1 - O(p)$, e is an edge on a path from s to t.

$\epsilon \leftarrow p^2$; $\delta \leftarrow p$

$|\hat{\theta}\rangle \leftarrow$ `WitnessGeneration`$(\mathcal{P}_{G_{st}}, O_x, \epsilon, \delta)$ (Algorithm 1)

if $|\hat{\theta}\rangle \neq$ *"Failure"* **then**

  $e \leftarrow$result of Measuring $|\hat{\theta}\rangle$ in the standard basis

Return "Failure"

If `WitnessGeneration`$(\mathcal{P}_{G_{st}}, O_x, \epsilon, \delta)$ (see Algorithm 1) does not fail, which happens with probability $1 - O(\delta) = 1 - O(p)$, then by Theorem 10,

$$|\hat{\theta}\rangle = |\theta^*\rangle / \||\theta^*\rangle\| + |\eta\rangle \tag{40}$$

for some $|\eta\rangle$ such that $\||\eta\rangle\|^2 = O(\epsilon)$ and from Equation (6), $|\theta^*\rangle = \frac{1}{2}\sum_{e\in\overrightarrow{E}} \theta^*(e)|e\rangle$ where θ^* is the optimal unit st-flow in $G(x)$, so $\||\theta^*\rangle\| = \sqrt{\mathcal{R}_{s,t}(G(x))}$.

Let $P_{E(x),s,t}$ be the projection onto the set of edges in $\overrightarrow{E}(x)$ that are on (self-avoiding) paths from s to t. The probability that we measure such an edge when we measure $|\hat{\theta}\rangle$ in the standard basis is the square of

$$\left\|P_{E(x),s,t}|\hat{\theta}\rangle\right\| \geq \left\|P_{E(x),s,t}|\theta^*\rangle / \||\theta^*\rangle\|\right\| - \left\|P_{E(x),s,t}|\eta\rangle\right\| = 1 - O(\sqrt{\epsilon}), \tag{41}$$

where we have used the triangle inequality, and the fact that $P_{E(x),s,t}|\theta^*\rangle = |\theta^*\rangle$, by Lemma 7. Continuing, we have probability

$$\left\|P_{E(x),s,t}|\hat{\theta}\rangle\right\|^2 \geq \left(1 - O(\sqrt{\epsilon})\right)^2 = 1 - O(\sqrt{\epsilon}). \tag{42}$$

Thus our total probability of success of measuring an edge on a path is $(1-O(\delta))(1-O(\sqrt{\epsilon}))$. Since we are setting ϵ to p^2 and δ to p, our total probability of success is $1 - O(p)$.

Let $\hat{q}$ be the output distribution of Algorithm 2. By the relationship between total variation distance and trace norm, we have that $d(\hat{q}, q_{G(x),s,t})$, the total variation distance between $\hat{q}$ and $q_{G(x),s,t}$, is at most the trace norm of $|\hat{\theta}\rangle$ and $|\theta^*\rangle/\||\theta^*\rangle\|$ (see e.g. [33]) so

$$\begin{aligned} d(\hat{q}, q_{G(x),s,t}) &\leq \sqrt{1 - \left|\langle\hat{\theta}|\theta^*\rangle / \||\theta^*\rangle\|\right|^2} \\ &= \sqrt{1 - \left|\langle\hat{\theta}|\hat{\theta}\rangle - \langle\hat{\theta}|\eta\rangle\right|^2} \\ &\leq \sqrt{1 - (1 - \||\eta\rangle\|)^2} \\ &\leq \sqrt{2\,\||\eta\rangle\|} = O(\epsilon^{1/4}) = O(\sqrt{p}). \end{aligned} \tag{43}$$

By Theorem 10, the expected query complexity of `WitnessGeneration`, and thus Algorithm 2 is

$$\widetilde{O}\left(\sqrt{\frac{w_+(x)\widetilde{W}_-}{\epsilon}}\log\left(\frac{1}{\delta}\right)\right) = \widetilde{O}\left(\frac{\sqrt{R_{s,t}(G(x))n}}{p}\right) \tag{44}$$

where we have used the fact that, for $\mathcal{P}_{G_{st}}$, $w_+(x) = R_{s,t}(G(x))$ and $\widetilde{W}_- = O(n^2)$ [7, 21]; and set ϵ to p^2 and δ to p, as in Algorithm 2. ◀

We can use Theorem 19 to prove the following separation between the quantum and classical query complexity of finding an edge on a path:

▶ **Theorem 20.** *Let $G = (V, E)$ with $s, t \in V$ be an n-vertex complete graph, and suppose we are promised that $G(x)$ has a path of length L for $L \in [3, n/4]$ between s and t (L may depend on x and need not be known ahead of time). Then st-EDGE$_G(x)$ can be solved in $\widetilde{O}(n\sqrt{L})$ expected quantum queries on input x, while any classical algorithm has query complexity $\Omega(n^2)$.*

Proof. For the quantum algorithm, we apply Theorem 19 with bounded probability of error $p = \Omega(1)$, and use the fact that $R_{s,t}(G) = O(L)$.

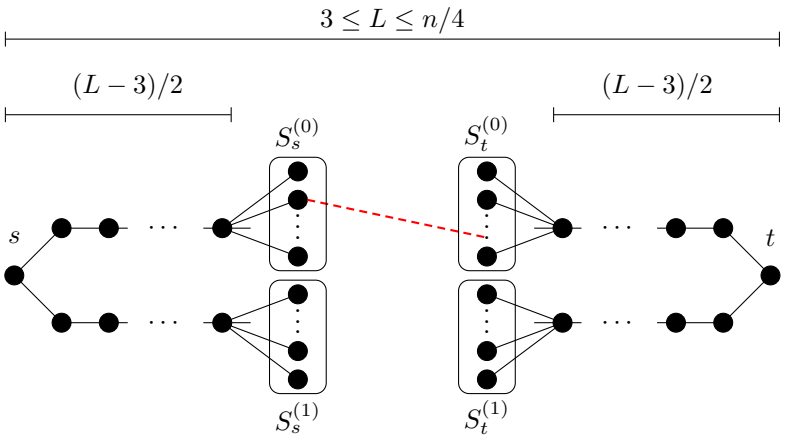


Figure 1 The solid black lines show the edges that are present in $G(x)$ for any x. In addition, $G(x)$ contains a single edge between a vertex in $S_s^{(b)}$ and $S_t^{(b)}$, where $b = \sigma_1^*$, as in the dashed red edge, resulting in a single path of length L.

For the classical lower bound, we reduce the following problem to path edge finding: Given a string x of $N = 2^\ell$ bits, $(x_\sigma)_{\sigma\in\{0,1\}^\ell}$ such that there is a unique σ^* with $x_{\sigma^*} = 1$, output σ_1^*. That is, we would like to output the first bit of the index of the unique 1-valued bit of x. By an adversary argument similar to a standard OR lower bound, the bounded error randomized query complexity of this problem is $\Omega(N)$. We will show how to solve this problem with an algorithm for finding a path edge on a graph like the one depicted in Figure 1.

For $x \in \{0,1\}^N$, let $G(x)$ be a graph on $n = \Theta(2^{\ell/2})$ vertices in which there is a unique st-path of length L, for some odd L, as shown in Figure 1. The vertex s is connected by a path of length $(L-3)/2$ to a vertex that is additionally connected to a set of $2^{(\ell-1)/2}$ vertices, $S_s^{(0)} = \{u_{0,\sigma} : \sigma \in \{0,1\}^{(\ell-1)/2}\}$. In a symmetric manner, s is also connected by another disjoint path of length $(L-3)/2$ to a vertex that is additionally connected to a

set of $2^{(\ell-1)/2}$ vertices, $S_s^{(1)} = \{u_{1,\sigma} : \sigma \in \{0,1\}^{(\ell-1)/2}\}$. In the same way, t is connected by a pair of disjoint paths of length $(L-3)/2$ to a pair of vertices, additionally connected to $S_t^{(0)} = \{v_{0,\sigma} : \sigma \in \{0,1\}^{(\ell-1)/2}\}$ and $S_t^{(1)} = \{v_{1,\sigma} : \sigma \in \{0,1\}^{(\ell-1)/2}\}$ respectively. All edges described so far (the black edges in Figure 1) are always present in $G(x)$ (we simulate querying the associated input bits by just outputting 1). We now describe edges whose presence in $G(x)$ is determined by x. For $b \in \{0,1\}$, there is a potential edge between every pair of vertices $u_{b,\sigma} \in S_s^{(b)}$ and $v_{b,\sigma'} \in S_t^{(b)}$, with the label $x_{b\sigma\sigma'}$, meaning exactly one of these is present in $G(x)$ – the one with $\sigma^* = b\sigma\sigma'$. All remaining possible edges are never present in $G(x)$ (we simulate querying their associated input bits by just outputting 0).

We can find the first bit of σ^* by running the edge finding algorithm on $G(x)$. Assuming the output is correct, there are the following possibilities:

1. If the algorithm outputs an edge from the middle part of the graph, then it must be the one labelled by x_{σ^*}, so σ^* is learned entirely.
2. If the algorithm outputs an edge from the left-hand side of the graph, it is on a path between s and $S_s^{(b)}$ for some $b \in \{0,1\}$, and we know that $\sigma_1^* = b$.
3. If the algorithm outputs an edge from the right-hand side of the graph, it is on a path between t and $S_t^{(b)}$ for some $b \in \{0,1\}$, and we know that $\sigma_1^* = b$.

In all cases, we have learned σ_1^*. This gives a lower bound on path-edge finding of $\Omega(N) = \Omega(2^\ell) = \Omega(n^2)$. ◀

4.2 Finding an *st*-cut set

Given a graph $G(x)$ containing a path from s to t, an *st*-cut set is a set of edges in $G(x)$ such that when those edges are removed from $G(x)$, there is no longer a path from s to t. The *st*-cut set problem is that of finding an *st*-cut set. This problem has applications to detecting weak points in networks in order to figure out how to strengthen a network, or conversely, for sabotaging networks.

We first note that for graphs with a single *st*-path, Theorem 19 can immediately be used to find an *st*-cut set, since any edge on the path is an *st*-cut set. However, we can also analyze more complex situations, as the following, in which we have an upper bound on the effective resistance of the graph, and a lower bound on the optimal unit *st*-flow going through any edge in the *st*-cut set:

▶ **Theorem 21.** *For functions $R, g : \mathbb{N} \to \mathbb{R}_{>0}$, let $G = (V, E)$ with $s, t \in V$ be a family of n-vertex simple graphs, and suppose we are additionally promised that $R_{s,t}(G(x)) \leq R(n)$, and there exists an st-cut set $C \subseteq E(x)$ such that for each $\{u, v\} \in C$, $\theta^*(u, v)^2 \geq g(n)$ where θ^* is the optimal unit st-flow in $G(x)$. Then there is a quantum algorithm that outputs a set C' such that $C \subseteq C'$ with bounded error, and has worst-case query complexity $\widetilde{O}\left(\frac{R(n)^2 n}{g(n)^{3/2}}\right)$.*

We can assume without loss of generality that the C in Theorem 21 is a minimal *st*-cut. While we are not guaranteed that the set C' output by the algorithm referred to in Theorem 21 is minimal, it is still an *st*-cut as long as it contains C, since its removal will disconnect s and t.

To prove Theorem 21, we will use the following variation of the well-known "coupon collector" problem.

▶ **Lemma 22.** *Consider repeatedly sampling a random variable Z on a finite set $\mathcal{S}$. Let $C \subseteq \mathcal{S}$ be such that for each $e \in C$, $\Pr[Z = e] \geq B$. Let T be the number of samples to Z before we have sampled each element of C at least once. Then $\mathbb{E}[T] = O\left(\frac{\log |C|}{B}\right)$.*

Proof. For $i \in \{1, \ldots, |C|\}$, the probability that Z is a new element of C, after $i-1$ elements have already been collected, is $p_i \geq (|C| - (i-1))B$. Let T_i be the number of samples to Z after sampling $(i-1)$ elements of C, until we sample i elements of C, so T_i is a geometric random variable with

$$\mathbb{E}[T_i] = 1/p_i \leq ((|C| - (i-1))B)^{-1}. \tag{45}$$

From this we can compute

$$\mathbb{E}[T] = \sum_{i=1}^{|C|} \mathbb{E}[T_i] \leq \sum_{i=1}^{|C|} \frac{1}{(|C| - (i-1))B} = \frac{1}{B} \sum_{j=1}^{|C|} \frac{1}{j} = \Theta\left(\frac{\log |C|}{B}\right). \tag{46}$$

◀

Proof of Theorem 21. We use parameters T' and ϵ, to be defined shortly, and $\delta = 1/4$. Our strategy is to repeatedly run `WitnessGeneration`$(\mathcal{P}_{G_{st}}, O_x, \epsilon, \delta)$ (Algorithm 1) to produce an approximate witness state, and then measure the resultant state in the standard basis to get an edge e, which we add to C'. We repeat this T' times, before outputting C'.

Let Z be the random variable on $E \cup \{\text{Failure}\}$ representing the measured output of one call to Algorithm 1. We set $\epsilon = \Theta\left(\frac{g(n)}{R(n)}\right)$ small enough so that if the algorithm does not fail, we produce a state $|\theta^*\rangle/\||\theta^*\rangle\| + |\eta\rangle$ where $\||\eta\rangle\|^2 \leq g(n)/R(n)$ (see Equation (40) and following discussion). Then the probability that we sample an edge $e' \in C$ when we measure in the standard basis is

$$\begin{aligned} \left\|\langle e'| \left(|\theta^*\rangle/\||\theta^*\rangle\| + |\eta\rangle\right)\right\|^2 &= \|2\theta^*(e')/\sqrt{R_{s,t}(G(x))} - \langle e'|\eta\rangle\|^2 \\ &\geq \|2\sqrt{g(n)/R(n)} - \sqrt{g(n)/R(n)}\|^2 \\ &= \Omega(g(n)/R(n)). \end{aligned} \tag{47}$$

Since the probability of one call to Algorithm 1 not failing is $1 - \delta = \Omega(1)$, for every $e' \in C$, we have $\Pr[Z = e'] \geq B$ for some $B = \Omega(g(n)/R(n))$. Thus, by Lemma 22, the expected number of calls to Algorithm 1 before $C \subseteq C'$ is at most:

$$\mathbb{E}[T] = O\left(\frac{R(n)}{g(n)} \log |C|\right) = O\left(\frac{R(n)}{g(n)} \log n\right). \tag{48}$$

By Markov's inequality, if we set $T' = 100\mathbb{E}[T]$, the algorithm will succeed with bounded error.

By Theorem 10, each call to Algorithm 1 has expected query complexity

$$\widetilde{O}\left(\sqrt{\frac{R_{s,t}(G(x))n^2}{\epsilon}}\right) = \widetilde{O}\left(n\sqrt{\frac{R(n)}{g(n)/R(n)}}\right) = \widetilde{O}\left(\frac{nR(n)}{\sqrt{g(n)}}\right), \tag{49}$$

so the total expected query complexity is

$$\widetilde{O}\left(T'\frac{nR(n)}{\sqrt{g(n)}}\right) = \widetilde{O}\left(\frac{R(n)}{g(n)}\frac{nR(n)}{\sqrt{g(n)}}\right) = \widetilde{O}\left(\frac{nR(n)^2}{g(n)^{3/2}}\right). \tag{50}$$

We can get a worst case algorithm by stopping after 100 times the expected number of steps, if the algorithm is still running, and outputting the current C'. We have no guarantee on the correctness of C' in that case, but by Markov's inequality, this only happens with probability $1/100$. ◀

We can use Theorem 21 to prove the following result for finding an st-cut set in a particular family of graphs with expander subgraphs and a single st-cut edge.

▶ **Corollary 23.** *Let $G = (V, E)$ with $s, t \in V$ be a family of n-vertex graphs, and suppose we are additionally promised that $G(x)$ consists of two disjoint, d-regular (for $d \geq 3$), constant expansion subgraphs, each on $n/2$ vertices, where s and t are always put in separate subgraphs, plus a single additional edge connecting the two subgraphs. Then there is a quantum algorithm that finds the st-cut edge with bounded error in worst-case $\widetilde{O}(n)$ queries, while any classical algorithm has query complexity $\Omega(n^2)$.*

Proof. For a classical algorithm, even if the algorithm had complete knowledge of the two subgraphs, there would be $\Omega(n^2)$ possible locations for the connecting edge, reducing the problem to search, requiring $\Omega(n^2)$ queries.

For the quantum algorithm, note that the maximum effective resistance between any two points in a d-regular (for $d \geq 3$), constant expansion graph on n-vertices is $O(1/d)$ [10]. Thus $R_{s,t}(G(x)) = \Omega(1)$. Additionally, since there is only one edge e' connecting the two subgraphs, the optimal unit st-flow on e', $\theta^*(e')$, must be equal to 1.

Applying Theorem 21 with $R(n) = O(1)$ and $g(n) = \Omega(1)$, we get a worst-case bounded error quantum query complexity $\widetilde{O}(n)$. ◀

4.3 Path Finding

In this section, we consider the problem of finding an st-path in $G(x)$, which we denote st-PATH$_G(x)$. That is, given query access to a string x that determines a subgraph $G(x) = (V, E(x))$ of an n-vertex graph G, as described in Section 2.3 (if G is a complete graph, x is just the adjacency matrix of $G(x)$), with $s, t \in V$ such that there is at least one path from s to t in $G(x)$, output a path from s to t. A path is a sequence of *distinct* vertices $\vec{u} = (u_0, \ldots, u_\ell)$ such that $s = u_0$, $t = u_\ell$, and for all $i \in [\ell]$, $(u_{i-1}, u_i) \in \overrightarrow{E}(G(x))$.

To solve st-PATH$_G$, one might expect that we could simply apply Algorithm 2 multiple times, storing each edge's endpoints and identifying vertices of the endpoints of found edges to reduce the size of the graph, until a path is found. However, such an algorithm could run into challenges that could produce slow running times. For example, in a graph where there are many st-paths, the algorithm could spend too much time sampling edges from different paths, rather than focusing on completing a single path. In the case of a single st-path, such a strategy would not take advantage of the fact that once one edge on the path is found, the problem reduces to two connectivity subproblems (from s to the found edge, and from t to the found edge) that each typically have significantly smaller query complexities than the original problem.

Thus we develop two algorithms that allow us to prove tighter expected query complexity bounds than Ref. [18] for the case of short longest st-paths, one in the case of a single st-path, and one for generic graphs.

Before getting into quantum algorithms for path detection, we note the following corollary of Theorem 20, via a reduction to path finding from path-edge finding, that characterizes the classical query complexity of path finding in the case of short longest st-paths:

▶ **Corollary 24.** *Let $G = (V, E)$ with $s, t \in V$ be an n-vertex complete graph and suppose we are promised that $G(x)$ has a path of length L for $L \in [3, n/4]$ between s and t. Then st-PATH$_G(x)$ has randomized query complexity $\Omega(n^2)$.*

4.3.1 Graph with a Single Path

When the graph $G(x)$ is known to have a single st-path, we will we use a divide-and-conquer algorithm to find the path. To show that the divide-and-conquer approach is useful, we first consider the simpler algorithm (as described above) that uses Theorem 19 to find an edge $\{u, v\}$ on the path, and then once that edge is found, the algorithm is run on a new graph where vertices u and v are identified. This process is continued until the edge $\{s, t\}$ is found. Thus if the length of the path is initially L, after an edge is found, the path length will be $L-1$, and then $L-2$ in the next iteration, etc. Ignoring error, and assuming the algorithm finds an edge in each round, by Theorem 19, the query complexity at the ith round will be $\widetilde{O}(n\sqrt{L-i})$. Over the course of the L rounds, the total query complexity will be

$$\sum_{i=0}^{L-1} \widetilde{O}(n\sqrt{L-i}) = \widetilde{O}\left(nL^{3/2}\right). \tag{51}$$

For $L \geq n^{2/3}$, this algorithm does not even outperform the best classical algorithm, and for $L \geq n^{1/3}$ it does not outperform the quantum algorithm of Ref. [18].

We instead consider the following divide-and-conquer approach, described in detail in Algorithm 3. We use Algorithm 2 to find a set of edges, some of which are very likely to be on the path. Then we use Lemma 8 to verify which of those edges is actually on the path, and Lemma 9 to ensure we choose an edge near the center of the path, so we are left with two subproblems of approximately half the size. Finally two recursive calls find the unique path from s to the found edge, and the unique path from t to the found edge.

▶ **Theorem 25.** *Let $p \geq 0$, and $G = (V, E)$ with $s, t \in V$ be a family of n-vertex graphs, and suppose we are promised that $G(x)$ contains a single st-path of some length L (L may depend on x and need not be known ahead of time). Then there is a quantum algorithm (Algorithm 3) that with probability $1 - O(p)$ solves st-PATH$_G(x)$ and uses $\widetilde{O}(nL^{1+o(1)} \log^2(1/p))$ expected queries on input x.*

Proof. We first analyze the probability of error, then we prove the correctness of Algorithm 3, assuming that no errors are made, and finally, we analyze the query complexity.

We will stop the algorithm after $O(n)$ recursive calls. Since each recursive call returns an edge, and any path has length at most n, this termination will not affect the success probability. We then bound our probability of error by $O(p/n^4) = O(p)$, by showing that the failure probability in each recursive call is $O(p/n^5)$.

We say a failure occurs (in some recursive call) if any of the following happens:

1. Any one of the at most 4ℓ `PathDetection` algorithms errs. This has probability $O(\ell\delta) = O(p/n^5)$, by our choice of $\delta = p/(\ell n^5)$.
2. One of the at most $O(\ell)$ calls to `WitnessSizeEst` produces an estimate that is not within the desired relative error. This has probability $O(\ell\delta) = O(p/n^5)$.
3. None of the ℓ iterations of `EdgeFinder` produces an edge that is on the st-path, and moreover, that is within $(\varepsilon_3 - \varepsilon_2)L = \sqrt{\varepsilon_1}L$ of the middle of the path. The absence of this type of failure is sufficient to guarantee that the condition on Line 20 will be satisfied, as long as `WitnessSizeEst` is also successful.

We analyze the probability of the last event, assuming the first two do not occur. Let $e_0, \ldots, e_{L-1}$ denote the path edges, in order, in the unique st-path in $G(x)$. For one of the ℓ runs of `EdgeFinder`, the probability that it does not output "Failure" is ε_1. Conditioned on

Algorithm 3 `SinglePathFinder`(O_x, p, G, s, t).

Input : Failure tolerance $p > 0$, oracle O_x for the graph $G(x) = (V, E(x))$, $s, t \in V$ such that there is a unique path from s to t.

Output : With probability $1 - O(p)$, a set of edges whose vertices form a path from s to t in $G(x)$.

`// Base Cases`

if $s = t$ **then** Return $\emptyset$

if $\{s, t\} \in E(x)$ **then** Return $\{s, t\}$

`// Finding Possible Edges on Path`

$\varepsilon_1 \leftarrow \frac{1}{\log n}$ `// Any` $\varepsilon_1 = o(1)$ `that is inverse polylog`(n) `would suffice`

$S \leftarrow \emptyset$

$\ell \leftarrow \frac{2\log(n^5/p)}{\varepsilon_1}$ **for** $i = 1$ **to** ℓ **do**

  $e \leftarrow$ `EdgeFinder`$(O_x, \varepsilon_1, G, s, t)$ (Algorithm 2)

  if $e \neq$ *"Failure" and* $e = (u, v) \in \overrightarrow{E}(x)$ **then** $S = S \cup \{(u, v), (v, u)\}$

`// Finding a possible edge that is actually on a path`

$\delta \leftarrow p/(\ell n^5)$

for $(u, v) \in S$ **do**

  Initialize `PathDetection`$(O_x, G^-_{\{u,v\}}, s, u, \delta)$ (Lemma 8)

  Initialize `PathDetection`$(O_x, G^-_{\{u,v\}}, v, t, \delta)$

$flag \leftarrow$ True

while *flag* **do**

  Run in parallel each `PathDetection` algorithm initialized in the prior **for** loop, until each algorithm applies O_x once or terminates (or do nothing for those algorithms that have terminated previously)

  for $(u, v) \in S$ **do**

    if `PathDetection`$(O_x, G^-_{\{u,v\}}, s, u, \delta)$ *and* `PathDetection`$(O_x, G^-_{\{u,v\}}, v, t, \delta)$ *have both terminated in this iteration of the* ***while*** *loop and both detected paths* **then**

      $\varepsilon_2 \leftarrow \sqrt{\varepsilon_1}$, $\varepsilon_3 \leftarrow 2\sqrt{\varepsilon_1}$

      $\tilde{k} \leftarrow$`WitnessSizeEst`$(O_x, G, s, u, \varepsilon_2, \delta)$ (Lemma 9) `// estimate of dist.` s `to` u

      if $|\tilde{k} - L/2| \leq \varepsilon_3 L$ **then**

        $(u^*, v^*) \leftarrow (u, v)$

        $flag \leftarrow$ False

`// Recursive call`

Return $\{(u^*, v^*)\} \cup$ `SinglePathFinder`$(O_x, p, G, s, u^*) \cup$ `SinglePathFinder`(O_x, p, G, v^*, t)

the output of `EdgeFinder` not being "Failure," by Theorem 19, we sample from a distribution $\hat{q}$ that is $\sqrt{\varepsilon_1}$-close in total variation distance to the uniform distribution over edges on the st-path. Thus, the probability that we sample an edge in the set

$$R = \{e_k : k \in [L/2 - (\varepsilon_3 - \varepsilon_2)L, L/2 + (\varepsilon_3 - \varepsilon_2)L]\}, \tag{52}$$

where e_k is the k^{th} path edge, is:

$$\hat{q}(R) \geq \frac{|R|}{L} - \sqrt{\varepsilon_1} = 2(\varepsilon_3 - \varepsilon_2) - \sqrt{\varepsilon_1} = 2(2\sqrt{\varepsilon_1} - \sqrt{\varepsilon_1}) - \sqrt{\varepsilon_1} = \sqrt{\varepsilon_1}. \tag{53}$$

Thus, using $\varepsilon_1 \leq 1/2$, each of the ℓ samples has probability at least $(1 - \varepsilon_1)\sqrt{\varepsilon_1} \geq \sqrt{\varepsilon_1}/2$ of being a path edge in the correct range, R. Using Hoeffding's bound, the probability that none of them is a path edge in the correct range is thus at most:

$$e^{-2\ell(\sqrt{\varepsilon_1}/2)^2} = e^{-\ell\varepsilon_1/2} = e^{-\log(n^5/p)} = O(n^{-5}p) \tag{54}$$

by our choice of $\ell = 2\log(n^5/p)/\varepsilon_1$. The total probability of failure in one round is thus at most $O(p/n^5)$.

We prove correctness using induction on L, the length of the path, assuming no failure occurs. For the base case, if $L \in \{0, 1\}$, we will correctly return the path in Lines 1 and 2.

For the inductive case, let $L' \geq 1$. We assume `SinglePathFinder` works correctly for all lengths L such that $0 \leq L \leq L'$. Now consider a graph with $L = L' + 1$. Then assuming no failure, we will sample at least one edge (u, v) in the set $R = \{e_k : k \in [L/2 - (\varepsilon_3 - \varepsilon_2)L, L/2 + (\varepsilon_3 - \varepsilon_2)L]\}$ (not doing so is a failure of the type specified by Item 3 in the list above). Then if there are no errors in the `PathDetection` algorithms, Line 16 will be satisfied when (u, v) corresponds to an edge in the path where u is closer to s and v is closer to t. This is because we have removed $\{u, v\}$ from the graph when we are running `PathDetection`, and since there is a unique st-path, there will only be a path from s to u and not from s to v, and likewise for t.

Then for every edge (u, v) that we have correctly found using `PathDetection` to be on a path, we apply `WitnessSizeEst` (see Lemma 9) to estimate $R_{s,u}(G(x))$. If $(u, v) = e_k$, then $e_0, \ldots, e_{k-1}$ is the unique su-path in G, and it has length k, and so $R_{s,u}(G(x)) = k$, and thus `WitnessSizeEst` is actually estimating k. Assuming $(u, v) \in R$, (and we know this holds for at least one such edge), we have $|k - L/2| \leq (\varepsilon_3 - \varepsilon_2)L$. Then since we assume `WitnessSizeEst` does not fail, it outputs an estimate $\tilde{k}$ of k, such that $|\tilde{k} - k| \leq \varepsilon_2 k \leq \varepsilon_2 L$. Together, these conditioned imply $|\tilde{k} - L/2| \leq \varepsilon_3 L$, which will trigger the **while** loop to halt. It is possible that we will break out of the loop for an edge not in R, but at the least we know that if no failure occurs, we will will certainty break out of the **while** loop with an edge (u^*, v^*) on the path.

Now that we have the edge (u^*, v^*), to find the rest of the path, we just need to find the rest of the path from s to u^* and from v^* to t. But both of these problems will have path lengths between 0 and L', so by inductive assumption, the recursive calls in Line 23 will be correct, and will return the edges on the paths.

Turning to our analysis of the expected query complexity, we first bound the contribution to the expected query complexity in the case of a failure. As just discussed, a failure occurs with probability $O(p/n^4)$. Even in case of failure, each of our $O(n\log(n/p)) = O(n^2\log(1/p))$ calls to `EdgeFinder`, `PathDetection`, and `WitnessSizeEst` still has expected query complexity at most $\widetilde{O}(n^{1.5}(1/\varepsilon_1 + 1/\varepsilon_2^{3/2})\log(1/\delta)) = O(n^2\log(1/p))$ (for *any* x), for a total query cost of $O(n^4\log^2(1/p))$. Thus, the error case contributes an additive $O(p\log^2(1/p)) = O(1)$ to the expected query complexity.

Next, we create a recurrence relation for the expected query complexity, assuming no failure occurs. Let $\mathbb{E}[T_L]$ be the expected query complexity of Algorithm 3 on a graph with n vertices, when there is a single path, and that path has length L. For $k \in \{0, \ldots, L-1\}$, let $\tilde{q}_L(k)$ be the probability that the path edge that we find, (u^*, v^*), is e_k. Because we assume no subroutine call fails, we can assume that $\tilde{k}$ is an estimate of k with relative error ε_2, so $|\tilde{k}-k| \leq \varepsilon_2 k \leq \varepsilon_2 L$. From the conditional statement in Line 20, we also have $|\tilde{k}-L/2| \leq \varepsilon_3 L$. Taken together, these imply:

$$|k - L/2| \leq (\varepsilon_2 + \varepsilon_3)L = (\sqrt{\varepsilon_1} + 2\sqrt{\varepsilon_1})L = 3\sqrt{\varepsilon_1}L. \tag{55}$$

Thus with certainty (assuming no failure occurs), we will exit the **while** loop with $(u^*, v^*) = e_k$, for $k \in [(1/2 - 3\sqrt{\varepsilon_1})L, (1/2 + 3\sqrt{\varepsilon_1})L]$, so:

$$\mathbb{E}[T_L] = \widetilde{O}(\ell n\sqrt{L}/\varepsilon_1) + \widetilde{O}(\ell n\sqrt{L}\log(1/\delta)) + \widetilde{O}\left(\ell \frac{n\sqrt{L}}{\varepsilon_2^{3/2}}\log(1/\delta)\right) + \sum_{k=\lceil(1/2-3\sqrt{\varepsilon_1})L\rceil}^{\lfloor(1/2+3\sqrt{\varepsilon_1})L\rfloor} \tilde{q}_L(k)\left(\mathbb{E}[T_k] + \mathbb{E}[T_{L-k-1}]\right), \tag{56}$$

where the first three terms come from: (1) running `EdgeFinder` (Algorithm 2, Theorem 19) ℓ times; (2) at most $O(\ell)$ parallel `PathDetection` (Lemma 8) algorithms; and (3) running `WitnessSizeEst` (Lemma 9) $O(\ell)$ times; and the final term from the two recursive calls.

To get a function that is strictly increasing in L, let $T'_L := \max_{k \leq L} \mathbb{E}[T_k]$, so in particular $\mathbb{E}[T_L] \leq T'_L$, and T'_L also satisfies the recursion in Equation (56) (with $=$ replaced by $\leq$). Then we have, for any $k \in [(1/2 - 3\sqrt{\varepsilon_1})L, (1/2 + 3\sqrt{\varepsilon_1})L]$,

$$\mathbb{E}[T_k] + \mathbb{E}[T_{L-k-1}] \leq 2T'_{(1/2+3\sqrt{\varepsilon_1})L}. \tag{57}$$

Continuing from Equation (56), and using $1/\varepsilon_1 = \log n$ and $1/\varepsilon_2 = 1/\sqrt{\varepsilon_1} = \sqrt{\log n}$, $\ell = 2\log(n^5/p)/\varepsilon_1 = O(\log(1/p)\log^2 n)$, and $\log(1/\delta) = O(\log(\ell n/p)) = \log(1/p)\mathrm{polylog}(n, \log(1/p))$, we get

$$\mathbb{E}[T_L] \leq T'_L \leq \widetilde{O}\left(n\sqrt{L}\log^2(1/p)\right) + 2T_{(1/2+3\sqrt{\varepsilon_1})L}. \tag{58}$$

To analyze this recurrence, we add up the queries made in every recursive call. At the i^{th} level of recursion, there are 2^i recursive calls, and each one makes $\widetilde{O}\left(n\sqrt{L/b^i}\log^2(1/p)\right)$ queries itself, where $b = (1/2 + 3\sqrt{\varepsilon_1})^{-1}$, before recursing further. Thus

$$\begin{aligned}\mathbb{E}[T_L] &\leq \widetilde{O}\left(n\sqrt{L}\log^2(1/p)\right) + \sum_{i=1}^{\log_b L} 2^i\sqrt{\frac{L}{b^i}} \cdot \widetilde{O}\left(n\log^2(1/p)\right) \\ &\leq \widetilde{O}\left(n\sqrt{L}\log^2(1/p)\right) + \widetilde{O}\left(n\sqrt{L}\log^2(1/p)\right)\left(2/\sqrt{b}\right)^{\log_b L}.\end{aligned} \tag{59}$$

Letting $\eta := \frac{1}{1+\frac{1}{6\sqrt{\varepsilon_1}}} = O(1/\sqrt{\log n})$ since $\varepsilon_1 = 1/\log n$, so that $b = 2(1-\eta)$, we have:

$$\begin{aligned}\log\left(2/\sqrt{b}\right)^{\log_b L} &= \left(1 - \frac{1}{2}\log b\right)\frac{\log L}{\log b} = \left(\frac{1}{\log b} - \frac{1}{2}\right)\log L \\ &= \left(\frac{1}{1 - \log\frac{1}{1-\eta}} - \frac{1}{2}\right)\log L = \left(\frac{1}{2} + \frac{\log\frac{1}{1-\eta}}{1 - \log\frac{1}{1-\eta}}\right)\log L \\ \text{so } \left(2/\sqrt{b}\right)^{\log_b L} &= L^{\frac{1}{2}+o(1)},\end{aligned} \tag{60}$$

where we used $\frac{\log \frac{1}{1-\eta}}{1-\log \frac{1}{1-\eta}} = o(1)$, since $\log \frac{1}{1-\eta} = o(1)$, which follows from $\eta = o(1)$. Thus, continuing from Equation (59), we have:

$$\mathbb{E}[T_L] = \widetilde{O}\left(n\sqrt{L}\log^2(1/p)\right) L^{\frac{1}{2}+o(1)} = \widetilde{O}\left(nL^{1+o(1)}\log^2(1/p)\right). \tag{61}$$

We note that while our approach in Theorem 25 outperforms the simpler, non-divide-and-conquer algorithm analyzed in Equation (51), it performs worse than the algorithm of Ref. [18] for graphs with $L = \Omega(n^{1/2-o(1)})$. Thus, one could run Algorithm 3 until $O\left(n^{3/2}\right)$ queries had been made, and then switch to the algorithm of Ref. [18].

4.3.2 Path Finding in Arbitrary Graphs

When $G(x)$ is not known to only have one st-path, while it is possible that an algorithm similar to Algorithm 3 would solve st-$\text{PATH}_G(x)$, we have not been able to bound the running time effectively. This is because in the case of a single path, once you find an intermediate edge on the path, the longest paths from s and t to that edge must be shorter than the length of the longest path from s to t. This ensures that subproblems take shorter time than the original problem. With multiple paths, we no longer have that guarantee.

However, we provide an alternative approach that, while not as fast as Algorithm 3, still provides an improvement over the algorithm of [18] for graphs in which all (self-avoiding) paths from s to t are short. Our approach does not make use of our path-edge sampling algorithm as a subroutine, and instead uses the path detection algorithm of Lemma 8 to decide whether there are paths through various subgraphs, and then uses that information to find each edge in a path in order from s to t. In this way, we avoid the problem of subproblems being larger than the original problem, since if the longest path from s to t has length L, and the first edge we find on the path is (s, u), then longest path from u to t that doesn't go through s must have length at most $L - 1$. However, we lose the advantage of a divide-and-conquer approach.

To find the first edge on a path, we use a group testing approach. We divide the neighbors of s in G into two sets, S_1 and S_2 and run path detection algorithms in parallel on two subgraphs of $G(x)$, one with edges from s removed, except those to vertices in S_1 (that is, $G^-_{\{\{s,u\}\in E : u\in S_1\}}$), and one with edges from s removed, except those to vertices in S_2. We will detect which of these subgraphs contains a path, and we will know there is a path whose first edge goes from s to a vertex in the corresponding set (S_1 or S_2). Then we divide that set into half again, and repeat, until we have narrowed down our set to one vertex u, that must be the first vertex on a path from s to t.

At this point we have learned the first edge on a path from s to t. We then consider G^-_s, which is G with vertex s removed, and recursively iterate this procedure to learn the first edge on a path from u to t. Using this approach, we obtain the following result:

▶ **Theorem 26.** *Let $p \geq 0$, and $G = (V, E)$ with $s, t \in V$ be a family of n-vertex graphs, and suppose we are promised that there is a path from s to t in $G(x)$. On input x, if the longest st-path in $G(x)$ has length L (L need not be known ahead of time), there is a quantum algorithm that returns the edges on a path with probability $1-O(p)$ and uses $\widetilde{O}(nL^{3/2}\log(1/p))$ expected queries.*

A detailed description of the algorithm and the proof of Theorem 26 can be found in the full version of this work [26, Section 4.3.2].

References

1 Scott Aaronson. Open problems related to quantum query complexity. *ACM Transactions on Quantum Computing*, 2(4):1–9, 2021.

2 Noel T. Anderson, Jay-U Chung, Shelby Kimmel, Da-Yeon Koh, and Xiaohan Ye. Improved quantum query complexity on easier inputs. *arXiv preprint arXiv:2303.00217*, 2023.

3 Simon Apers and Stephen Piddock. Elfs, trees and quantum walks. *arXiv preprint arXiv:2211.16379*, 2022.

4 Salman Beigi and Leila Taghavi. Quantum speedup based on classical decision trees. *Quantum*, 4:241, 2020.

5 Salman Beigi, Leila Taghavi, and Artin Tajdini. Time- and query-optimal quantum algorithms based on decision trees. *ACM Transactions on Quantum Computing*, 3(4):1–31, 2022.

6 Aleksandrs Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS 2012)*, pages 207–216. IEEE, 2012.

7 Aleksandrs Belovs and Ben W. Reichardt. Span programs and quantum algorithms for st-connectivity and claw detection. In *Proceedings of the 20th Annual European Symposium on Algorithms (ESA 2012)*, pages 193–204. Springer, 2012.

8 Chris Cade, Ashley Montanaro, and Aleksandrs Belovs. Time and space efficient quantum algorithms for detecting cycles and testing bipartiteness. *Quantum Information & Computation*, 18(1-2):18–50, 2018.

9 Titouan Carette, Mathieu Laurière, and Frédéric Magniez. Extended learning graphs for triangle finding. *Algorithmica*, 82(4):980–1005, 2020.

10 Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, and Prasoon Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, 1996.

11 Denis X. Charles, Kristen E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22:93–113, 2007. doi:10.1007/s00145-007-9002-x.

12 Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC 2003)*, pages 59–68, 2003.

13 Andrew M. Childs, Matthew Coudron, and Amin Shiraz Gilani. Quantum algorithms and the power of forgetting. *arXiv preprint arXiv:2211.12447*, 2022.

14 Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354, 1998. doi:10.1098/rspa.1998.0164.

15 Arjan Cornelissen, Stacey Jeffery, Maris Ozols, and Alvaro Piedrafita. Span programs and quantum time complexity. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, pages 26:1–26:14, 2020.

16 Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematial Cryptology*, 8:209–247, 2014. doi:10.1515/jmc-2012-0015.

17 Kai DeLorenzo, Shelby Kimmel, and R. Teal Witter. Applications of the Quantum Algorithm for st-Connectivity. In *Proceedings of the 14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, volume 135 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:14, 2019. doi:10.4230/LIPIcs.TQC.2019.6.

18 Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006.

19 Steven D. Galbraith and Frederik Vercauteren. Computational problems in supersingular elliptic curve isogenies. *Quantum Information Processing*, 17(265), 2018. doi:10.1007/s11128-018-2023-6.

20 Dmitry Grinko, Julien Gacon, Christa Zoufal, and Stefan Woerner. Iterative quantum amplitude estimation. *npj Quantum Information*, 7(1):52, March 2021. doi:10.1038/s41534-021-00379-1.

21 Tsuyoshi Ito and Stacey Jeffery. Approximate Span Programs. *Algorithmica*, 81(6):2158–2195, 2019. doi:10.1007/s00453-018-0527-1.

22 Michael Jarret, Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum Algorithms for Connectivity and Related Problems. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:13, 2018. doi:10.4230/LIPIcs.ESA.2018.49.

23 Stacey Jeffery. Quantum subroutine composition. *arXiv preprint arXiv:2209.14146*, 2022.

24 Stacey Jeffery. Span programs and quantum space complexity. *Theory of Computing*, 18(1):1–49, 2022.

25 Stacey Jeffery and Shelby Kimmel. Quantum algorithms for graph connectivity and formula evaluation. *Quantum*, 1:26, 2017. doi:10.22331/q-2017-08-17-26.

26 Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum algorithm for path-edge sampling. *arXiv preprint arXiv:2303.03319*, 2023.

27 Stacey Jeffery and Sebastian Zur. Multidimensional quantum walks, with application to k-distinctness. *arXiv preprint arXiv:2208.13492*, 2022.

28 Mauricio Karchmer and Avi Wigderson. On span programs. In *Proceedings of the 8th Annual IEEE Conference on Structure in Complexity Theory*, pages 102–111, 1993.

29 A. Yu Kitaev. Quantum measurements and the Abelian Stabilizer Problem. *arXiv:quant-ph/9511026*, 1995. arXiv:quant-ph/9511026.

30 Troy Lee, Frédéric Magniez, and Miklos Santha. A learning graph based quantum query algorithm for finding constant-size subgraphs. *Chicago Journal of Theoretical Computer Science*, 18(1), 2011. doi:10.4086/cjtcs.2012.010.

31 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum Query Complexity of State Conversion. In *Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 344–353, 2011. doi:10.1109/FOCS.2011.75.

32 Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via Quantum Walk. *SIAM Journal on Computing*, 40(1):142–164, 2011. doi:10.1137/090745854.

33 Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

34 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)*, pages 560–569, 2011. doi:10.1137/1.9781611973082.44.

35 Ben W. Reichardt. Span programs are equivalent to quantum query algorithms. *SIAM Journal on Computing*, 43(3):1206–1219, 2014. doi:10.1137/100792640.

36 Seiichiro Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410:5285–5297, 2009. doi:10.1016/j.tcs.2009.08.030.

Improved Approximations for Extremal Eigenvalues of Sparse Hamiltonians

Daniel Hothem
Quantum Algorithms and Applications Collaboratory, Sandial National Laboratories, Livermore, CA, USA

Ojas Parekh
Quantum Algorithms and Applications Collaboratory, Sandial National Laboratories, Albuquerque, NM, USA

Kevin Thompson
Quantum Algorithms and Applications Collaboratory, Sandial National Laboratories, Albuquerque, NM, USA

Abstract

We give a classical $1/(qk+1)$-approximation for the maximum eigenvalue of a k-sparse fermionic Hamiltonian with strictly q-local terms, as well as a $1/(4k+1)$-approximation when the Hamiltonian has both 2-local and 4-local terms. More generally we obtain a $1/O(qk^2)$-approximation for k-sparse fermionic Hamiltonians with terms of locality at most q. Our techniques also yield analogous approximations for k-sparse, q-local qubit Hamiltonians with small hidden constants and improved dependence on q.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Mathematics of computing → Approximation algorithms

Keywords and phrases Approximation algorithms, Extremal eigenvalues, Sparse Hamiltonians, Fermionic Hamiltonians, Qubit Hamiltonians

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.6

Related Version *Previous Version*: `https://arxiv.org/abs/2301.04627`

Funding This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, National Quantum Information Science Research Centers, Exploratory Research for Extreme Scale Science program. Support is also acknowledged from the Accelerated Research in Quantum Computing program under the same office.

Acknowledgements We thank Yaroslav Herasymenko for an insightful contribution to Lemma 8. This article has been authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan `https://www.energy.gov/downloads/doe-public-access-plan`.

Table 1 Main results contrasted with the previous state of the art.

Hamiltonian		Our result	Previous result
fermionic	k-sparse, strictly q-local fermionic	$1/(qk+1)$	$1/\mathcal{O}(q^2k^2)$[5]
	k-sparse, $4,2$-local	$1/(4k+1)$	$1/\mathcal{O}(k^2)$[5]
	k-sparse, q-local	$1/\mathcal{O}(qk^2)$	N/A
qubit	k-sparse, strictly q-local	$1/(qk+1)$	$3^{-q/2}/(4qk)$[3]
	k-sparse, 2-local	$1/(2k+1)$	$1/(24\text{k})$[3]
	k-sparse, q-local	$1/\mathcal{O}(qk^2)$	$3^{-q/2}/(4qk)$[3]

1 Introduction

Finding the ground state energy of systems of particles is a fundamental problem of quantum mechanics. Finding the ground state energies of local Hamiltonians is believed to be difficult for both classical and quantum computers [7, 8]. Instead, it is often easier to find classical and quantum approximations to these ground state energies. In this paper, we consider approximations to the extremal eigenvalues of a local, k-sparse fermionic Hamiltonian:

$$H = \sum_{\Gamma} H_\Gamma c^\Gamma.$$

Here H is a fermionic Hamiltonian with real coefficients H_Γ, where ignoring phase factors, each term c^Γ is a product of q Majorana operators (i.e., H is q-local with q even) and each Majorana operator appears in at most k non-zero terms (i.e., H is k-sparse). We let $m = \sum_\Gamma |H_\Gamma|$.

Our main technical contribution is a carefully designed graph G, whose vertices correspond to the terms in H. We are able to construct states that achieve better approximations than in previous works by finding a suitably large independent set in G. This work is similar to (but distinct from) recent work by Herasymenko, Stroeks, Helsen, and Terhal [5] in which a similar graph is used to find *diffuse sets* of Majorana monomials from which they construct a state. Herasymenko et al. also work with a graph whose vertices correspond to the terms in H; however, their edge set is different. Our new edge set also allows us to generalize our results beyond the $q = 4, 2$ case handled by Herasymenko et al., and to prove better approximation ratios.

Table 1 summarizes our main results. We also list the previously known best results. The table is split into two sections: (1) fermionic Hamiltonians and (2) qubit Hamiltonians. Although our work does not focus on qubit Hamiltonians, our proof ideas furnish results that improve upon the previously known best results (see Sections 2 and 6).

2 Contextualizing our results

Bravyi, Gosset, Koenig, and Temme [2] were the first to suggest approximation algorithms for the largest eigenvalue of fermionic Hamiltonians using fermionic Gaussian states, achieving a $1/\mathcal{O}(n\log(n))$-approximation ratio for generic 4-local fermionic Hamiltonians. They also asked whether Gaussian states might provide a constant-factor approximation. Among other results, Hastings and O'Donnell [4] subsequently demonstrated that Gaussian states offer at best a $1/\Omega(\sqrt{n})$-approximation for a class of 4-local fermionic Hamiltonians, known as the Sachdev-Ye-Kitaev (SYK) model. Hamiltonians in the SYK model are dense 4-local Hamiltonians, hence the work of Hastings and O'Donnell left open the possibility of a

constant-factor Gaussian approximation algorithm for models with sparse Hamiltonians. Sparse Hamiltonians are a natural class of Hamiltonians to study. Examples, such as the Fermi-Hubbard model, are ubiquitous [6].

Recent work by Herasymenko, Stroeks, Helsen, and Terhal [5] proves the existence of such constant-factor approximations. They show that $\lambda_{\max}(H) \geq m/Q$, where $\lambda_{\max}(H)$ is the largest eigenvalue of H and $Q = q(q-1)(k-1)^2 + q(k-1) + 2$ for a k-sparse strictly q-local fermionic Hamiltonian. Herasymenko et al. also prove an improved ratio of $Q = 12(k+1)^2 + 4(k-1) + 2$ when specializing to k-sparse, 4, 2-local fermionic Hamiltonians. Our work directly improves upon these results (see Table 1). Our work also removes the conditions on system size present in Herasymenko et al. This leads to immediate improvements in Herasymenko et al.'s work on the sparse SYK model. All of the above results are obtained by efficient classical algorithms producing descriptions of Gaussian states. We refer the reader to [5] for further background, motivation, and applications to the SYK model. Finally, Herasymenko et al.'s result do not extend to k-sparse, q-local fermionic Hamiltonians (i.e., where all terms have locality *at most* q). To our knowledge, our $1/\mathcal{O}(qk^2)$ approximation is the first of this kind. It remains an open question whether this may be improved to $1/\mathcal{O}(qk)$.

Results of the above flavor were obtained for traceless k-sparse qubit Hamiltonians with constant locality by Harrow and Montanaro [3], who show that $\lambda_{\max}(H) \geq \Omega(m/k)$ using product states, where k-sparse and m are defined analogously as above; bounds upon which our ideas give a constant-factor improvement (see Table 1). They also give an improved bound with respect to the operator norm instead of the maximum eigenvalue: $\|H\| \geq \Omega(m/\sqrt{k})$. In the fermionic case, we give a 2-local example with $\lambda_{\max}(H) = \|H\| = \Theta(m/k)$, showing that such an improvement is not possible (see Section 7) and that our result for the strictly q-local case is tight. As noted in Table 1, our techniques also apply to the Hamiltonians considered by Harrow and Montanaro, yielding approximation guarantees with small hidden constants and improved dependence on q.

3 Preliminaries

In this section, we provide the necessary preliminaries for the rest of the work. We begin with an overview of fermionic Hamiltonians before providing the necessary background on Gaussian states. This section draws upon [5, 4, 1].

3.1 Fermionic Hamiltonians

Fermionic Hamiltonians describe systems of fermionic particles, such as electrons. For our purposes, it is easiest to express a fermionic Hamiltonian in terms of Majorana operators. Throughout, we use the notation $[n] := \{1, \ldots, n\}$. We also use the notation $\mathcal{E} = \{\Gamma \subseteq [n] \mid H_\Gamma \neq 0\}$ to denote the set of non-zero terms in a Hamiltonian.

▶ **Definition 1.** *Given n fermionic modes, a set of $2n$ traceless and Hermitian operators $\{c_i\}_{i=1}^{2n}$ are Majorana operators if they satisfy $c_i c_j + c_j c_i = 2\delta_{ij}$ for all $i, j \in [2n]$.*

▶ **Definition 2.** *Let $\{c_i\}$ be a collection of $2n$ Majorana operators endowed with an ordering (say the lexicographic ordering). A fermionic Hamiltonian has the form:*

$$H = \sum_{\Gamma \subseteq [2n]} H_\Gamma c^\Gamma, \tag{1}$$

where $\Gamma \subseteq [2n]$ has even order, c^Γ is the product of Majorana operators appearing in Γ (ordered lexicographically), and the $H_\Gamma \in \mathbb{R}$. Note that c^Γ may contain an additional pre-factor of i in order to satisfy hermiticity (e.g., when $|\Gamma| = 2$).

▶ **Definition 3.** *If H is a fermionic Hamiltonian defined in terms of $2n$ Majorana operators, then H is q-local if there exists $q \in \mathbb{N}$ such that each non-zero term in H has locality at most q, that is, for all $\Gamma \subseteq [2n]$ with $H_\Gamma \neq 0$, $|\Gamma| \leq q$. H is strictly q-local if $|\Gamma| = q$ for all non-zero summands.*

▶ **Definition 4.** *Let H be a fermionic Hamiltonian on $2n$ Majorana operators. Then H is k-sparse if each Majorana operator c_i appears in at most k non-zero terms, that is, for all $i \in [2n]$, $|\{\Gamma \in \mathcal{E} \mid i \in \Gamma\}| \leq k$.*

3.2 Gaussian states

First note that for any real, orthogonal matrix $R \in O(2n)$, the transformation

$$\tilde{c}_i = \sum_{j=1}^{2n} R_{ij} c_j, \tag{2}$$

gives rise to a new set of $[2n]$ Majorana operators $\{\tilde{c}_i\}$.

▶ **Definition 5.** *Let $\{c_i\}$ be a set of $[2n]$ Majorana operators, $R \in O(2n)$, and $\{\tilde{c}_i\}$ defined as in 2. For any assignment $\lambda_1, \ldots, \lambda_n \in [-1, 1]$, the following state is a (mixed) fermionic Gaussian state:*

$$\rho = \frac{1}{2^n} \prod_{j=1}^{n} \left(\mathbb{I} + i\lambda_j \tilde{c}_{2j-1} \tilde{c}_{2j} \right) \tag{3}$$

The state ρ is pure when $\lambda_j \in \{\pm 1\}$ for all $j \in [n]$.

Fermionic Gaussian states exhibit several nice properties. Not only are they the ground states of homogeneous 2-local fermionic Hamiltonians [1], but their higher-order correlates are efficiently computable from their *correlation matrix*. If ρ is defined as in Definition 5, then the correlation matrix M of ρ is the real, antisymmetric $2n \times 2n$ matrix with entries defined as:

$$M_{ij} = \frac{i}{2} Tr(\rho[c_i, c_j]). \tag{4}$$

The higher-order correlates of ρ can be computed via Wick's formalism:

$$Tr(c^\Gamma \rho) = Pf(M_\Gamma), \tag{5}$$

where M_Γ is the $|\Gamma| \times |\Gamma|$ submatrix of M containing only the ordered rows and columns in Γ and $Pf(\cdot)$ is the matrix Pfaffian. Finally, for any Gaussian state ρ, the set $\{\lambda_j\}$ and M are connected by the following lemma:

▶ **Lemma 6** (Bra05). *For any Gaussian state ρ with correlation matrix M, there exists some $R \in O(2n)$ such that the adjoint action of $O(2n)$ on M block-diagonalizes M into the following form:*

$$M = R \bigoplus_{j=1}^{n} \begin{pmatrix} 0 & \lambda_j \\ -\lambda_j & 0 \end{pmatrix} R^T, \tag{6}$$

where the λ_j are the same as in Definition 5. Thus, every real, anti-symmetric matrix M is the correlation matrix for some Gaussian state ρ.

4 Main approximation algorithm

In this section, we demonstrate our main technical ideas by proving an approximation ratio for k-sparse Hamiltonians with both 4-local and 2-local terms. We chose this specific case as it highlights all of our technical ideas, while also being the most physically interesting case. In Section 6 we show how these ideas generalize to the other cases described in Table 1.

▶ **Theorem 7.** *There is a classical polynomial time algorithm that, given as input the weights* $\{H_\Gamma\}$ *of some* k*-sparse and* $4,2$*-local fermionic Hamiltonian* H*, returns a description of a quantum state* ρ *achieving energy*

$$Tr(H\rho) \geq \frac{1}{4k+1}\sum_{\Gamma}|H_\Gamma| \geq \frac{1}{4k+1}\lambda_{max}(H).$$

Proof. Define a graph $G=(V,E)$ with vertices corresponding to the nonzero terms in the Hamiltonian (i.e., $V=\mathcal{E}$). The graph G may contain vertices corresponding to 2-local or 4-local terms. We include an edge $(v_\Gamma, v_{\Gamma'}) \in E$ if and only if one of the following conditions is met:

(i) c^Γ and $c^{\Gamma'}$ share one or more Majorana operators (i.e., $\Gamma \cap \Gamma' \neq \emptyset$), or
(ii) Γ and Γ' are disjoint and $\Gamma \cup \Gamma' \in \mathcal{E}$.

If there are m nonzero terms in the Hamiltonian then the graph G has m vertices, and the degree of a vertex in the graph is at most $4k$. We can see the latter as follows. Fix some vertex v_Γ. By construction,

$$\deg(v_\Gamma) = |\{(\Gamma,\Gamma') \in \mathcal{E}\times\mathcal{E} \mid \Gamma \text{ and } \Gamma' \text{ satisfy (i) or (ii)}\}|. \tag{7}$$

We consider two cases:

- Γ is 4-local. Consider an edge $(v_\Gamma, v_{\Gamma'})$. As H contains no 6-local or 8-local terms, $\Gamma \cap \Gamma' \neq \emptyset$. As H is k sparse, there are at most $4k$ Γ' for which this can occur.
- Γ is 2-local. Let a equal the number of 4-local Hamiltonian terms overlapping with Γ, and let b equal the number of 2-local terms overlapping with Γ. We claim that the degree of v_Γ is at most $2a+b$.
  There are b 2-local Γ' satisfying (i) with Γ. Each 2-local Γ' satisfying (ii) results in a unique 4-local $\Gamma\cup\Gamma' \in \mathcal{E}$ overlapping with Γ, hence there at most a such Γ'. Finally, no 4-local Γ' may satisfy (ii), and there are a 4-local Γ' satisfying (i).
  Since Γ overlaps with at most $2k$ Γ', we have $a+b \leq 2k$ so that $2a+b \leq 4k$.

By Brooks' Theorem we can in polynomial time find a coloring of the vertices of G with at most $4k+1$ colors. This means we can partition the vertices into at most $4k+1$ independent sets, $\{S_1,...,S_t\}$, with one of these sets having at least a $1/(4k+1)$ fraction of the sum of the absolute values of the weights:

$$\sum_\Gamma |H_\Gamma| = \sum_{S_i}\sum_{\Gamma\in S_i}|H_\Gamma| \leq (4k+1)\max_i \sum_{\Gamma\in S_i}|H_\Gamma|. \tag{8}$$

It follows from Equation (8) that

$$\max_i \sum_{\Gamma\in S_i}|H_\Gamma| \geq \frac{1}{(4k+1)}\sum_\Gamma |H_\Gamma|.$$

Define $S_j = \arg\max_j \sum_{\Gamma\in S_j}|H_\Gamma|$, and consider the following state:

$$\rho = \frac{1}{2^n}\prod_{\Gamma\in S_j}(\mathbb{I} + \text{sign}(H_\Gamma)c^\Gamma). \tag{9}$$

We claim that ρ is a valid quantum state and obtains objective $\sum_{\Gamma\in S_j}|H_\Gamma|$. By definition, ρ is proportional to a projector on a stabilizer state with stabilizer generators given by c^Γ for $\Gamma\in S_j$: Observe that $[c^\Gamma, c^{\Gamma'}]=0$ for all $\Gamma,\Gamma'\in S_j$ since S_j is an independent set. Hence, ρ is the product of commuting projectors and must be positive semidefinite.

To see that ρ obtains the desired objective, we first expand the product in Equation (9) as a sum and consider products of two or more terms, $\sigma=\prod_p c^{\Gamma_p}$ for $\Gamma_p\in S_j$. If any of the Γ_p are 4-local or $p\geq 3$, σ cannot be proportional to a term of H since the $\Gamma\in S_j$ are disjoint, and no cancellation in products of Majorana operators can occur. The remaining case is a product of two 2-local operators. For any such $\Gamma,\Gamma'\in S_j$, by (ii) and because S_j is an independent set, the product $c^\Gamma c^{\Gamma'}$ cannot be proportional to $c^{\Gamma''}$ for any $\Gamma''\in\mathcal{E}$.

Hence we have

$$\begin{aligned} &Tr(\mathbb{I}\rho)=1,\\ &Tr(c^\Gamma\rho)=\text{sign}(H_\Gamma)\quad\forall\Gamma\in S_j,\text{ and}\\ &Tr(c^\Gamma\rho)=0\quad\forall\Gamma\in\mathcal{E}\setminus S_j. \end{aligned}$$

This yields the desired claim that ρ is a normalized state for which

$$Tr(H\rho)=\sum_\Gamma H_\Gamma Tr(c^\Gamma\rho)=\sum_{\Gamma\in S_j}H_\Gamma Tr(c^\Gamma\rho)=\sum_{\Gamma\in S_j}|H_\Gamma|\geq\frac{1}{4k+1}\sum_\Gamma|H_\Gamma|. \qquad \blacktriangleleft$$

5 Conversion to a Gaussian state

The ρ constructed in Theorem 7 is, in fact, a mixture of Gaussian states. This is proven in the following lemma. This implies the existence of a Gaussian state with at least the same objective as ρ.

▶ **Lemma 8.** *The state ρ defined in Equation (9) is a mixture of Gaussian states.*

Proof. For each $\Gamma\in S_j$ let M_Γ be the perfect matching of the operators in Γ induced by the lexicographic ordering of Γ, and let M be a perfect matching of the Majorana operators in $\{c_1,\ldots c_{2n}\}\setminus\{c_i\mid\exists\Gamma\in S_j\text{ with }i\in\Gamma\}$ induced by the lexicographic ordering. Define the following Gaussian state:

$$\rho'(z)=\frac{1}{2^n}\prod_{\Gamma\in S_j}\prod_{gh\in M_\Gamma}(\mathbb{I}+z_{gh}\ ic_gc_h)\prod_{rs\in M}(\mathbb{I}+z_{rs}\ ic_rc_s), \tag{10}$$

where all $z_{gh},z_{rs}\in\{\pm1\}$.

Consider the state $\rho''=\mathbb{E}_z[\rho'(z)]$ where for each Γ the set $\{z_{gh}\}_{gh\in M_\Gamma}$ is uniformly random distributed over $\{\pm1\}^{|M_\Gamma|}$ subject to the constraint:

$$\text{sign}\left[\left(\prod_{gh\in M_\Gamma}z_{gh}\ ic_gc_h\right)c^\Gamma\right]=\text{sign}(H_\Gamma)\quad\forall\Gamma\in S_j, \tag{11}$$

where $\text{sign}(\pm\mathbb{I})$ is defined as ±1. In other words, $\{z_{gh}\}_{gh\in M_\Gamma}$ is chosen as the uniform distribution over strings in $\{\pm1\}^{|M_\Gamma|}$ which satisfy Equation (11). We will assume further that $\{z_{gh}\}_{gh\in M_\Gamma}$ is independent of all other $\{z_{gh}\}_{gh\in M_{\Gamma'}}$ and that each z_{rs} for $rs\in M$ is uniform and independent of all other random variables.

We claim that $\rho = \rho''$. Begin by using independence to push the expectation past the first and third products in Equation (10):

$$\rho'' = \frac{1}{2^n} \prod_{\Gamma \in S_j} \Big(\mathbb{E}_z \Big[\prod_{gh \in M_\Gamma} (\mathbb{I} + z_{gh}\ ic_g c_h) \Big] \Big) \prod_{rs \in M} \Big(\mathbb{E}_z \Big[(\mathbb{I} + z_{rs}\ ic_r c_s) \Big] \Big), \tag{12}$$

We first focus on the final product. Observe that:

$$\prod_{rs \in M} \Big(\mathbb{E}_z \Big[(\mathbb{I} + z_{rs}\ ic_r c_s) \Big] \Big) = \mathbb{I} \tag{13}$$

This follows from the independence of the $\{z_{rs} \mid rs \in M\}$ and because $\mathbb{E}_z[z_{rs}] = 0$ for all $rs \in M$. Hence:

$$\rho'' = \frac{1}{2^n} \prod_{\Gamma \in S_j} \Big(\mathbb{E}_z \Big[\prod_{gh \in M_\Gamma} (\mathbb{I} + z_{gh}\ ic_g c_h) \Big] \Big). \tag{14}$$

For fixed $\Gamma \in S_j$, we claim that:

$$\mathbb{E}_z \Big[\prod_{gh \in M_\Gamma} (\mathbb{I} + z_{gh}\ ic_g c_h) \Big] = \mathbb{I} + \operatorname{sign}(H_\Gamma) c^\Gamma. \tag{15}$$

Lemma 8 follows immediately from Equation (15). For any strict subset $\Gamma' \subsetneq \Gamma$, define

$$M_{\Gamma' \cap \Gamma} := \{gh \in M_\Gamma : g \in \Gamma', h \in \Gamma'\}.$$

We may then expand the left-hand side of Equation (15) as:

$$\mathbb{E}_z \Big[\prod_{gh \in M_\Gamma} (\mathbb{I} + z_{gh}\ ic_g c_h) \Big] = \mathbb{I} + \sum_{\Gamma' \subsetneq \Gamma} \mathbb{E}_z \Big[\prod_{gh \in M_{\Gamma' \cap \Gamma}} z_{gh}\ ic_g c_h \Big] + \mathbb{E}_z \Big[\prod_{gh \in M_\Gamma} z_{gh}\ ic_g c_h \Big] \tag{16}$$

$$= \mathbb{I} + \operatorname{sign}(H_\Gamma) c^\Gamma \tag{17}$$

The final expectation in Equation (16) evaluates to $\operatorname{sign}(H_\Gamma) c^\Gamma$ due to constraint 11. The sum of expectations in Equation (16) disappears as the marginal distribution of the z when restricted to a matching on a strict subset $\Gamma' \subsetneq \Gamma$ of size $|M_{\Gamma' \cap \Gamma}| = p$ is totally uniform over $\{\pm 1\}^p$. Therefore $\mathbb{E}_z[z_{gh}] = 0$ for any such matching. ◀

Although $\rho'(z)$ in Lemma 8 is a Gaussian state for any z, the state ρ'' is a mixture of Gaussian states by definition. However, we may derandomize the choice of z to obtain a Gaussian state. We only require pairwise independence of the elements of z, hence using standard derandomization approaches, we can obtain a Gaussian state $\rho'(z)$ in polynomial time such that $Tr(H\rho'(z)) \geq Tr(H\rho'')$.

6 Extensions

In this section, we demonstrate how our core approach in the proof of Theorem 7 leads to improved classical approximation algorithms for the ground state energies of various sparse, local Hamiltonians. Each case is dealt with as its own corollary to Theorem 7.

▶ **Corollary 9** (Strictly q-local Hamiltonians.). *Let H be a k-sparse, strictly q-local fermionic Hamiltonian. There exists a classical polynomial time algorithm that, given $\{H_\Gamma\}$ as input, outputs a description of a quantum state ρ achieving energy*

$$Tr(H\rho) \geq \frac{1}{qk+1} \sum_\Gamma |H_\Gamma| \geq \frac{\lambda_{max}(H)}{qk+1}. \tag{18}$$

Proof. In this case we only need to include edges in G between v_Γ and $v_{\Gamma'}$ precisely when condition (i) holds, since (ii) is vacuous. Consequently we may omit the second case below Equation (7) and simply bound the degree as qk. We then effectively replace "4" with q in the remaining proof. ◀

▶ **Corollary 10** (Hamiltonians with bounded locality.). *Let H be a k-sparse, q-local fermionic Hamiltonian. There exists a classical polynomial time algorithm that, given $\{H_\Gamma\}$ as input, outputs a description of a quantum state ρ achieving energy*

$$Tr(H\rho) \geq \frac{1}{Cqk^2} \sum_{\Gamma} |H_\Gamma| \geq \frac{\lambda_{max}(H)}{Cqk^2}, \tag{19}$$

for some constant $C \in \mathbb{R}$.

Proof. In this case we need an appropriate generalization of condition (ii) from Theorem 7. Let us start by defining G using only the condition (i); the maximum possible degree in G is qk. The purpose of (ii) in the proof is to ensure that for Γ, Γ' in the independent set S_j, $c^\Gamma c^{\Gamma'}$ cannot be proportional to $c^{\Gamma''}$ for any $\Gamma'' \in \mathcal{E}$. Note that if this happens, then Γ'' must contain both Γ and Γ'. Thus it would suffice for our independent set S_j in G to satisfy the additional property that no $v_\Gamma, v_{\Gamma'} \in S_j$ could have a common neighbor $v_{\Gamma''} \in V$ with $\Gamma, \Gamma' \subset \Gamma''$. We could satisfy this by adding an edge in G between all pairs v_Γ and $v_{\Gamma'}$ with such a common neighbor. By k-sparsity, the vertex v_Γ has at most k neighbors $v_{\Gamma''}$ in G with $\Gamma \subset \Gamma''$. Since any such $v_{\Gamma''}$ has degree at most qk, the degree of v_Γ increases by at most $k(qk-1)$, and maximum degree in the resulting graph G' is $O(qk^2)$. Applying Brooks' Theorem in G' produces the desired approximation. ◀

▶ **Corollary 11** (Qubit Hamiltonians). *Consider a k-sparse, q-local qubit Hamiltonian H defined analogously to the fermionic Hamiltonian in Definition 2. Given the appropriate assumptions on the locality of H, there exists a classical polynomial time algorithm that, given as inputs the weights $\{H_\Gamma\}$, outputs a description of a quantum state ρ achieving energy at least:*

Hamiltonian	Energy
strictly q-local	$1/(qk+1)$
k-sparse, 2-local	$1/(2k+1)$
k-sparse, q-local	$1/\mathcal{O}(qk^2)$

Proof. For qubit Hamiltonians, condition (i) in Theorem 7 is modified to cover any pair of local terms which involve the same qubit, while condition (ii) is modified to be "Γ and Γ' do not involve the same qubit." Our results for k-sparse and: (i) strictly q-local, (ii) 2-local, and (iii) q-local qubit Hamiltonians follow from this modification and considering Corollary 9, Theorem 7, and Corollary 10 respectively. ◀

7 Optimality of our strictly q-local result

For k-sparse H where all terms are q-local, since $\|H\| \geq \lambda_{\max}(H)$, our results show that

$$\|H\| \geq \lambda_{\max}(H) \geq \frac{m}{qk+1},$$

where we recall $m = \sum_\Gamma |H_\Gamma|$ and $\|\cdot\|$ denotes the operator norm. We give an explicit family of fermionic 2-local n-sparse Hamiltonians $\{H_n\}_{n=1}^\infty$ demonstrating this bound is asymptotically tight (i.e., cannot be improved for all q and k, up to constant factors).

Each H_n is expressed as a sum of monomials in $2n$ Majorana operators $\{c_1, c_2, ..., c_{2n}\}$ satisfying the usual canonical anti-commutation relations. For each n, partition $[2n]$ evenly into $A = \{1, ..., n\}$ and $B = \{n+1, ..., 2n\}$. Then:

$$H_n := \sum_{a\in A, b\in B} ic_a c_b = i\left(\sum_{a\in A} c_a\right)\left(\sum_{b\in B} c_b\right).$$

The eigenvalues of H_n are easy to determine, define $R \in O(2n)$ as some orthogonal matrix satisfying:

$$R_{a,1} = 1/\sqrt{n} \quad \forall a \in A \text{ and } R_{b,2} = 1/\sqrt{n} \quad \forall b \in B.$$

Note that this is well defined since the first two columns are orthonormal. We can then define a new set of Majorana operators (also satisfying the canonical anti-commutation relations) by:

$$\tilde{c}_i = \sum_{i=1}^{2n} R_{j,i} c_j.$$

In particular, we have

$$\tilde{c_1} = \frac{1}{\sqrt{n}} \sum_{a\in A} c_a \text{ and } \tilde{c_2} = \frac{1}{\sqrt{n}} \sum_{b\in B} c_b,$$

so

$$H = ni\tilde{c_1}\tilde{c_2}.$$

Since $i\tilde{c_1}\tilde{c_2}$ is Hermitian and satisfies $(i\tilde{c_1}\tilde{c_2})^2 = \mathbb{I}$, it has eigenvalues in $\{\pm 1\}$. Thus the eigenvalues of H_n are $\{\pm n\}$. Note that H_n is n-sparse, $m = n^2$, and $\|H_n\| = \lambda_{\max}(H_n)$ so that

$$\|H_n\| = \lambda_{\max}(H_n) = n = \Theta\left(\frac{n^2}{2n+1}\right) = \Theta\left(\frac{m}{qk+1}\right).$$

References

1. Sergey Bravyi. Lagrangian representation for fermionic linear optics. *Quantum Information & Computation.*, 5(3):216–238, 2005. doi:10.5555/2011637.2011640.
2. Sergey Bravyi, David Gosset, Robert Koenig, and Kristan Temme. Approximation algorithms for quantum many-body problems. *Journal of Mathematical Physics*, 60, 2019. doi:10.1063/1.5085428.
3. Aram W. Harrow and Ashley Montanaro. Extremal eigenvalues of local Hamiltonians. *Quantum*, 1:6, 2017. doi:10.22331/q-2017-04-25-6.
4. Matthew B. Hastings and Ryan O'Donnell. Optimizing strongly interacting fermionic hamiltonians. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 776–789, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3519960.
5. Yaroslav Herasymenko, Maarten Stroeks, Jonas Helsen, and Barbara Terhal. Optimizing sparse fermionic hamiltonians, 2022. arXiv:2211.16518.

6 John Hubbard. Electron correlations in narrow energy bands. *Proceedings of the Royal Society R. Society A*, 276:238–257, 1963. doi:10.1098/rspa.1963.0204.

7 Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*, pages 372–383, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/978-3-540-30538-5_31.

8 Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47 in Graduate Studies in Mathematics. American Mathematical Society, 2002. doi:10.1090/gsm/047.

Improved Algorithm and Lower Bound for Variable Time Quantum Search

Andris Ambainis
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

Martins Kokainis
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

Jevgēnijs Vihrovs
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

Abstract

We study variable time search, a form of quantum search where queries to different items take different time. Our first result is a new quantum algorithm that performs variable time search with complexity $O(\sqrt{T}\log n)$ where $T=\sum_{i=1}^{n} t_i^2$ with t_i denoting the time to check the i^{th} item. Our second result is a quantum lower bound of $\Omega(\sqrt{T\log T})$. Both the algorithm and the lower bound improve over previously known results by a factor of $\sqrt{\log T}$ but the algorithm is also substantially simpler than the previously known quantum algorithms.

2012 ACM Subject Classification Theory of computation → Quantum query complexity; Theory of computation → Quantum complexity theory

Keywords and phrases quantum search, amplitude amplification

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.7

Related Version *Full Version*: https://arxiv.org/abs/2302.06749

Funding This research was supported by the ERDF project 1.1.1.5/18/A/020.

Acknowledgements We thank Krišjānis Prūsis for useful discussions on the lower bound proof. The authors are grateful to the anonymous referees for the helpful comments and suggestions.

1 Introduction

We study variable time search [2], a form of quantum search in which the time needed for a query depends on which object is being queried. Variable time search and its generalization, variable time amplitude [3] amplification, are commonly used in quantum algorithms. For example,

- Ambainis [3] used variable time amplitude amplification to improve the running time of HHL quantum algorithm for solving systems of linear equations [12] from $\widetilde{O}(\kappa^2)$ (where κ is the condition number of the system) to $\widetilde{O}(\kappa^{1+o(1)})$ in different contexts;
- Childs et al. [8] used variable time amplitude amplification to design a quantum algorithm for solving systems of linear equations with an exponentially improved dependence of the running time on the required precision;
- Le Gall [15] used variable time search to construct the best known quantum algorithm for triangle finding, with a running time $\widetilde{O}(n^{5/4})$ where n is the number of vertices;
- De Boer et al. [10] used variable time search to optimize the complexity of quantum attacks against a post-quantum cryptosystem;
- Glos et al. [11] used variable time search to develop a quantum speedup for a classical dynamic programming algorithm.
- Schrottenloher and Stevens [16] used variable time amplitude amplification to transform a classical nested search into a quantum algorithm, with applications to quantum attacks on AES.

In those applications, the oracle for the quantum search is a quantum algorithm whose running time depends on the item that is being queried. For example, we might have a graph algorithm that uses quantum search to find a vertex with a certain property and the time t_v to check the property may depend on the degree of the vertex v.

In such situations, using standard quantum search would mean that we run the checking algorithm for the maximum possible time $t_{\max} = \max_v t_v$. If most times t_v are substantially smaller, this results in suboptimal quantum algorithms.

A more efficient strategy is to use the variable time quantum search algorithm [2]. It has two variants: the "known times" variant when times t_v for checking various v are known in advance and can be used to design the algorithm and the "unknown times" variant in which t_v are only discovered when running the algorithm. In the "known times" case, VTS (variable time search) has complexity $O(\sqrt{T})$ where $T = \sum_v t_v^2$ and there is a matching lower bound [2].

For the "unknown times" case, the complexity of the variable time search increases to $O(\sqrt{T}\log^{1.5} T)$ and the quantum algorithm becomes substantially more complicated. Since almost all of the applications of VTS require the "unknown times" setting, it may be interesting to develop a simpler quantum algorithm.

In more detail, the "unknown times" search works by first running the query algorithm for a small time T_1 and then amplifying v for which the query either returns a positive result or does not finish in time T_1. This is followed by running the query algorithm for longer time T_2, T_3, ... and each time, amplifying v for which the query either returns a positive result or does not finish in time T_i. To determine the necessary amount of amplification, quantum amplitude estimation is used. This results in a complex algorithm consisting of interleaved amplification and estimation steps. This complex structure contributes to the complexity of the algorithm, via log factors and may also lead to large constants hidden under the big-O.

In this paper, we develop a simple algorithm for variable time search that uses only amplitude amplification. Our algorithm achieves the complexity of $O(\sqrt{T}\log n)$ where T is an upper bound for $\sum_v t_v^2$ provided to the algorithm. (Unlike in the "known times" model, we do not need to provide $t_1, \ldots, t_n$ but only an estimate for T.) This also improves over the previous algorithm by a $\sqrt{\log}$ factor.

To summarize, the key difference from the earlier algorithms [2, 3] is that the earlier algorithms would use amplitude estimation (once for each amplification step) to determine the optimal schedule for amplitude amplification for this particular $t_1, \ldots, t_n$. In contrast, we use one fixed schedule for amplitude amplification (that depends only on the estimate for T and not on $t_1, \ldots, t_n$). While this schedule may be slightly suboptimal, the losses from it being suboptimal are less than savings from not performing multiple rounds of amplitude estimations. This also leads to the quantum algorithm being substantially simpler.

Our second result is a lower bound of $\Omega(\sqrt{T\log T})$, showing that a complexity of $\Theta(\sqrt{T})$ is not achievable. The lower bound is by creating a query problem which can be solved by variable time search and using the quantum adversary method to show a lower bound for this problem. In particular, this proves that "unknown times" search is more difficult than "known times" search (which has the complexity of $\Theta(\sqrt{T})$).

2 Model, definitions, and previous results

We consider the standard search problem in which the input consists of variables $x_1, \ldots, x_n \in \{0,1\}$ and the task is to find $i : x_i = 1$ if such i exists.

Our model is a generalization of the usual quantum query model. We model a situation when the variable x_i is computed by a query algorithm Q_i which is initialized in the state $|0\rangle$ and, after t_i steps, outputs the final state $|x_i\rangle|\psi_i\rangle$ for some unknown $|\psi_i\rangle$. (For most of the paper, we restrict ourselves to the case when Q_i always outputs the correct x_i. The bounded error case is discussed briefly at the end of this section.) In the first $t_i - 1$ steps, Q_i can be in arbitrary intermediate states.

The goal is to construct an algorithm A that finds $i : x_i = 1$ (if such i exists). The algorithm A can run the query Q_i for a chosen t, with Q_i outputting x_i if $t_i \leq t$ or * (an indication that the computation is not complete) if $t_i > t$. The complexity of A is the amount of time that is spent running the query algorithms Q_i. Transformations that does not involve running Q_i do not count towards the complexity.

More formally, we assume that, for any T, there is a circuit C_T which, on an input $\sum_{i=1}^n |i\rangle \otimes |0\rangle$ outputs

$$\sum_{i=1}^{n} |i\rangle \otimes |y_i\rangle \otimes |\psi_i\rangle ,$$

where $y_i = x_i$ if $t_i \leq T$ and $y_i = *$ if $t_i > T$. The state $|\psi_i\rangle$ contains intermediate results of the computation and can be arbitrary. An algorithm A for variable time search consists of two types of transformations:

- circuits C_T for various T;
- transformations U_i that are independent of $x_1, \ldots, x_n$.

If there is no intermediate measurements, an algorithm A is of the form

$$U_k C_{T_k} U_{k-1} \ldots U_1 C_{T_1} U_0$$

and its complexity is defined as $T_1 + T_2 + \ldots + T_k$. In the general case, an algorithm is a sequence

$$U_0, C_{T_1}, U_1, \ldots, C_{T_k}, U_k$$

with intermediate measurements. Depending on the outcomes of those measurements, the algorithm may stop and output the result or continue with the next transformations. The complexity of the algorithm is defined as $p_1 T_1 + \ldots + p_k T_k$ where p_i is the probability that C_{T_i} is performed. (One could also allow U_i and T_i to vary depending on the results of previous measurements but this will not be necessary for our algorithm.)

If there exists $i : x_i = 1$, A must output one of such i with probability at least $2/3$. If $x_i = 0$, A must output "no such i" with probability at least $2/3$.

Known vs. unknown times. This model can be studied in two variants. In the "known times" variant, the times t_i for each $i \in [n]$ are known in advance and can be used to design the search algorithm. In the "unknown times" variant, the search algorithm should be independent of the times t_i, $i \in [n]$.

The complexity of the variable time search is characterized by the parameter $T = \sum_{i=1}^n t_i^2$. We summarize the previously known results below.

▶ **Theorem 2.1** ([2, 3]).

(a) Algorithm – known times: *For any $t_1, \ldots, t_n$, there is a variable time search algorithm $A_{t_1,\ldots,t_n}$ with the complexity $O(\sqrt{T})$.*

(b) Algorithm – unknown times: *There is a variable time search algorithm A with the complexity $O(\sqrt{T}\log^{1.5} T)$ for the case when $t_1, \ldots, t_n$ are not known in advance.*

(c) Lower bound – known times. *For any $t_1, \ldots, t_n$ and any variable time search algorithm $A_{t_1,\ldots,t_n}$, its complexity must be $\Omega(\sqrt{T})$.*

Parts (a) and (c) of the theorem are from [2]. Part (b) is from [3], specialized to the case of search.

In the recent years there have been attempts to reproduce and improve the aforementioned results by other means. In [9], the authors obtain a variant of Theorem 2.1(a) by converting the original algorithms into span programs, which then are composed and subsequently converted back to a quantum algorithm. More recently, [14] gives variable time quantum walk algorithm (which generalizes variable time quantum search) by employing a recent technique of multidimensional quantum walks. While the focus of these two papers is on developing very general frameworks, our focus is on making the variable time search algorithm simpler.

Concurrently and independently of our work, a similar algorithm for variable time amplitude amplification was presented in [16], which also relies on recursive nesting of quantum amplitude amplifications.

Variable time search with bounded error inputs. We present our results for the case when the queries Q_i are perfect (have no error) but our algorithm can be extended to the case if Q_i are bounded error algorithms, at the cost of an extra logarithmic factor.

Let k be the maximum number of calls to C_T's in an algorithm A. Then, it suffices that each C_T outputs a correct answer with a probability $1 - o(1/k^2)$. This can be achieved by repeating C_T $O(\log k)$ times and taking the majority of answers.

Possibly, this logarithmic factor can be removed using methods similar to ones for search with bounded error inputs in the standard (not variable time) setting [13].

3 Algorithm

We proceed in two steps. We first present a simple algorithm for the case when a sufficiently good bound on the number of solutions $m = |i : x_i = 1|$ are known (Section 3.2). We then present an algorithm for the general case that calls the simple algorithm multiple times, with different estimates for the parameter ℓ corresponding to m (Section 3.3).

Both algorithms require an estimate T for which $\sum_{i=1}^{n} t_i^2 \leq T$, with the complexity depending on T.

3.1 Tools and methods

Before presenting our results, we describe the necessary background about quantum amplitude amplification [6].

Amplitude amplification – basic construction. Assume that we have an algorithm A that succeeds with a small probability and it can be verified whether A has succeeded. Amplitude amplification is a procedure for increasing the success probability. Let

$$A|0\rangle = \sin\alpha\,|\psi_{\text{succ}}\rangle + \cos\alpha\,|\psi_{\text{fail}}\rangle\,.$$

Then, there is an algorithm $A(k)$ that involves $k+1$ applications of A and k applications of A^{-1} such that

$$A(k)\,|0\rangle = \sin\left((2k+1)\alpha\right)|\psi_{\text{succ}}\rangle + \cos\left((2k+1)\alpha\right)|\psi_{\text{fail}}\rangle\,.$$

Knowledge of α is not necessary (the way how $A(k)$ is obtained from A is independent of α).

Amplitude amplification – amplifying to success probability $1-\delta$. If α is known then one can choose $k = \lfloor \frac{\pi}{4\alpha} \rfloor$ to amplify to a success probability close to 1 (since $(2k+1)\alpha$ will be close to $\frac{\pi}{2}$). If the success probability of A is ϵ, then $\sin\alpha \approx \sqrt{\epsilon}$ and $k \approx \frac{\pi}{4\sqrt{\epsilon}}$.

For unknown α, amplification to success probability $1-\delta$ for any $\delta > 0$ can be still achieved, via a more complex algorithm. Namely, for any $\epsilon, \delta \in (0,1)$ and any A, one can construct an algorithm $A(\epsilon,\delta)$ such that:

- $A(\epsilon,\delta)$ invokes A and A^{-1} $O(\frac{1}{\sqrt{\epsilon}}\log\frac{1}{\delta})$ times;
- If A succeeds with probability at least ϵ, $A(\epsilon,\delta)$ succeeds with probability at least $1-\delta$.

To achieve this, we first note that performing $A(k)$ for a randomly chosen $k \in \{1,\ldots,M\}$ for an appropriate $M = O\left(\frac{1}{\sqrt{\epsilon}}\right)$ and measuring the final state gives a success probability that is close to 1/2 (as observed in the proof of Theorem 3 in [6]). Repeating this procedure $O\left(\log\frac{1}{\delta}\right)$ times achieves the success probability of at least $1-\delta$.

3.2 Algorithm with a fixed number of stages

Now we present an informal overview of the algorithm when tight bounds on the number of solutions $m = |i : x_i = 1|$ is known. We will define a sequence of times $T_1, T_2, \ldots$ and procedures $A_1, A_2, \ldots$. We choose $T_1 = 3\sqrt{T/n}$ (this ensures that at most $n/9$ of indices $i \in [n]$ have $t_i \geq T_1$) and $T_2 = 3T_1$, $T_3 = 3T_2$, ... until d for which $T_d \geq \sqrt{T}$. The procedure A_1 creates the superposition $\sum_{i=1}^{n}\frac{1}{\sqrt{n}}|i\rangle$ and runs the checking procedure C_{T_1}, obtaining state of the form $\sum_{i=1}^{n}\frac{1}{\sqrt{n}}|i,a_i\rangle$, where $a_i \in \{0,1,*\}$, with $*$ denoting a computation that did not terminate. The subsequent procedures A_j are defined as $A_j = C_{T_j}A_{j-1}(1)$, i.e., we first amplify the parts of the state with outcomes 1 or $*$ and then run the checking procedure C_{T_j}.

We express the final state of A_{j-1} as

$$\sin\alpha_{j-1}\,|\psi_{\text{succ}}\rangle + \cos\alpha_{j-1}\,|\psi_{\text{fail}}\rangle\,,$$

where $|\psi_{\text{succ}}\rangle$ consists of those indices $i \in [n]$ which are either 1 or are still unresolved $*$ (and thus have the potential to turn out to be '1'). Then the amplitude amplification part triples the angle α_{j-1}, i.e., amplifies both the "good" and "unresolved" states by a factor of $\sin(3\alpha_{j-1})/\sin(\alpha_{j-1}) \approx 3$. We will show that $\ell = \lceil \log_9 \frac{n}{m} \rceil$ stages are sufficient, i.e., the procedure A_ℓ the amplitude at the "good" states (if they exist) is sufficiently large.

We note that the idea of recursive tripling via amplitude amplification has been used in other contexts. It has been used to build an algorithm for bounded-error search in [13]; more recently, the recursive tripling trick has also been used in, e.g., [7]. Furthermore, the repeated tripling of the angle α also explains the scaling factor 3 when defining the sequence $T_1, T_2, T_3 \ldots$

A formal description follows.

We assume an estimate $T \geq \sum_i t_i^2$ to be known and set

$$T_1 = 3\sqrt{T/n},\ T_2 = 3T_1,\ \ldots, T_d = 3T_{d-1},$$

with $d \in \mathbb{N}$ s.t. $T_{d-1} < \sqrt{T} \leq T_d$ (equivalently, $9^{d-1} < n \leq 9^d$).

Let $\mathcal{M} = \{i \in [n] : x_i = 1\}$, $m = |\mathcal{M}|$. We assume that we know ℓ for which m belongs to the interval $\left[\frac{n}{9^\ell}, \frac{n}{9^{\ell-1}}\right)$ (so that $\ell = \lceil \log_9 \frac{n}{m} \rceil$).

Under those assumptions, we now describe a variable time search algorithm with parameters T, ℓ.

Algorithm 1 VTS algorithm with a fixed number of stages.

Parameters: T, n, ℓ, δ.
Run the amplified algorithm $A(0.04, \delta)$ where A is the procedure defined below and we amplify the part of the state for the second register contains '1'
procedure A
    Run A_ℓ ▷ (defined below)
    Run $C_{T_{\ell+1}}$ (or C_{T_d} if $\ell = d$)
end procedure
Measure the state
if The second register is '1' **then**
    Output i from the first register
else
    Output `No solutions.`
end if
procedure A_j ▷ $j \in [d]$
    if $j = 1$ **then**
        Create the state $\sum_{i=1}^n \frac{1}{\sqrt{n}} |i\rangle$
        Run C_{T_1}, obtaining state of the form $\sum_{i=1}^n \frac{1}{\sqrt{n}} |i, a_i\rangle$ where $a_i \in \{0, 1, *\}$.
    else
        Perform the amplified algorithm $A_{j-1}(1)$, amplifying the basis states with 1 or * in the second register
        if $j < \ell$ **then**
            Run C_{T_j}.
        end if
    end if
end procedure

▶ **Lemma 3.1.** *Algorithm 1 with parameter $\ell = \lceil \log_9 \frac{n}{m} \rceil$ finds an index $i \in \mathcal{M}$ with probability at least $1 - \delta$ in time $O\left(\sqrt{\frac{T}{m}} \log \frac{n}{m} \log \frac{1}{\delta}\right)$.*

Proof. By $\mathcal{S}_j$ we denote the sets of those indices whose amplitudes will be amplified after running A_j, namely, the set of indices for which the query either returns a positive result or does not finish in time T_j:

$$\mathcal{S}_j = \{i \in [n] \ : \ (T_j < t_i) \vee (t_i \leq T_j \wedge x_i = 1)\}, \quad j = 0, 1, 2, \ldots, d,$$

where $T_0 := 0$. We note that the sets $\mathcal{S}_j$ form a decreasing sequence[1], i.e.,

$$[n] = \mathcal{S}_0 \supseteq \mathcal{S}_1 \supseteq \mathcal{S}_2 \supseteq \ldots \supseteq \mathcal{S}_{d-1} \supseteq \mathcal{S}_d = \mathcal{M}.$$

We shall denote the cardinality of $\mathcal{S}_j$ by s_j; then

$$n = s_0 \geq s_1 \geq \ldots \geq s_d = m.$$

[1] Since each i s.t. $t_i \leq T_j \wedge x_i = 1$ either satisfies $t_i \leq T_{j-1} \wedge x_i = 1$ or $t_i > T_{j-1}$; in both cases $i \in \mathcal{S}_{j-1}$.

We express the final state of A_j as

$$\sin\alpha_j\,|\psi_{\mathrm{succ},j}\rangle + \cos\alpha_j\,|\psi_{\mathrm{fail},j}\rangle$$

where $|\psi_{\mathrm{succ},j}\rangle$ consists of basis states with $|i\rangle$, $i \in \mathcal{S}_j$, in the first register and $|\psi_{\mathrm{fail},j}\rangle$ consists of basis states with $|i\rangle$, $i \notin \mathcal{S}_j$, in the first register.

We begin by describing how the cardinality of $\mathcal{S}_j$ is related to the amplitude $\sin\alpha_j$ (the proof is deferred to Appendix A).

▶ **Lemma 3.2.** *For all $j = 1, 2, \ldots, \ell$,*

$$\sin^2\alpha_j = \frac{s_j}{n}\prod_{k=1}^{j-1}\left(\frac{\sin(3\alpha_k)}{\sin\alpha_k}\right)^2. \tag{1}$$

Moreover, for any $i \in \mathcal{S}_j$, the amplitude at $|i,1\rangle$ (or $|i,*\rangle$, if $t_i > T_j$) equals $\frac{\sin\alpha_j}{\sqrt{s_j}}$.

Equation (1) and the trigonometric identity

$$\sin(3\alpha) = (3 - 4\sin^2\alpha)\sin\alpha$$

allows to obtain (for $j = 1, 2, \ldots, \ell$)

$$\frac{\sin(3\alpha_j)}{\sin\alpha_j} = 3 - 4\sin^2\alpha_j = 3 - \frac{4s_j\cdot 9^{j-1}}{n}\prod_{k=1}^{j-1}\left(\frac{\sin(3\alpha_k)}{3\sin\alpha_k}\right)^2 \geq 3 - \frac{4s_j}{n}\cdot 9^{j-1}, \tag{2}$$

where the inequality is justified by the observation $\left|\frac{\sin(3\alpha)}{3\sin\alpha}\right| \leq 1$. This allows to estimate

$$\sin\alpha_\ell = \sqrt{\frac{s_\ell}{n}}\prod_{j=1}^{\ell-1}\frac{\sin(3\alpha_j)}{\sin\alpha_j} \geq 3^{\ell-1}\sqrt{\frac{s_\ell}{n}}\prod_{j=1}^{\ell-1}\left(1 - \frac{4s_j}{27n}\cdot 9^j\right), \tag{3}$$

as long as each factor on the RHS is positive. We argue that it is indeed the case; moreover, the whole product is lower-bounded by a constant (the proof is deferred to Appendix A):

▶ **Lemma 3.3.** *The following claims hold:*

C-1 *Each factor on the RHS of* (3) *is positive:* $\frac{9^j s_j}{n} \leq \frac{9}{4}$, *thus*

$$\left(1 - \frac{4s_j}{27n}\cdot 9^j\right) \geq \frac{2}{3}, \quad \text{for all } j \in [\ell - 1].$$

C-2 *The product* $\prod_{j=1}^{\ell-1}\left(1 - \frac{4s_j}{27n}\cdot 9^j\right)$ *is lower bounded by* $2/3$.

C-3 $9^\ell s_\ell \geq 9^\ell s_d \geq n$.

From (3) and Lemma 3.3 it is evident that $\sin\alpha_\ell \geq \frac{2}{9}\sqrt{\frac{9^\ell s_\ell}{n}} \geq \frac{2}{9}$.

However, after running A_ℓ, there still could be some unresolved indices i with $t_i > T_\ell$ and some of these unresolved indices may correspond to $x_i = 0$. Our next argument is that running $C_{T_{\ell+1}}$, i.e., the checking procedure for $3T_\ell$ steps, resolves sufficiently many indices in $\mathcal{M}$. This argument, however, is necessary only for $\ell < d$; for $\ell = d$, one runs C_{T_d} instead of $C_{T_{\ell+1}}$ and the same estimate (4) of the success probability applies, with $8m/9$ replaced by m. Also notice that in Algorithm 1 we skipped running C_{T_ℓ} at the end of A_ℓ and immediately proceeded with running $C_{T_{\ell+1}}$ instead. In the analysis, this detail is omitted for convenience (since it is equivalent to running C_{T_j} at the end of each procedure A_j and additionally running $C_{T_{\ell+1}}$ after A_ℓ).

By the choice of ℓ we have $\sqrt{\frac{T}{m}} \leq T_\ell = \sqrt{\frac{9^\ell T}{n}}$ and $T_{\ell+1} \geq 3\sqrt{\frac{T}{m}}$. Notice that at most $m/9$ of the indices $i \in [n]$ can satisfy $t_i^2 > T_{\ell+1}^2$ (otherwise, the sum over those indices already exceeds $\frac{m}{9} \cdot \frac{9T}{m} = T$). Consequently, after running the checking procedure $C_{T_{\ell+1}}$, at least $8m/9$ of the indices in $\mathcal{M}$ will be resolved to '1'. By Lemma 3.2, the amplitude at each of the respective states $|i, 1\rangle$ is equal to $\frac{\sin \alpha_\ell}{\sqrt{s_\ell}}$, therefore the probability to measure '1' in the second register is at least

$$\frac{8m}{9} \cdot \frac{\sin^2 \alpha_\ell}{s_\ell} \geq \frac{8m}{9} \cdot \frac{9^\ell}{n} \left(\frac{1}{3} \prod_{j=1}^{\ell-1} \left(1 - \frac{4s_j}{27n} \cdot 9^j \right) \right)^2 \geq \frac{8}{9} \cdot \left(\frac{2}{9} \right)^2 > 0.04, \tag{4}$$

where the first inequality follows from (3) and the second inequality is due to **C-2** and **C-3**.

We conclude that the procedure A finds an index $i \in \mathcal{M}$ with probability at least 0.04; its running time is easily seen to be

$$T_{\ell+1} + T_\ell + 3\left(T_{\ell-1} + 3\left(T_{\ell-2} + \ldots + 3\left(T_2 + 3T_1\right)\right)\right) = (3+\ell)T_\ell,$$

which for our choice of ℓ is of order

$$O\left(\log \frac{n}{m} \sqrt{9^\ell \frac{T}{n}} \right) = O\left(\log \frac{n}{m} \sqrt{\frac{T}{m}} \right).$$

Use $O(\log \frac{1}{\delta})$ rounds amplitude amplification to amplify the success probability of A to $1 - \delta$, concluding the proof. ◀

3.3 Algorithm for the general case

When the cardinality of $|\mathcal{M}|$ is not known in advance, we run Algorithm 1 with increasing values of ℓ (which corresponds to exponentially decreasing guesses of m) until either $i : x_i = 1$ is found or we conclude that no such i exists. Algorithm 1 also suffers the "soufflé problem" [5] in which iterating too much (choosing ℓ in Algorithm 1 larger than its optimal value) may "overcook" the state and decrease the success probability. For this reason, before running Algorithm 1 with the next value of ℓ, we re-run it with all the previous values of ℓ to ensure that the probability of running Algorithm 1 with too large ℓ is small. This ensures that the algorithm stops in time $O\left(\sqrt{\frac{T}{m}} \log \frac{n}{m}\right)$ with high probability. Formally, we make the following claim:

▶ **Lemma 3.4.** *If $\mathcal{M}$ is nonempty, Algorithm 2 finds an index $i \in \mathcal{M}$ with probability at least 5/6 with complexity $O\left(\sqrt{\frac{T}{m}} \log \frac{n}{m}\right)$. If $\mathcal{M}$ is empty, Algorithm 2 outputs `No solutions.` with complexity $O\left(\sqrt{T} \log n\right)$.*

Proof of Lemma 3.4. Let $\delta = 1/6$; let us remark that each procedure B_k runs in time $O\left(k3^k\sqrt{T/n}\right)$.

Let us consider the case when $m = |\mathcal{M}| > 0$; denote $\ell := \lceil \log_9 \frac{n}{m} \rceil$. The probability of B_k, $k \neq \ell$, finding an index $i \in \mathcal{M}$ is lower-bounded by 0; the probability of B_ℓ finding an index $i \in \mathcal{M}$ is lower-bounded by $1 - \delta$.

Hence, the total complexity of the algorithm stages $j = 1, 2, \ldots, \ell$, is of order

$$\sqrt{\frac{T}{n}} \sum_{j=1}^{\ell} \sum_{k=1}^{j} k3^k = \sqrt{\frac{T}{n}} \sum_{j=1}^{\ell} (\ell + 1 - j) j 3^j \asymp \ell 3^\ell \sqrt{\frac{T}{n}}.$$

and the last step B_ℓ finds $i \in \mathcal{M}$ with probability at least $1 - \delta$.

Algorithm 2 VTS algorithm for arbitrary number of solutions m.

Parameters: T, n.
Let B_k stand for Algorithm 1 with parameters T, n, k and $\delta = 1/6$.
for $j = 1, 2, \ldots, d$ **do**
  for $k = 1, 2, \ldots, j$ **do**
    Run B_k
    If B_k returned $i \in \mathcal{M}$, output this i and quit
  end for
end for
Output `No solutions.`

With probability at most δ, the last step fails to find $i \in \mathcal{M}$, and then Algorithm 2 proceeds with $j = \ell + 1$ and runs the sequence B_1, B_2, …, B_ℓ, where the last step finds $i \in \mathcal{M}$ with (conditional) probability at least $1 - \delta$ (conditioned on the failure to find $i \in \mathcal{M}$ in the previous batch). The complexity of this part is of order

$$\delta\sqrt{\frac{T}{n}}\left(\sum_{j=1}^{\ell} j3^j\right) \asymp \delta\sqrt{\frac{T}{n}}\ell 3^\ell,$$

where the δ factor reflects the fact the respective procedures are invoked with probability δ.

With (total) probability at most δ^2, the algorithm still has not found $i \in \mathcal{M}$. Then Algorithm 2 runs the sequence $B_{\ell+1}$, B_1, B_2, …, B_ℓ (i.e., finishes with $j = \ell + 1$ and continues with $j = \ell + 2$), where the last step finds $i \in \mathcal{M}$ with (conditional) probability at least $1 - \delta$. The complexity of this part is of order

$$\delta^2\sqrt{\frac{T}{n}}\left((\ell+1)3^{\ell+1} + \ell 3^\ell\right) \asymp \delta^2\sqrt{\frac{T}{n}}(\ell+1)3^{\ell+1}.$$

With (total) probability at most δ^3, the algorithm still has not found $i \in \mathcal{M}$. Then Algorithm 2 runs the sequence $B_{\ell+1}$,$B_{\ell+2}$, B_1, B_2, …, B_ℓ, where the last step finds $i \in \mathcal{M}$ with (conditional) probability at least $1 - \delta$. The complexity of this part is of order

$$\delta^3\sqrt{\frac{T}{n}}\left((\ell+1)3^{\ell+1} + (\ell+2)3^{\ell+2} + \ell 3^\ell\right) \asymp \delta^3\sqrt{\frac{T}{n}}(\ell+2)3^{\ell+2},$$

and so on.

For $j = d$, the final batch $B_1, B_2, \ldots, B_\ell$ is invoked with probability at most $\delta^{d-\ell}$; with conditional probability at most δ we still fail to find $i \in \mathcal{M}$ and run the remaining sequence $B_{\ell+1}, \ldots, B_d$ (which can completely fail finding any $i \in \mathcal{M}$ as it has no non-trivial lower bounds on the success probability). The complexity of the latter sequence is of order

$$\delta^{d+1-\ell}\sqrt{\frac{T}{n}}\left((\ell+1)3^{\ell+1} + (\ell+2)3^{\ell+2} + \ldots + d3^d\right) \asymp \delta^{d+1-\ell}\sqrt{\frac{T}{n}}\, d\, 3^d.$$

We see that Algorithm 2 fails with probability at most $\delta^{d+1-\ell}$; since $\ell \le d$, this is upper-bounded by $\delta = 1/6$. The total complexity of the algorithm is of order

$$
\begin{aligned}
&3^\ell\sqrt{\frac{T}{n}}\left(\ell+3\delta^2(\ell+1)+9\delta^3(\ell+2)+\ldots+(3\delta)^{d-\ell}\cdot d\delta\right)\\
&<3^\ell\sqrt{\frac{T}{n}}\left(\ell+\ell\,3\delta^2\sum_{i=0}^{\infty}(3\delta)^i+3\delta^2\sum_{i=1}^{\infty}i(3\delta)^{i-1}\right)\\
&=3^\ell\sqrt{\frac{T}{n}}\left(\ell+\ell\frac{3\delta^2}{1-3\delta}+\frac{3\delta^2}{(1-3\delta)^2}\right)\asymp\ell 3^\ell\sqrt{\frac{T}{n}},
\end{aligned}
$$

since $3\delta = 1/2$. Since $3^\ell \asymp \sqrt{\frac{n}{m}}$ and $\ell \asymp \log\frac{n}{m}$, we conclude that the complexity of the algorithm is $O\left(\sqrt{\frac{T}{m}}\log\frac{n}{m}\right)$, as claimed.

Let us consider the case when $\mathcal{M}$ is empty; then with certainty each B_j fails to output any i, and Algorithm 2 correctly outputs `No solutions`. In this case, the complexity of the algorithm is of order

$$
\sqrt{\frac{T}{n}}\sum_{j=1}^{d}\sum_{k=1}^{j}k3^k=\sqrt{\frac{T}{n}}\sum_{j=1}^{d}(d+1-j)j3^j\asymp d3^d\sqrt{\frac{T}{n}}\asymp\sqrt{T}\log n,
$$

since $3^d \asymp \sqrt{n}$. ◀

4 Lower bound

For the improved lower bound, we consider a query problem which can be solved with variable time search. Let $g : \{0,1,\star\}^m \to \{0,1\}$ be a partial function defined on the strings with exactly one non-$\star$ value, which is the value of the function. The function f we examine then is the composition of OR_n with g. We note that g is also known in the literature as pSEARCH, which has been used for quantum lower bounds in cryptographic applications [4].

For any $i \in [n]$, if the index of the non-$\star$ element in the corresponding instance of g is $j_i \in [m]$, then we can find this value in $O(\sqrt{j_i})$ queries using Grover's search. This creates an instance of the variable search problem with unknown times $t_i = \sqrt{j_i}$. By examining only inputs with fixed $T = \sum_{i=1}^n t_i^2 = \sum_{i=1}^n j_i$ and the restriction of f on these inputs f_T, we are able to prove a $\Omega(\sqrt{T\log T})$ query lower bound using the weighted quantum adversary bound [1]. Since any quantum algorithm for the variable time search also solves f_T, this gives the required lower bound.

▶ **Theorem 4.1.** *Any algorithm that solves variable time search with unknown times t_i requires time $\Omega(\sqrt{T\log T})$, where $T = \sum_{i\in[n]} t_i^2$.*

We note that the lower bound of Theorem 4.1 contains a factor of $\sqrt{\log T}$ while the upper bound of Lemma 3.4 contains a factor of $\log n$. There is no contradiction between these two results as the lower bound uses inputs with $T = \Theta(n\log n)$ and for those inputs $\log T = (1+o(1))\log n$.

Proof of Theorem 4.1. Consider a partial function $f : D \to \{0,1\}$, where $D \subset \{\star,0,1\}^{[n]\times[m]}$, defined as follows. An input $x \in D$ if for each $i \in [n]$ there is a unique $j \in [m]$ such that $x_{i,j} \neq \star$; denote this j by $j_{x,i}$. Then $f(x) = 1$ iff there exists an i such that $x_{i,j_{x,i}} = 1$.

Suppose that x is given by query access to $x_{i,j}$. For any i, we can check whether $x_{i,j_{x,i}} = 1$ in $O(\sqrt{j_{x,i}})$ queries with certainty in the following way. There is a version of Grover's search that detects a marked element out of N elements in $O(\sqrt{N})$ queries with certainty, if the

number of marked elements is either 0 or 1 [6]. By running this algorithm for the first N elements, where we iterate over $N = 1, 2, \ldots, 2^{\lceil \log_2 j_{x,i} \rceil}$, we will detect whether $x_{i,j_{x,i}} = 1$ in $O(\sqrt{j_{x,i}})$ queries with certainty.

Letting $t_i = \sqrt{j_{x,i}}$ and $T = \sum_{i\in[n]} t_i^2$, we get an instance of a variable search problem. Now fix any value of T and examine only inputs with such T. Denote f restricted on T by f_T. If the quantum query complexity of f_T is $\mathrm{Q}(f_T)$, then any algorithm that solves variable time search must require at least $\Omega(\mathrm{Q}(f_T))$ time. In the following, we will prove that $\mathrm{Q}(f_T) = \Omega(\sqrt{T \log T})$.

Adversary bound

We will use the relational version of the quantum adversary bound [1]. Let $X \subseteq f_T^{-1}(0)$ and $Y \subseteq f_T^{-1}(1)$ and $R : X \times Y \to \mathbb{R}_{\geq 0}$ be a weight function. For any input $x \in X$, define $w(x) = \sum_{y\in Y} R(x, y)$ and for any $i \in [n]$, $j \in [m]$, define $w(x, i, j) = \sum_{y\in Y, x_{i,j}\neq y_{i,j}} R(x, y)$. Similarly define $w(y)$ and $w(y, i, j)$. Then

$$\mathrm{Q}(f_T) = \Omega\left(\min_{\substack{x\in x, y\in Y \\ i\in[n], j\in[m] \\ R(x,y)>0 \\ x_{i,j}\neq y_{i,j}}} \sqrt{\frac{w(x)w(y)}{w(x,i,j)w(y,i,j)}}\right).$$

Input sets

Here we define the subsets of inputs X and Y. First, let k be the smallest positive integer such that $T \leq 2^k k$ and k is a multiple of 4. Denote $d = 2^k$, then $k = \log_2 d$ and $T = \Theta(d \log d)$. An input z from either X or Y must then satisfy the following conditions.

- for each $p \in \left[0, \frac{k}{2}\right]$, there are exactly $\frac{d}{2^p}$ indices i such that $j_{z,i} \in [2^p, 2^{p+1})$; we will call the set of such indices the *p-th block* of z;
- moreover, for each p and each $\ell \in [0, 2^p)$, there are exactly $\frac{d}{2^{2p}}$ indices i such that $j_{z,i} = 2^p + \ell$.

Consequently, we examine inputs with $n = 2^k + 2^{k-1} + \ldots 2^{\frac{k}{2}}$ and $m = 2^{\frac{k}{2}+1} - 1$. Additionally, an input y belongs to Y only if there is a unique i such that $y_{i,j_{y,i}} = 1$. For this i, we also require $j_{y,i} \geq 2^{\frac{k}{4}+1}$: equivalently this means that i belongs to a block with $p > \frac{k}{4}$.

We verify the value of $T' = \sum_{i\in[n]} t_i^2$ for these inputs. If i belongs to the p-th block of an input z, then $j_{z,i} = \Theta(2^p)$, as $j_{z,i} \in [2^p, 2^{p+1})$. Then

$$T' = \sum_{i\in[n]} t_i^2 = \sum_{i\in[n]} j_{z,i} = \sum_{p\in\left[0,\frac{k}{2}\right]} \frac{d}{2^p} \cdot 2^p = d\left(\frac{k}{2} + 1\right) = \Theta(T).$$

Note that since $\mathrm{Q}(f_{T'}) \leq \mathrm{Q}(f_T)$, a lower bound on $\mathrm{Q}(f_{T'})$ in terms of T will also give us a lower bound on $\mathrm{Q}(f_T)$. In the remainder of the proof, we will thus lower bound $\mathrm{Q}(f_{T'})$.

Relation

For an index $i \in [n]$ of an input z that belongs to the p-th block, we define an *index weight* $W_{z,i} = 2^p$. Then we also define values

- $B_p = \frac{d}{2^p} \cdot 2^p = d$ is the total index weight of the p-th block;
- $J_p = \frac{d}{2^{2p}} \cdot 2^p = \frac{d}{2^p}$ is the total index weight in the p-th block for any $j_{z,i} \in [2^p, 2^{p+1})$.

Note that these values do not depend on the input.

For the relation, we will call the p-th block *light* if $p \in \left[0, \frac{k}{4}\right]$ and *heavy* if $p \in \left(\frac{k}{4}, \frac{k}{2}\right]$. Two inputs $x \in X$ and $y \in Y$ have $R(x, y) > 0$ iff:

- there are exactly two indices $i_0, i_1 \in [n]$ such that $j_{x,i_b} \neq j_{y,i_b}$;
- i_0 is from some light block p_0 and i_1 is from some heavy block p_1 of y; let $j_0 = j_{y,i_0}$ and $j_1 = j_{y,i_1}$;
- $y_{i_0,j_0} = 0$, $y_{i_1,j_1} = 1$.
- $x_{i_0,j_1} = x_{i_1,j_0} = 0$.

Then let the weight in the relation be

$$R(x,y) = W_{y,i_0} W_{y,i_1} = W_{x,i_1} W_{x,i_0} = 2^{p_0} 2^{p_1}.$$

Figure 1 illustrates the structure of the inputs and the relation.

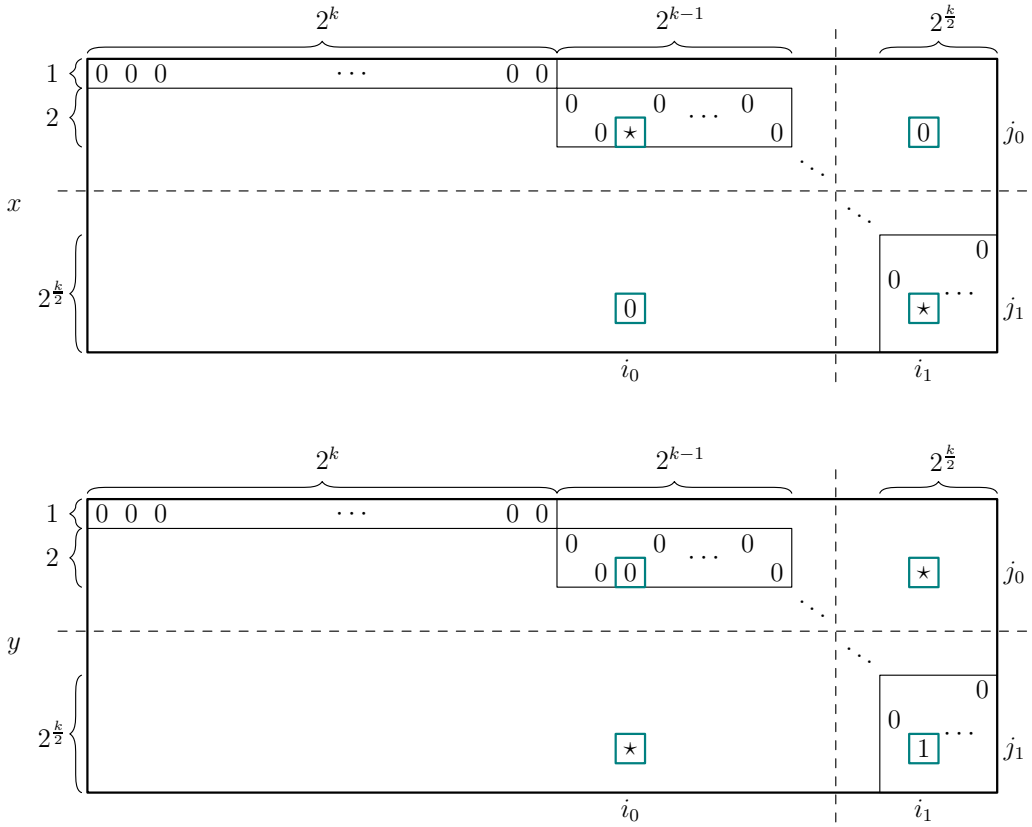


Figure 1 An example of two inputs $x \in X$ and $y \in Y$ in the relation. Inputs x and y differ only in the 4 highlighted positions. All of the empty cells contain $\star$, not shown for readability. For y, the non-$\star$ symbols of the light blocks are located in the left upper area separated by the dashed lines, while the non-$\star$ symbols of the heavy blocks are located in the lower right area. Note that for x, i_0 is in a heavy block and i_1 is in a light block.

Lower bound

Now we will calculate the values for the adversary bound. Fix two inputs $x \in X$ and $y \in Y$ with $R(x,y) > 0$. First, since for x the index i_1 can be any index from any light block and i_0 can be any index from any heavy block,

$$w(x) = \left(\sum_{p_0 \in \left[0, \frac{k}{4}\right]} B_{p_0} \right) \cdot \left(\sum_{p_1 \in \left(\frac{k}{4}, \frac{k}{2}\right]} B_{p_1} \right) = \Theta(d^2 k^2).$$

For $w(y)$, note that p_1 is uniquely determined by the position of the unique symbol 1 in y. However, the choice for i_0 is not additionally constrained, hence

$$w(y) = \left(\sum_{p_0 \in \left[0, \frac{k}{4}\right]} B_{p_0} \right) \cdot 2^{p_1} = \Theta(dk2^{p_1}).$$

Therefore, the nominator in the ratio in the adversary bound is

$$w(x)w(y) = \Theta(d^3k^3 2^{p_1}).$$

Now note the following important property: if $x_{i,j} \neq y_{i,j}$, then one of $x_{i,j}$ and $y_{i,j}$ is $\star$, and the other is either 0 or 1. There are in total exactly 4 positions (i, j) where x and y differ. We will examine each case separately.

(a) $i = i_0$, $j = j_0$. In this case $x_{i,j} = \star$ and $y_{i,j} = 0$.
For x, i_1 is not fixed but j_0 is known and hence also p_0 is known. Therefore, the total index weight from the light blocks is J_{p_0}. On the other hand, the positions of i_0 and, therefore, also p_1 are fixed. Thus,

$$w(x, i, j) = J_{p_0} \cdot 2^{p_1} = \frac{d}{2^{p_0}} \cdot 2^{p_1}.$$

For y, both i_0 and i_1 are fixed, hence

$$w(y, i, j) = 2^{p_0} \cdot 2^{p_1} < d,$$

since $p_0 + p_1 \leq \frac{k}{4} + \frac{k}{2} < k$. Overall,

$$w(x, i, j)w(y, i, j) < \frac{d}{2^{p_0}} \cdot 2^{p_1} \cdot d = \frac{d^2 \cdot 2^{p_1}}{2^{p_0}}.$$

(b) $i = i_0$, $j = j_1$. In this case $x_{i,j} = 0$ and $y_{i,j} = \star$.
For x, now the position i_0 is fixed, but i_1 can be chosen without additional constraints. The index i_0 uniquely defines the value of p_1. Hence,

$$w(x, i, j) = \left(\sum_{p_0 \in \left[0, \frac{k}{4}\right]} B_{p_0} \right) \cdot 2^{p_1} = \Theta(dk2^{p_1}).$$

For y, similarly as in the previous case, we have i_0 and i_1 fixed, thus

$$w(y, i, j) = 2^{p_0} \cdot 2^{p_1} < d.$$

Then

$$w(x, i, j)w(y, i, j) = O(dk2^{p_1} \cdot d) = O(d^2k2^{p_1}).$$

(c) $i = i_1$, $j = j_0$. In this case $x_{i,j} = 0$ and $y_{i,j} = \star$.
For x, i_1 is fixed, so it uniquely determines p_0. The index i_0 can be chosen without additional restrictions. Hence,

$$w(x, i, j) = 2^{p_0} \cdot \left(\sum_{p_1 \in \left(\frac{k}{4}, \frac{k}{2}\right]} B_{p_1} \right) = \Theta(2^{p_0} \cdot dk).$$

For y, i_0 is not fixed but j_0 is fixed, which also fixes p_0. Therefore, the total index weight from the light blocks is J_{p_0}. On the other hand, i_1 and p_1 are fixed for y by the position of the symbol 1, thus

$$w(y,i,j) = J_{p_0} \cdot 2^{p_1} = \frac{d}{2^{p_0}} \cdot 2^{p_1}.$$

Their product is

$$w(x,i,j)w(y,i,j) = \Theta\left(2^{p_0} \cdot dk \cdot \frac{d}{2^{p_0}} \cdot 2^{p_1}\right) = \Theta(d^2k2^{p_1}).$$

(d) $i = i_1$, $j = j_1$. In this case $x_{i,j} = \star$ and $y_{i,j} = 1$.
For x, i_1 is fixed, hence p_0 is also fixed; i_0 is not fixed, but $j_1 = j$ and p_1 is uniquely defined. Hence,

$$w(x,i,j) = 2^{p_0} \cdot J_{p_1} = 2^{p_0} \cdot \frac{d}{2^{p_1}}.$$

For y, the position of the symbol 1 must necessarily change, hence

$$w(y,i,j) = w(y) = \Theta(dk2^{p_1}).$$

The product then is

$$w(x,i,j)w(y,i,j) = \Theta\left(2^{p_0} \cdot \frac{d}{2^{p_1}} \cdot dk2^{p_1}\right) = \Theta(d^2k2^{p_0}) = O(d^2k2^{p_1}),$$

as $p_0 \le \frac{k}{4} < p_1$.

We can see that in all cases the denominator in the ratio of the adversary bound is $O(d^2k2^{p_1})$. Therefore,

$$\frac{w(x)w(y)}{w(x,i,j)w(y,i,j)} = \Omega\left(\frac{d^3k^32^{p_1}}{d^2k2^{p_1}}\right) = \Omega(dk^2) = \Omega(d\log^2 d)$$

and since $\log T = \Theta(\log(d\log d)) = \Theta(\log d + \log\log d) = \Theta(\log d)$, we have

$$\mathrm{Q}(f_T) \ge \mathrm{Q}(f_{T'}) = \Omega\left(\sqrt{d\log^2 d}\right) = \Omega\left(\sqrt{T\log T}\right). \qquad \blacktriangleleft$$

5 Conclusion

In this paper, we developed a new quantum algorithm and a new quantum lower bound for variable time search. Our quantum algorithm has complexity $O(\sqrt{T}\log n)$, compared to $O(\sqrt{T}\log^{1.5} T)$ for the best previously known algorithm (quantum variable time amplitude amplification [3] instantiated to the case of search). It also has the advantage of being simpler than previous quantum algorithms for variable time search. If the recursive structure is unrolled, our algorithm consists of checking algorithms C_{T_i} for various times T_i interleaved with Grover diffusion steps. Thus, the structure is the essentially same as for regular search and the main difference is that C_{T_i} for different i are substituted at different query steps.

We note that our algorithm has a stronger assumption about T: we assume that an upper bound estimate $T \ge \sum_{i=1}^n t_i^2$ is provided as an input to the algorithm and the complexity depends on this estimate T, rather than the actual $\sum_{i=1}^n t_i^2$. Possibly, this assumption can be removed by a doubling strategy that tries values of T that keep increasing by a factor of 2 but the details remain to be worked out.

Our quantum lower bound is $\Omega(\sqrt{T \log T})$ which improves over the previously known $\Omega(\sqrt{T})$ lower bound. This shows that variable time search for the "unknown times" case (when the times $t_1, \ldots, t_n$ are not known in advance and cannot be used to design the quantum algorithm) is more difficult than for the "known times" case (which can be solved with complexity $\Theta(\sqrt{T})$).

A gap between the upper and lower bounds remains but is now just a factor of $\sqrt{\log T}$. Possibly, this is due to the lower bound using a set of inputs for which an approximate distribution of values t_i is fixed. In such a case, the problem may be easier than in the general case, as an approximately fixed distribution of t_i can be used for algorithm design.

References

1 Scott Aaronson. Lower bounds for local search by quantum arguments. *SIAM Journal on Computing*, 35(4):804–824, 2006. arXiv:0307149. doi:10.1137/S0097539704447237.

2 Andris Ambainis. Quantum search with variable times. *Theory of Computing Systems*, 47(3):786–807, 2010. arXiv:0609168. doi:10.1007/s00224-009-9219-1.

3 Andris Ambainis. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In Christoph Dürr and Thomas Wilke, editors, *29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012)*, volume 14 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 636–647. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012. arXiv:1010.4458. doi:10.4230/LIPIcs.STACS.2012.636.

4 Aleksandrs Belovs, Gilles Brassard, Peter Høyer, Marc Kaplan, Sophie Laplante, and Louis Salvail. Provably secure key establishment against quantum adversaries. In Mark M. Wilde, editor, *12th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2017)*, volume 73 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. arXiv:1704.08182. doi:10.4230/LIPIcs.TQC.2017.3.

5 Gilles Brassard. Searching a quantum phone book. *Science*, 275(5300):627–628, 1997. doi:10.1126/science.275.5300.627.

6 Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. arXiv:0005055. doi:10.1090/conm/305/05215.

7 Sourav Chakraborty, Arkadev Chattopadhyay, Peter Høyer, Nikhil S. Mande, Manaswi Paraashar, and Ronald de Wolf. Symmetry and quantum query-to-communication simulation. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science (STACS 2022)*, volume 219 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:23, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. arXiv:2012.05233. doi:10.4230/LIPIcs.STACS.2022.20.

8 Andrew M. Childs, Robin Kothari, and Rolando D. Somma. Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. *SIAM Journal on Computing*, 46(6):1920–1950, 2017. arXiv:1511.02306. doi:10.1137/16M1087072.

9 Arjan Cornelissen, Stacey Jeffery, Maris Ozols, and Alvaro Piedrafita. Span programs and quantum time complexity. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. arXiv:2005.01323. doi:10.4230/LIPIcs.MFCS.2020.26.

10 Koen de Boer, Léo Ducas, Stacey Jeffery, and Ronald de Wolf. Attacks on the AJPS Mersenne-based cryptosystem. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography*, pages 101–120, Cham, 2018. Springer International Publishing. Preprint: https://eprint.iacr.org/2017/1171. doi:10.1007/978-3-319-79063-3_5.

11 Adam Glos, Martins Kokainis, Ryuhei Mori, and Jevgēnijs Vihrovs. Quantum speedups for dynamic programming on n-dimensional lattice graphs. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*, volume 202 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 50:1–50:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. arXiv:2104.14384. doi:10.4230/LIPIcs.MFCS.2021.50.

12 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103:150502, 2009. arXiv:0811.3171. doi:10.1103/PhysRevLett.103.150502.

13 Peter Høyer, Michele Mosca, and Ronald de Wolf. Quantum search on bounded-error inputs. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming*, pages 291–299, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. arXiv:0304052. doi:10.1007/3-540-45061-0_25.

14 Stacey Jeffery. Quantum subroutine composition, 2022. arXiv:2209.14146.

15 François Le Gall. Improved quantum algorithm for triangle finding via combinatorial arguments. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 216–225. IEEE, 2014. arXiv:1407.0085. doi:10.1109/FOCS.2014.31.

16 André Schrottenloher and Marc Stevens. A quantum analysis of nested search problems with applications in cryptanalysis. Cryptology ePrint Archive, Paper 2022/761, 2022. URL: https://eprint.iacr.org/2022/761.

A Proofs of Lemmas 3.2 and 3.3

▶ **Lemma 3.2.** *For all $j = 1, 2, \ldots, \ell$,*

$$\sin^2 \alpha_j = \frac{s_j}{n} \prod_{k=1}^{j-1} \left(\frac{\sin(3\alpha_k)}{\sin \alpha_k} \right)^2 . \tag{1}$$

Moreover, for any $i \in \mathcal{S}_j$, the amplitude at $|i, 1\rangle$ (or $|i, *\rangle$, if $t_i > T_j$) equals $\frac{\sin \alpha_j}{\sqrt{s_j}}$.

Proof. For each j express the final state of A_j in the canonical basis as

$$\sum_{i=1}^{n} \beta_{ij} |i, a_{ij}\rangle ,$$

where $a_{ij} \in \{0, 1, *\}$ and $a_{ij} = 0$ iff $x_i = 0$ and $t_i \leq T_j$ (i.e., iff $i \notin \mathcal{S}_j$). Initially, $\beta_{i0} = n^{-1/2}$ for all i. Then

$$\sin^2 \alpha_j = \sum_{i \in \mathcal{S}_j} |\beta_{ij}|^2 ,$$

for all j. To see how the amplitude $\beta_{i(j+1)}$ is related to β_{ij}, consider how the state evolves under A_{j+1}:

- the final state of A_j is

  $$\sum_{i \in [n] \setminus \mathcal{S}_j} \beta_{ij} |i, 0\rangle + \sum_{i \in \mathcal{S}_j} \beta_{ij} |i, a_{ij}\rangle ,$$

  by the definition of β_{ij}; moreover, $a_{ij} \in \{1, *\}$ for all $i \in \mathcal{S}_j$.
- Amplitude amplification $A_j(1)$ results in the state

  $$\sum_{i \in [n] \setminus \mathcal{S}_j} \frac{\cos(3\alpha_j)}{\cos \alpha_j} \beta_{ij} |i, 0\rangle + \sum_{i \in \mathcal{S}_j} \frac{\sin(3\alpha_j)}{\sin \alpha_j} \beta_{ij} |i, a_{ij}\rangle .$$

- An application of $C_{T_{j+1}}$ transforms this state to

$$\sum_{i\in[n]\setminus\mathcal{S}_j}\frac{\cos(3\alpha_j)}{\cos\alpha_j}\beta_{ij}\left|i,0\right\rangle+\sum_{i\in\mathcal{S}_j\setminus\mathcal{S}_{j+1}}\frac{\sin(3\alpha_j)}{\sin\alpha_j}\beta_{ij}\left|i,0\right\rangle+\sum_{i\in\mathcal{S}_{j+1}}\frac{\sin(3\alpha_j)}{\sin\alpha_j}\beta_{ij}\left|i,a_{i(j+1)}\right\rangle.$$

  We conclude that

$$\beta_{i(j+1)}=\begin{cases}\beta_{ij}\frac{\sin(3\alpha_j)}{\sin\alpha_j}, & i\in\mathcal{S}_j,\\ \beta_{ij}\frac{\cos(3\alpha_j)}{\cos\alpha_j}, & i\in[n]\setminus\mathcal{S}_j.\end{cases}\tag{5}$$

In particular, for any $j\in[\ell]$ and $i\in\mathcal{S}_j$ we have

$$\beta_{ij}=\frac{1}{\sqrt{n}}\prod_{k=1}^{j-1}\frac{\sin(3\alpha_k)}{\sin\alpha_k},$$

since each such i is in $\mathcal{S}_k$, $k\le j-1$, thus, by (5), the respective amplitude gets multiplied by $\frac{\sin(3\alpha_k)}{\sin\alpha_k}$ at each step. This establishes the second part of the lemma (that the amplitudes β_{ij} are all equal for any $i\in\mathcal{S}_j$). For the first part, we arrive at

$$\sin^2\alpha_j=\sum_{i\in\mathcal{S}_j}|\beta_{ij}|^2=\sum_{i\in\mathcal{S}_j}\prod_{k=1}^{j-1}\left(\frac{\sin(3\alpha_k)}{\sin\alpha_k}\right)^2=\frac{s_j}{n}\prod_{k=1}^{j-1}\left(\frac{\sin(3\alpha_k)}{\sin\alpha_k}\right)^2.$$ ◀

▶ **Lemma 3.3.** *The following claims hold:*

C-1 *Each factor on the RHS of* (3) *is positive:* $\frac{9^j s_j}{n}\le\frac{9}{4}$, *thus*

$$\left(1-\frac{4s_j}{27n}\cdot 9^j\right)\ge\frac{2}{3},\quad \textit{for all } j\in[\ell-1].$$

C-2 *The product* $\prod_{j=1}^{\ell-1}\left(1-\frac{4s_j}{27n}\cdot 9^j\right)$ *is lower bounded by* $2/3$.

C-3 $9^\ell s_\ell\ge 9^\ell s_d\ge n$.

Proof. We will prove the following inequality:

$$\sum_{j=1}^{\ell-1}s_j9^j<\frac{9n}{4}.\tag{6}$$

Then **C-1** will immediately follow, since each term on (6) is nonnegative. Furthermore, also **C-2** follows from (6) via the generalized Bernoulli's inequality:

$$\prod_{j=1}^{\ell-1}\left(1-\frac{4s_j}{27n}\cdot 9^j\right)\ge 1-\frac{4}{27n}\sum_{j=1}^{\ell-1}s_j9^j\ge 1-\frac{4}{27n}\cdot\frac{9n}{4}=\frac{2}{3}.$$

First we observe that

$$\sum_{j=1}^{d}\sum_{i\in\mathcal{S}_{j-1}\setminus\mathcal{S}_j}t_i^2=\sum_{i\in[n]\setminus\mathcal{M}}t_i^2<\sum_{i\in[n]}t_i^2\le T.$$

Notice that each set difference $\mathcal{S}_{j-1}\setminus\mathcal{S}_j$ can be characterized as follows:

$$\mathcal{S}_{j-1}\setminus\mathcal{S}_j=\{i\in[n]\ :\ (T_{j-1}<t_i\le T_j)\wedge x_i=0\}.$$

Therefore all t_i^2 s.t. $i \in \mathcal{S}_{j-1} \setminus \mathcal{S}_j$ satisfy the bound

$$t_i^2 \geq T_{j-1}^2 = \begin{cases} \frac{9^{j-1}T}{n}, & j > 1, \\ 0, & j = 1. \end{cases}$$

Thus we obtain the following inequality:

$$\frac{T}{n} \sum_{j=2}^{d} 9^{j-1} \left|\mathcal{S}_{j-1} \setminus \mathcal{S}_j\right| < \sum_{j=1}^{d} \sum_{i \in \mathcal{S}_{j-1} \setminus \mathcal{S}_j} t_i^2 < T$$

or

$$\sum_{k=1}^{d-1} 9^k \left(s_k - s_{k+1}\right) < n. \tag{7}$$

We also expand $9^\ell s_\ell$ as follows, taking into account $s_d = m$:

$$9^\ell s_\ell = 9^\ell \left(s_\ell - s_{\ell+1}\right) + \frac{1}{9} \cdot 9^{\ell+1} \left(s_{\ell+1} - s_{\ell+2}\right) + \ldots + \frac{1}{9^{d-1-\ell}} \cdot 9^{d-1} \left(s_{d-1} - s_d\right) + 9^\ell m.$$

From this equality, taking into account $s_k - s_{k+1} \geq 0$, we can upper bound $9^\ell s_\ell$ as

$$9^\ell s_\ell \leq \sum_{k=\ell}^{d-1} 9^k \left(s_k - s_{k+1}\right) + 9^\ell m \tag{8}$$

Rewrite (7) as

$$s_1 + \frac{8}{9} \sum_{k=1}^{\ell-1} 9^k s_k - 9^{\ell-1} s_\ell + \sum_{k=\ell}^{d-1} 9^k \left(s_k - s_{k+1}\right) < n$$

and apply (8) to obtain

$$s_1 + \frac{8}{9} \sum_{k=1}^{\ell-1} 9^k s_k + \sum_{k=\ell}^{d-1} 9^k \left(s_k - s_{k+1}\right) < n + 9^{\ell-1} s_\ell \leq n + \frac{1}{9} \sum_{k=\ell}^{d-1} 9^k \left(s_k - s_{k+1}\right) + 9^{\ell-1} m$$

$$\frac{8}{9} \sum_{k=1}^{\ell-1} 9^k s_k + \frac{8}{9} \sum_{k=\ell}^{d-1} 9^k \left(s_k - s_{k+1}\right) < n - s_1 + 9^{\ell-1} m$$

$$8 \sum_{k=1}^{\ell-1} 9^k s_k < 9n - 9s_1 + 9^\ell m < 9n + 9^\ell m.$$

By the choice of ℓ we have $9^{\ell-1} \leq \frac{n}{m}$, therefore we arrive at

$$8 \sum_{k=1}^{\ell-1} 9^k s_k < 9n + 9\frac{n}{m} \cdot m = 18n,$$

which is equivalent to (6).

Finally, to show **C-3**, we recall that $s_\ell \geq s_d = m$. Again by the choice of ℓ, $9^\ell \geq \frac{n}{m}$. Consequently,

$$9^\ell s_\ell \geq \frac{n}{m} \cdot m = n,$$

as claimed. ◀

Fully Device-Independent Quantum Key Distribution Using Synchronous Correlations

Nishant Rodrigues
Department of Computer Science, University of Maryland, College Park, MD, USA
Joint Center for Quantum Information and Computer Science, College Park, MD, USA

Brad Lackey
Microsoft Quantum, Redmond, WA, USA

Abstract

We derive a device-independent quantum key distribution protocol based on synchronous correlations and their Bell inequalities. This protocol offers several advantages over other device-independent schemes including symmetry between the two users and no need for pre-shared randomness. We close a "synchronicity" loophole by showing that an almost synchronous correlation inherits the self-testing property of the associated synchronous correlation. We also pose a new security assumption that closes the "locality" (or "causality") loophole: an unbounded adversary with even a small uncertainty about the users' choice of measurement bases cannot produce any almost synchronous correlation that approximately maximally violates a synchronous Bell inequality.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols; Hardware → Quantum communication and cryptography; Theory of computation → Quantum information theory

Keywords and phrases quantum cryptography, device independence, key distribution, security proofs, randomness

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.8

1 Introduction

Quantum key distribution (QKD) allows two parties to establish a shared classical secret key using quantum resources. The two main requirements of QKD are (1) Correctness: the two parties, Alice and Bob, get the same key; and (2) Security: an adversary Eve gets negligible information about the key. Device-independent quantum key distribution (DI-QKD) is entanglement-based, and aims to prove security of QKD based solely on the correctness of quantum mechanics, separation of devices used by the two parties, and passing of statistical tests known as Bell violations [23, 16]. These protocols are usually specified by a non-local game, characterized by a conditional probability distribution or *correlation* $p(y_A, y_B \,|\, x_A, x_B)$. Intuitively, Alice and Bob obtain or generate random inputs x_A and x_B respectively, and the correlation describes the likelihood their entangled quantum devices return outputs y_A and y_B to each respectively. We will be interested in symmetric correlations and so will take $x_A, x_B \in X$ and $y_A, y_B \in Y$ where X and Y are finite sets; for our protocol specifically $X = \{0, 1, 2\}$ and $Y = \{0, 1\}$.

In general, security of a DI-QKD scheme relies on the monogamy of entanglement. The key result is that maximally entangled quantum states are separable within any larger quantum system. In cryptographic terms, if Alice and Bob share a maximally entangled state then the results of measurements they make on this state will be uncorrelated to any other measurement results an adversary can perform. Hence, presuming the correctness of quantum mechanics, no adversary can have any information about key bits Alice and Bob may generate through this process. Generally, a DI-QKD protocol will involve two types of

rounds: testing rounds where Alice and Bob (publicly) share their inputs and output results for performing statistics tests, and data rounds where they obtain shared secret bits. The goal of the testing rounds is to produce a certificate that Alice and Bob are operating on maximally entangled states.

Most current DI-QKD schemes are based on the CHSH inequality, a linear inequality in $p(y_A, y_B \,|\, x_A, x_B)$, which if satisfied characterizes classical statistics within a quantum system. Hence a violation of this inequality is a certificate of quantum behavior. This inequality exhibits "rigidity" in that the only quantum state that produces a maximal violation of the inequality is (up to natural equivalences) a Bell pair. Thus the goal of the testing rounds in a DI-QKD protocol is to statistically verify that the system produces a maximal violation of the CHSH inequality.

In a non-local game Alice and Bob may preshare an entangled resource in each round, but are not allowed any communication between receiving or generating their inputs x_A and x_B and measuring the system to obtain their outputs y_A and y_B. This is typically called a "nonsignaling" condition, leading to *nonsignaling correlations* which include all quantum strategies. If (even classical) communication between Alice and Bob is possible, then it is simple to classically simulate a correlation that produces a maximal violation of the CHSH inequality, and hence any certificates of quantumness or entanglement are void [22]. This *locality* or *causality loophole* in the security proof is challenging to avoid; the only known means to close it is by having Alice and Bob acausally separated during each round: bounds on the speed of light prevent such communication [11, 9, 21].

A *synchronous correlation* is one such that $p(y_A, y_B \mid x, x) = 0$ whenever $y_A \neq y_B$ and $x \in X$. That is, whenever Alice and Bob input the same value they are guaranteed to receive the same outputs, although that value may be nondeterministic. These correlations have recently become popular owing to their use in the resolution of the Connes Embedding Conjecture and Tsirl'son's Problem [12], but have also been used to generalize combinatorial properties to the quantum setting [14, 17, 13].

We present a fully device-independent QKD protocol based on synchronous correlations. This protocol is symmetric, in that roles of Alice and Bob are completely interchangeable. This is an advantage over other DI-QKD protocols based on the CHSH inequality [23] (which is neither symmetric nor synchronous) as sender versus receiver roles do not need to be negotiated. Additionally, as Alice and Bob select their inputs independently they do not need pre-shared secret bits to decide upon testing versus data rounds.

The mathematical framework needed to prove device-independent security of this protocol was laid out in [20], where four analogues of the Bell/CHSH inequality for synchronous correlations were given. In this work we focus only on one of these, $J_3(p) \geq 0$ (see Equation (3) below). As well, bounds on quantum violations of these were given ($J_3(p) \geq -\frac{1}{8}$), and rigidity of correlations that achieve a maximal violation proven. The two critical analyses needed to complete a proof of security for our DI-QKD protocol are as follows. First, we must prove that if the system is observed to be close to the maximal violation then it is close to the ideal system, which measures a Bell pair. Then, we provide an alternative security assumption that bypasses the causality loophole.

We tackle the first of these through two theorems. For context, Alice and Bob will select their inputs from $X = \{0, 1, 2\}$ and each measure a quantum system that produces a bit output from $Y = \{0, 1\}$. The ideal system, that produces $J_3(p) = -\frac{1}{8}$, involves measuring a Bell pair using three specific projection-valued measures $\{\hat{E}_y^x\}_{y=0,1}$ for $x = 0, 1, 2$ given in Equation (1) below. Any synchronous quantum correlation that achieves $J_3(p) = -\frac{1}{8}$ must

have $E_y^x = \hat{E}_y^x \otimes \mathbb{1}$, and hence the measurements have no influence on the larger system. In Section 3 we show that if we take a *synchronous* quantum system that is close to achieving maximal J_3 violation, then it must be close to the ideal system in trace norm.

Unfortunately this introduces a "synchronicity" loophole: rigidity holds among synchronous correlations, but are there asynchronous correlations with $J_3 = -\frac{1}{8}$ that cannot certify maximal entanglement? In Section 4, we close this loophole using recent work on "almost synchronous" correlations [24]. This leads to our complete DI-QKD scheme given as Algorithm 1 below, where in addition to verifying a Bell violation one also bounds the total amount of asynchronicity of the correlation, S, as defined in Equation (7).

In Section 5 we use the Entropy Accumulation Theorem (EAT) [8] to bound the the min-entropy of the outputs given an adversary's side-information. This allows us to derive the key rate of Algorithm 1.

Finally, in Section 6, we pose a new security assumption to close the *causality* or *locality* loophole: the adversary Eve may have unlimited communication and computational power, yet she has imperfect knowledge of Alice and Bob's inputs. Informally, given nonnegative values $\lambda \leq \frac{1}{8}$ and $\mu \leq 1$ there exists a bound ϵ_{max} such that if Eve's uncertainty about Alice and Bob's inputs is greater than ϵ_{max} then there is no device she can create where Alice and Bob's expected Bell violation J_3 and asynchronicity S satisfy $-\frac{1}{8} \leq J_3 \leq -\frac{1}{8} + \lambda$ and $0 \leq S \leq \mu$.

2 Preliminaries

We present some definitions that will be used in the protocol later. Like other device-independent schemes, our protocol is expressed in terms of a nonlocal game, which is characterized by a conditional probability distribution (or correlation) $p(y_A, y_B|x_A, x_B)$ where $x_A, x_B \in X$ and $y_A, y_B \in Y$ are from finite sets X and Y. By a nonlocal game we mean the players Alice and Bob will receive inputs $x_A, x_B \in X$ from a referee and will produce outputs $y_A, y_B \in Y$. These are then adjudicated by the referee against some criterion, synchronicity in our case. Alice and Bob are not allowed to communicate once they receive their inputs, which is characterized by the famous nonsignaling conditions on the correlation [19, 6].

▶ **Definition 1.** *A correlation is* synchronous *if* $p(y_A, y_B \mid x, x) = 0$ *whenever* $x \in X$ *and* $y_A \neq y_B \in Y$. *A correlation is* symmetric *if* $p(y_A, y_B \mid x_A, x_B) = p(y_B, y_A \mid x_B, x_A)$.

Unlike nonlocal games such as the CHSH or Magic Square games, or their generalizations [15, 18, 6, 3, 7], it is straightforward for Alice and Bob to create a perfect winning strategy for synchronicity. Prior to the games they agree on some function $f : X \to Y$, then regardless of how the referee selects $x_A, x_B \in X$, they output $y_A = f(x_A)$ and $y_B = f(x_B)$. Hence the "value" of any synchronous game (Alice's and Bob's expected success probability) is always 1, and so value plays no role in the following.

The analysis of nonlocal games relies on understanding the set of local (or "classical" or "hidden variables") correlations like the one above within the set of quantum correlations. While a general quantum correlation has the form $p(y_A, y_B \mid x_A, x_B) = \mathrm{tr}(\rho(E_{y_A}^{x_A} \otimes F_{y_B}^{x_B}))$ for a density operator ρ and sets of positive operator-valued measures $\{E_y^x\}_{y \in Y}$ and $\{F_y^x\}_{y \in Y}$ on Hilbert spaces $\mathfrak{H}_A$ and $\mathfrak{H}_B$, a synchronous quantum correlation is always a convex combination of so-called "tracial" states [17, 20] of the form $p(y_A, y_B \mid x_A, x_B) = \frac{1}{d}\mathrm{tr}(E_{y_A}^{x_A} E_{y_B}^{x_B})$ where $d = \dim \mathfrak{H}_A$.

For input and output $X = \{0, 1, 2\}$ and $Y = \{0, 1\}$, respectively, there are four Bell inequalities for synchronous hidden variables theories. That is, the synchronous classical correlations (among general nonsignaling synchronous correlations) are characterized by four

inequalities $J_0, J_1, J_2, J_3 \geq 0$ where each $J_i = J_i(p)$ is a linear combination of the correlation components $p(y_A, y_B \mid x_A, x_B)$. For this work, we will focus only on one of these as given in Equation (3) below (see also [20]).

Synchronous quantum correlations can violate the inequality $J_3 \geq 0$. However one can show an analogue of Tsirl'son bound, in that any synchronous quantum correlation must have $J_3 \geq -\frac{1}{8}$. This follows from Equation (6) below. Of particular interest are correlations that maximize this quantum violation. Like CHSH or Magic Square games, one can show a rigidity result: there is a unique synchronous quantum correlation with $J_3 = -\frac{1}{8}$, which involves a maximally entangled state shared between Alice and Bob. One can then use principal decompositions, or two projections theory, to convert this into a self-test for certifying a single EPR pair, the basis for device-independence.

We denote the binary entropy function by $h(p) = -p\log(p) - (1-p)\log(1-p)$ for $p \in [0,1]$. The von Neumann entropy of a quantum state ρ is given by $H(\rho) = -\mathrm{tr}(\rho\log(\rho))$. Given two operators ρ_1 and ρ_2, we say $\rho_1 \geq \rho_2$ if $\rho_1 - \rho_2 \geq 0$.

▶ **Definition 2.** *For a bipartite quantum state $\rho_{AB} \in \mathfrak{H}_A \otimes \mathfrak{H}_B$, the min-entropy of A conditioned on B is:*

$$H_{\min}(A \mid B)_{\rho_{AB}} = \max\{s \in \mathbb{R} : \exists \sigma_B \in \mathcal{D}(\mathfrak{H}_B) \text{ such that } 2^{-s}\mathrm{id}_A \otimes \sigma_B \geq \rho_{AB}\}$$

where $\mathcal{D}(\mathfrak{H}_B)$ is the set of density operators in $\mathfrak{H}_B$.

The ϵ-smooth version of the conditional min-entropy considers states that are ϵ-close to ρ_{AB}. The notion of closeness that is typically used is the purified distance $P(\rho,\sigma) = \sqrt{1 - F(\rho,\sigma)^2}$, where $F(\rho,\sigma)$ is the fidelity between states ρ and σ.

▶ **Definition 3.** *For a bipartite quantum state $\rho_{AB} \in \mathfrak{H}$, the ϵ-smooth min-entropy of A conditioned on B is defined as:*

$$H^{\epsilon}_{\min}(A \mid B)_{\rho_{AB}} = \max_{\substack{\tilde{\rho}_{AB} \in \mathcal{S}(\mathfrak{H}) \\ P(\rho_{AB}, \tilde{\rho}_{AB}) \leq \epsilon}} H_{\min}(A \mid B)_{\tilde{\rho}_{AB}}$$

The quantum ϵ-smooth max-entropy is defined as:

$$H^{\epsilon}_{\max}(A \mid B)_{\rho_{AB}} = \log \inf_{\substack{\tilde{\rho}_{AB} \in \mathcal{S}(\mathfrak{H}) \\ P(\rho_{AB}, \tilde{\rho}_{AB}) \leq \epsilon}} \sup_{\sigma_B} \left|\left| \tilde{\rho}_{AB}^{\frac{1}{2}} \sigma_B^{-\frac{1}{2}} \right|\right|_1^2 .$$

where $\mathcal{S}(\mathfrak{H})$ is the set of sub-normalized states in $\mathfrak{H}$ and $||A||_\alpha = \mathrm{tr}\left(\left(\sqrt{A^\dagger A}\right)^\alpha\right)^{\frac{1}{\alpha}}$.

3 A synchronous DI-QKD protocol

We present a synchronous device-independent quantum key distribution protocol that is symmetric with respect to Alice and Bob, each party performing the same tasks.

Suppose Alice and Bob share an EPR pair. Each draws $x_A, x_B \in X = \{0,1,2\}$ respectively, and measures according to $\{\hat{E}_y^{x_A}\}_{y\in Y}$ and $\{\hat{E}_y^{x_B}\}_{y\in Y}$ to get outputs $y_A, y_B \in Y = \{0,1\}$, where the projection-valued measures $\{\hat{E}_y^x\}_{y\in\{0,1\}}$ for $x \in \{0,1,2\}$ are:

$$\begin{aligned}
&\hat{E}_1^0 = |\phi_0\rangle\langle\phi_0|,\ \hat{E}_0^0 = \mathbb{1} - \hat{E}_1^0, \quad \text{where } |\phi_0\rangle = |1\rangle \\
&\hat{E}_1^1 = |\phi_1\rangle\langle\phi_1|,\ \hat{E}_0^1 = \mathbb{1} - \hat{E}_1^1, \quad \text{where } |\phi_1\rangle = \tfrac{\sqrt{3}}{2}|0\rangle + \tfrac{1}{2}|1\rangle \\
&\hat{E}_1^2 = |\phi_2\rangle\langle\phi_2|,\ \hat{E}_0^2 = \mathbb{1} - \hat{E}_1^2, \quad \text{where } |\phi_2\rangle = \tfrac{\sqrt{3}}{2}|0\rangle - \tfrac{1}{2}|1\rangle
\end{aligned} \tag{1}$$

The likelihood of Alice's and Bob's results are characterized by the correlation [20]

$$p(y_A, y_B \mid x_A, x_B) = \frac{1}{2}\mathrm{tr}(\hat{E}_{y_A}^{x_A}\hat{E}_{y_B}^{x_B}).$$

In particular, this strategy produces a synchronous quantum correlation with correlation matrix:

$$[p(y_A, y_B|x_A, x_B)] = \begin{array}{c} \\ \frac{1}{8} \end{array} \begin{array}{c} \begin{array}{ccccccccc} (0,0) & (0,1) & (0,2) & (1,0) & (1,1) & (1,2) & (2,0) & (2,1) & (2,2) \end{array} \\ \begin{pmatrix} 4 & 1 & 1 & 1 & 4 & 1 & 1 & 1 & 4 \\ 0 & 3 & 3 & 3 & 0 & 3 & 3 & 3 & 0 \\ 0 & 3 & 3 & 3 & 0 & 3 & 3 & 3 & 0 \\ 4 & 1 & 1 & 1 & 4 & 1 & 1 & 1 & 4 \end{pmatrix} \end{array} \begin{array}{c} (0,0) \\ (0,1) \\ (1,0) \\ (1,1) \end{array} \tag{2}$$

One can verify this correlation yields a maximal violation of the Bell inequality, $J_3 = -\frac{1}{8}$, where

$$\begin{aligned} J_3 \quad = \quad & 1 - \tfrac{1}{4}\big(p(0,1 \mid 0,1) + p(1,0 \mid 0,1) + p(0,1 \mid 1,0) + p(1,0 \mid 1,0) \\ & + p(0,1 \mid 0,2) + p(1,0 \mid 0,2) + p(0,1 \mid 2,0) + p(1,0 \mid 2,0) \\ & + p(0,1 \mid 1,2) + p(1,0 \mid 1,2) + p(0,1 \mid 2,1) + p(1,0 \mid 2,1) \big). \end{aligned} \tag{3}$$

This correlation is rigid in that any synchronous quantum correlation that achieves $J_3 = -\frac{1}{8}$ must have implemented the strategy above. This follows from our Theorem 4 below. In particular, this maximal violation of J_3 is a self-test of the device to detect interference from adversary: Alice and Bob can certify that their devices hold maximally entangled pairs, and by monogamy of entanglement can establish that Eve doesn't have any information about their inputs.

Our protocol extends the above scenario to n rounds. It is important to note that the observable for our synchronous Bell inequality (3) only involves correlations where Alice and Bob use different inputs. Critically, neither Alice nor Bob must pre-select which rounds will used for testing versus key generation. Upon revealing their choices of bases, testing rounds are given by those where they selected different bases and key generation rounds where they selected the same basis. In particular, they need not have any pre-shared randomness.

Of course no physical device adheres to a theoretical model perfectly, so in practice one still must perform standard information reconciliation and privacy amplification on the results.

Once the n rounds of the protocol are over, Alice and Bob communicate their basis selection over an authenticated classical channel. When they chose different bases (i.e. $x_A \neq x_B$), they exchange their measurement outcomes and use those to compute J_3. If the value of J_3 deviates too much from $-\frac{1}{8}$, they abort. The protocol is synchronous, therefore $y_A = y_B$ whenever $x_A = x_B$ and those can be used as the raw key bits for further standard privacy amplification and information reconciliation.

Our first main result is our technical rigidity statement that synchronous quantum correlations near $J_3 = -\frac{1}{8}$ have the desired security. Informally, after splitting off a space $\mathfrak{L}$ of small relative dimension, the correlation's projections are near (in trace norm) the ideal one, which separates Alice and Bob performing the perfect protocol on $\mathbb{C}^2$, and Eve and all other parties receiving no information having measurement outcomes from $\mathbb{1}_{\mathfrak{K}}$.

▶ **Theorem 4.** *Let $p(y_A, y_B \mid x_A, x_B) = \frac{1}{d}\mathrm{tr}(E_{y_A}^{x_A} E_{y_B}^{x_B})$ be a synchronous quantum correlation with maximally entangled state, where $\{E_y^x\}$ is a projection-valued measure on a d-dimensional Hilbert space $\mathfrak{H}$. Suppose $J_3(p) \leq -\frac{1}{8} + \lambda$. Then on $\mathfrak{H} = \mathfrak{L} \oplus (\mathbb{C}^2 \otimes \mathfrak{K})$ there exists a projection-value measure $\{\tilde{E}_y^x\}$ where (1) $\tilde{E}_y^x = L_y^x + \hat{E}_y^x \otimes \mathbb{1}_{\mathfrak{K}}$, (2) $\frac{\dim \mathfrak{L}}{\dim \mathfrak{H}} \leq 8\lambda$, and (3) $\frac{1}{3}\sum_{x,y} \frac{1}{d}\mathrm{tr}\left(\left(E_y^x - \tilde{E}_y^x\right)^2\right) \leq 8\lambda$. In particular, the expected statistical difference*

$$\frac{1}{3}\sum_{x,y}\left|p(y, y \mid x, x) - \frac{1}{2}\right| \leq \frac{1}{3}\left(\sqrt{8}\sqrt{\lambda} + 32\lambda\right).$$

Proof. We begin by defining the ± 1-valued observables $M_x = E_0^x - E_1^x$, so $M_x^2 = \mathbb{1}$, and following customary notation write

$$a_x = \frac{1}{d}\mathrm{tr}(M_x) \text{ and } c_{x_A x_B} = \frac{1}{d}\mathrm{tr}(M_{x_A} M_{x_B}).$$

Similarly denote $\tilde{M}_x = \tilde{E}_0^x - \tilde{E}_1^x$. Notice $E_0^x = \frac{1}{2}(\mathbb{1} + M_x)$ and $E_1^x = \frac{1}{2}(\mathbb{1} - M_x)$ so

$$\frac{1}{3}\sum_{x,y}\frac{1}{d}\mathrm{tr}\left(\left(E_y^x - \tilde{E}_y^x\right)^2\right) = \frac{1}{6}\sum_x \frac{1}{d}\mathrm{tr}\left(\left(M_x - \tilde{M}_x\right)^2\right).$$

Now define $\Delta := M_0 + M_1 + M_2$, and compute

$$\begin{aligned}
\Delta^2 &= M_0^2 + M_1^2 + M_2^2 + M_0M_1 + M_1M_0 + M_0M_2 + M_2M_0 + M_1M_2 + M_2M_1 \\
&= 3\mathbb{1} + M_0M_1 + M_1M_0 + (M_0 + M_1)M_2 + M_2(M_0 + M_1) \qquad (4)\\
&= \mathbb{1} + M_0M_1 + M_1M_0 + (M_0 + M_1 + M_2)M_2 + M_2(M_0 + M_1 + M_2) \\
&= \mathbb{1} + M_0M_1 + M_1M_0 + \Delta M_2 + M_2\Delta \qquad (5)
\end{aligned}$$

We have Δ^2 relates to J_3, and hence we obtain the following bound:

$$\begin{aligned}
\frac{1}{d}\mathrm{tr}(\Delta^2) &= \frac{1}{d}\mathrm{tr}\left(M_0^2 + M_1^2 + M_2^2 + 2M_0M_1 + 2M_0M_2 + 2M_1M_2\right) \\
&= \frac{3}{d}\mathrm{tr}\left(\mathbb{1}\right) + \frac{2}{d}\mathrm{tr}\left(M_0M_1 + M_0M_2 + M_1M_2\right) \\
&= 3 + 2(c_{01} + c_{02} + c_{12}) = 1 + 2(1 + c_{01} + c_{02} + c_{12}) = 1 + 8J_3 \\
&\leq 1 + 8\left(-\frac{1}{8} + \lambda\right) = 8\lambda \qquad (6)
\end{aligned}$$

Using two projections theory [2, 10, 5], we have a decomposition of the Hilbert space $\mathfrak{H}$

$$\mathfrak{H} = \mathfrak{L}_{00} \oplus \mathfrak{L}_{01} \oplus \mathfrak{L}_{10} \oplus \mathfrak{L}_{11} \oplus \bigoplus_{j=1}^{k} \mathfrak{H}_j,$$

where $\dim(\mathfrak{L}_{\alpha\beta}) = l_{\alpha\beta}$ for $\alpha, \beta \in \{0, 1\}$, and $\dim(\mathfrak{H}_j) = 2$, where the projections E_0^0 and E_0^1 take the form:

$$\begin{aligned}
E_0^0 &= 0_{l_{00}} \oplus 0_{l_{01}} \oplus \mathbb{1}_{l_{10}} \oplus \mathbb{1}_{l_{11}} \oplus \bigoplus_{j=1}^{k} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \\
E_0^1 &= 0_{l_{00}} \oplus \mathbb{1}_{l_{01}} \oplus 0_{l_{10}} \oplus \mathbb{1}_{l_{11}} \oplus \bigoplus_{j=1}^{k} \begin{pmatrix} \cos^2\theta_j & \sin\theta_j\cos\theta_j \\ \sin\theta_j\cos\theta_j & \sin^2\theta_j \end{pmatrix}.
\end{aligned}$$

That is, we can express

$$M_0 = -\mathbb{1}_{\mathfrak{L}_{00}} \oplus -\mathbb{1}_{\mathfrak{L}_{01}} \oplus \mathbb{1}_{\mathfrak{L}_{10}} \oplus \mathbb{1}_{\mathfrak{L}_{11}} \oplus \bigoplus_{j=1}^{k} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

$$M_1 = -\mathbb{1}_{\mathfrak{L}_{00}} \oplus \mathbb{1}_{\mathfrak{L}_{01}} \oplus -\mathbb{1}_{\mathfrak{L}_{10}} \oplus \mathbb{1}_{\mathfrak{L}_{11}} \oplus \bigoplus_{j=1}^{k} \begin{pmatrix} \cos 2\theta_j & \sin 2\theta_j \\ \sin 2\theta_j & -\cos 2\theta_j \end{pmatrix}.$$

Now let us define $\tilde{M}_0, \tilde{M}_1, \tilde{M}_2$ as follows. Note that our ideal projections $\hat{E}_0^1, \hat{E}_1^1$ correspond to angle $\hat{\theta} = \frac{2\pi}{3}$, and without loss of generality we can assume[1] $|\theta_j - \hat{\theta}| \leq \frac{\pi}{6}$.

$$\tilde{M}_0 = M_0 = -\mathbb{1}_{\mathfrak{L}_{00}} \oplus -\mathbb{1}_{\mathfrak{L}_{01}} \oplus \mathbb{1}_{\mathfrak{L}_{10}} \oplus \mathbb{1}_{\mathfrak{L}_{11}} \oplus \bigoplus_{j=1}^{k} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

$$\tilde{M}_1 = -\mathbb{1}_{\mathfrak{L}_{00}} \oplus \mathbb{1}_{\mathfrak{L}_{01}} \oplus -\mathbb{1}_{\mathfrak{L}_{10}} \oplus \mathbb{1}_{\mathfrak{L}_{11}} \oplus \bigoplus_{j=1}^{k} \begin{pmatrix} \cos 2\hat{\theta} & \sin 2\hat{\theta} \\ \sin 2\hat{\theta} & -\cos 2\hat{\theta} \end{pmatrix},$$

$$\tilde{M}_2 = \mathbb{1}_{\mathfrak{L}_{00}} \oplus \mathbb{1}_{\mathfrak{L}_{01}} \oplus -\mathbb{1}_{\mathfrak{L}_{10}} \oplus -\mathbb{1}_{\mathfrak{L}_{11}} \oplus \bigoplus_{j=1}^{k} \begin{pmatrix} -1-\cos 2\hat{\theta} & -\sin 2\hat{\theta} \\ -\sin 2\hat{\theta} & 1+\cos 2\hat{\theta} \end{pmatrix}.$$

As desired, $\tilde{M}_x = (L_0^x - L_1^x) + \hat{M}_x \otimes \mathbb{1}_{\mathbb{C}^k}$, where the $\{L_y^x\}$ are the projection onto the summands $\mathfrak{L}_{\mu\nu}$.

First we bound the dimension of each $\mathfrak{L}_{\mu\nu}$. Consider (4) for Δ^2. If $|\psi_{01}\rangle \in \mathfrak{L}_{01}$, then

$$\begin{aligned}\langle\psi_{01}|\Delta^2|\psi_{01}\rangle &= \langle\psi_{01}|(3\mathbb{1} + M_0M_1 + M_1M_0 + (M_0+M_1)M_2 + M_2(M_0+M_1)|\psi_{01}\rangle \\ &= 3 - 1 - 1 + 0 + 0 = 1.\end{aligned}$$

The same equality holds for $|\psi_{10}\rangle \in \mathfrak{L}_{10}$, namely $\langle\psi_{10}|\Delta^2|\psi_{10}\rangle = 1$.

For a vector $|\psi_{00}\rangle$ in $\mathfrak{L}_{00}$ we again use (4) to get $\langle\psi_{00}|\Delta^2|\psi_{00}\rangle = 3+1+1-4\langle\psi_{00}|M_2|\psi_{00}\rangle$.

Now from Cauchy-Schwarz, and that $M_2^2 = \mathbb{1}$, we have

$$|\langle\psi_{00}|M_2|\psi_{00}\rangle| \leq |\langle\psi_{00}|\psi_{00}\rangle|^{\frac{1}{2}} |\langle\psi_{00}|M_2^2|\psi_{00}\rangle|^{\frac{1}{2}} = 1$$

and thus $\langle\psi_{00}|\Delta^2|\psi_{00}\rangle \geq 1$. Similarly for $|\psi_{11}\rangle$ in $\mathfrak{L}_{11}$ we have

$$\langle\psi_{11}|\Delta^2|\psi_{11}\rangle = 5 + 4\langle\psi_{11}|M_2|\psi_{11}\rangle \geq 5 - 4|\langle\psi_{11}|M_2|\psi_{11}\rangle| \geq 1.$$

Putting everything together, since $\langle\psi_{\alpha\beta}|\Delta^2|\psi_{\alpha\beta}\rangle \geq 1$ on each $\mathfrak{L}_{\alpha\beta}$, for $\alpha, \beta \in \{0,1\}$, summing over bases of the respective spaces

$$\frac{l}{d} = \frac{1}{d}(l_{00} + l_{01} + l_{10} + l_{11}) \leq \frac{1}{d}\sum_{j=1}^{l}\langle\psi_j|\Delta^2|\psi_j\rangle \leq \frac{1}{d}\mathrm{tr}(\Delta^2) \leq 8\lambda.$$

where the second-to-last inequality follows from Δ^2 being positive semidefinite.

This immediately provides the claimed bound on the statistical difference from uniform. We can explicitly bound the quantities $|a_0|$ and $|a_1|$ as follows:

$$|a_0| = \frac{1}{d}|\mathrm{tr}(M_0)| = \frac{1}{d}|-l_{00} - l_{01} + l_{10} + l_{11}| \leq \frac{l}{d} \leq 8\lambda$$

$$|a_1| = \frac{1}{d}|\mathrm{tr}(M_1)| = \frac{1}{d}|-l_{00} + l_{01} - l_{10} + l_{11}| \leq \frac{l}{d} \leq 8\lambda.$$

[1] Direct examination of (1) reveals that any θ_j is within $\frac{\pi}{6}$ of the image of some E_y^x; the bound we prove is symmetric in x, y we may reorder the labeling in each $\mathfrak{H}_j$ so that θ_j is close to E_0^1 with $\hat{\theta} = \frac{2\pi}{3}$.

Using Cauchy-Schwarz, we bound $|a_2|$. As

$$a_0 + a_1 + a_2 = \frac{1}{d}\mathrm{tr}(\Delta) \leq \left(\frac{1}{d}\mathrm{tr}(\Delta^2)\right)^{\frac{1}{2}} \left(\frac{1}{d}\mathrm{tr}(\mathbb{1}^2)\right)^{\frac{1}{2}} \leq \sqrt{8\lambda},$$

we have $a_2 \leq \sqrt{8\lambda} - a_0 - a_1$ and therefore $|a_2| \leq \sqrt{8\lambda} + |a_0| + |a_1| \leq \sqrt{8}\sqrt{\lambda} + 16\lambda$.

Finally we bound each $\frac{1}{d}\mathrm{tr}\left(\left(M_x - \tilde{M}_x\right)^2\right)$. Note $M_0 - \tilde{M}_0 = 0$ by construction. Then

$$\begin{aligned}\frac{1}{d}\mathrm{tr}\left(\left(M_1 - \tilde{M}_1\right)^2\right) &= \frac{1}{d}\sum_j \mathrm{tr}\left(\begin{pmatrix} \cos 2\theta_j - \cos 2\hat{\theta} & \sin 2\theta_j - \sin 2\hat{\theta} \\ \sin 2\theta_j - \sin 2\hat{\theta} & -\cos 2\theta_j + \cos 2\hat{\theta} \end{pmatrix}^2\right) \\ &= \frac{1}{d}\sum_j (4 - 4\cos(2(\theta_j - \hat{\theta}))) \;=\; \frac{8}{d}\sum_j \sin^2(\theta_j - \hat{\theta})\end{aligned}$$

To bound this, we note that on any $\mathfrak{H}_j$:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} \cos 2\theta_j & \sin 2\theta_j \\ \sin 2\theta_j & -\cos 2\theta_j \end{pmatrix} + \begin{pmatrix} \cos 2\theta_j & \sin 2\theta_j \\ \sin 2\theta_j & -\cos 2\theta_j \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = 2\cos 2\theta_j \cdot \mathbb{1}_{\mathfrak{H}_j}.$$

From this we obtain

$$\left[\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + \begin{pmatrix} \cos 2\theta_j & \sin 2\theta_j \\ \sin 2\theta_j & -\cos 2\theta_j \end{pmatrix}\right]^2 = 4\cos^2\theta_j \mathbb{1}_{\mathfrak{H}_j}.$$

Hence there exists a basis $\{|\psi_0\rangle, |\psi_1\rangle\}$ of $\mathfrak{H}_j$ such that

$$(M_0 + M_1)|\psi_0\rangle = 2\cos\theta_j|\psi_0\rangle \text{ and } (M_0 + M_1)|\psi_1\rangle = -2\cos\theta_j|\psi_1\rangle.$$

Therefore again from (4) we have

$$\begin{aligned}\langle\psi_0|\Delta^2|\psi_0\rangle &= 3 + 2\cos 2\theta_j + 4\cos\theta_j\langle\psi_0|M_2|\psi_0\rangle \\ \langle\psi_1|\Delta^2|\psi_1\rangle &= 3 + 2\cos 2\theta_j - 4\cos\theta_j\langle\psi_1|M_2|\psi_1\rangle.\end{aligned}$$

In particular, $\langle\psi_0|\Delta^2|\psi_0\rangle + \langle\psi_1|\Delta^2|\psi_1\rangle \geq 6 + 4\cos 2\theta_j - 8|\cos\theta_j|$. It is straightforward to show for $\theta \in \left[\frac{2\pi}{3} - \frac{\pi}{6}, \frac{2\pi}{3} + \frac{\pi}{6}\right]$ that $6 + 4\cos 2\theta - 8|\cos\theta| \geq 4\sin^2\left(\theta - \frac{2\pi}{3}\right)$. And hence we obtain the bound

$$\begin{aligned}\frac{1}{d}\mathrm{tr}(\Delta^2) &\geq \frac{1}{d}\sum_j (6 + 4\cos 2\theta_j - 8|\cos\theta_j|) \\ &\geq \frac{1}{d}\sum_j 4\sin^2(\theta_j - \hat{\theta}) \;=\; \frac{1}{2d}\mathrm{tr}\left(\left(M_1 - \tilde{M}_1\right)^2\right).\end{aligned}$$

In particular, $\frac{1}{d}\mathrm{tr}\left(\left(M_1 - \tilde{M}_1\right)^2\right) \leq 16\lambda$.

Finally, note $\tilde{M}_0 + \tilde{M}_1 + \tilde{M}_2 = -\mathbb{1}_{\mathfrak{L}_{00}} \oplus \mathbb{1}_{\mathfrak{L}_{01}} \oplus -\mathbb{1}_{\mathfrak{L}_{10}} \oplus \mathbb{1}_{\mathfrak{L}_{11}}$. By Jensen's inequality

$$\begin{aligned}\frac{1}{d}\mathrm{tr}\left(\left(M_2 - \tilde{M}_2\right)^2\right) &= \frac{1}{d}\mathrm{tr}\left(\left(\Delta - (-\mathbb{1}_{\mathfrak{L}_{00}} \oplus \mathbb{1}_{\mathfrak{L}_{01}} \oplus -\mathbb{1}_{\mathfrak{L}_{10}} \oplus \mathbb{1}_{\mathfrak{L}_{11}}) + (\tilde{M}_1 - M_1)\right)^2\right) \\ &\leq \frac{1}{d}\mathrm{tr}\left(\Delta^2\right) + \frac{1}{d}\mathrm{tr}\left(\mathbb{1}_{\mathfrak{L}}\right) + \frac{1}{d}\mathrm{tr}\left(\left(\tilde{M}_1 - M_1\right)^2\right) \leq 32\lambda.\end{aligned}$$

Therefore, $\frac{1}{3}\sum_{x,y}\frac{1}{d}\mathrm{tr}\left(\left(E_y^x - \tilde{E}_y^x\right)^2\right) \leq 8\lambda$ as desired.

It is straightforward to get a bound on the statistical difference to any synchronous quantum correlation close to $J_3 = -\frac{1}{8}$. Every synchronous quantum correlation is a convex sum of synchronous quantum correlations with maximally entangled states, and so we may write $p = \sum_j c_j p_j$ where p_j is as in the theorem above. Say $J_3(p_j) \leq -\frac{1}{8} + \lambda_j$, and so

$$J_3(p) = \sum_j c_j J_3(p_j) \leq -\frac{1}{8} + \sum_j c_j \lambda_j = -\frac{1}{8} + \lambda$$

where we define $\lambda = \sum_j c_j \lambda_j$. With two uses of Jensen's inequality,

$$\begin{aligned} \frac{1}{3} \sum_{x,y} \left| p(y, y \,|\, x, x) - \frac{1}{2} \right| &\leq \frac{1}{3} \sum_{j,x,y} c_j \left| p_j(y, y \,|\, x, x) - \frac{1}{2} \right| \\ &\leq \sum_j c_j (C\sqrt{\lambda_j} + C'\lambda_j) \leq C\sqrt{\lambda} + C'\lambda. \end{aligned}$$

◀

Unfortunately, this does not yet produce a fully device-independent protocol as we still suffer from a "synchronicity" loophole. We discuss this loophole and close the loophole in the next section.

4 Measure of asynchronicity

That $J_3 = -\frac{1}{8}$ can be achieved by a unique synchronous quantum correlation, which necessarily can only be realized through a maximally entangled state, provides the device-independent security of the above QKD scheme. However this opens a "synchronicity" security loophole: can a (asynchronous) quantum device simulate $J_3 = -\frac{1}{8}$ without using maximally entangled states (and hence potentially leak information about the derived shared keys)? Fortunately a recent work shows that the same results apply to "almost" synchronous correlations [24]. This allows us to close this synchronicity loophole by also bounding the asynchronicity of the observed correlation.

▶ **Definition 5.** *The* asynchronicity *with respect to a basis choice $x \in X$ and set of measurement outcomes Y is $S_x(p) = \sum_{y_A \neq y_B} p(y_A, y_B \,|\, x, x)$. The* total *(or* expected*) asynchronicity is*

$$S(p) = \frac{1}{|X|} \sum_{x \in X} S_x(p) \tag{7}$$

In [24], this measure is called the "default to synchronicity" and denoted δ_{sync}. While the expected asynchronicity is the average likelihood of an asynchronous result where the inputs are sampled uniformly at random, all results here and in [24], apply to the case where the expectation is computed over inputs sampled with respect to some other fixed distribution. To bound the asynchronicity, we modify the scheme in Section 3 so that for some data rounds where Alice and Bob have selected the same inputs they still reveal their output, stated as Algorithm 1 below.

Here we state the main result [24, Theorem 3.1] in the notation used above. Note that this theorem refers to symmetric (albeit asynchronous) correlations, which is the natural setting as every synchronous quantum correlation is symmetric. This implies a special form for the projections in the correlation, involving the transpose with respect to the natural basis given by the Schmidt-decomposition of the entangled state used in the correlation.

▶ **Theorem 6** (Vidick). *There are universal constants $c, C > 0$ such that the following holds. Let X and Y be finite sets and p a symmetric quantum correlation with input set X, measurement results Y, and asynchronicity $S = S(p)$. Write $p(y_A, y_B \mid x_A, x_B) = \langle\psi|E_{y_A}^{x_A} \otimes (E_{y_B}^{x_B})^T|\psi\rangle$ where $\{E_y^x\}_{y\in Y}$ is a POVM on a finite-dimensional Hilbert space $\mathfrak{H}$ and $|\psi\rangle$ a state on $\mathfrak{H}\otimes\mathfrak{H}$. Let $|\psi\rangle = \sum_{j=1}^r \sqrt{\sigma_j} \sum_{m=1}^{d_j} |\phi_{j,m}^A\rangle\otimes|\phi_{j,m}^B\rangle$ be the Schmidt decomposition, and write $|\psi_j\rangle = \frac{1}{\sqrt{d_j}} \sum_{m=1}^{d_j} |\phi_{j,m}^A\rangle \otimes |\phi_{j,m}^B\rangle$. Then*

1. *$\mathfrak{H} = \bigoplus_{j=1}^r \mathfrak{H}_j$ with $|\psi_j\rangle$ being maximally entangled on $\mathfrak{H}_j \otimes \mathfrak{H}_j$;*
2. *there is a projective measurement $\{E_y^{j,x}\}_{y\in Y}$ on each $\mathfrak{H}_j$ so that*

$$p_j(y_A, y_B \mid x_A, x_B) = \langle\psi_j|E_{y_A}^{j,x_A} \otimes (E_{y_B}^{j,x_B})^T|\psi_j\rangle = \frac{1}{d_j}\mathrm{tr}(E_{y_A}^{j,x_A} E_{y_B}^{j,x_B})$$

is a synchronous quantum correlation and $p \approx \sum_{j=1}^r d_j\sigma_j p_j$ in that:

$$\frac{1}{|X|} \sum_{x\in X} \sum_{y\in Y} \sum_{j=1}^r \frac{1}{d_j} \sum_{m=1}^{d_j} \langle\phi_{j,m}^A| \left(E_y^x - E_y^{j,x}\right)^2 |\phi_{j,m}^A\rangle \leq CS^c.$$

As indicated in [24, §4.1], this result can be used to transfer rigidity from synchronous to almost synchronous correlations. As $\sum_j d_j\sigma_j = 1$, we also transfer the bound on the statistical difference from uniform to convex sums in this theorem exactly as in the previous section. As for the full correlation we rephrase Lemma 2.10 of [24] in the context of Theorem 6 as follows.

▶ **Corollary 7.** *Let $p(y_A, y_B \mid x_A, x_B) = \langle\psi|E_{y_A}^{x_A} \otimes (E_{y_B}^{x_B})^T|\psi\rangle$ be a quantum correlation with asynchronocity S as in Theorem 6, and let $\bar{p} = \sum_{j=1}^r d_j\sigma_j p_j$ with*

$$\frac{1}{|X|} \sum_{x\in X} \sum_{y\in Y} \sum_{j=1}^r \frac{1}{d_j} \sum_{m=1}^{d_j} \langle\phi_{j,m}^A| \left(E_y^x - E_y^{j,x}\right)^2 |\phi_{j,m}^A\rangle = \gamma$$

as given in Theorem 6. Then

$$\frac{1}{|X|^2} \sum_{x_A,x_B,y_A,y_B} |p(y_A, y_B \mid x_A, x_B) - \bar{p}(y_A, y_B \mid x_A, x_B)| \leq 3S + 4\sqrt{\gamma}.$$

Note that this bound on the statistical difference directly bounds $J_3(p)$ in terms of the convex sum of the analogous $J_3(p_j)$. Note that J_3, as seen in (3), is an affine function so $J_3(\bar{p}) = \sum_{j=1}^r \sigma_j d_j J_3(p_j)$ using the notation of Theorem 6 above. Then immediately from Corollary 7, $|J_3(p) - J_3(\bar{p})| \leq \frac{27}{4}S + 9\sqrt{\gamma}$. In turn from Theorem 6 we have $\gamma \leq CS^c$, and so there are different universal constants C', c' so that

$$|J_3(p) - J_3(\bar{p})| \leq C'S^{c'}. \tag{8}$$

▶ **Corollary 8.** *Let $p(y_A, y_B \mid x_A, x_B) = \langle\psi|E_{y_A}^{x_A} \otimes (E_{y_B}^{x_B})^T|\psi\rangle$ be a quantum correlation as in Theorem 6 and suppose $J_3(p) = -\frac{1}{8} + \lambda$. Then the Hilbert space decomposes as $\mathfrak{H} = \bigoplus_{j=1}^r \mathfrak{H}_j = \bigoplus_{j=1}^r (\mathfrak{L}_j \oplus (\mathbb{C}^2 \otimes \mathfrak{K}_j))$ where $\frac{\dim\mathfrak{L}_j}{\dim\mathfrak{H}_j} \leq 8\lambda_j$. On each summand we have projection-valued measures $\{\tilde{E}_y^{j,x}\}$ such that $\tilde{E}_y^{j,x} = L_y^{j,x} + \hat{E}_y^x \otimes \mathbb{1}_{\mathfrak{K}_j}$ and*

$$\frac{1}{3} \sum_{x,y} \sum_{j=1}^r \sigma_j d_j \left(\frac{1}{d_j} \sum_{m=1}^{d_j} \langle\phi_{j,m}^A|(E_y^x - \tilde{E}_y^{j,x})^2|\phi_{j,m}^A\rangle \right) \leq C_1 S^c + C_2\lambda$$

for universal constants c, C_1, C_2.

Proof. Given $\{E_y^x\}$ as above, we obtain projections $\{E_y^{j,x}\}$ defining synchronous correlations p_j from Theorem 6. Write $J_3(p_j) = -\frac{1}{8} + \lambda_j$. From Theorem 4, we obtain the given decomposition of the Hilbert space and projection-valued measures $\{\tilde{E}_y^{j,x}\}$ where

1. $\tilde{E}_y^{j,x} = L_y^{j,x} + \hat{E}_y^x \otimes \mathbb{1}_{\mathfrak{K}_j}$,
2. $\frac{\dim \mathfrak{L}_j}{\dim \mathfrak{H}_j} \leq 8\lambda_j$, and
3. $\frac{1}{3}\sum_{x,y} \frac{1}{d_j}\sum_{m=1}^{d_j} \langle \phi_{j,m}^A | (E_y^{j,x} - \tilde{E}_y^{j,x})^2 | \phi_{j,m}^A \rangle \leq C_2 \lambda_j$.

Then using the notation and (8) above $|J_3(p) - J_3(\bar{p})| = \left|\lambda - \sum_{j=1}^r \sigma_j d_j \lambda_j\right| \leq C' S^{c'}$ and thus

$$\frac{1}{3}\sum_{x,y}\sum_{j=1}^r \sigma_j d_j \left(\frac{1}{d_j}\sum_{m=1}^{d_j} \langle \phi_{j,m}^A | (E_y^{j,x} - \tilde{E}_y^{j,x})^2 | \phi_{j,m}^A \rangle\right) \leq C_2 \sum_{j=1}^r \sigma_j d_j \lambda_j = C_2\lambda + C_2 C' S^{c'}.$$

On the other hand,

$$\begin{aligned}
&\frac{1}{3}\sum_{x,y}\sum_{j=1}^r \sigma_j d_j \left(\frac{1}{d_j}\sum_{m=1}^{d_j} \langle \phi_{j,m}^A | (E_y^{x} - E_y^{j,x})^2 | \phi_{j,m}^A \rangle\right) \\
&\quad \leq \frac{1}{3}\sum_{x,y}\sum_{j=1}^r \left(\frac{1}{d_j}\sum_{m=1}^{d_j} \langle \phi_{j,m}^A | (E_y^{x} - E_y^{j,x})^2 | \phi_{j,m}^A \rangle\right) \leq C'' S^{c''}
\end{aligned}$$

directly from Theorem 6. So by Jensen's inequality

$$\begin{aligned}
&\frac{1}{3}\sum_{x,y}\sum_{j=1}^r \sigma_j d_j \left(\frac{1}{d_j}\sum_{m=1}^{d_j} \langle \phi_{j,m}^A | (E_y^{x} - \tilde{E}_y^{j,x})^2 | \phi_{j,m}^A \rangle\right) \\
&\quad \leq \frac{2}{3}\sum_{x,y}\sum_{j=1}^r \sigma_j d_j \left(\frac{1}{d_j}\sum_{m=1}^{d_j} \langle \phi_{j,m}^A | (E_y^{x} - E_y^{j,x})^2 | \phi_{j,m}^A \rangle\right) \\
&\qquad + \frac{2}{3}\sum_{x,y}\sum_{j=1}^r \sigma_j d_j \left(\frac{1}{d_j}\sum_{m=1}^{d_j} \langle \phi_{j,m}^A | (E_y^{j,x} - \tilde{E}_y^{j,x})^2 | \phi_{j,m}^A \rangle\right) \\
&\quad \leq 2C_1 S^c + 2C_2\lambda
\end{aligned}$$

for some universal constant C_1. ◀

5 Security and key-rate analysis

Our synchronous fully device-independent quantum key distribution protocol is stated in Algorithm 1. For an honest, but possibly noisy implementation of the protocol, we assume that Alice and Bob perform measurements $E_{y_A}^{x_A^i} \otimes E_{y_B}^{x_B^i}$ on the state ρ_{AB}. We assume a depolarization channel and take ρ_{AB} to be the state $(1-\nu)|\Phi^+\rangle\langle\Phi^+| + \frac{\nu}{4}\mathbb{1}$, where $\nu \in [0,1]$ is the depolarization noise and $|\Phi^+\rangle$ is the EPR pair. Using measurements according to Equation (1), we get $J_3 = -\frac{1}{8} + \frac{3}{8}\nu$, and $S = \frac{\nu}{2}$. A general framework for analyzing device-independent protocols was laid out in [4], which we use to show completeness and soundness of our protocol.

Algorithm 1 Synchronous QKD Protocol.

Input:

$\lambda \in [0, \frac{1}{8})$:	Allowed error in J_3 violation
$\mu \in [0, \mu_0]$:	Allowed error in asynchronicity S with μ_0 being a pre-decided threshold
$n \in \mathbb{N}$:	Total number of rounds
$m \in \mathbb{N}$:	Parameter for choosing asynchronicity check rounds. $\kappa := \frac{1}{m}$
$\gamma \in (0, 1]$:	Expected fraction of test rounds
$\delta^{J_3}_{est} \in (0, 1)$:	Width of statistical interval for the J_3 test
$\delta^{S}_{est} \in (0, 1)$:	Width of statistical interval for the S test
EC:	Error Correction protocol
PA:	Privacy Amplification protocol

1 **For** $i \in [n]$:
2 Alice and Bob draw $x^i_A \leftarrow X$, $x^i_B \leftarrow X$ according to Equation (9)
3 They produce outputs y^i_A and y^i_B using $\{E^{x^i_A}_y\}$ and $\{E^{x^i_B}_y\}$ respectively
4 They share their inputs x^i_A and x^i_B.
5 **Error Correction**: Alice and Bob use error correction protocol EC to obtain outputs $\tilde{Y}_A$ and $\tilde{Y}_B$. They abort if the error correction protocol aborts.
6 **Parameter Estimation**:
7 Bob estimates the J_3 violation in rounds where $x^i_A \neq x^i_B$, i.e. he sets $R_i = 1$ if $\tilde{y}^i_A \neq y^i_B$ else 0. He aborts if $\sum_i R_i < \left[\gamma\left(\frac{3}{4} - \frac{2}{3}\lambda\right) - \delta^{J_3}_{est}\right] \cdot n$
8 He also estimates the asynchronicity S in rounds where $x^i_A = x^i_B$ and $i \pmod m = 0$, i.e. he sets $Q_i = 1$ if $\tilde{y}^i_A \neq y^i_B$ else 0 in those rounds. He aborts if $\sum_i Q_i < \left[\kappa(1-\gamma)\mu - \delta^{S}_{est}\right] \cdot n$
9 **Privacy Amplification**: Alice and Bob use privacy amplification protocol PA to create final keys K_A and K_B using $\tilde{y}^i_A$ and $\tilde{y}^i_B$ where $x^i_A = x^i_B$ and $i \pmod m \neq 0$.

▶ **Lemma 9** (Completeness). *Let ϵ^c_{EC} be the completeness error of the EC protocol, and ϵ_{EC} be the probability that the EC protocol does not abort but Alice and Bob hold different outputs post error correction. Then, Protocol 1 has completeness error $\epsilon^c_{QKD} \leq exp\left(-2n\left((\delta^S_{est})^2) + (\delta^{J_3}_{est})^2\right)\right) + \epsilon^c_{EC} + \epsilon_{EC}$.*

Proof. The protocol either aborts in the error correction step or the parameter estimation step. The probability of aborting during the J_3 and S tests can be bounded using Hoeffding's inequality as follows:

$$\Pr\left(\sum_j R_j > \left[\gamma\left(\frac{3}{4} - \frac{2}{3}\lambda\right) - \delta^{J_3}_{est}\right] \cdot n \bigwedge \sum_j Q_j > \left[\kappa(1-\gamma)\mu - \delta^S_{est}\right] \cdot n\right) \leq exp\left(-2n\left((\delta^S_{est})^2) + (\delta^{J_3}_{est})^2\right)\right).$$

The rest of the proof follows analogously to [4, Lemma 5.2 and Eq. 5.2] ◀

We use the Entropy Accumulation Theorem (EAT) [8], to bound the min-entropy of Alice and Bob's outputs with respect to an adversary Eve's side information. To that effect, we define Ω as the event that Alice and Bob do not abort the protocol in the parameter estimation step. The EAT yields a bound on the min-entropy, given we find an appropriate *min-tradeoff* function.

We state the min-entropy bound in the following theorem.

▶ **Theorem 10.** *Let $\rho_{Y_AY_BX_AX_BTE}$ be the joint state of Alice, Bob and Eve's system along with the register T for indicating testing versus data rounds, and let Ω be the event that the protocol does not abort during parameter estimation. We write $\rho_{|\Omega}$ for the state of the system conditioned on Ω. Let $\epsilon_{EA}, \epsilon_s \in (0,1)$. Then either*

1. *The protocol aborts with probability greater than $1-\epsilon_{EA}$, or*
2. $H_{\min}^{\epsilon_s}(Y_AY_B|X_AX_BTE)_{\rho|\Omega} > n \cdot \mathsf{OPT}(\epsilon_s, \epsilon_{EA})$, *where* OPT *is defined as follows:*

$$g(p) = \begin{cases} 1 - h\left(3 - 4\frac{p(1)}{\gamma}\right) & \frac{p(1)}{\gamma} \in \left[\frac{2}{3}, \frac{3}{4}\right] \\ 1 & \frac{p(1)}{\gamma} \in \left[\frac{3}{4}, 1\right], \end{cases}$$

$$f_{\min}^{J_3}(p, p_t) = \begin{cases} g(p) & \text{if } p(1) \leq p_t(1) \\ \frac{d}{dp(1)} g(p)\Big|_{p_t} \cdot p(1) + g(p_t) - \frac{d}{dp(1)} g(p)\Big|_{p_t} \cdot p_t(1) & \text{if } p(1) > p_t(1), \end{cases}$$

$$f_{EAT} = n f_{\min}^{J_3}(p, p_t) - \frac{2}{\sqrt{n}} \left(\log 9 + \left\lceil \frac{d}{dp(1)} g(p) \right\rceil\right) \sqrt{1 - 2\log(\epsilon_s \cdot \epsilon_{EA})},$$

$$\mathsf{OPT}(\epsilon_s, \epsilon_{EA}) = \max_{\frac{2}{3} < \frac{p_t(1)}{\gamma} < \frac{3}{4}} f_{EAT}(p, p_t, \epsilon_s, \epsilon_{EA}).$$

Before we state the proof, we develop some key ideas and prove some lemmas that will be used in the proof. In round $i \in [n]$, Alice and Bob draw from a local biased distribution with $p_0, p_1, p_2 \in [0,1]$:

$$x_i = \begin{cases} i \pmod 3 & \text{with probability } p_0, \\ i+1 \pmod 3 & \text{with probability } p_1, \\ i+2 \pmod 3 & \text{with probability } p_2. \end{cases} \tag{9}$$

Without loss of generality we may assume that the total number of rounds is a multiple of 3, i.e. $n = 3N$ for some N. There are two cases in which they perform a testing round – first for testing the violation of the Bell inequality J_3, and second to test the asynchronicity of the protocol. Let γ be the probability of performing a J_3 test. Thus we have $\gamma = p(x_A \neq x_B)$.

$$\gamma = p(x_A \neq x_B) = \frac{1}{3}\sum_{i=0}^{2} p(x_A^i \neq x_B^i) = 2(p_0p_1 + p_0p_2 + p_1p_2).$$

For the J_3 test we define a random variable R_i as follows:

$$R_i = \begin{cases} 1 & \text{if } y_A^i \neq y_B^i \text{ and } x_A^i \neq x_B^i, \\ 0 & \text{if } y_A^i = y_B^i \text{ and } x_A^i \neq x_B^i, \\ \perp & \text{if } x_A^i = x_B^i. \end{cases}$$

The probability that $R_i = 1$ is given by

$$p(R_i = 1) = p(y_A^i \neq y_B^i \wedge x_A^i \neq x_B^i) = \sum_{i}^{3N} \sum_{\substack{y_A^i \neq y_B^i \\ x_A^i \neq x_B^i}} p(y_A^i, y_B^i \mid x_A^i, x_B^i) \cdot p(x_A^i, x_B^i) \cdot \frac{1}{3N}$$

$$= \frac{1}{3}\sum_{i=0}^{2} \sum_{\substack{y_A^i \neq y_B^i \\ x_A^i \neq x_B^i}} p(y_A^i, y_B^i \mid x_A^i, x_B^i) \cdot p(x_A^i, x_B^i)$$

$$= \frac{1}{3}(p_0p_1 + p_0p_2 + p_1p_2) \sum_{\substack{y_A \neq y_B \\ x_A \neq x_B}} p(y_A, y_B \mid x_A, x_B)$$

$$= \frac{1}{3}(p_0p_1 + p_0p_2 + p_1p_2)(4 - 4J_3) = \gamma\left(\frac{2}{3} - \frac{2}{3}J_3\right).$$

Similarly, we define a random variable Q_i corresponding to the asyncronicity. We reserve every mth key generation round to perform an asynchronicity check i.e. if $i = 0 \pmod m$ for i such that $x_A^i = x_B^i$. We denote by $\kappa = 1/m$ the fraction of asynchronicity check rounds. We have

$$Q_i = \begin{cases} 1 & \text{if } y_A^i \neq y_B^i \text{ and } x_A^i = x_B^i \text{ and } i = 0 \pmod m, \\ 0 & \text{if } y_A^i = y_B^i \text{ and } x_A^i = x_B^i, \\ \perp & \text{if } x_A^i \neq x_B^i. \end{cases}$$

The probability that $Q_i = 1$ is given by

$$\begin{aligned} p(Q_i = 1) &= p(y_A^i \neq y_B^i \wedge x_A^i = x_B^i \wedge i = 0 \pmod m) \\ &= \frac{1}{m} \sum_{i}^{3N} \sum_{x_A^i = x_B^i} \sum_{y_A^i \neq y_B^i} p(y_A^i, y_B^i \mid x_A^i, x_B^i) \cdot p(x_A^i, x_B^i) \cdot \frac{1}{3N} \\ &= \frac{\kappa}{3} \sum_{i=0}^{2} \sum_{\substack{y_A^i \neq y_B^i \\ x_A^i = x_B^i}} p(y_A^i, y_B^i \mid x_A^i, x_B^i) \cdot p(x_A^i, x_B^i) \\ &= \frac{\kappa}{3}(p_0^2 + p_1^2 + p_2^2) \sum_{\substack{y_A \neq y_B \\ x_A = x_B}} p(y_A, y_B \mid x_A, x_B) = \frac{\kappa}{3}(1-\gamma) \cdot 3S = \kappa(1-\gamma)S. \end{aligned}$$

Thus if $p(x_A \neq x_B) = \gamma$, then the probability that we are in a testing round (J_3 or S), i.e. $T_i = 1$ is given by $\gamma + \kappa(1-\gamma)$. We can tune γ arbitrarily by choosing p_0, p_1 and p_2 appropriately.

Before proving Theorem 10, we first show a bound on the mutual information between Alice's output and Eve's system. Following the outline in [1], we assume that Eve provides Alice and Bob a Bell diagonal state with eigenvalues $\lambda_{\Phi^+}, \lambda_{\Phi^-}, \lambda_{\Psi^+}, \lambda_{\Psi^-}$ corresponding to the Bell states

$$\begin{aligned} |\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \\ |\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \qquad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \end{aligned}$$

We may write the Bell diagonal state as

$$\rho_\lambda = \begin{pmatrix} \lambda_{\Phi^+} & & & \\ & \lambda_{\Psi^-} & & \\ & & \lambda_{\Phi^-} & \\ & & & \lambda_{\Psi^-} \end{pmatrix} \tag{10}$$

The following lemma provides a bound on the mutual information between Alice's output and Eve's system. This bound is then used in the proof of the theorem in bounding the min-entropy of Alice and Bob's outputs in the protocol conditioned on Eve's side information.

▶ **Lemma 11.** *Let Y_A^i be Alice's output in round $i \in [n]$, and E be Eve's register. If Eve provides Alice and Bob the Bell diagonal state ρ_λ in Equation* (10)*, with eigenvalues ordered as $\lambda_{\Phi^+} \geq \lambda_{\Psi^-}$ and $\lambda_{\Phi^-} \geq \lambda_{\Psi^+}$, we have*

$$\chi(Y_A^i : E) \leq h(\lambda_{\Phi^-}).$$

Proof. For Alice and Bob's measurement operators $E_{y_A}^{x_A}$ and $F_{y_B}^{x_B}$, the probability of getting outputs (y_A, y_B) given inputs (x_A, x_B) and state ρ is given by the Born rule, $p(y_A, y_B \,|\, x_A, x_B) = \mathrm{tr}\left(\left(E_{y_A}^{x_A} \otimes F_{y_B}^{x_B}\right)\rho\right)$. For the Bell diagonal state ρ_λ, this probability may be expanded as follows:

$$\begin{aligned}
p(y_A, y_B \,|\, x_A, x_B) &= \mathrm{tr}\left((E_{y_A}^{x_A} \otimes F_{y_B}^{x_B})\rho_\lambda\right) \\
&= \lambda_{\Phi^+}\mathrm{tr}\left((E_{y_A}^{x_A} \otimes F_{y_B}^{x_B})|\Phi^+\rangle\langle\Phi^+|\right) + \lambda_{\Phi^-}\mathrm{tr}\left((E_{y_A}^{x_A} \otimes F_{y_B}^{x_B})|\Phi^-\rangle\langle\Phi^-|\right) \\
&\quad + \lambda_{\Psi^+}\mathrm{tr}\left((E_{y_A}^{x_A} \otimes F_{y_B}^{x_B})|\Psi^+\rangle\langle\Psi^+|\right) + \lambda_{\Psi^-}\mathrm{tr}\left((E_{y_A}^{x_A} \otimes F_{y_B}^{x_B})|\Psi^-\rangle\langle\Psi^-|\right) \\
&= \lambda_{\Phi^+}\mathrm{tr}\left((E_{y_A}^{x_A} \otimes F_{y_B}^{x_B})|\Phi^+\rangle\langle\Phi^+|\right) + \lambda_{\Phi^-}\mathrm{tr}\left((E_{y_A}^{x_A} \otimes ZF_{y_B}^{x_B}Z)|\Phi^+\rangle\langle\Phi^+|\right) \\
&\quad + \lambda_{\Psi^+}\mathrm{tr}\left((E_{y_A}^{x_A} \otimes XF_{y_B}^{x_B}X)|\Phi^+\rangle\langle\Phi^+|\right) + \lambda_{\Psi^-}\mathrm{tr}\left((E_{y_A}^{x_A} \otimes YF_{y_B}^{x_B}Y)|\Phi^+\rangle\langle\Phi^+|\right) \\
&= \frac{\lambda_{\Phi^+}}{2}\mathrm{tr}\left(E_{y_A}^{x_A}\overline{F_{y_A}^{x_A}}\right) + \frac{\lambda_{\Phi^-}}{2}\mathrm{tr}\left(E_{y_A}^{x_A}\overline{ZF_{y_A}^{x_A}Z}\right) \\
&\quad + \frac{\lambda_{\Psi^+}}{2}\mathrm{tr}\left(E_{y_A}^{x_A}\overline{XF_{y_A}^{x_A}X}\right) + \frac{\lambda_{\Psi^-}}{2}\mathrm{tr}\left(E_{y_A}^{x_A}\overline{YF_{y_A}^{x_A}Y}\right)
\end{aligned}$$

Using this probability, we can compute the values of J_3 and S. One can show that choosing $E_{y_A}^{x_A} = \overline{F_{y_B}^{x_B}}$ is the optimal choice for minimizing J_3 and S simultaneously, but we skip the proof here. We define projection operators using variables $\theta_1, \theta_2, \gamma_1$ and γ_2 which we later optimize:

$$\begin{aligned}
E_0^0 &= |\phi_0\rangle\langle\phi_0| \quad \text{with } |\phi_0\rangle = |0\rangle \\
E_0^1 &= |\phi_1\rangle\langle\phi_1| \quad \text{with } |\phi_1\rangle = \cos\theta_1|0\rangle + e^{i\gamma_1}\sin\theta_1|1\rangle \\
E_0^2 &= |\phi_2\rangle\langle\phi_2| \quad \text{with } |\phi_2\rangle = \cos\theta_2|0\rangle + e^{i\gamma_2}\sin\theta_2|1\rangle
\end{aligned}$$

and where the corresponding $E_1^x = \mathbb{1} - E_0^x$ for $x \in \{0, 1, 2\}$. Computing the asynchronicity S directly according to Equation (7) we get

$$S = \frac{\lambda_{\Phi^-}}{3}\left[\sin^2(2\theta_1) + \sin^2(2\theta_2)\right] + \frac{\lambda_{\Psi^+}}{3}\left[3 - (\sin^2(2\theta_1) + \sin^2(2\theta_2)\right] + \lambda_{\Psi^-}$$

The λ_{Ψ^-} term doesn't depend on θ_1 and θ_2, so we may take $\lambda_{\Psi^-} = 0$ since we want to minimize S. Further, since $\sin^2(2\theta_1) + \sin^2(2\theta_2) \geq 0$ and $\lambda_{\Phi^-} \geq \lambda_{\Psi^+}$, we may take $\lambda_{\Psi^+} = 0$. Next we define δ_1 and δ_2 to be the deviation in angles from the angles in the optimal strategy defined in Equation (1) (the optimal angles are given by $\theta_1 = \frac{\pi}{3}$ and $\theta_2 = -\frac{\pi}{3}$). Thus the equations we obtain for J_3 and S using $\theta_1 = \frac{\pi}{3} + \delta_1$ and $\theta_2 = -\frac{\pi}{3} + \delta_2$ are:

$$\begin{aligned}
J_3 = &-(2\lambda_{\Phi^-} - 1)\cos\left(\frac{\pi}{3} + \delta_1\right)\cos\left(-\frac{\pi}{3} + \delta_2\right)\sin\left(\frac{\pi}{3} + \delta_1\right)\sin\left(-\frac{\pi}{3} + \delta_2\right) \\
&+ \cos^2\left(\frac{\pi}{3} + \delta_1\right)\cos^2\left(-\frac{\pi}{3} + \delta_2\right)
\end{aligned}$$

Since we want to minimize J_3, we minimize the term independent of the factor λ_{Φ^-}. We call this term c_{J_3} and find that this term is

$$\begin{aligned}
c_{J_3} &= \cos\left(\frac{\pi}{3} + \delta_1\right)\cos\left(-\frac{\pi}{3} + \delta_2\right)\sin\left(\frac{\pi}{3} + \delta_1\right)\sin\left(-\frac{\pi}{3} + \delta_2\right) \\
&\quad + \cos^2\left(\frac{\pi}{3} + \delta_1\right)\cos^2\left(-\frac{\pi}{3} + \delta_2\right) \\
&= \cos\left(\frac{2\pi}{3} + \delta_1 - \delta_2\right)\cos\left(\frac{\pi}{3} + \delta_1\right)\cos\left(\frac{\pi}{3} - \delta_2\right)
\end{aligned}$$

Minimizing c_{J_3} for δ_1 and δ_2 we find that $\delta_1 = \frac{\delta_2}{2}$, and $\delta_2 \in \{0, \frac{2\pi}{3}, \frac{4\pi}{3}\}$. The solutions $\delta_2 = \frac{2\pi}{3}$ and $\delta_2 = \frac{4\pi}{3}$ are equivalent to $\delta_2 = 0$, so we only consider the latter solution. This suggests that in order for Eve to minimize J_3, her strategy must match the ideal strategy developed in Equation (1). Using $\delta_1 = \delta_2 = 0$, we get

$$\begin{aligned} J_3 &= -\frac{1}{8} + \frac{3}{8}\lambda_{\Phi^-} \\ S &= \frac{1}{2}\lambda_{\Phi^-} \end{aligned} \tag{11}$$

From [1, Lemma 5], we have

$$\begin{aligned} \chi(Y_A^i : E) &\leq H([\lambda_{\Phi^+}, \lambda_{\Phi^-}, \lambda_{\Psi^+}, \lambda_{\Psi^-}]) - h(\lambda_{\Phi^+} + \lambda_{\Phi^-}) \\ &= h(\lambda_{\Phi^-}) = \begin{cases} h(\frac{1}{3} + \frac{8}{3}J_3) \\ h(2S) \end{cases} \end{aligned}$$

Where the second to last equality follows because $\lambda_{\Psi^+} = \lambda_{\Psi^-} = 0$, thus $H([\lambda_{\Phi^+}, \lambda_{\Phi^-}]) = h(\lambda_{\Phi^-})$, and $h(\lambda_{\Phi^+} + \lambda_{\Phi^-}) = h(1) = 0$ ◀

Proof of Theorem 10. In similar fashion to [4, Theorem 4.1], we need to find a min-tradeoff function in order to apply the EAT. From Lemma 11, we have $\chi(Y_A^i : E|X_A^i = 0) \leq h\left(\frac{1}{3} + \frac{8}{3}J_3\right)$. Thus

$$H(Y_A^i|X_A^i X_B^i E) \geq 1 - h\left(\frac{1}{3} + \frac{8}{3}J_3\right) \tag{12}$$

Inserting this back into Equation (12), we get

$$H(Y_A^i|X_A^i X_B^i E) \geq 1 - h\left(\frac{1}{3} + \frac{8}{3}\left(1 - \frac{3}{2}\frac{p(R_i = 1)}{\gamma}\right)\right) = 1 - h\left(3 - 4\frac{p(R_i = 1)}{\gamma}\right)$$

For $\frac{p(1)}{\gamma} \in \left[\frac{2}{3}, 1\right]$, let

$$g(p) = \begin{cases} 1 - h\left(3 - 4\frac{p(1)}{\gamma}\right) & \frac{p(1)}{\gamma} \in \left[\frac{2}{3}, \frac{3}{4}\right] \\ 1 & \frac{p(1)}{\gamma} \in \left[\frac{3}{4}, 1\right] \end{cases}$$

We note that we only define $g(p)$ in the regime $\frac{p(1)}{\gamma} \in \left[\frac{2}{3}, 1\right]$ since that range is operationally relevant. The function can be extended to values of $\frac{p(1)}{\gamma} \in \left[0, \frac{2}{3}\right]$ for completeness but is not necessary for the purposes of the proof. The function $g(p)$ has unbounded gradient at $\frac{p(1)}{\gamma} = \frac{3}{4}$, and therefore needs to be modified using the "cutting-and-gluing" trick of [4] in order to define a min-tradeoff function that can be used in the EAT. To that effect, we define two functions l_1 and l_2 over a point p_t that can be later optimized:

$$l_1(p_t) = \left\lceil \frac{d}{dp(1)} g(p) \Big|_{p_t} \right\rceil, \qquad l_2(p_t) = g(p_t) - l_1(p_t) \cdot p_t(1)$$

and define $f_{\min}^{J_3}$ as follows:

$$f_{\min}^{J_3}(p, p_t) = \begin{cases} g(p) & \text{if } p(1) \leq p_t(1) \\ l_1(p_t) \cdot p(1) + l_2(p_t) & \text{if } p(1) > p_t(1) \end{cases}$$

Applying the EAT with min-tradeoff function $f_{\min}^{J_3}(p, p_t)$ for any p_t such that $\frac{2}{3} < \frac{p_t(1)}{\gamma} < \frac{3}{4}$, and plugging in $\frac{p(1)}{\gamma} = \frac{p(R_i=1)}{\gamma} = \frac{2}{3} - \frac{2}{3}J_3$, we get the bound on the smooth min-entropy $H_{\min}^{\epsilon_s}(Y_A Y_B \mid X_A X_B E)_{\rho|\Omega}$ ◀

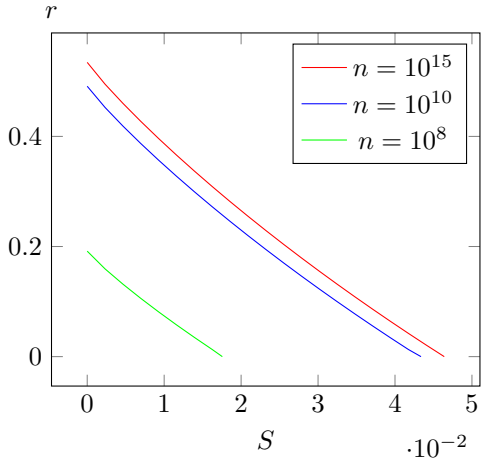


Figure 1 Values of $r = l/n$ against S

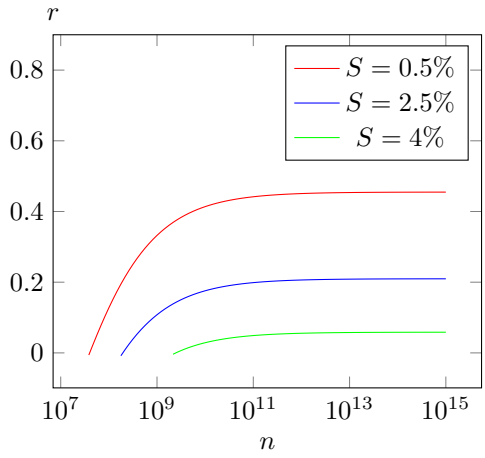


Figure 2 Values of $r = l/n$ against n

The soundness proof for the protocol follows identically to [4, Lemmas 5.3 and 5.4]. The key length l generated at the end of Protocol 1 is derived analogously to [4, Theorem 5.1 and Eq 5.4] which for completeness we state here:

$$l = n \cdot \mathsf{OPT}(\epsilon_s/4, \epsilon_{EA} + \epsilon_{EC}) - \mathsf{leak}_{\mathsf{EC}} - 3\log\left(1 - \sqrt{1 - (\epsilon_s/4)^2}\right) - \gamma \cdot n - \sqrt{n} 2\log 7\sqrt{1 - 2\log(\epsilon_s/4 \cdot (\epsilon_{EA} + \epsilon_{EC}))} - 2\log(1/\epsilon_{PA}) \tag{13}$$

where $\mathsf{leak}_{\mathsf{EC}}$ is discussed in detail in [4, §5.5.1 and Eq 5.9].

Based on Theorem 10 and [4, Theorem 5.1], we plot the key rate, defined as $r = \frac{l}{n}$. In Figure 1, we plot the key rate against the asynchronicity (referred to as the bit-error rate in [4]), and in Figure 2 we plot the key-rate against the total number of rounds while keeping asynchronicity constant. For large n, we are able to tolerate asynchronicity of up to 4.6% before the key-rate goes to 0. We use the values $\epsilon_{EC} = 10^{-10}, \epsilon_{EA} = \epsilon^s_{QKD} = 10^{-5}, \epsilon^c_{QKD} = 10^{-2}, p_0 = 0.97, p_1 = p_2 = 0.015$ and $\delta^{J_3}_{est} = 10^{-3}$ to plot the key rate curves in Figures 1 and 2.

6 Causality Loophole

In this section we describe what is called the *causality* or *locality* loophole common to device independent quantum key distribution protocols that use non-local games, and propose a solution to the loophole using a new security assumption.

As seen in Section 4, the bound for the Bell inequality $J_3 \geq -\frac{1}{8}$ is sharp and rigid only among synchronous quantum correlations. There exist more powerful synchronous nonsignaling strategies that violate those bounds. Furthermore, if classical communication is allowed between the parties in the protocol, even greater violations can be achieved. This is the *causality loophole*: unless Alice and Bob are acausally separated, then the statistics for the synchronous Bell inequalities can simply be simulated using classical communication.

In order to resolve the causality loophole in our protocol we pose a new security assumption: instead of limiting Eve's computational power or limiting the communication she can perform, we assume that she has imperfect knowledge of the basis Alice and Bob use in the protocol. We state this more formally:

Let ϵ be Eve's uncertainty about Alice and Bob's inputs. Without loss of generality, we assume that this is symmetric across all basis selections. For $x', x \in \{0,1,2\}$ we have

$$\Pr\{\text{Eve guesses basis } x' \mid \text{Alice (or Bob) selects basis } x\} = \begin{cases} 1-\epsilon & \text{when } x' = x \\ \\ \frac{\epsilon}{2} & \text{when } x' \neq x. \end{cases} \tag{14}$$

In greater generality, we model the basis selection that Alice and Bob use for their inputs as a classical-quantum state on $\mathbb{C}^3 \otimes \mathbb{C}^3 \otimes \mathfrak{H}_E$, corresponding to Alice, Bob, and Eve respectively. Alice and Bob's states are classical while Eve can have quantum side information which she may use to produce a correlation for a cheating strategy. We denote this state by ρ_{ABE}. For inputs $x_A, x_B \in \{0,1,2\}$ for Alice and Bob respectively, we have $\rho_{ABE} = |x_A\rangle\langle x_A| \otimes |x_B\rangle\langle x_B| \otimes \rho_E^{x_A,x_B}$, where $\rho_E^{x_A,x_B}$ quantifies Eve's side information. Based on (14) above we further decompose

$$\begin{aligned}\rho_E^{x_A,x_B} = \Big((1-\epsilon)^2\sigma_{x_A,x_B} + (1-\epsilon)\frac{\epsilon}{2}(\sigma_{x_A,x_B\oplus 1} + \sigma_{x_A,x_B\oplus 2} + \sigma_{x_A\oplus 1,x_B} + \sigma_{x_A\oplus 2,x_B}) \\ + \frac{\epsilon^2}{4}(\sigma_{x_A\oplus 1,x_B\oplus 1} + \sigma_{x_A\oplus 1,x_B\oplus 2} + \sigma_{x_A\oplus 2,x_B\oplus 1} + \sigma_{x_A\oplus 2,x_B\oplus 2})\Big),\end{aligned}$$

where we denote $x_A \oplus i \coloneqq x_A + i \pmod 3$, and similarly for x_B. Writing Eve's guess for Alice's input by z_A and for Bob's input by z_B, the σ_{z_A,z_B} for $z_A, z_B \in \{0,1,2\}$ are densities containing Eve's side information depending on her guess for x_A and x_B respectively. With these, we also allow Eve to have unlimited computational power and communication to produce outputs (y_A, y_B). We denote the resulting conditional probability distribution as $\Pr\{(y_A, y_B \mid z_A, z_B)\}_{\sigma_{z_A,z_B}}$. As this is also a correlation, Eve has her own Bell term which we denote by $\tilde{J}_3$ and her own asynchronicity term which we denote by $\tilde{S}$.

Eve's goal is to program Alice and Bob's devices such that the device outputs pass statistical tests for estimating Bell violation and asynchronicity. The following theorem shows that Eve's uncertainty ϵ is upper-bounded by a function of the allowed errors in Alice and Bob's Bell and asynchronicity terms. If Eve's uncertainty exceeds a certain threshold then there does not exist a distribution $\Pr\{(y_A, y_B \mid z_A, z_B)\}_{\sigma_{z_A,z_B}}$ she can use to provide outputs to Alice and Bob that still pass their Bell and asynchronicity checks. We state the theorem formally as follows.

▶ **Theorem 12.** *Let $0 \leq \lambda < \frac{1}{8}$ be the allowed error in Alice and Bob's J_3 term, and $0 \leq \mu$ be their asynchronicity bound. On Eve's side, let $\tilde{J}_3$ and $\tilde{S}$ be analogous Bell inequality and asynchronicity terms for her correlation. Let ϵ be Eve's uncertainty about Alice and Bob's inputs as given in Equation (14), and δ be such that $0 \leq \delta$. If $\epsilon > \epsilon_{max}^{\delta}$, where*

$$\epsilon_{max}^{\delta} = \frac{2}{3} - \frac{2}{3}\left(\frac{\sqrt{144(\delta-1)\lambda + 64\lambda^2 + 6(36\delta + 8\lambda - 9)\mu - 72\mu^2 - 162\delta + 81}}{6\mu - 18\delta - 8\lambda + 9}\right),$$

then Eve's asynchronicity $\tilde{S} < \delta$.

Proof. For inputs $x_A, x_B \in \{0,1,2\}$, the correlation that Alice and Bob use to compute key bits and self-test their devices is then given by:

$$p(y_A, y_B \mid x_A, x_B) = \tag{15}$$
$$\sum_{z_A,z_B} \Pr\{y_A, y_B \mid z_A, z_B\}_{\sigma_{z_A,z_B}} \cdot \left\{ \begin{array}{ll} 1-\epsilon & \text{for } z_A = x_A \\ \\ \frac{\epsilon}{2} & \text{otherwise} \end{array} \right\} \cdot \left\{ \begin{array}{ll} 1-\epsilon & \text{for } z_B = x_B \\ \\ \frac{\epsilon}{2} & \text{otherwise} \end{array} \right\}$$

We begin by deriving expressions for the expected values of J_3 and S.

$$\begin{aligned}\langle 1 - J_3 \rangle = \frac{1}{4}\big(&p(0,1 \mid 0,1) + p(1,0 \mid 0,1) + p(0,1 \mid 1,0) + p(1,0 \mid 1,0) \\ &+ \; p(0,1 \mid 0,2) + p(1,0 \mid 0,2) + p(0,1 \mid 2,0) + p(1,0 \mid 2,0) \\ &+ \; p(0,1 \mid 1,2) + p(1,0 \mid 1,2) + p(0,1 \mid 2,1) + p(1,0 \mid 2,1)\big) \\ = \big(1 - \epsilon + \tfrac{3}{4}\epsilon^2\big)\,(1 - \tilde{J}_3) &+ \big(\tfrac{3}{2}\epsilon - \tfrac{9}{8}\epsilon^2\big)\,\tilde{S} \end{aligned} \tag{16}$$

A similar computation for S gives us:

$$\begin{aligned}\langle S \rangle = \frac{1}{3}\big(&p(0,1 \mid 0,0) + p(1,0 \mid 0,0) + p(0,1 \mid 1,1) \\ &+ p(1,0 \mid 1,1) + p(0,1 \mid 2,2) + p(1,0 \mid 2,2)\big) \\ = \big(1 - 2\epsilon + \tfrac{3}{2}\epsilon^2\big)\,\tilde{S} &+ \big(\tfrac{4}{3}\epsilon - \epsilon^2\big)\,(1 - \tilde{J}_3) \end{aligned} \tag{17}$$

Using Equations (16) and (17), we can solve for $\tilde{J}_3$ and $\tilde{S}$ as:

$$\begin{bmatrix} 1 - \tilde{J}_3 \\ \tilde{S} \end{bmatrix} = \begin{bmatrix} 1 - \epsilon + \frac{3}{4}\epsilon^2 & \frac{3}{2}\epsilon - \frac{9}{8}\epsilon^2 \\ \frac{4}{3}\epsilon - \epsilon^2 & 1 - 2\epsilon + \frac{3}{2}\epsilon^2 \end{bmatrix}^{-1} \begin{bmatrix} \frac{9}{8} - \lambda \\ \mu \end{bmatrix}$$

We get solutions:

$$\tilde{J}_3 = \frac{(3\epsilon^2 - 4\epsilon)(3 - 6\mu + 8\lambda) + 16\lambda - 2}{4\,(3\epsilon - 2)^2} \qquad \tilde{S} = \frac{(3\epsilon^2 - 4\epsilon)(6\mu - 8\lambda + 9) + 24\mu}{6\,(3\epsilon - 2)^2}. \tag{18}$$

Plugging $\tilde{S} = \delta$ in Equation (18), and solving for ϵ gives us:

$$\epsilon^{\delta}_{max} = \frac{2}{3} - \frac{2}{3}\left(\frac{\sqrt{144(\delta - 1)\lambda + 64\lambda^2 + 6(36\delta + 8\lambda - 9)\mu - 72\mu^2 - 162\delta + 81}}{6\mu - 18\delta - 8\lambda + 9}\right) \tag{19}$$

◀

▶ **Corollary 13.** *For $\epsilon > \epsilon^0_{max}$, there is no correlation Eve can use to produce a cheating strategy against Alice and Bob.*

Proof. Plugging in $\delta = 0$ in Equation (19),

$$\epsilon_{max} \coloneqq \epsilon^0_{max} = \frac{2}{3} - \frac{2}{3}\left(\frac{\sqrt{64\lambda^2 + 6(8\lambda - 9)\mu - 72\mu^2 - 144\lambda + 81}}{6\mu - 8\lambda + 9}\right).$$

If Eve's uncertainty $\epsilon > \epsilon_{max}$, then $\tilde{S} < 0$, and since no correlation can have negative asynchronicity, no such $\Pr\{(y_A, y_B \mid z_A, z_B)\}_{\sigma_{z_A, z_B}}$ exists. ◀

By the corollary above, we conclude that Eve's uncertainty cannot grow too much before her asynchronicity becomes negative, therefore resulting in an infeasible strategy. Fixing a reasonable threshold for the error allowed in the Bell term, say $\lambda = 0.05$, we plot values of ϵ_{max} against varying values of Alice and Bob's allowed asynchronicity μ in Figure 3. The plot shows that even for allowed asynchronicity $\mu = 5\%$, Eve must have close to perfect certainty $\approx 97\%$ about Alice and Bob's inputs. Thus even with unlimited computational and communication power, when $\epsilon > \epsilon_{max}$, no correlation exists to perfectly simulate statistics that pass Alice and Bob's Bell and asynchronicity checks.

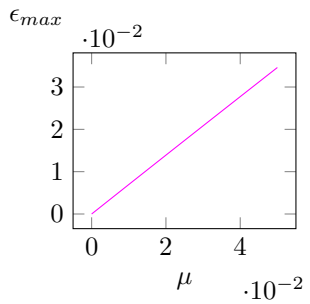


Figure 3 Values of μ vs. ϵ_{max} for which $\tilde{S}$ is non-negative

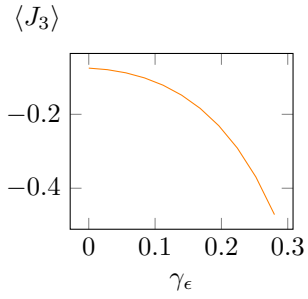


Figure 4 Values of $\langle J_3 \rangle$ vs. γ_ϵ for $\mu = 0.05, \lambda = 0.05$

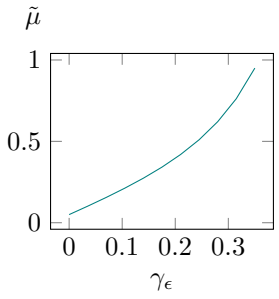


Figure 5 Values of $\langle S \rangle = \tilde{\mu}$ vs. γ_ϵ for $\mu = 0.05, \lambda = 0.05$

We further examine the regime where Eve's uncertainty $\epsilon > \epsilon_{max}$. In this case the best Eve can do in order to provide Alice and Bob an expected asynchronicity value $\langle S \rangle$ close to μ, is to use a synchronous correlation herself, i.e. $\tilde{S} = 0$. Fixing $\tilde{S} = 0$, we plot $\langle J_3 \rangle$ as Eve's uncertainty exceeds ϵ_{max}. Let $\gamma_\epsilon \coloneqq \epsilon - \epsilon_{max}$ denote how much Eve's uncertainty is above the maximum. Figure 4 shows that even with a lot of uncertainty, Eve can make $\langle J_3 \rangle$ as close to $-\frac{1}{8}$ as she likes. Since Eve is not restricted to quantum strategies, she can in fact violate the $-\frac{1}{8}$ bound. However, providing a $\langle J_3 \rangle$ value smaller than $-\frac{1}{8}$ is not in her best interest since Alice and Bob check if their estimated J_3 is in $[-\frac{1}{8}, -\frac{1}{8} + \lambda]$.

As a result, detecting Eve's interference depends *only* on the asynchronicity check. Since Eve's $\tilde{S} = 0$, she has to provide a value for Alice and Bob's $\langle S \rangle = \tilde{\mu}$ that is strictly larger than their decided error threshold μ. We use Equation (17) to plot the effect of increasing ϵ past ϵ_{max} on $\langle S \rangle = \tilde{\mu}$ for a fixed λ and μ. Figure 5 shows the comparison between γ_ϵ and $\tilde{\mu}$ for $\mu = 0.05$ and $\lambda = 0.05$. In our analysis the choice of 0.05 for both λ and μ is arbitrary, and is made to demonstrate the effect of increasing Eve's uncertainty ϵ on the expected value $\langle S \rangle$. Alice and Bob may pick any reasonable error values for their J_3 and S terms without affecting the following calculations. From Figure 5, we see that $\tilde{\mu}$ increases sharply as γ_ϵ increases, which in turn implies that Alice and Bob's asynchronicity test always fails except with negligible probability. We show this using a straightforward Chernoff argument and bounding the probability that Alice and Bob's output is asynchronous in fewer than a μ fraction of the asynchronicity check rounds. Formally, let's assume Alice and Bob have m asynchronicity check rounds. Let A_i be a $\{0, 1\}$ random variable denoting whether their output is asynchronous in round $i \in [m]$. Since Eve provides an asynchronous output with probability $\tilde{\mu}$, we have

$$A_i = \begin{cases} 1 & \text{with probability } \tilde{\mu}, \\ 0 & \text{otherwise.} \end{cases}$$

Let $A_S = \sum_i A_i$. Therefore $\langle A_S \rangle = \sum_i \langle A_i \rangle = m\tilde{\mu}$. Using a Chernoff bound we get

$$\Pr(A_S \leq m\mu) \leq \exp\left(-\frac{(\tilde{\mu} - \mu)^2 k}{2\tilde{\mu}}\right).$$

Alice and Bob can thus make this probability arbitrarily small by picking an appropriate value m for the number of asynchronicity check rounds they perform.

References

1 Antonio Acín, Nicolas Brunner, Nicolas Gisin, Serge Massar, Stefano Pironio, and Valerio Scarani. Device-independent security of quantum cryptography against collective attacks. *Physical Review Letters*, 98(23):230501, 2007.

2 Werner Oskar Amrein and Kalyan B Sinha. On pairs of projections in a Hilbert space. *Linear algebra and its applications*, 208:425–435, 1994.

3 Alex Arkhipov. Extending and characterizing quantum magic games. *arXiv preprint arXiv:1209.3819*, 2012.

4 Rotem Arnon-Friedman, Renato Renner, and Thomas Vidick. Simple and tight device-independent security proofs. *SIAM Journal on Computing*, 48(1):181–225, 2019.

5 Albrecht Böttcher and Ilya M Spitkovsky. A gentle guide to the basics of two projections theory. *Linear Algebra and its Applications*, 432(6):1412–1459, 2010.

6 Richard Cleve, Peter Høyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *Computational Complexity, 2004. Proceedings. 19th IEEE Annual Conference on*, pages 236–249. IEEE, 2004.

7 Andrea Coladangelo and Jalex Stark. Robust self-testing for linear constraint system games. *arXiv preprint arXiv:1709.09267*, 2017.

8 Frederic Dupuis, Omar Fawzi, and Renato Renner. Entropy accumulation. *Communications in Mathematical Physics*, 379(3):867–913, 2020.

9 Marissa Giustina, Marijn AM Versteegh, Sören Wengerowsky, Johannes Handsteiner, Armin Hochrainer, Kevin Phelan, Fabian Steinlechner, Johannes Kofler, Jan-Åke Larsson, Carlos Abellán, et al. Significant-loophole-free test of bell's theorem with entangled photons. *Physical review letters*, 115(25):250401, 2015.

10 Paul R Halmos. Two subspaces. *Transactions of the American Mathematical Society*, 144:381–389, 1969.

11 Bas Hensen, Hannes Bernien, Anaïs E Dréau, Andreas Reiserer, Norbert Kalb, Machiel S Blok, Just Ruitenberg, Raymond FL Vermeulen, Raymond N Schouten, Carlos Abellán, et al. Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526(7575):682–686, 2015.

12 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP* = RE. *arXiv preprint arXiv:2001.04383*, 2020.

13 Se-Jin Kim, Vern Paulsen, and Christopher Schafhauser. A synchronous game for binary constraint systems. *Journal of Mathematical Physics*, 59(3):032201, 2018.

14 Laura Mancinska and David Roberson. Graph homomorphisms for quantum players. In *9th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

15 N David Mermin. Simple unified form for the major no-hidden-variables theorems. *Physical review letters*, 65(27):3373, 1990.

16 Carl A Miller and Yaoyun Shi. Robust protocols for securely expanding randomness and distributing keys using untrusted quantum devices. *Journal of the ACM (JACM)*, 63(4):1–63, 2016.

17 Vern I Paulsen, Simone Severini, Daniel Stahlke, Ivan G Todorov, and Andreas Winter. Estimating quantum chromatic numbers. *Journal of Functional Analysis*, 270(6):2188–2222, 2016.

18 Asher Peres. Incompatible results of quantum measurements. *Physics Letters A*, 151(3-4):107–108, 1990.

19 Sandu Popescu and Daniel Rohrlich. Quantum nonlocality as an axiom. *Foundations of Physics*, 24(3):379–385, 1994.

20 Nishant Rodrigues and Brad Lackey. Nonlocal games, synchronous correlations, and Bell inequalities. *arXiv preprint arXiv:1707.06200v4*, 2020.

21 Lynden K Shalm, Evan Meyer-Scott, Bradley G Christensen, Peter Bierhorst, Michael A Wayne, Martin J Stevens, Thomas Gerrits, Scott Glancy, Deny R Hamel, Michael S Allman, et al. Strong loophole-free test of local realism. *Physical review letters*, 115(25):250402, 2015.

22 Ben F Toner and Dave Bacon. Communication cost of simulating Bell correlations. *Physical Review Letters*, 91(18):187904, 2003.

23 Umesh Vazirani and Thomas Vidick. Fully device-independent quantum key distribution. *Physical Review Letters*, 113(14):Art–No, 2014.

24 Thomas Vidick. Almost synchronous quantum correlations. *arXiv preprint arXiv:2103.02468*, 2021.

Rewindable Quantum Computation and Its Equivalence to Cloning and Adaptive Postselection

Ryo Hiromasa
Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Japan

Akihiro Mizutani
Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Japan

Yuki Takeuchi
NTT Communication Science Laboratories, NTT Corporation, Atsugi, Japan

Seiichiro Tani
NTT Communication Science Laboratories, NTT Corporation, Atsugi, Japan
International Research Frontiers Initiative (IRFI), Tokyo Institute of Technology, Japan

Abstract

We define rewinding operators that invert quantum measurements. Then, we define complexity classes RwBQP, CBQP, and AdPostBQP as sets of decision problems solvable by polynomial-size quantum circuits with a polynomial number of rewinding operators, cloning operators, and adaptive postselections, respectively. Our main result is that $\mathsf{BPP}^{\mathsf{PP}} \subseteq \mathsf{RwBQP} = \mathsf{CBQP} = \mathsf{AdPostBQP} \subseteq \mathsf{PSPACE}$. As a byproduct of this result, we show that any problem in PostBQP can be solved with only postselections of outputs whose probabilities are polynomially close to one. Under the strongly believed assumption that $\mathsf{BQP} \not\supseteq \mathsf{SZK}$, or the shortest independent vectors problem cannot be efficiently solved with quantum computers, we also show that a single rewinding operator is sufficient to achieve tasks that are intractable for quantum computation. In addition, we consider rewindable Clifford and instantaneous quantum polynomial time circuits.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Quantum computing, Postselection, Lattice problems

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.9

Related Version *Full Version*: https://arxiv.org/abs/2206.05434 [23]

Funding *Akihiro Mizutani*: Supported by JST, ACT-X Grant Number JPMJAX210O, Japan.
Yuki Takeuchi: Supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant Number JPMXS0118067394 and JPMXS0120319794 and JST [Moonshot R&D – MILLENNIA Program] Grant Number JPMJMS2061.
Seiichiro Tani: Supported by the Grant-in-Aid for Transformative Research Areas No.JP20H05966 of JSPS, JST [Moonshot R&D – MILLENNIA Program] Grant Number JPMJMS2061, and the Grant-in-Aid for Scientific Research (A) No.JP22H00522 of JSPS.

Acknowledgements We thank Yasuhiro Takahashi and Yusuke Aikawa for helpful discussions. We also thank Tomoyuki Morimae for fruitful discussions and pointing out Refs. [4, 5] to us.

1 Introduction

1.1 Background and Our Contribution

It is believed that universal quantum computers outperform their classical counterparts. There are two approaches to strengthening this belief. The first is to introduce tasks that seem intractable for classical computers but can be efficiently solved with quantum computers. For example, no known efficient classical algorithm can solve the integer factorization, but Shor's quantum algorithm [33] can do it efficiently. The second approach is to consider what

happens if classical computers can efficiently simulate the behaviors of quantum computers. So far, sampling tasks have often been considered in this approach [22]. It has been shown that if any probability distribution obtained from some classes of quantum circuits (e.g., instantaneous quantum polynomial time (IQP) circuits [12]) can be efficiently simulated with classical computers, then PH collapses to its second [17, 27] or third level [12, 3, 26, 24, 13, 34, 35, 18, 20, 25, 10, 21, 11], or BQP is in the second level of PH [28]. Since the collapse of PH and the inclusion of BQP in PH are considered to be unlikely, these results imply quantum advantages.

In this paper, we take the second approach. If efficient classical simulation of quantum measurements is possible, then the measurements become invertible because classical computation is reversible. From the analogy of the rewinding technique used in zero-knowledge (see e.g., [37, 7, 36]), we call such measurements rewindable measurements. They make quantum computation genuinely reversible and *incredibly* powerful. More formally, the following rewinding operator R becomes possible. R receives a post-measurement n-qubit quantum state $(|z\rangle\langle z| \otimes I^{\otimes n-1})|\psi\rangle$ with $z \in \{0,1\}$ and a classical description $\mathcal{D}$ of a pre-measurement quantum state $|\psi\rangle$ and outputs the quantum state $|\psi\rangle$:

$$R\left(\left(|z\rangle\langle z| \otimes I^{\otimes n-1}\right)|\psi\rangle \otimes |\mathcal{D}\rangle\right) = |\psi\rangle, \tag{1}$$

where $I \equiv |0\rangle\langle 0| + |1\rangle\langle 1|$ is the two-dimensional identity operator. As an important point, R requires the classical description $\mathcal{D}$ as an input. If it requires only $(|z\rangle\langle z| \otimes I^{\otimes n-1})|\psi\rangle$ as an input, the output state cannot be uniquely determined. For example, in the case of both $|\psi\rangle = |0\rangle|+\rangle$ and $(|0\rangle|+\rangle + |1\rangle|-\rangle)/\sqrt{2}$, the post-measurement state is $|0\rangle|+\rangle$ for $z = 0$, where $|\pm\rangle \equiv (|0\rangle \pm |1\rangle)/\sqrt{2}$. To circumvent this problem, we require the classical description $\mathcal{D}$ as information about $|\psi\rangle$. As a concrete example, the classical descriptions of $|0\rangle|+\rangle$ and $(|0\rangle|+\rangle + |1\rangle|-\rangle)/\sqrt{2}$ are $I \otimes H$ and $CZ(H \otimes H)$, respectively. Here, $H \equiv |+\rangle\langle 0| + |-\rangle\langle 1|$ is the Hadamard gate, $CZ \equiv |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z$ is the controlled-Z (CZ) gate, and $Z \equiv |0\rangle\langle 0| - |1\rangle\langle 1|$ is the Pauli-Z operator. These descriptions are proper because $|0\rangle|+\rangle$ and $(|0\rangle|+\rangle + |1\rangle|-\rangle)/\sqrt{2}$ can be prepared by applying $I \otimes H$ and $CZ(H \otimes H)$ on the fixed initial state $|00\rangle$, respectively. Furthermore, we define rewinding operators for only pure states, that is their functionality is arbitrary for mixed states. Due to this restriction, we can avoid contradictions with an ordinary interpretation of mixed states (see Sec. 3) and the no-signaling principle (see the full paper [23]).

It is strongly believed that the rewinding of measurements cannot be performed in ordinary quantum mechanics, i.e., the superposition is destroyed by measurements, and it cannot be recovered after measurements. One may think that if the rewinding were possible against this belief, it could add extra computation power to universal quantum computers. We show that this expectation is indeed correct. More formally, we define RwBQP (BQP with rewinding) as a set of decision problems solvable by polynomial-size quantum circuits with a polynomial number of rewinding operators and show $\mathsf{BQP} \subseteq \mathsf{BPP}^{\mathsf{PP}} \subseteq \mathsf{RwBQP}$.

The rewinding operator can be considered a probabilistic postselection. By just repeating measurements and rewinding operations until the target outcome is obtained, we can efficiently simulate the postselection with high probability if the output probability of the target outcome is at least the inverse of some polynomial. However, the original postselection enables us to deterministically obtain a target outcome even if the probability is exponentially small [2]. In this case, the above simple repeat-until-success approach requires an exponential number of rewinding operations on average. Surprisingly, we show that it is possible to exponentially mitigate probabilities of nontarget outcomes with a polynomial number of rewinding operators. By using this mitigation protocol, we can obtain the target outcome with high probability even if the output probability of the target outcome is exponentially small. In this sense, the rewinding and postselection are equivalent. More formally, we show

that RwBQP is equivalent to the class AdPostBQP (BQP with adaptive postselection) of decision problems solvable by polynomial-size quantum circuits with a polynomial number of adaptive postselections. Here, an adaptive postselection is a projector $|b\rangle\langle b|$ such that the value of $b \in \{0,1\}$ depends on previous measurement outcomes. From this equivalence, we also obtain RwBQP $\subseteq$ PSPACE.

The rewinding is also related to cloning. By strengthening our rewinding operator in Eq. (1), we define the cloning operator C as follows:

$$C|\mathcal{D}\rangle = |\psi\rangle. \tag{2}$$

Unlike Eq. (1), this operator does not require the post-measurement state $(|z\rangle\langle z| \otimes I^{\otimes n-1})|\psi\rangle$. Since it is easy to duplicate the classical description $\mathcal{D}$, we can efficiently duplicate $|\psi\rangle$, i.e., generate $|\psi\rangle^{\otimes 2}$ by simply applying $C \otimes C$ on $|\mathcal{D}\rangle^{\otimes 2}$. Although the ordinary cloning operator $\tilde{C}$ is defined such that $\tilde{C}(|\psi\rangle|0^m\rangle) = |\psi\rangle^{\otimes 2}$ for some $m \in \mathbb{N}$ [38], we define C as an operator whose input is the classical description $\mathcal{D}$ of $|\psi\rangle$ rather than $|\psi\rangle$ itself. This makes sense because we can always obtain a classical description of $|\psi\rangle$ in our setting[1]. Note that it could be difficult to realize C in quantum polynomial time because $|\psi\rangle$ might be prepared by using measurements. More precisely, it may be defined as a quantum state prepared when the measurement outcome is 0, e.g., $|\psi\rangle = U_2(I \otimes |0\rangle\langle 0| \otimes I^{\otimes n-2})U_1|0^n\rangle$ for some unitary operators U_1 and U_2. We show that RwBQP is also equivalent to the class CBQP (BQP with cloning) of decision problems solvable by polynomial-size quantum circuits with a polynomial number of cloning operators. That is, the difference between Eqs. (1) and (2) does not matter to the computation power. The following theorem summarizes our main results explained above:

▶ **Result 1** (Theorem 16). $\mathsf{BPP}^{\mathsf{PP}} \subseteq \mathsf{RwBQP} = \mathsf{CBQP} = \mathsf{AdPostBQP} \subseteq \mathsf{PSPACE}$.

The computation power of the cloning has been addressed in [4] as an open problem. Result 1 gives lower and upper bounds on our class CBQP, and it seems to be a reasonable approach to capturing the power of cloning.

All the above results assume that rewinding operators can be utilized a polynomial number of times. Under the strongly believed assumption that the shortest independent vectors problem (SIVP) [30] cannot be efficiently solved with universal quantum computers, we show that a single rewinding operator is sufficient to achieve a task that is intractable for universal quantum computation:

▶ **Result 2** (Informal Version of Theorem 22). *Assume that there is no polynomial-time quantum algorithm that solves the SIVP. Then, there exists a problem such that it can be efficiently solved with a constant probability if a single rewinding operator is allowed for quantum computation, but the probability is super-polynomially small if it is not allowed.*

We also show a superiority of a single rewinding operator under a different assumption:

▶ **Result 3** (Informal Version of Corollary 24). *Let* $\mathsf{RwBQP}(1)$ *be* RwBQP *with a single rewinding operator. Then,* $\mathsf{RwBQP}(1) \supset \mathsf{BQP}$ *unless* $\mathsf{BQP} \supseteq \mathsf{SZK}$.

It is strongly believed that BQP does not include SZK. At least, we can say that it is hard to show $\mathsf{BQP} \supseteq \mathsf{SZK}$ because there exists an oracle A such that $\mathsf{BQP}^A \not\supseteq \mathsf{SZK}^A$ [1]. For example, by assuming that the decision version of SIVP, gapSIVP, is hard for universal quantum computation, Result 3 implies that a single rewinding operator is sufficient to

[1] Since we are interested in complexity classes, we consider only quantum states generated from quantum circuits in uniform families.

achieve a task that is intractable for universal quantum computation. This is because the gapSIVP (with an appropriate parameter) is in SZK [29]. As a difference from Result 2, Result 3 shows the superiority of a single rewinding operator for promise problems.

As simple observations, we also consider the effect of rewinding operators for restricted classes of quantum circuits. It has been shown that polynomial-size Clifford circuits are classically simulatable [19]. We show that such circuits with rewinding operators are still classically simulatable (for details, see the full paper [23]). It is also known that IQP circuits are neither universal nor classically simulatable under plausible complexity-theoretic assumptions [13]. We show that IQP circuits with rewinding operators can efficiently solve any problem in RwBQP (for details, see the full paper [23]).

Our mitigation protocol used to show AdPostBQP $\subseteq$ RwBQP also has an application for PostBQP [2], which is a class of decision problems solvable by polynomial-size quantum circuits with non-adaptive postselections. By slightly modifying our mitigation protocol and replacing rewinding operators with postselections, we obtain the following corollary:

▶ **Result 4** (Corollary 20). *For any polynomial function $p(|x|)$ in the size $|x|$ of an instance x,* PP = PostBQP *holds even if only non-adaptive postselections of outputs whose probabilities are at least $1 - \Omega(1/p(|x|))$ are allowed.*

The equality PP = PostBQP was originally shown in [2] by using postselections of outputs whose probabilities may be exponentially small. Result 4 shows that such postselections can be replaced with those of outputs whose probabilities are polynomially close to one. This result is *optimal* in the sense that polynomially many repetitions of non-adaptive postselections of outputs whose probabilities are $1 - 1/f(|x|)$ with a super-polynomial function $f(|x|)$ can be simulated in quantum polynomial time. It is worth mentioning that when the probabilities are at least some constant, the above replacement is obvious in PostBPP (or $\mathsf{BPP}_{\mathsf{path}}$). This is because any behavior of a probabilistic Turing machine can be represented as a binary tree such that each path is chosen with probability 1/2. However, in its quantum analogue PostBQP, it was open as to whether such replacement is possible even if the probabilities are at least some constant.

Other related works and open problems are introduced in the full paper [23].

1.2 Overview of Techniques

To obtain Result 1, we show (i) RwBQP $\subseteq$ CBQP; (ii) CBQP $\subseteq$ AdPostBQP; (iii) AdPostBQP $\subseteq$ RwBQP; (iv) $\mathsf{BQP}^{\mathsf{PP}} \subseteq$ RwBQP, which immediately means $\mathsf{BPP}^{\mathsf{PP}} \subseteq$ RwBQP because $\mathsf{BPP}^{\mathsf{PP}} \subseteq \mathsf{BQP}^{\mathsf{PP}}$; and (v) AdPostBQP $\subseteq$ PSPACE. The first inclusion (i) is obvious from the definitions of the rewinding operator R and cloning operator C (see Eqs. (1) and (2)). The fifth inclusion (v) can also be easily shown by using the Feynman path integral that is used to show BQP $\subseteq$ PSPACE [9]. In BQP, measurements are only performed at the end of quantum circuits. On the other hand, in AdPostBQP, intermediate ordinary and postselection measurements are also allowed. However, this difference does not matter in showing the inclusion in PSPACE.

The second inclusion (ii) is a natural consequence from the simple observation that postselections can simulate the cloning operator C. On the other hand, the third inclusion (iii) is nontrivial because we have to efficiently simulate postselection by using only a polynomial number of rewinding operators. To this end, we give an efficient protocol to exponentially mitigate the amplitude of a nontarget state by using a polynomial number of rewinding operators. Let $|\psi\rangle = \sqrt{2^{-p(n)}}|\psi_t\rangle + \sqrt{1-2^{-p(n)}}|\psi_t^{\perp}\rangle$, where $p(n)$ is some polynomial in the size n of a given AdPostBQP problem, $|\psi_t\rangle$ is a target state that we would like to postselect, and $\langle\psi_t|\psi_t^{\perp}\rangle = 0$. By using our mitigation protocol, from $|\psi\rangle$, we can obtain

$\sqrt{2^{-p(n)}}|\psi_t\rangle + \sqrt{2^{-p(n)}(1-2^{-p(n)})}|\psi_t^\perp\rangle$ up to a normalization factor. Since $2^{-p(n)}$ is larger than $2^{-p(n)}(1-2^{-p(n)})$, we now obtain $|\psi_t\rangle$ with probability of at least $1/2$. By repeating these procedures, we can simulate the postselection of $|\psi_t\rangle$ with high probability.

Our mitigation protocol is also useful in showing the fourth inclusion (iv). First, from $\mathsf{PP} = \mathsf{PostBQP}$ [2], we obtain $\mathsf{PP} \subseteq \mathsf{RwBQP}$ by using our mitigation protocol. Then, we show that the completeness-soundness gap in RwBQP can be amplified to a constant exponentially close to 1, and RwBQP is closed under composition. By combining these results, we obtain $\mathsf{BQP}^{\mathsf{PP}} \subseteq \mathsf{RwBQP}^{\mathsf{RwBQP}} = \mathsf{RwBQP}$. Note that $\mathsf{BQP}^{\mathsf{PP}} \subseteq \mathsf{RwBQP}^{\mathsf{PP}}$ is obvious from the definition of RwBQP (see Def. 9).

We show Result 2 as follows. Cojocaru *et al.* have shown that under the hardness of SIVP, there exists a family $\mathcal{F} \equiv \{f_K\}_{K\in\mathcal{K}}$ of functions that is collision resistant against quantum computers, i.e., no polynomial-time quantum algorithm can output a collision pair (x, x') such that $x \neq x'$ and $f_K(x) = f_K(x')$ [15]. Here, $\mathcal{K}$ is a finite set of parameters uniquely specifying each function (see Sec. 2.2 for details). We show that a collision pair can be output with a constant probability if only one rewinding operator is given. From the construction of $\mathcal{F}$, the last bits of collision pairs are different, i.e., there exist x_0 and x_1 such that $x = (x_0, 0)$ and $x' = (x_1, 1)$. Using the idea in [14], we can efficiently prepare

$$\frac{|x_0\rangle|0\rangle + |x_1\rangle|1\rangle}{\sqrt{2}} \tag{3}$$

for some output value $y = f_K(x) = f_K(x')$. Note that since the preparation of Eq. (3) includes a measurement, if we perform it again, we will obtain a quantum state in Eq. (3) for a different output value y', and hence it is difficult to simultaneously obtain x and x' for the same y. When we can use a rewinding operator, the situation changes. By measuring the state in Eq. (3), we can obtain x_0 or x_1. For simplicity, suppose that we obtain x_0. Then, by performing the rewinding operator R on $|x_0\rangle|0\rangle$ and a classical description of Eq. (3), we can prepare the quantum state in Eq. (3) for the *same* y. From this state, we can obtain x_1 with probability $1/2$. As an important point, since the last bits of x and x' differ, a single rewinding operator (i.e., the rewinding of a single qubit) is sufficient to find a collision pair with a constant probability.

Finally, we show Result 3. To this end, we show that a SZK-complete problem is in $\mathsf{RwBQP}(1)$ by using a technique inspired by [5].

2 Preliminaries

In this section, we review some preliminaries that are necessary to understand our results. In Sec. 2.1, we introduce a complexity class $\mathsf{PostBQP}$ and explain the postselection. In Sec. 2.2, we introduce the SIVP and a collision-resistant and $\delta - 2$ regular family of functions.

2.1 Quantum complexity class

In this subsection, we review $\mathsf{PostBQP}$ and explain the postselection. Then, we clarify a difference between $\mathsf{PostBQP}$ and our class $\mathsf{AdPostBQP}$ (see Def. 11). Note that we assume that readers are familiar with classical complexity classes [8]. $\mathsf{PostBQP}$ is defined as follows:

▶ **Definition 5** ($\mathsf{PostBQP}$ [2]). *A promise problem $L = (L_{\text{yes}}, L_{\text{no}}) \subseteq \{0,1\}^*$ is in $\mathsf{PostBQP}$ if and only if there exist polynomials n and q and a uniform family $\{U_x\}_x$ of polynomial-size quantum circuits, such that*

- $\Pr[p = 1] \geq 1/2^q$
- *when $x \in L_{\text{yes}}$, $\Pr[o = 1 \mid p = 1] \geq 2/3$*
- *when $x \in L_{\text{no}}$, $\Pr[o = 1 \mid p = 1] \leq 1/3$,*

where o and p are called output and postselection registers, respectively. Here, for any $z_1 \in \{0,1\}$ and $z_2 \in \{0,1\}$,

$$\Pr[p = z_2] \equiv \langle 0^n | U_x^\dagger \left(I \otimes |z_2\rangle\langle z_2| \otimes I^{\otimes n-2} \right) U_x |0^n\rangle \tag{4}$$

$$\Pr[o = z_1 \mid p = z_2] \equiv \frac{\langle 0^n | U_x^\dagger \left(|z_1 z_2\rangle\langle z_1 z_2| \otimes I^{\otimes n-2} \right) U_x |0^n\rangle}{\Pr[p = z_2]}. \tag{5}$$

In this definition, "polynomial" means the one in the length $|x|$ of the instance x.

From Def. 5, we notice that the postselection is to apply a projector. In PostBQP, it is allowed to apply the projector $|1\rangle\langle 1|$ to the qubit in the postselection register at the end of a quantum circuit. Therefore, PostBQP is a set of promise problems solvable by polynomial-size quantum circuits (in uniform families) with a single non-adaptive postselection[2]. On the other hand, in AdPostBQP, we allow the application of a polynomial number of intermediate measurements and projectors. This means that the value $b \in \{0,1\}$ of a projector $|b\rangle\langle b|$ can depend on previous measurement outcomes, while it is determined before executing a quantum circuit in PostBQP.

2.2 SIVP

The SIVP with approximation factor γ (SIVP_γ) is defined as follows:

▶ **Definition 6** (SIVP_γ)**.** *Let n be any natural number and γ (≥ 1) be any real number. Given an n bases of a lattice L, output a set of n linearly independent lattice vectors of length at most $\gamma \cdot \lambda_n(L)$. Here, γ can depend on n, and $\lambda_n(L)$ is the nth successive minimum of L (i.e., the smallest r such that L has n linearly independent vectors of norm at most r).*

Since there is no known polynomial-time quantum algorithm to solve SIVP_γ for polynomial approximation factor, it is used as a basis of the security of lattice-based cryptography [30].

The hardness of the SIVP is also used to construct families of collision-resistant functions against universal quantum computers. From [15], we can immediately obtain the following theorem:

▶ **Theorem 7** (adapted from [15])**.** *Let n be any natural number, $q = 2^{5\lceil \log n \rceil + 21}$, $m = 23n + 5n\lceil \log n \rceil$, $\mu = 2mn\sqrt{23 + 5\log n}$, and $\mu' = \mu/m$, where $\lceil \cdot \rceil$ is the ceiling function. Let $K \equiv (A, As_0 + e_0) \in \mathcal{K}$ with $\mathcal{K}$ being the multiset $\{(A, As_0 + e_0)\}_{A \in \mathbb{Z}_q^{n \times m}, s_0 \in \mathbb{Z}_q^n, e_0 \in \chi'^m}$, where $\mathbb{Z}_q^{n \times m}$ be the set of $n \times m$ matrices each of whose entry is chosen from $\mathbb{Z}_q \equiv \{0, 1, \ldots, q-1\}$, and χ' is the set of integers bounded in absolute value by μ'. Assume that there is no polynomial-time quantum algorithm that solves $\mathrm{SIVP}_{p(n)}$ for some polynomial $p(n)$ in n. Then, the family $\mathcal{F} \equiv \{f_K : \mathbb{Z}_q^n \times \chi^m \times \{0,1\} \to \mathbb{Z}_q^m\}_{K \in \mathcal{K}}$ of functions*

$$f_K(s, e, c) \equiv As + e + c \cdot (As_0 + e_0) \pmod q, \tag{6}$$

where χ is the set of integers bounded in absolute value by μ, is collision resistant[3] and δ-2 regular[4] for a constant δ.

---

[2] Note that a polynomial number of postselections are allowed if they can be unified as a single non-adaptive postselection.

[3] Let $\mathcal{F} \equiv \{f_K : \mathcal{D} \to \mathcal{R}\}_{K \in \mathcal{K}}$ be a function family. We say that $\mathcal{F}$ is collision resistant if for any polynomial-time quantum algorithm A, which receives K and outputs two bit strings $x, x' \in \mathcal{D}$, the probability $\Pr_K[A(K) = (x, x')$ such that $x \neq x'$ and $f_K(x) = f_K(x')]$ is super-polynomially small. Note that K is chosen from $\mathcal{K}$ uniformly at random, and the probability is also taken over the randomness in A.

[4] Let $\mathcal{F} \equiv \{f_K : \mathcal{D} \to \mathcal{R}\}_{K \in \mathcal{K}}$ be a function family. For a fixed K, we say that $y \in \mathcal{R}$ has two preimages if there exist exactly two different inputs $x, x' \in \mathcal{D}$ such that $f(x) = f(x') = y$. Let $\mathcal{Y}_K^{(2)}$ be the set of y having two preimages for K. The function family $\mathcal{F}$ is said to be δ-2 regular when $\Pr_{K,x}[f_K(x) \in \mathcal{Y}_K^{(2)}] \geq \delta$, where K and x are chosen from $\mathcal{K}$ and $\mathcal{D}$, respectively, uniformly at random.

From Eq. (6), the function f_K has a collision pair[5] $(s, e, 1)$ and $(s + s_0, e + e_0, 0)$, and Theorem 7 shows that it is difficult to find them simultaneously. This function family will be used to show that a single rewinding operator is sufficient to achieve a task that seems difficult for universal quantum computers.

Note that in [15], the matrix A is constructed so that it has a trapdoor to efficiently invert $As + e$, and its distribution is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$. In Theorem 7, we consider a simplified variant of the function family of [15] in which the matrix A is chosen uniformly at random.

3 Computational Complexity of Rewinding

In this section, we show Results 1 and 4. To this end, first, we define the rewinding operator R and cloning operator C as follows:

▶ **Definition 8** (Rewinding and Cloning Operators). *Let n be any natural number, Q be any n-qubit linear operator composed of unitary operators and the Z-basis projective operators $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$, $\mathcal{D}$ be a classical description of the linear operator Q, and I be the single-qubit identity operator. The rewinding and cloning operators R and C are maps from a quantum state to a quantum state such that for any $s \in \{0,1\}$, when $\left(|s\rangle\langle s| \otimes I^{\otimes n-1}\right) Q|0^n\rangle \neq 0$,*

$$R\left(\frac{\left(|s\rangle\langle s| \otimes I^{\otimes n-1}\right) Q|0^n\rangle}{\sqrt{\langle 0^n|Q^\dagger \left(|s\rangle\langle s| \otimes I^{\otimes n-1}\right) Q|0^n\rangle}} \otimes |\mathcal{D}\rangle\right) = \frac{Q|0^n\rangle}{\sqrt{\langle 0^n|Q^\dagger Q|0^n\rangle}} \tag{7}$$

$$C|\mathcal{D}\rangle = \frac{Q|0^n\rangle}{\sqrt{\langle 0^n|Q^\dagger Q|0^n\rangle}}. \tag{8}$$

For other input states, the functionality of R and C is undefined, that is outputs are arbitrary n-qubit states. Particularly when it depends on a value of classical bits whether R and C are applied, we call them classically controlled rewinding and cloning operators, respectively.

Note that since the linear operator Q may include projective operators (e.g., $Q = U_2(I \otimes |0\rangle\langle 0| \otimes I^{\otimes n-2})U_1$ for some n-qubit unitary operators U_1 and U_2), in general, $Q^\dagger Q \neq I^{\otimes n}$. An example of classically controlled rewinding operators is an operator such that if a measurement outcome is 0, the identity operator is applied to the post-measurement state, but if the outcome is 1, the rewinding operator R is applied to it. Classically controlled rewinding and cloning operators play an important role in giving our main result.

Simply speaking, the rewinding operator R rewinds the state projected onto $|s\rangle$ to the state before the measurement. As an important point, Def. 8 implies that the rewinding operator R only works for pure states. The following contradiction for an ordinary interpretation of mixed states occurs without the restriction to pure states. Suppose that we measure a maximally mixed state $I/2$ in the computational basis[6], and then obtain the measurement outcome 0. In this case, it is natural that even if we rewind this measurement and perform the same measurement again, the outcome is always 0. However, if we define the rewinding operator R so that it also works for mixed states, then we can obtain $I/2$ from $|0\rangle$ with the rewinding operator, and the measurement on it may output 1. In other words, if the rewinding operator works for a mixed state ρ, we can measure ρ again and again, and thus

[5] Since q is larger than μ, the second element $e + e_0$ of the second input may not be in the set χ^m. Therefore, the probability of f_K having a collision pair is not 1.

[6] We sometimes call the Z basis the computational basis.

we obtain its information as much as we want without changing ρ. This situation contradicts with the natural interpretation that mixed states arise due to the lack of knowledge about them. Furthermore, the restriction to pure states would be useful in circumventing the contradiction with the no-signaling principle as explained in the full paper [23]. From Def. 8, it is easily observed that the cloning operator C can efficiently simulate the rewinding operator R.

By using the rewinding and cloning operators and postselections, we define three complexity classes – RwBQP (BQP with rewinding), CBQP (BQP with cloning), and AdPostBQP (BQP with adaptive postselection) – as follows:

▶ **Definition 9** (RwBQP and CBQP). *Let n and k be any natural number, ℓ be a polynomial in n, and $0 \leq s < c \leq 1$. A promise problem $L = (L_{\rm yes}, L_{\rm no}) \subseteq \{0,1\}^*$ is in $\mathsf{RwBQP}(c,s)(k)$ if and only if there exists a polynomial-time deterministic Turing machine that receives 1^n as an input and generates a ℓ-bit description $\tilde{\mathcal{D}}$ of an operator Q_n such that it consists of a polynomial number of elementary gates in a universal gate set, single-qubit measurements in the computational basis, and k (classically controlled) rewinding operators R defined in Def. 8 and satisfies, for the instance $x \in \{0,1\}^n$ and a polynomial m, that*

- *if $x \in L_{\rm yes}$, $\left|\left|\left(|1\rangle\langle 1| \otimes I^{\otimes n+m+\ell-1}\right) Q_n\left(|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle\right)\right|\right|^2 \geq c$*
- *if $x \in L_{\rm no}$, $\left|\left|\left(|1\rangle\langle 1| \otimes I^{\otimes n+m+\ell-1}\right) Q_n\left(|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle\right)\right|\right|^2 \leq s$.*

Here, $|||v\rangle|| \equiv \sqrt{\langle v|v\rangle}$ for any vector $|v\rangle$, and "polynomial" is the abbreviation of "polynomial in n." Particularly, for the set $\mathrm{poly}(n)$ of all polynomial functions, we denote $\bigcup_{k\in\mathrm{poly}(n)} \mathsf{RwBQP}(2/3,1/3)(k)$ as RwBQP.

By replacing R with the cloning operator C defined in Def. 8, $\mathsf{CBQP}(c,s)(k)$ and CBQP are defined in a similar way.

To perform a rewinding operator R to recover an intermediate state $|\psi\rangle$, a classical description $\mathcal{D}$ of $|\psi\rangle$ is neccessary. It can always be generated from $\tilde{\mathcal{D}}$ and measurement outcomes obtained before preparing $|\psi\rangle$. As in the case of BQP, computations performed to solve RwBQP problems can be written as uniform families of quantum circuits. A concrete circuit diagram of a RwBQP computation is given in Appendix A.

From Def. 9, we immediately obtain the following lemma:

▶ **Lemma 10.** $\mathsf{RwBQP} \subseteq \mathsf{CBQP}$.

Proof. The only difference between RwBQP and CBQP is whether the rewinding or cloning operator is allowed. Since the cloning operator C can exactly simulate the rewinding operator R, this lemma holds. ◀

▶ **Definition 11** (AdPostBQP). *Let n be any natural number, ℓ be a polynomial in n, and $0 \leq s < c \leq 1$. A promise problem $L = (L_{\rm yes}, L_{\rm no}) \subseteq \{0,1\}^*$ is in $\mathsf{AdPostBQP}(c,s)$ if and only if there exists a polynomial-time deterministic Turing machine that receives 1^n as an input and generates a ℓ-bit description $\tilde{\mathcal{D}}$ of an operator Q_n such that it consists of a polynomial number of elementary gates in a universal gate set, single-qubit measurements in the computational basis, and single-qubit projectors $|1\rangle\langle 1|$ and satisfies, for the instance $x \in \{0,1\}^n$ and a polynomial m, that*

- *if $x \in L_{\rm yes}$, $\left|\left|\left(|1\rangle\langle 1| \otimes I^{\otimes n+m+\ell-1}\right) \mathcal{N}[Q_n\left(|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle\right)]\right|\right|^2 \geq c$*
- *if $x \in L_{\rm no}$, $\left|\left|\left(|1\rangle\langle 1| \otimes I^{\otimes n+m+\ell-1}\right) \mathcal{N}[Q_n\left(|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle\right)]\right|\right|^2 \leq s$,*

where $\mathcal{N}[\cdot]$ denotes the normalization of the vector in the square brackets. Here, "polynomial" is the abbreviation of "polynomial in n." Note that for $1 \leq i \leq n$, a projector $|1\rangle\langle 1|_i$ on the ith qubit can be applied only when a quantum state $|\psi\rangle$ to be applied satisfies

$$||(|1\rangle\langle 1|_i)\,|\psi\rangle||^2 \geq 2^{-p(n)} \tag{9}$$

for a polynomial $p(n)$ in n. Particularly, we denote AdPostBQP$(2/3, 1/3)$ *as* AdPostBQP*.*

Note that Q_n can include adaptive postselections because depending on previous measurement outcomes, we can decide whether X is applied before and after applying $|1\rangle\langle 1|$. Here, $X \equiv |1\rangle\langle 0| + |0\rangle\langle 1|$ is the Pauli-X operator. It is worth mentioning that it is unknown whether the adaptive postselection can be efficiently done in PostBQP as discussed in [2]. Indeed, if it is possible, SZK $\subseteq$ PP should be immediately obtained from the argument in [4, 5], while it is a long-standing problem. Eq. (9) can be automatically satisfied by using standard gate sets whose elementary gates involve only square roots of rational numbers. From Defs. 9 and 11, we notice that the main difference between RwBQP, CBQP, and AdPostBQP is whether the rewinding or cloning operators or projectors are allowed.

From Def. 11, we immediately obtain the following lemma:

▶ **Lemma 12.** AdPostBQP $\subseteq$ PSPACE*.*

Proof. The proof is essentially the same as that of BQP $\subseteq$ PSPACE [9]. The details are given in Appendix B. ◀

The following three corollaries also hold:

▶ **Corollary 13.** RwBQP*,* CBQP*, and* AdPostBQP *are closed under complement.*

▶ **Corollary 14.** RwBQP $=$ RwBQP$(1 - 2^{-p(n)}, 2^{-p(n)})$, CBQP $=$ CBQP$(1 - 2^{-p(n)}, 2^{-p(n)})$, *and* AdPostBQP $=$ AdPostBQP$(1 - 2^{-p(n)}, 2^{-p(n)})$ *for any polynomial function $p(n)$ in the size n of a given instance x.*

▶ **Corollary 15.** RwBQP*,* CBQP*, and* AdPostBQP *are closed under composition.*

Since they are obvious from Defs. 9 and 11 and can be shown by using standard techniques, proofs are given in Appendix C.

In the rest of this section, we consider a relation between the rewinding, cloning, and postselection (i.e., RwBQP, CBQP, and AdPostBQP), and also obtain lower and upper bounds on them. More formally, we show the following theorem:

▶ **Theorem 16.** BPP$^{\text{PP}}$ $\subseteq$ RwBQP $=$ CBQP $=$ AdPostBQP $\subseteq$ PSPACE*.*

Proof. This theorem can be obtained by showing (i) RwBQP $\subseteq$ CBQP; (ii) CBQP $\subseteq$ AdPostBQP; (iii) AdPostBQP $\subseteq$ RwBQP; (iv) BQP$^{\text{PP}}$ $\subseteq$ RwBQP, which immediately means BPP$^{\text{PP}}$ $\subseteq$ RwBQP because BPP$^{\text{PP}}$ $\subseteq$ BQP$^{\text{PP}}$; and (v) AdPostBQP $\subseteq$ PSPACE. The inclusions (i) and (v) are already shown in Lemmas 10 and 12, respectively. The remaining inclusions (ii), (iii), and (iv) will be shown in Lemma 17 and Corollary 19. ◀

To simplify our argument in proofs of Lemma 17 and Theorem 18, we particularly consider the universal gate set $\{X, CH, CCZ\} \cup \{H_k \mid k \in \mathbb{Z}, -p(|x|) \leq k \leq p(|x|)\}$ with a polynomial $p(|x|)$ in the instance size $|x|$ of a given problem. Here, $CH \equiv |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes H$ is the controlled-Hadamard gate, $CCZ \equiv |0\rangle\langle 0| \otimes I^{\otimes 2} + |1\rangle\langle 1| \otimes CZ$ is the controlled-controlled-Z (CCZ) gate, and H_k is the generalized Hadamard gate such that $H_k|0\rangle = (|0\rangle + 2^k|1\rangle)/\sqrt{1+4^k}$ and $H_k|1\rangle = (2^k|0\rangle - |1\rangle)/\sqrt{1+4^k}$. Therefore, H_0 is the ordinary Hadamard gate H, and

hence, from [32], our gate set is universal. By using our universal gate set, we can make output probabilities of any Pauli-Z measurement in any polynomial-size quantum circuit 0 or at least $2^{-q(|x|)}$ for some polynomial $q(|x|)$. Due to this property, we can postselect any outcome for any polynomial-size quantum circuit [see Eq. (9)], which simplifies a proof of Lemma 17. Furthermore, by using this gate set, we can perform all quantum operations required in a proof of Theorem 18 without any approximation. Note that our argument can also be applied to other universal gate sets such as $\{H, T, CZ\}$ with $T \equiv |0\rangle\langle 0| + e^{i\pi/4}|1\rangle\langle 1|$ by using the Solovay-Kitaev algorithm [16].

We show the second inclusion (ii) (for the proof, see Appendix D):

▶ **Lemma 17.** $\mathsf{CBQP} \subseteq \mathsf{AdPostBQP}$.

As the first step to obtain inclusions (iii) and (iv), we show the following theorem:

▶ **Theorem 18.** $\mathsf{RwBQP} \supseteq \mathsf{PP}$.

The proof is given in Appendix E.

From Theorem 18, we obtain the following corollary (for the proof, see Appendix H):

▶ **Corollary 19.** $\mathsf{BQP}^{\mathsf{PP}} \subseteq \mathsf{AdPostBQP} \subseteq \mathsf{RwBQP}$.

As the most important difference between the proof of Theorem 18 and that of $\mathsf{PostBQP} \supseteq \mathsf{PP}$ in [2], we have not used postselections of outputs whose probabilities are exponentially small by proposing the mitigation protocol (see Appendix E). To state the difference more explicitly on the technical level, we show the following corollary (for the proof, see Appendix I):

▶ **Corollary 20.** *For any polynomial function $p(|x|)$ in the size $|x|$ of an instance x,* $\mathsf{PP} = \mathsf{PostBQP}$ *holds even if only non-adaptive postselections of outputs whose probabilities are at least $1 - \Omega(1/p(|x|))$ are allowed.*

4 Restricted Rewindable Quantum Computation

In Sec. 3, a polynomial number of rewinding operators was available. If the number is restricted to a constant, the question is: how is the rewinding useful for universal quantum computation? We show that a single rewinding operator is sufficient to solve the following problem with a constant probability, which seems hard for universal quantum computation:

▶ **Definition 21** (Collision-finding Problem)**.** *Given the function family $\mathcal{F} \equiv \{f_K\}_{K\in\mathcal{K}}$ in Theorem 7 and a parameter K for $\mathcal{F}$, output a pair (x, x') with $x, x' \in \mathbb{Z}_q^n \times \chi^m \times \{0,1\}$ such that (i) $x \neq x'$ and (ii) $f_K(x) = f_K(x')$.*

▶ **Theorem 22.** *Assume that a rewinding operator can be applied in one step, and there is no polynomial-time quantum algorithm solving* $\mathrm{SIVP}_{p(n)}$ *for some polynomial $p(n)$ in n. Then, the problem in Def. 21 can be solved with probability of at least $\delta/2(1 - o(1))$ by uniformly generated polynomial-size quantum circuits with a single rewinding operator, but it cannot be achieved without rewinding operators. Here, the probability is taken over the uniform distribution on $\mathcal{K}$ and the randomness used in a quantum circuit to solve the problem.*

The proof is given in Appendix J.

We also show a superiority of a single rewinding operator under a different assumption. To this end, we use the statistical difference (SD) problem, which is SZK-complete [31], and show the following theorem:

▶ **Theorem 23.** *The SD problem defined in Def. 25 is in* $\mathsf{RwBQP}(1/2-2^{-O(n^c)}, 2\cdot 2^{-O(n^c)})(1)$, *where n and c are the problem size and some positive constant as defined in Def. 25.*

The proof is given in Appendix K. From Theorem 23, we obtain the following corollary:

▶ **Corollary 24.** *Let* $\mathsf{RwBQP}(1) \equiv \bigcup_{1/(c-s)\in \mathrm{poly}(|x|)} \mathsf{RwBQP}(c,s)(1)$ *for the set* $\mathrm{poly}(|x|)$ *of all polynomial functions in the size $|x|$ of an instance x. Then,* $\mathsf{RwBQP}(1) \supset \mathsf{BQP}$ *unless* $\mathsf{BQP} \supseteq \mathsf{SZK}$.

Proof. From Theorem 23, if $\mathsf{RwBQP}(1) \subseteq \mathsf{BQP}$, then $\mathsf{SZK} \subseteq \mathsf{BQP}$. ◀

The implication of Theorem 22 and Corollary 24 is given in Appendix L. As simple observations, in the full paper [23], we consider the computational capability of rewinding operators for two restricted classes of quantum circuits: Clifford and IQP circuits.

References

1. S. Aaronson. Quantum lower bound for the collision problem. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 635–642, 2002.
2. S. Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 461(2063):3473–3482, 2005.
3. S. Aaronson and A. Arkhipov. The computational complexity of linear optics. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, pages 333–342, 2011.
4. S. Aaronson, A. Bouland, J. Fitzsimons, and M. Lee. The space "just above" BQP, 2014. arXiv:1412.6507.
5. S. Aaronson, A. Bouland, J. Fitzsimons, and M. Lee. The space "just above" BQP. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 271–280, 2016.
6. D. S. Abrams and S. Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for *NP*-complete and # *P* problems. *Phys. Rev. Lett.*, 81(18):3992, 1998.
7. A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 474–483, 2014.
8. S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
9. E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.
10. S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, 2018.
11. A. Bouland, B. Fefferman, C. Nirkhe, and U. Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15(2):159–163, 2019.
12. M. J. Bremner, R. Jozsa, and D. J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):459–472, 2011.
13. M. J. Bremner, A. Montanaro, and D. J. Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Phys. Rev. Lett.*, 117(8):080501, 2016.
14. A. Cojocaru, L. Colisson, E. Kashefi, and P. Wallden. Qfactory: Classically-instructed remote secret qubits preparation. In *Proceedings of the 25th Annual International Conference on the Theory and Application of Cryptology and Information Security*, pages 615–645, 2019.
15. A. Cojocaru, L. Colisson, E. Kashefi, and P. Wallden. On the possibility of classical client blind quantum computing. *Cryptography*, 5(1):3, 2021.

16 C. M. Dawson and M. A. Nielsen. The solovay-kitaev algorithm. *Quantum Information and Computation*, 6(1):81–95, 2006.

17 K. Fujii, H. Kobayashi, T. Morimae, H. Nishimura, S. Tamate, and S. Tani. Impossibility of classically simulating one-clean-qubit model with multiplicative error. *Phys. Rev. Lett.*, 120(20):200502, 2018.

18 X. Gao, S.-T. Wang, and L.-M. Duan. Quantum supremacy for simulating a translation-invariant ising spin model. *Phys. Rev. Lett.*, 118(4):040502, 2017.

19 D. Gottesman. The heisenberg representation of quantum computers. In *Group22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics*, pages 32–43, 1999.

20 C. S. Hamilton, R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex. Gaussian boson sampling. *Phys. Rev. Lett.*, 119(17):170501, 2017.

21 D. Hangleiter, J. Bermejo-Vega, M. Schwarz, and J. Eisert. Anticoncentration theorems for schemes showing a quantum speedup. *Quantum*, 2:65, 2018.

22 A. W. Harrow and A. Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203–209, 2017.

23 R. Hiromasa, A. Mizutani, Y. Takeuchi, and S. Tani. Rewindable quantum computation and its equivalence to cloning and adaptive postselection, 2022. arXiv:2206.05434.

24 A. P. Lund, A. Laing, S. Rahimi-Keshari, T. Rudolph, J. L. O'Brien, and T. C. Ralph. Boson sampling from a gaussian state. *Phys. Rev. Lett.*, 113(10):100502, 2014.

25 J. Miller, S. Sanders, and A. Miyake. Quantum supremacy in constant-time measurement-based computation: A unified architecture for sampling and verification. *Phys. Rev. A*, 96(6):062320, 2017.

26 T. Morimae, K. Fujii, and J. F. Fitzsimons. Hardness of classically simulating the one-clean-qubit model. *Phys. Rev. Lett.*, 112(13):130502, 2014.

27 T. Morimae, Y. Takeuchi, and H. Nishimura. Merlin-Arthur with efficient quantum Merlin and quantum supremacy for the second level of the Fourier hierarchy. *Quantum*, 2:106, 2018.

28 T. Morimae, Y. Takeuchi, and S. Tani. Sampling of globally depolarized random quantum circuit, 2019. arXiv:1911.02220.

29 C. Peikert and V. Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *Proceedings of the 28th International Cryptology Conference*, pages 536–553, 2008.

30 O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 84–93, 2005.

31 A. Sahai and S. Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2):196–249, 2003.

32 Y. Shi. Both toffoli and controlled-not need little help to do universal quantum computation, 2002. arXiv:quant-ph/0205115.

33 P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

34 Y. Takahashi, S. Tani, T. Yamazaki, and K. Tanaka. Commuting quantum circuits with few outputs are unlikely to be classically simulatable. *Quantum Information and Computation*, 16(3&4):251–270, 2016.

35 Y. Takeuchi and Y. Takahashi. Ancilla-driven instantaneous quantum polynomial time circuit for quantum supremacy. *Phys. Rev. A*, 94(6):062336, 2016.

36 D. Unruh. Computationally binding quantum commitments. In *Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 497–527, 2016.

37 J. Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

38 W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

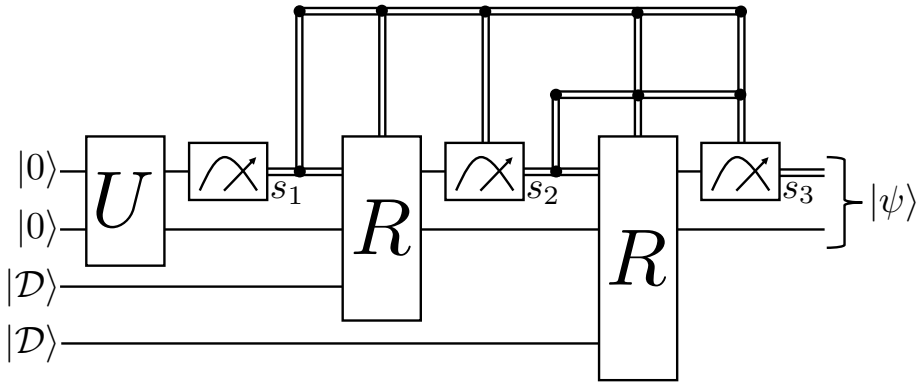


Figure 1 Concrete example of RwBQP computation. $\mathcal{D}$, U, R, and $|\psi\rangle$ are a classical description of $U|00\rangle$, a two-qubit unitary operator, the rewinding operator, and the output state, respectively. More precisely, when $U = \prod_i u_i$ for elementary quantum gates u_i in a universal gate set, $\mathcal{D}$ is a bit string representing $\prod_i u_i$. Note that since $|\psi\rangle$ is prepared by using only the unitary operator U, its classical description $\mathcal{D}$ does not include projectors and can be generated from only $\tilde{\mathcal{D}}$. Meter symbols represent the Pauli-Z measurements, and $s_i \in \{0,1\}$ is the ith measurement outcome for $1 \leq i \leq 3$. We represent $|s_i\rangle$ as a classical bit s_i to emphasize that it can be copied. When the first measurement outcome s_1 is 1, the first rewinding operator R is applied. On the other hand, when $s_1 = 0$, we do not apply R, because the target state is obtained. Since the second and third measurements and the second rewinding operator are applied only when $s_1 = 1$, they are also conditioned on s_1. In a similar way, since it is not necessary to apply the second rewinding operator if $s_2 = 0$, the second rewinding operator and the third measurement are also conditioned on s_2. Finally, $|\psi\rangle$ becomes the target state when $s_1 = 0$, $(s_1, s_2) = (1, 0)$, or $(s_1, s_2, s_3) = (1, 1, 0)$.

A Example of RwBQP Computation

Due to the addition of rewinding operators, it may be difficult to imagine quantum circuits used in RwBQP. To clarify them, as an example, we give a concrete circuit diagram for the following RwBQP computation. Suppose that we would like to prepare a two qubit state $(|0\rangle\langle 0| \otimes I)U|00\rangle$ (up to normalization) for a two-qubit unitary operator U. To this end, we use at most two classically controlled rewinding operators. More precisely, the rewinding operator R is applied if and only if the measurement outcome is 1. This computation can be depicted as a fixed quantum circuit in Fig. 1.

B Proof of Lemma 12

We give a proof of Lemma 12.

Proof. To show this lemma, it is sufficient to show that the acceptance probability

$$p_{\mathrm{acc}} = \left|\left|\left(|1\rangle\langle 1| \otimes I^{\otimes n+m+\ell-1}\right) \mathcal{N}\left[Q_n\left(|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle\right)\right]\right|\right|^2 \tag{10}$$

can be computed in polynomial space. To this end, we use the Feynman path integral. Let k be some polynomial in n. By using q_i that is an elementary gate in a universal gate set, a single-qubit measurement in the computational basis, or a single-qubit postselection onto $|1\rangle\langle 1|$ for $1 \leq i \leq k$, we can decompose $\mathcal{N}[Q_n(\cdot)]$ as $\mathcal{N}[Q_n(\cdot)] = \prod_{i=1}^{k} q_i(\cdot)$. Let $N \equiv n+m+\ell$.

Therefore,

$$\begin{aligned}&\left(|1\rangle\langle 1|\otimes I^{\otimes N-1}\right)\mathcal{N}[Q_n\left(|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle\right)]\\ =\ &\left[|1\rangle\langle 1|\otimes\left(\sum_{d\in\{0,1\}^{N-1}}|d\rangle\langle d|\right)\right]q_k\left[\prod_{i=1}^{k-1}\left(\sum_{s^{(i)}\in\{0,1\}^N}|s^{(i)}\rangle\langle s^{(i)}|\right)q_i\right]|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle.\end{aligned} \tag{11}$$

Let s be a shorthand notation of a $(k-1)N$-bit string $s^{(1)}s^{(2)}\dots s^{(k-1)}$. By defining

$$g(s,d)\equiv\langle 1|\langle d|\left[q_k\left(\prod_{i=1}^{k-1}|s^{(i)}\rangle\langle s^{(i)}|q_i\right)\right]|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle, \tag{12}$$

p_{acc} can be written as

$$\sum_{s,\tilde{s}\in\{0,1\}^{(k-1)N},d\in\{0,1\}^{N-1}}g(s,d)g^*(\tilde{s},d). \tag{13}$$

Sine q_i is just a constant-size matrix, each term $g(s,d)g^*(\tilde{s},d)$ can be computed in polynomial space[7]. Therefore, Eq. (13) can also be computed in polynomial space. ◀

C Proofs of Corollaries 13, 14, and 15

The proof of Corollary 13 is as follows:

Proof. Since proofs are essentially the same for all three classes, we only write a concrete proof for RwBQP. Let $\overline{L}$ be the complement of L. From Def. 9, when $x\in\overline{L}_{\mathrm{yes}}$ (i.e., $x\in L_{\mathrm{no}}$),

$$\left|\left|\left(|1\rangle\langle 1|\otimes I^{\otimes n+m+\ell-1}\right)\left(X\otimes I^{\otimes n+m+\ell-1}\right)Q_n\left(|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle\right)\right|\right|^2\geq 2/3. \tag{14}$$

On the other hand, when $x\in\overline{L}_{\mathrm{no}}$ (i.e., $x\in L_{\mathrm{yes}}$),

$$\left|\left|\left(|1\rangle\langle 1|\otimes I^{\otimes n+m+\ell-1}\right)\left(X\otimes I^{\otimes n+m+\ell-1}\right)Q_n\left(|x\rangle|0^m\rangle|\tilde{\mathcal{D}}\rangle\right)\right|\right|^2\leq 1/3. \tag{15}$$

Therefore, coRwBQP $\subseteq$ RwBQP. By using the same argument, we can also show coRwBQP $\supseteq$ RwBQP and thus RwBQP = coRwBQP. ◀

The proof of Corollary 14 is as follows:

Proof. Since proofs are essentially the same for all three classes, we only write a concrete proof for RwBQP. By repeating the same RwBQP computation m times and taking the majority vote on the outcomes, due to the Chernoff bound [8], the error probability is improved from 1/3 to $2^{-q(m)}$ for a positive polynomial function $q(m)$ in m. Therefore, by setting m so that $q(m)\geq p(n)$, we obtain this corollary. ◀

The proof of Corollary 15 is as follows:

Proof. Since proofs are essentially the same for all three classes, we only write a concrete proof for RwBQP. From Def. 9, when a polynomial-time algorithm calls another polynomial-time algorithm as a subroutine, the resultant algorithm can still be realized in polynomial time. Since the RwBQP computation has some error probability, a remaining concern is that errors may accumulate every time polynomial-time algorithms are called. However, the accumulation of errors is negligible from Corollary 14. As a result, we obtain $\mathsf{RwBQP}^{\mathsf{RwBQP}}=\mathsf{RwBQP}$. ◀

[7] It may not be able to be computed in polynomial time, because q_i may be a measurement.

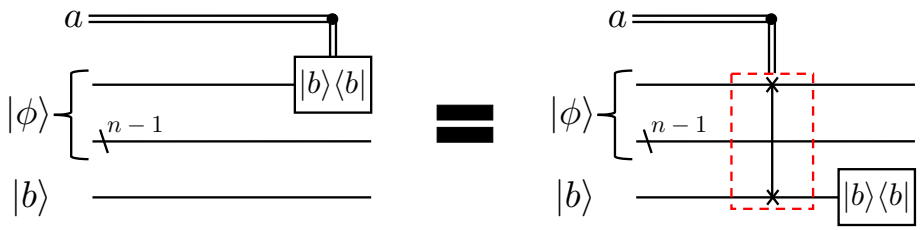


Figure 2 Replacement of a classically controlled projector with the classically controlled SWAP gate. $|\phi\rangle$ is a quantum state immediately before applying $P^{(i)} = |b\rangle\langle b|$, where $b \in \{0,1\}$. The SWAP gate is depicted as a vertical line enclosed by a dotted red rectangle. The projector $|b\rangle\langle b|$ and the SWAP gate is applied only when $a \in \{0,1\}$ is 1.

D Proof of Lemma 17

We give a proof of Lemma 17.

Proof. To obtain Lemma 17, it is sufficient to show that for any polynomial-size linear operator Q and its classical description $\mathcal{D}$, the cloning operator C can be simulated in quantum polynomial time by using the postselection. That is, our purpose is to perform the cloning operator C on the input state $|\mathcal{D}\rangle$. Let m be the number of Z-basis projective operators included in Q. By using n-qubit unitary operators $\{U^{(i)}\}_{i=1}^{m+1}$ and Z-basis projective operators $\{P^{(i)}\}_{i=1}^{m}$, $Q = U^{(m+1)} \prod_{i=1}^{m} (P^{(i)} U^{(i)})$. We can obtain the classical description $\mathcal{D}$ of Q by measuring the state $|\mathcal{D}\rangle$ in the Pauli-Z basis. The description $\mathcal{D}$ informs us about whether $P^{(i)}$ is $|0\rangle\langle 0|$ or $|1\rangle\langle 1|$ and how to construct $U^{(i)}$ from $\{X, H_k, CH, CCZ\}$ for all i. Therefore, by using the postselection, we can prepare $Q|0^n\rangle$ (up to normalization) in quantum polynomial time. When we would like to apply $U^{(i)}$, we just apply it. On the other hand, when we apply $P^{(i)}$, we use the postselection. Since we assume the universal gate set $\{X, H_k, CH, CCZ\}$, the postselection is possible in any case. These efficient procedures simulate the non classically-controlled cloning operator C.

We next show that the above procedures can also be applied to simulate a classically controlled cloning operator. To this end, a classically controlled postselection is necessary. Suppose that when $a \in \{0,1\}$ is 1, we would like to apply the cloning operator C. On the other hand, when $a = 0$, we do not apply C. Note that without loss of generality, we can assume that C is controlled by a single bit a because C is applied or not. Only when $a = 1$, we must apply $P^{(i)}$ to simulate the classically controlled cloning operator. Let $P^{(i)} = |b\rangle\langle b|$ for $b \in \{0,1\}$. Such classically controlled $P^{(i)}$ can be simulated by adding an ancillary qubit $|b\rangle$ and applying the classically controlled SWAP gate as shown in Fig. 2. Classically controlled quantum gates are allowed in AdPostBQP computation because any classically controlled quantum gate can be realized by combining elementary quantum gates in a universal gate set. In conclusion, we obtain CBQP $\subseteq$ AdPostBQP. ◀

E Proof of Theorem 18

We give a proof of Theorem 18.

Proof. We show Theorem 18 by replacing the postselection used in the proof of PP $\subseteq$ PostBQP in [2] with a polynomial number of rewinding operators. To this end, we consider the following PP-complete problem [2]: let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function computable in classical polynomial time, and $s \equiv \sum_{x \in \{0,1\}^n} f(x)$. Decide $0 < s < 2^{n-1}$ or $s \geq 2^{n-1}$. Note that it is promised that one of them is definitely satisfied.

To solve this problem with an exponentially small error probability using rewinding operators, first, we prepare

$$\sqrt{p}|\psi_t^{\perp}\rangle + \sqrt{1-p}|\psi_t\rangle, \tag{16}$$

where

$$p \equiv \frac{2\alpha^2(2^n-s)^2+\beta^2 4^n}{2[(2^n-s)^2+s^2]} \tag{17}$$

$$|\psi_t^{\perp}\rangle \equiv \frac{\sqrt{2}\alpha(2^n-s)|0\rangle+\beta 2^n|1\rangle}{\sqrt{2\alpha^2(2^n-s)^2+\beta^2 4^n}}\otimes|0\rangle \tag{18}$$

$$|\psi_t\rangle \equiv \frac{\sqrt{2}\alpha s|0\rangle+\beta(2^n-2s)|1\rangle}{\sqrt{2\alpha^2 s^2+\beta^2(2^n-2s)^2}}\otimes|1\rangle \tag{19}$$

for positive real numbers α and β such that $\alpha^2+\beta^2=1$ and $\beta/\alpha=2^k$, where k is an integer whose absolute value is upper bounded by n. As shown in Appendix F, this preparation can be done in quantum polynomial time with probability of at least $1-1/2^n$.

If the second qubit in Eq. (16) is projected onto $|1\rangle$, we can obtain

$$|\phi_{\beta/\alpha}\rangle \equiv \frac{\sqrt{2}\alpha s|0\rangle+\beta(2^n-2s)|1\rangle}{\sqrt{2\alpha^2 s^2+\beta^2(2^n-2s)^2}}. \tag{20}$$

Aaronson has shown that if n copies of $|\phi_{\beta/\alpha}\rangle$ can be prepared for all $-n\le k\le n$, then we can decide whether $0<s<2^{n-1}$ or $s\ge 2^{n-1}$ with an exponentially small error probability p_{err} in quantum polynomial time [2].

However, since p may be exponentially close to 1, the efficient preparation of Eq. (20) is difficult without postselection. We resolve this problem by using rewinding operators. Our idea is to amplify the probability of $|1\rangle$ being observed by mitigating the probability of $|0\rangle$ being observed. We propose the following mitigation protocol:

1. Set $i=0$ and $c=0$.

2. By using the state in Eq. (16), prepare
$$\sqrt{p_i}|\psi_t^{\perp}\rangle|+\rangle + \sqrt{1-p_i}|\psi_t\rangle|0\rangle, \tag{21}$$
where $p_0=p$, and measure the last register in the Pauli-Z basis. Let z be the measurement outcome. Furthermore, replace c with $c+1$.

3. Depending on the values of z, i, and c, perform one of following steps:

a. When $z=0$, replace i with $i+1$, reset c to 0, and obtain
$$\sqrt{p_{i+1}}|\psi_t^{\perp}\rangle + \sqrt{1-p_{i+1}}|\psi_t\rangle, \tag{22}$$
where
$$p_{i+1}=\frac{p_i}{2-p_i}. \tag{23}$$
If $i+1<2n+3$, do step 2 by using the state in Eq. (22). On the other hand, if $i+1=2n+3$, output the state in Eq. (22) and halt the mitigation protocol.

b. When $z=1$ and $c<3n$, apply the rewinding operator R and do step 2 again for the same i.

c. When $z=1$ and $c=3n$, answer $0<s<2^{n-1}$ or $s\ge 2^{n-1}$ uniformly at random, and halt the mitigation protocol.

In this protocol, i and c count how many times the mitigation succeeds and how many times the mitigation fails for a single i, respectively. From Eq. (23),

$$\frac{1-p_{i+1}}{p_{i+1}}=2\frac{1-p_i}{p_i}, \tag{24}$$

and hence we succeed in mitigating the amplitude of the nontarget state $|\psi_t^{\perp}\rangle$.

From Appendix G,

$$\sqrt{p_{2n+3}}|\psi_t^{\perp}\rangle + \sqrt{1-p_{2n+3}}|\psi_t\rangle \tag{25}$$

is output with probability of at least $1-5n/8^n$. Since $(1-p_{2n+3})/p_{2n+3} = 2^{2n+3}(1-p)/p \geq 1$ as shown in Appendix G, we can obtain the outcome 1 with probability of at least $1/2$ by measuring the second qubit in Eq. (25). If we obtain 0, we do the same measurement again by using the rewinding operator. Therefore, by repeating this procedure n times, we obtain the outcome 1 with probability of at least $1-1/2^n$. In total, with probability of at least

$$p_{\mathrm{suc}} \equiv \left[\left(1-\frac{1}{2^n}\right)^2\left(1-\frac{5n}{8^n}\right)\right]^{n(2n+1)}, \tag{26}$$

we obtain n copies of $|\phi_{\beta/\alpha}\rangle$ for all $-n \leq k \leq n$. As a result, we can correctly decide whether $0 < s < 2^{n-1}$ or $s \geq 2^{n-1}$ in polynomial time with probability of at least $p_{\mathrm{suc}}(1-p_{\mathrm{err}})$ that is exponentially close to 1. ◀

F Preparation of State in Eq. (16)

Although the procedure in this appendix has been proposed in [2], we explain it for the completeness of our paper. First, we prepare

$$\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}(H^{\otimes n}|x\rangle)|f(x)\rangle \tag{27}$$

in quantum polynomial time. Then, we measure all n qubits in the first register in the Pauli-Z basis. We repeat these procedures until we obtain the outcome 0^n or the repetition number reaches n. Since the probability[8] of 0^n being output in each repetition is at least $1/2$, we can obtain at least one 0^n with probability of at least $1-1/2^n$. When the measurement outcome is 0^n, we obtain

$$|\psi\rangle \equiv \frac{(2^n-s)|0\rangle + s|1\rangle}{\sqrt{(2^n-s)^2+s^2}}. \tag{28}$$

From this state, for any positive real numbers α and β such that $\alpha^2+\beta^2=1$ and $\beta/\alpha = 2^k$, where k is an integer whose absolute value is upper bounded by n, we can prepare

$$CH[(\alpha|0\rangle+\beta|1\rangle)|\psi\rangle] = \alpha|0\rangle|\psi\rangle + \beta|1\rangle H|\psi\rangle = \sqrt{p}|\psi_t^{\perp}\rangle + \sqrt{1-p}|\psi_t\rangle \tag{29}$$

in quantum polynomial time.

G Success Probability of Our Mitigation Protocol

We show that the probability that the state in Eq. (25) is output in our mitigation protocol is at least $1-5n/8^n$ and that $(1-p_{2n+3})/p_{2n+3} \geq 1$. For clarity, we show a schematic diagram of our mitigation protocol in Fig. 3.

[8] In previous calculations [2, 6], this probability is lower bounded by $1/4$. However, by calculating it more precisely, we tighten the lower bound.

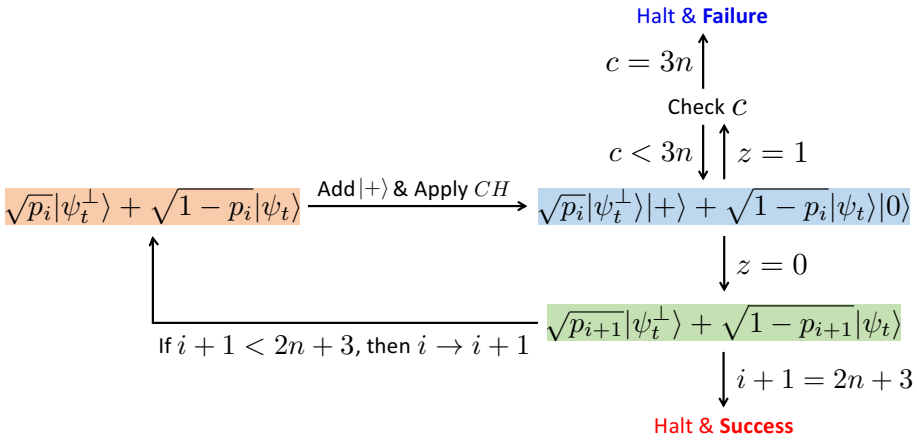


Figure 3 Schematic diagram of our mitigation protocol. **Failure** and **Success** indicate step (c) and the case that the state in Eq. (25) is output, respectively.

When the outcome 0 is obtained by measuring the last register of Eq. (21) in the Z basis, the amplitude of the nontarget state $|\psi_t^\perp\rangle$ is mitigated with the factor $1/\sqrt{2}$ (up to normalization) because

$$\left(I^{\otimes 2} \otimes \langle 0|\right)\left(\sqrt{p_i}|\psi_t^\perp\rangle|+\rangle + \sqrt{1-p_i}|\psi_t\rangle|0\rangle\right) = \sqrt{\frac{p_i}{2}}|\psi_t^\perp\rangle + \sqrt{1-p_i}|\psi_t\rangle. \tag{30}$$

Therefore, for any i, the probability q_i that the outcome 0 is obtained by measuring the last register of Eq. (21) in the Z basis is

$$q_i = \frac{p2^{-(i+1)} + (1-p)}{1-(1-2^{-i})p}, \tag{31}$$

where we have used $p_0 = p$. Therefore, for any i, the probability that we obtain 0 by measuring the last register of Eq. (21) in the Z basis before or at $c = 3n$ is

$$1 - (1-q_i)^{3n} \geq 1 - (1-q_0)^{3n} = 1 - \left(\frac{p}{2}\right)^{3n}. \tag{32}$$

Our purpose is to sufficiently mitigate the amplitude of $|\psi_t^\perp\rangle$ so that we obtain the outcome 1 by measuring the second register of Eq. (22) in the Z basis with probability of at least $1/2$. To this end, it is sufficient to run our mitigation protocol until $i = N$ such that

$$\frac{1-p_N}{p_N} \geq 1. \tag{33}$$

From Eq. (24), this condition can be satisfied by setting

$$N = \left\lceil \log\left(\frac{p}{1-p}\right)\right\rceil. \tag{34}$$

By combining Eqs. (32) and (34), the probability that the state in Eq. (25) is output in our mitigation protocol (i.e., the probability of our mitigation protocol reaching to **Success** in Fig. 3) is at least

$$\left[1 - \left(\frac{p}{2}\right)^{3n}\right]^N \geq 1 - \left[\log\left(\frac{p}{1-p}\right) + 1\right]\left(\frac{p}{2}\right)^{3n} \tag{35}$$

$$\geq 1 - \left[\log\left(\frac{p}{1-p}\right) + 1\right]\left(\frac{1}{2}\right)^{3n}. \tag{36}$$

Since $\log[p/(1-p)]$ is a monotonically increasing function of p in the range of $0 < p < 1$, the remaining task is to upper bound p. (Recall that our goal in this appendix is to show that Eq.(36) is lower bounded by $1 - 5n/8^n$ and $N \leq 2n + 3$.)

From the simple observation that $2[(2^n - s)^2 + s^2]$, which is the denominator of p in Eq. (17), is a symmetric convex downward function that becomes minimum at $s = 2^{n-1}$, and that $(2^n - s)^2$ in the numerator of p is a monotonically decreasing function in the range of $0 < s \leq 2^n$, the value of s maximizing p (for any α, β, and n) is between 1 and 2^{n-1}. To upper bound p, we separately consider three cases: (i) $1 \leq s \leq (1 - 1/\sqrt{2})2^n$, (ii) $(1 - 1/\sqrt{2})2^n < s \leq 2^{n-1} - 1$, and (iii) $s = 2^{n-1}$.

(i) When $1 \leq s \leq (1 - 1/\sqrt{2})2^n$, the inequality $2(2^n - s)^2 \geq 4^n$ holds. Therefore, from Eq. (17),

$$\begin{aligned} p &\leq \frac{(2^n - s)^2}{(2^n - s)^2 + s^2} && (37)\\ &\leq \frac{(2^n - 1)^2}{(2^n - 1)^2 + 1} && (38)\\ &= 1 - \frac{1}{(2^n - 1)^2 + 1}. && (39) \end{aligned}$$

(ii) When $(1 - 1/\sqrt{2})2^n < s \leq 2^{n-1} - 1$, the inequality $2(2^n - s)^2 < 4^n$ holds, and hence

$$\begin{aligned} p &\leq \frac{4^n}{2[(2^n - s)^2 + s^2]} && (40)\\ &\leq \frac{4^n}{2[(2^{n-1} + 1)^2 + (2^{n-1} - 1)^2]} && (41)\\ &= 1 - \frac{1}{4^{n-1} + 1}. && (42) \end{aligned}$$

(iii) When $s = 2^{n-1}$,

$$\begin{aligned} p &= \frac{\alpha^2}{2} + \beta^2 && (43)\\ &\leq 1 - \frac{1}{2(4^n + 1)}, && (44) \end{aligned}$$

where we have used $2^{-n} \leq \beta/\alpha \leq 2^n$ and $\alpha^2 + \beta^2 = 1$.

From Eqs. (39), (42), and (44), $p \leq 1 - 1/[2(4^n + 1)] \equiv p_{\max}$, and hence

$$N \leq \log \frac{p}{1-p} + 1 \leq \log \frac{p_{\max}}{1 - p_{\max}} + 1 \leq 2n + 3. \quad (45)$$

This implies that Eq.(36) is lower bounded by

$$1 - \left[\log\left(\frac{p_{\max}}{1 - p_{\max}}\right) + 1\right]\left(\frac{1}{2}\right)^{3n} \geq 1 - \frac{2n+3}{2^{3n}} \geq 1 - \frac{5n}{8^n}. \quad (46)$$

H Proof of Corollary 19

We give a proof of Corollary 19.

Proof. From Lemmas 10 and 17, it is sufficient to show $\mathsf{BQP}^{\mathsf{PP}} \subseteq \mathsf{RwBQP}$ and $\mathsf{AdPostBQP} \subseteq \mathsf{RwBQP}$ to obtain Corollary 19. First, we show the former inclusion. From Def. 9, it is obvious that any process in BQP can be simulated by a process in RwBQP in polynomial time. Furthermore, from Corollary 14 and Theorem 18, the PP oracle can be replaced with the RwBQP oracle. Therefore, from Corollary 15, $\mathsf{BQP}^{\mathsf{PP}} \subseteq \mathsf{RwBQP}^{\mathsf{RwBQP}} = \mathsf{RwBQP}$.

To obtain the latter inclusion, we show that each postselection can be simulated by rewinding operators. From Def. 11, each postselection (i.e., each projector) acts on a single qubit. Therefore, we can write a quantum state immediately before a postselection as $\alpha|0\rangle_p|\psi_0\rangle + \beta|1\rangle_p|\psi_1\rangle$ for some quantum states $|\psi_0\rangle$ and $|\psi_1\rangle$ and complex numbers α and β such that $|\alpha|^2 + |\beta|^2 = 1$. Here, the subscript p denotes the postselection register. Although $|\beta|^2$ may be exponentially small, the postselection onto $|1\rangle$ can be simulated by using our mitigation protocol. ◀

I Proof of Corollary 20

We give a proof of Corollary 20.

Proof. We can obtain Corollary 20 by slightly modifying our mitigation protocol and showing that it can be realized with non-adaptive postselections of outputs whose probabilities are at least $q \equiv 1 - \Omega(1/p(|x|))$. For some natural number m, we prepare

$$\sqrt{p}|\psi_t^{\perp}\rangle \left(\sqrt{q}|0\rangle + \sqrt{1-q}|1\rangle\right)^{\otimes m} + \sqrt{1-p}|\psi_t\rangle|0\rangle^{\otimes m} \tag{47}$$

instead of the state in Eq. (21). By postselecting m qubits in the second register onto $|0\rangle$ one by one, we obtain

$$\frac{\sqrt{pq^m}|\psi_t^{\perp}\rangle + \sqrt{1-p}|\psi_t\rangle}{\sqrt{1-p+pq^m}}. \tag{48}$$

These m postselections are non-adaptive ones of outputs whose probabilities are at least q. If the amplitude of $|\psi_t\rangle$ in Eq. (48) is at least $\sqrt{1/2}$, we can obtain $|\psi_t\rangle$ with at least a constant probability, and hence we can solve the PP-complete problem. Such the amplitude is realized by setting $m \geq \log\left[p/(1-p)\right]/\log\left(1/q\right)$. Since $\log\left[p/(1-p)\right]/\log\left(1/q\right) \leq 2(n+1)/\log\left(1/q\right) \leq 2(n+1)O(p(|x|))$ from Appendix G, where n is at most a polynomial function in $|x|$, a polynomial number of postselections are sufficient in the above argument. ◀

J Proof of Theorem 22

The proof of Theorem 22 is as follows:

Proof. To solve the problem in Def. 21 with a constant probability, we use the idea used in [14]. We prepare the state

$$\frac{\sum_{s\in\mathbb{Z}_q^n, e\in\chi^m, d\in\{0,1\}} |s,e,d\rangle|f_K(s,e,d)\rangle}{\sqrt{2q^n(2\mu+1)^m}}, \tag{49}$$

where $1/\sqrt{2q^n(2\mu+1)^m}$ is the normalization factor (see Theorem 7). When there exists a natural number N satisfying $2^N = 2q^n(2\mu+1)^m$, this preparation is trivially possible in quantum polynomial time with unit probability. If this is not the case, we prepare

$$\frac{\sum_{(s,e,d)\in\mathbb{Z}_q^n\times\chi^m\times\{0,1\}} |s,e,d\rangle|f_K(s,e,d)\rangle|1\rangle + \sum_{(s,e,d)\notin\mathbb{Z}_q^n\times\chi^m\times\{0,1\}} |s,e,d\rangle|0^{m\log q}\rangle|0\rangle}{\sqrt{2^{\tilde{N}}}} \tag{50}$$

with unit probability, where $\tilde{N}$ is the smallest natural number satisfying $2^{\tilde{N}} \geq 2q^n(2\mu+1)^m$. If we obtain the outcome 1 by measuring the third register in the computational basis, we can prepare the state in Eq. (49). From $2q^n(2\mu+1)^m > 2^{\tilde{N}-1}$, the probability of 1 being observed is larger than 1/2. Therefore, by repeating these procedures, we can obtain the outcome 1 at least once with probability of at least $1 - o(1)$.

By measuring the second register in Eq. (49), we obtain a value of $f_K(s,e,d)$. From the δ-2 regularity of $\mathcal{F}$, the obtained output $f_K(s,e,d)$ has exactly two different preimages with probability of at least δ. When $f_K(s,e,d)$ has exactly two different preimages, the state of the first register becomes

$$\frac{|s,e,1\rangle + |s+s_0, e+e_0, 0\rangle}{\sqrt{2}}, \tag{51}$$

where $f_K(s,e,1) = f_K(s+s_0, e+e_0, 0)$. Then, we measure the state in Eq. (51) and obtain the values of $(s,e,1)$ or $(s+s_0, e+e_0, 0)$.

To obtain the other one with probability $1/2$, we would like to obtain the state in Eq. (51) again. It is possible by applying the rewinding operator R on $|s,e,1\rangle$ or $|s+s_0, e+e_0, 0\rangle$ and a classical description[9] of the state in Eq. (51). As an important point, since the state in Eq. (51) becomes $|s,e,1\rangle$ or $|s+s_0, e+e_0, 0\rangle$ by measuring only the last single qubit in the Z basis, a single rewinding operator is sufficient to rewind it.

On the other hand, if rewinding operator is not allowed, the probability of the problem being solved is super polynomially small from the collision resistance of the function family $\mathcal{F}$. ◀

K Proof of Theorem 23

In this proof, we use the statistical difference (SD) problem:

▶ **Definition 25** (Statistical Difference Problem [31]). *Given classical descriptions of two Boolean circuits $C_0, C_1 : \{0,1\}^n \to \{0,1\}^m$ with natural numbers n and m, let P_0 and P_1 be distributions of $C_0(x)$ and $C_1(x)$ with uniformly random inputs $x \in \{0,1\}^n$, respectively. Decide whether $D_{\mathrm{TV}}(P_0,P_1) < 2^{-O(n^c)}$ or $D_{\mathrm{TV}}(P_0,P_1) > 1 - 2^{-O(n^c)}$ for some positive constant c, where $D_{\mathrm{TV}}(\cdot,\cdot)$ is the total variation distance.*

We show that the RwBQP(1) machine can solve the SD problem with probabilities at least $1/2 - 2^{-O(n^c)}$ and $1 - 2\cdot 2^{-O(n^c)}$ when $D_{\mathrm{TV}}(P_0,P_1) < 2^{-O(n^c)}$ and $D_{\mathrm{TV}}(P_0,P_1) > 1 - 2^{-O(n^c)}$, respectively. To this end, we use an argument inspired by [5][10]. First, the RwBQP(1) machine prepares

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{b\in\{0,1\}, x\in\{0,1\}^n} |b\rangle|x\rangle|C_b(x)\rangle. \tag{52}$$

By measuring the last register in the computational basis, it obtains the outcome $y \in \{0,1\}^m$ and

$$\frac{|0\rangle\left(\sum_{x:C_0(x)=y}|x\rangle\right) + |1\rangle\left(\sum_{x:C_1(x)=y}|x\rangle\right)}{\sqrt{2^n(P_0(y)+P_1(y))}}, \tag{53}$$

where for $b \in \{0,1\}$, $P_b(y) = |\{x \in \{0,1\}^n : C_b(x) = y\}|/2^n$ is the probability of C_b outputting y for uniformly random inputs $x \in \{0,1\}^n$. This event occurs with probability $(P_0(y)+P_1(y))/2$. Then, it measures the first register in Eq. (53) in the computational basis

[9] More precisely, the classical description means a transcript of how to prepare the state in Eq. (51) from a tensor product of $|0\rangle$'s. Let V be a unitary that prepares the state in Eq. (49) from $|0\rangle$'s and ℓ be the number of qubits required in the first register in Eq. (49). Then, the classical description is $(I^{\otimes\ell} \otimes |f_K(s,e,d)\rangle\langle f_K(s,e,d)|)V$. Note that V can be decomposed into a polynomial number of elementary gates in a universal gate set.

[10] As a difference between their argument in [5] and ours, we replace their non-collapsing measurement with a single rewinding operator and an ordinary (i.e., a collapsing) measurement. Furthermore, although they use three non-collapsing measurements, we can perform the rewinding operator only once.

and obtain the outcome $b_1 \in \{0,1\}$. By using a single rewinding operator, it can perform the same measurement again and obtain another outcome $b_2 \in \{0,1\}$. Finally, it outputs 1 if $b_1 \neq b_2$. Otherwise, it outputs 0.

We now calculate error probabilities, i.e., probabilities of the machine outputting 0 and 1 when $D_{\mathrm{TV}}(P_0,P_1) < 2^{-O(n^c)}$ and $D_{\mathrm{TV}}(P_0,P_1) > 1-2^{-O(n^c)}$, respectively. First, we consider the case of $D_{\mathrm{TV}}(P_0,P_1) < 2^{-O(n^c)}$. The probability p_{err} of the machine outputting 0, i.e., that of $b_1 = b_2$ is

$$p_{\mathrm{err}} = \sum_{y\in\{0,1\}^m} \frac{P_0(y)+P_1(y)}{2} \frac{P_0(y)^2+P_1(y)^2}{(P_0(y)+P_1(y))^2}. \tag{54}$$

Let $\delta(y) \equiv \max\{P_0(y)-P_1(y), P_1(y)-P_0(y)\}$ and $P_{\min}(y) \equiv \min\{P_0(y),P_1(y)\}$. From Eq. (54),

$$p_{\mathrm{err}} = \frac{1}{2}\left(1+\sum_{y\in\{0,1\}^m} \delta(y)\frac{P_{\min}(y)+\delta(y)}{2P_{\min}(y)+\delta(y)}\right) \leq \frac{1}{2}\left(1+\sum_{y\in\{0,1\}^m}\delta(y)\right) < \frac{1}{2}+2^{-O(n^c)}, \tag{55}$$

where we have used $\sum_{y\in\{0,1\}^m}\delta(y) = 2D_{\mathrm{TV}}(P_0,P_1)$ in the last inequality.

Then, we consider the case of $D_{\mathrm{TV}}(P_0,P_1) > 1-2^{-O(n^c)}$. The probability p'_{err} of the machine outputting 1, i.e., that of $b_1 \neq b_2$ is

$$p'_{\mathrm{err}} = \sum_{y\in\{0,1\}^m} \frac{P_0(y)+P_1(y)}{2} \frac{2P_0(y)P_1(y)}{(P_0(y)+P_1(y))^2}. \tag{56}$$

Since $D_{\mathrm{TV}}(P_0,P_1) > 1-2^{-O(n^c)}$, there exists a set S such that $\sum_{y\in S}P_0(y) \geq 1-2^{-O(n^c)}$ and $\sum_{y\in S}P_1(y) \leq 2^{-O(n^c)}$. Let $\bar{S}$ be a complement of S. From Eq. (56),

$$p'_{\mathrm{err}} = \sum_{y\in S}\frac{P_0(y)P_1(y)}{P_0(y)+P_1(y)} + \sum_{y\in\bar{S}}\frac{P_0(y)P_1(y)}{P_0(y)+P_1(y)} \leq \sum_{y\in S}P_1(y) + \sum_{y\in\bar{S}}P_0(y) \leq 2\cdot 2^{-O(n^c)}, \tag{57}$$

where we have used $\sum_{y\in S\cup\bar{S}}P_0(y) = 1$ in the last inequality.

L Implication of Theorem 22 and Corollary 24

We first explain the implication of Theorem 22. In Sec. 3, we have shown the equivalence between the postselection and rewinding. In contrast, Theorem 22 may represent their difference. A possible approach to solving the problem in Def. 21 is to generate two copies of the state in Eq. (3) (more precisely, Eq. (51) in Appendix J) by using the postselection. As a straightforward way, this can be achieved by postselecting the second register in the state $\sum_x |x\rangle|f_K(x)\rangle$ (more precisely, Eq. (49) in Appendix J) onto the same $f_K(x)$. However, it requires the postselection of a polynomial number of qubits (or the postselection of states whose amplitudes are exponentially small), while a single qubit is sufficient for the rewinding. Furthermore, since we do not know which $f_K(x)$ has exactly two different preimages, the postselection applied to the second copy of $\sum_x |x\rangle|f_K(x)\rangle$ needs to be adaptive, i.e., it depends on $f_K(x)$ obtained from the first copy. On the other hand, a non-adaptive (i.e., non classically-controlled) rewinding operator is sufficient for solving the problem (with a constant probability). Although there may be other ways to solve the problem by using a non-adaptive postselection of a single qubit, the above discussion may imply that the rewinding is superior to the postselection in some situations where the number of qubits to be rewound or postselected is restricted, and copies (i.e., $(\sum_x |x\rangle|f_K(x)\rangle)^{\otimes 2}$) are not processed collectively.

We next explain a difference between Theorem 22 and Corollary 24. For example, by assuming that the decision version of SIVP, gapSIVP, is hard for universal quantum computation, Corollary 24 implies that a single rewinding operator is sufficient to achieve a task that is intractable for universal quantum computation. This is because the gapSIVP (with an appropriate parameter) is in SZK [29]. Therefore, Corollary 24 shows the superiority of a single rewinding operator for promise problems, while Theorem 22 shows it for the search problem.

Quantum Mass Production Theorems

William Kretschmer
University of Texas at Austin, TX, USA

Abstract

We prove that for any n-qubit unitary transformation U and for any $r = 2^{o(n/\log n)}$, there exists a quantum circuit to implement $U^{\otimes r}$ with at most $O(4^n)$ gates. This asymptotically equals the number of gates needed to implement just a *single* copy of a worst-case U. We also establish analogous results for quantum states and diagonal unitary transformations. Our techniques are based on the work of Uhlig [Math. Notes 1974], who proved a similar mass production theorem for Boolean functions.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory; Theory of computation → Circuit complexity

Keywords and phrases mass production, quantum circuit synthesis, quantum circuit complexity

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.10

Related Version *Previous Version*: https://arxiv.org/abs/2212.14399

Funding Supported by an NDSEG fellowship.

Acknowledgements Part of this work was done while the author attended the 2022 Extended Reunion for the Quantum Wave in Computing at the Simons Institute for the Theory of Computing. We thank Alex Meiburg for helpful discussions.

1 Introduction

If a computational task requires c resources, then common sense dictates that repeating the same task r times should require roughly $c \cdot r$ resources. In many settings, including query complexity [11] and communication complexity [12, 4], this intuition can be made rigorous: such results are known as *direct sum theorems*. Closely related are *direct product theorems*, which show that, with a fixed computational budget, the probability of successfully performing r independent tasks decays in r. We recommend [7, Chapter 1] for a good overview of the topic.

Nevertheless, direct sum and direct product theorems are not universal. Some computational settings exhibit a "mass production" phenomenon, in which the cost of performing the same task many times in parallel does *not* scale linearly with the number of repetitions. A well-known example [13, 7] is based on the circuit complexity of matrix-vector multiplication. For a matrix $M \in \{0,1\}^{n \times n}$, define $f_M : \{0,1\}^n \to \{0,1\}^n$ by $f_M(v) = Mv$, where addition and multiplication are taken mod 2. Then a simple counting argument implies that for most M, the complexity of implementing f_M via a Boolean circuit is at least $\Omega(n^2/\log n)$, as measured by the number of 2-bit AND, OR, and NOT gates. Yet, by observing that f_M^n (i.e. f_M repeated n times) is simply a matrix-matrix multiplication, we find that the cost of implementing f_M^n is only $O(n^\omega)$, where $\omega < 2.38$ is the exponent of matrix multiplication [3, 8] – substantially less than the naive bound of $O(n^3)$.

One might be left with the impression that such mass production phenomena can only occur for extremely special functions, like matrix multiplication, that have a particular algebraic or combinatorial structure. Remarkably, this intuition fails dramatically in the setting of Boolean circuit complexity. A theorem of Uhlig [17, 18, 19] shows that for *any*

Boolean function $f : \{0,1\}^n \to \{0,1\}$ and for any $r = 2^{o(n/\log n)}$, there exists a Boolean circuit implementing f^r with at most $O\left(\frac{2^n}{n}\right)$ gates. Asymptotically, this equals the number of gates needed to evaluate a worst-case f on a *single* input, by the well-known counting argument of Shannon [15]. In fact, Uhlig even showed that the leading constant in the big-O does not increase with r, and hence arbitrary Boolean functions can be mass produced with essentially no overhead.

1.1 This Work

In this work, we consider the natural question of whether a similar mass production phenomenon holds for quantum circuit complexity. Our question is well-motivated by recent works demonstrating that for certain learning tasks, algorithms with access to many copies of a quantum state on a quantum memory can be exponentially more powerful than algorithms that have access only to single copies of the state at a time [6, 10, 5]. Indeed, these results suggest that optimizing the complexity of mass producing quantum states and processes could have valuable applications. We also view our question as interesting from a purely theoretical perspective, especially considering that Uhlig's theorem for classical functions has recently found complexity-theoretic applications in characterizing the minimum circuit size problem [14, 9].

For simplicity, we consider quantum circuit complexity in the setting of qubit quantum circuits, using the universal gate set of arbitrary single-qubit gates plus CNOT gates with all-to-all connectivity. We also allow ancilla qubits initialized to $|0\rangle$, so long as they are reset to $|0\rangle$ at the end of the computation. We measure circuit complexity in terms of the CNOT count. This measure is justified by the fact that multiple-qubit gates are more error-prone and expensive to implement than single-qubit gates, and also by the observation that the number of single-qubit gates is related to the CNOT count by at most a factor of 4 in any irredundant circuit.

In analogy with Uhlig's theorem [17, 18, 19], our main result establishes mass production theorems for both quantum states and unitary transformations.

▶ **Theorem 1.** *Let $|\psi\rangle$ be an n-qubit quantum state, and let $r = 2^{o(n/\log n)}$. Then there exists a quantum circuit with at most $(1 + o(1))2^n$ CNOT gates to prepare $|\psi\rangle^{\otimes r}$.*

▶ **Theorem 2.** *Let U be an n-qubit unitary transformation, and let $r = 2^{o(n/\log n)}$. Then there exists a quantum circuit with at most $(5/2 + o(1))4^n$ CNOT gates to implement $U^{\otimes r}$.*

Note that the factor 2^n (respectively, 4^n), in Theorem 1 (respectively, Theorem 2) is optimal, because it asymptotically equals the number of CNOT gates needed to prepare a *single* copy of an arbitrary n-qubit state (respectively, to implement an arbitrary n-qubit unitary once), up to a small multiplicative constant [16]. Above, we made the leading constants explicit only to illustrate that they are not too large, and thus to demonstrate that these theorems have some hope of becoming practical. We leave a full optimization of these constants and the factors hidden in the $o(1)$ to future work.

1.2 Proof Overview

Our results build heavily on the simple proof of Uhlig's theorem given in [19], which we now briefly summarize. The proof proceeds by first showing that for an arbitrary $f : \{0,1\}^n \to \{0,1\}$, one can compute 2 copies of f using roughly $\frac{2^n}{n}$ gates – the same cost

as is needed to compute a single copy of a worst-case f. Then, Uhlig shows that we can generalize to a larger number of repetitions r by a straightforward recursive argument. So, we focus on the $r = 2$ case.

Fix a parameter k do be chosen later, and define for each $0 \leq i \leq 2^k - 1$ the function $f_i : \{0,1\}^{n-k} \to \{0,1\}$ to be the restriction of f obtained by fixing the first k bits to be the binary representation of i. So, for example,

$$f(\underbrace{0,0,\ldots,0}_{k \text{ times}}, x_{k+1}, \ldots, x_n) = f_0(x_{k+1}, \ldots, x_n).$$

Next, we define a set of functions $g_\ell : \{0,1\}^{n-k} \to \{0,1\}$ for each $0 \leq \ell \leq 2^k$ by:

- $g_0 = f_0$.
- $g_\ell = f_{\ell-1} \oplus f_\ell$ if $1 \leq \ell \leq 2^k - 1$.
- $g_{2^k} = f_{2^k-1}$.

Observe that

$$f_i = \bigoplus_{\ell=0}^{i} g_\ell = \bigoplus_{\ell=i+1}^{2^k} g_\ell. \tag{1}$$

Now, suppose that we have a pair of inputs $x, y \in \{0,1\}^n$ to f, and our goal is to evaluate $f(x)$ and $f(y)$ simultaneously. Let i and j denote the integers whose binary representations are the first k bits of x and y, respectively. Assume without loss of generality that $i \leq j$. Uhlig's idea is to evaluate $f(x)$ using the decomposition $f_i = \bigoplus_{\ell=0}^{i} g_\ell$ and $f(y)$ using $f_j = \bigoplus_{\ell=j+1}^{2^k} g_\ell$. The key observation is that in doing so, we only need to evaluate each g_ℓ at most once. The cost of computing $f(x)$ and $f(y)$ this way is dominated by computing the g_ℓs. So, the total size of the circuit is roughly

$$\left(2^k + 1\right)\left(\frac{2^{n-k}}{n-k}\right),$$

because there are $2^k + 1$ different g_ℓs, and each g_ℓ is a function on $n - k$ bits. For reasonable choices of k, this is asymptotically $(1 + o(1))\frac{2^n}{n}$, as desired.

Our main insight is that the same general approach generalizes straightforwardly from mass producing Boolean functions to mass producing diagonal unitary matrices, which we establish in Theorem 4. In one sense, the only conceptual change between our proof and Uhlig's is that we work with the group of complex units under multiplication, rather than the group $\{0,1\}$ under XOR. Nevertheless, our proof requires some care, as we do not deal with diagonal matrices directly. Rather, we mass produce the direct sum of a diagonal unitary with its inverse. In other words, for an n-qubit diagonal unitary U, we find it more convenient to work with the diagonal unitary on $n + 1$ qubits that applies U when the last qubit is $|0\rangle$, and $U^\dagger$ when the last qubit is $|1\rangle$. The intuitive reason why we require this change is that the XOR function is its own inverse, whereas multiplication by a complex unit is generally not.

Finally, once we have established Theorem 4 for diagonal unitary transformations, we obtain the mass production theorems for quantum states and general unitary transformations by using well-known decompositions of states and unitaries into diagonal gates [16].

2 Preliminaries

2.1 Basic Notation

We denote by $\mathbb{1}\{p\}$ the function that evaluates to 1 if proposition p is true, and 0 otherwise. If α is a complex number, we let α^* denote its complex conjugate. We denote by $\mathbb{T} := \{a + bi : |a|^2 + |b|^2 = 1\}$ the set of complex units. For a function $f : \{0,1\}^n \to \mathbb{T}$, denote by $\bar{f} : \{0,1\}^{n+1} \to \mathbb{T}$ the function defined by $\bar{f}(x,c) = f(x)^{1-2c}$, so that $\bar{f}$ evaluates to f when $c = 0$ and evaluates to f^* when $c = 1$. We freely identify a function $f : \{0,1\}^n \to \mathbb{T}$ with the corresponding diagonal unitary transformation U that acts as $U|x\rangle = f(x)|x\rangle$ on basis states $x \in \{0,1\}^n$.

We use standard notation for quantum circuits, including CNOT, Toffoli, and Fredkin gates. We also borrow a large amount of notation and terminology from [16], as we detail further below. We define the x-, y-, and z-axis rotations by:

$$R_x(\theta) = \begin{pmatrix} \cos(\theta/2) & i\sin(\theta/2) \\ i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix},$$
$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{pmatrix},$$
$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}.$$

2.2 Multiplexors

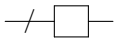
A *multiplexor* with s select qubits and d data qubits is a block-diagonal $(s+d)$-qubit unitary transformation that preserves every computational basis state $|x\rangle$ on the select qubits. For brevity, we call such a unitary an (s,d)-multiplexor. An $(s,1)$-multiplexor in which all of the diagonal blocks are R_z on the data qubit may alternatively be called a *multiplexed* R_z (analogously for R_x and R_y). Collectively, multiplexed R_x, R_y, and R_z are called *multiplexed rotations*. Observe that an $(s,1)$-multiplexed R_z is equivalent to a unitary implementing $\bar{f}$ for some $f : \{0,1\}^s \to \mathbb{T}$.

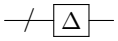
We require the following basic fact about implementing multiplexed rotations:

▶ **Proposition 3** ([16, Theorem 8]). *Let U be an $(n,1)$-multiplexed rotation. Then there exists a quantum circuit with at most 2^n CNOT gates to implement U.*

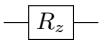
2.3 Generic Gates

As in [16], we use circuit diagrams containing *generic gates*. An equivalence of two circuit diagrams containing generic gates means that for any assignment of parameters to the generic gates on one side, there exists an assignment of parameters to the gates on the other side that makes the two circuits compute the same operator. We use the following notation for generic gates:

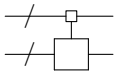

A generic unitary gate.

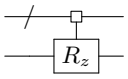


A generic diagonal unitary gate.



An R_z gate for some unspecified θ. Conventions for R_x and R_y are analogous.


A generic multiplexor, with select qubits on the upper register and data qubits on the lower register



A multiplexed R_z. Conventions for R_x and R_y are analogous.

3 Diagonal Unitaries and Multiplexors

We begin by generalizing the proof of Uhlig's theorem [19] to diagonal unitary matrices (or, more precisely, multiplexed R_z gates).

▶ **Theorem 4.** *Let $f : \{0,1\}^n \to \mathbb{T}$ and let $r = 2^{o(n/\log n)}$. Then there exists a quantum circuit with at most $(1+o(1))2^n$ CNOT gates to implement $\bar{f}^{\otimes r}$.*

Proof. Without loss of generality, let $r = 2^t$ for some $t = o(n/\log n)$. Our proof proceeds by induction on t: for fixed k (chosen later) and for every $n > k \cdot t$, we construct for each $f : \{0,1\}^n \to \mathbb{T}$ a circuit $\mathcal{C}_{f,n,k,t}$ computing $\bar{f}^{\otimes 2^t}$. We proceed in order: first we construct $\mathcal{C}_{f,n,k,1}$ for every n and f, then $\mathcal{C}_{f,n,k,2}$ for every n and f, then $\mathcal{C}_{f,n,k,3}$ for every n and f, and so on. Ultimately, we show that there exists a universal constant d such that the number of CNOT gates in $\mathcal{C}_{f,n,k,t}$, denoted $s_{n,k,t}$, satisfies the bound:

$$s_{n,k,t} \leq \left(2^k + 1\right)^t \left(2^{n-tk} + 2^t dn\right). \tag{2}$$

We begin by describing the construction of $\mathcal{C}_{f,n,k,1}$. For each $0 \leq i \leq 2^k - 1$, let $f_i : \{0,1\}^{n-k} \to \mathbb{T}$ denote the restriction of f obtained by fixing the first k bits to the binary representation of i. For each $0 \leq i \leq 2^k$, define $g_i : \{0,1\}^{n-k} \to \mathbb{T}$ by:

- $g_0 = f_0$.
- $g_\ell = f^*_{\ell-1} f_\ell$ if $1 \leq \ell \leq 2^k - 1$.
- $g_{2^k} = f^*_{2^k-1}$.

Observe that

$$f_i = \prod_{\ell=0}^{i} g_\ell = \prod_{\ell=i+1}^{2^k} g^*_\ell. \tag{3}$$

The key idea in the remainder of the proof is to evaluate $\bar{f}$ on a pair of inputs (x, y) using the two decompositions in (3), one each for x and y. Indeed, the following algorithm accomplishes this.

Algorithm 1 Evaluate $\bar{f}^{\otimes 2}$.

Input: $x, y \in \{0,1\}^n$, $c_x, c_y \in \{0,1\}$
Output: $\bar{f}(x, c_x) \cdot \bar{f}(y, c_y)$

$\alpha \coloneqq 1$
if $x \leq y$ **then** /* viewing x, y as integers w/ highest order bits x_1, y_1 */
  $m \coloneqq x$; $c_m \coloneqq c_x$ /* set $m = \min\{x, y\}$, $M = \max\{x, y\}$ */
  $M \coloneqq y$; $c_M \coloneqq c_y$
else
  $m \coloneqq y$; $c_m \coloneqq c_y$
  $M \coloneqq x$; $c_M \coloneqq c_x$
for $0 \leq \ell \leq 2^k$ **do**
  if $\ell \leq m_{[1:k]}$ **then** /* $x_{[i:j]}$ denotes bits i through j of x */
    Multiply α by $\bar{g}_\ell(m_{[k+1:n]}, c_m)$
  else if $\ell > M_{[1:k]}$ **then**
    Multiply α by $\bar{g}_\ell(M_{[k+1:n]}, 1 - c_M)$ /* note negation on c_M */
  else
    Multiply α by 1
return α

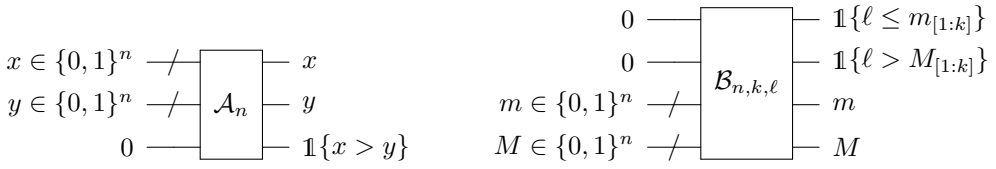
Here, the $\ell \leq m_{[1:k]}$ clause corresponds to the multiplication $\prod_{\ell=0}^{m_{[1:k]}} g_\ell$, while the $\ell > M_{[1:k]}$ clause corresponds to $\prod_{\ell=M_{[1:k]}}^{2^n} g_\ell^*$. An equivalent reformulation of Algorithm 1 is given below.

Algorithm 2 Evaluate $\bar{f}^{\otimes 2}$

Input: $x, y \in \{0,1\}^n$, $c_x, c_y \in \{0,1\}$
Output: $\bar{f}(x, c_x) \cdot \bar{f}(y, c_y)$

$\alpha \coloneqq 1$
if $x \leq y$ **then**
  $m \coloneqq x$; $c_m \coloneqq c_x$
  $M \coloneqq y$; $c_M \coloneqq c_y$
else
  $m \coloneqq y$; $c_m \coloneqq c_y$
  $M \coloneqq x$; $c_M \coloneqq c_x$
for $0 \leq \ell \leq 2^k$ **do**
  $a \coloneqq \mathbb{1}\{\ell \leq m_{[1:k]}\}$ /* at most one of a, b is nonzero */
  $b \coloneqq \mathbb{1}\{\ell > M_{[1:k]}\}$
  $z \coloneqq a \cdot m_{[k+1:n]} \oplus b \cdot M_{[k+1:n]}$
  $c \coloneqq a \cdot c_m \oplus b \cdot (1 - c_M)$
  Multiply α by $\bar{g}_\ell(z, c)$
  Multiply α by $g_\ell^*(0^{n-k})^{(1-a)\cdot(1-b)}$ /* undo added phase in case $a = b = 0$ */
return α

Algorithm 2 readily extends to a quantum circuit implementation. Define a pair of classical reversible circuits $\mathcal{A}_n$ and $\mathcal{B}_{n,k,\ell}$ whose input and output behavior are given in Figure 1. Using $\mathcal{A}_n$ and $\mathcal{B}_{n,k,\ell}$, via the same strategy as Algorithm 2, we obtain the quantum circuit $\mathcal{C}_{f,n,k,1}$ defined in Figure 2 that implements $\bar{f}^{\otimes 2}$.



(a) The circuit $\mathcal{A}_n$. **(b)** The circuit $\mathcal{B}_{n,k,\ell}$.

Figure 1 Inputs and outputs of reversible circuits $\mathcal{A}_n$ and $\mathcal{B}_{n,k,\ell}$.

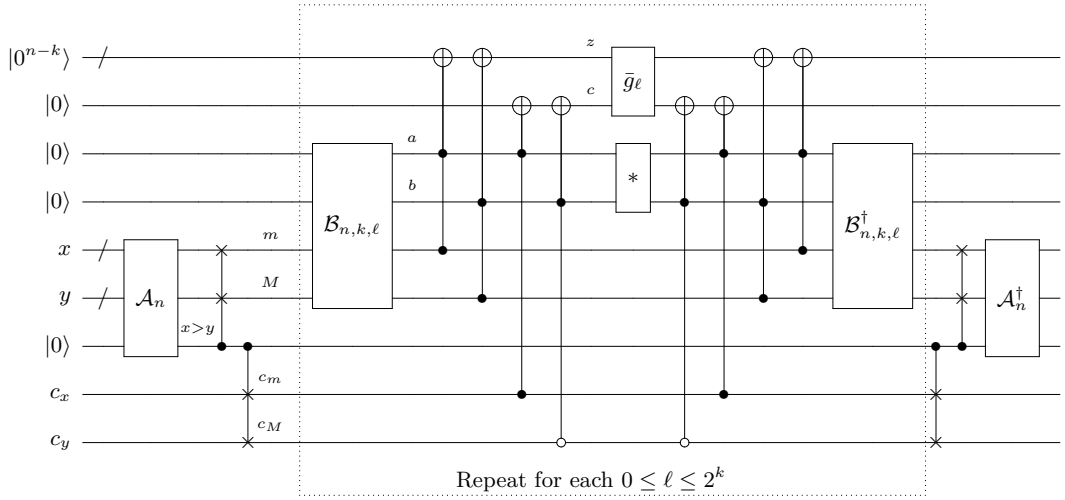


Figure 2 Circuit diagram of $\mathcal{C}_{f,n,k,1}$. The Toffoli gates with controls acting on the x and y registers are understood to be arrays of $n-k$ Toffoli gates between the corresponding qubits of the control and target registers. The gate marked $*$ adds a phase of $g^*_\ell(0^{n-k})$ if both qubits are $|0\rangle$ and otherwise does nothing. For convenience, several of the wires are labeled with the values they take on corresponding to variables in Algorithm 2.

By Proposition 3, for every ℓ, $\bar{g}_\ell$ can be implemented using at most 2^{n-k} CNOT gates, because $\bar{g}_\ell$ is equivalent to an $(n-k,1)$-multiplexed R_z. Moreover, it is easy to see that $\mathcal{A}_n$ and $\mathcal{B}_{n,k,\ell}$ can be implemented using at most $O(n)$ CNOT gates each, because comparison of two n-bit integers can be performed by a classical circuit of at most $O(n)$ gates. As a consequence, we conclude that there exists a constant d such that:

$$s_{n,k,1} \leq \left(2^k+1\right)\left(2^{n-k}+dn\right). \tag{4}$$

This is certainly less than the bound in (2), so this establishes the base case of the induction proof.

Now we proceed to the induction step on t. Suppose that for every $n > k \cdot (t-1)$, we have a circuit $\mathcal{C}_{f,n,k,t-1}$ computing $\bar{f}^{\otimes 2^{t-1}}$ with CNOT count bounded by

$$s_{n,k,t-1} \leq \left(2^k+1\right)^{t-1}\left(2^{n-(t-1)k}+2^{t-1}dn\right). \tag{5}$$

To construct $\mathcal{C}_{f,n,k,t}$, we start by first taking 2^{t-1} copies of $\mathcal{C}_{f,n,k,1}$. Then, for each $0 \leq \ell \leq 2^k$, we replace each of the 2^{t-1} sub-circuits that compute $\bar{g}_\ell$ with $\mathcal{C}_{g_\ell,n-k,k,t-1}$. Then, the number

of gates in $\mathcal{C}_{f,n,k,t}$ is bounded by:

$$\begin{aligned}
s_{n,k,t} &\leq \left(2^k+1\right)\left(s_{n-k,k,t-1}+2^{t-1}dn\right)\\
&\leq \left(2^k+1\right)\left(\left(2^k+1\right)^{t-1}\left(2^{n-k-(t-1)k}+2^{t-1}d(n-k)\right)+2^{t-1}dn\right)\\
&\leq \left(2^k+1\right)^t\left(2^{n-tk}+2^{t-1}dn\right)+\left(2^k+1\right)2^{t-1}dn\\
&\leq \left(2^k+1\right)^t\left(2^{n-tk}+2^t dn\right),
\end{aligned}$$

where the first line substitutes (5) for the cost of the $\bar{g}_\ell$'s and otherwise uses the same bound as (4) for the non-$\bar{g}_\ell$ gates, and the second line applies the induction hypothesis (5). This establishes the induction step, and thus (2) holds for every $n > k \cdot t$.

Choose $k = \lceil \log n \rceil$. Then:

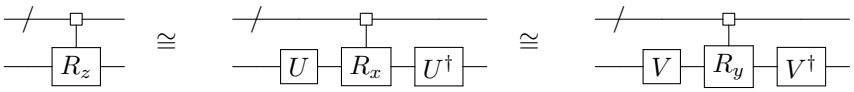
$$\begin{aligned}
s_{n,k,t} &\leq \left(2^k+1\right)^t\left(2^{n-tk}+2^t dn\right)\\
&= 2^{kt}\left(1+\frac{1}{2^k}\right)^t\left(2^{n-tk}+2^t dn\right)\\
&\leq 2^{kt}e^{t/2^k}\left(2^{n-tk}+2^t dn\right)\\
&\leq 2^{kt}(1+o(1))\left(2^{n-tk}+2^t dn\right)\\
&\leq 2^{kt}(1+o(1))\left(2^{n-tk}+o\left(2^{n-tk}\right)\right)\\
&\leq (1+o(1))2^n,
\end{aligned}$$

where we applied the exponential inequality in the third line, and used the assumption $t \leq o(n/\log n)$ in the fourth and fifth lines. This proves the theorem. ◀

Theorem 4 straightforwardly generalizes to arbitrary multiplexed rotations and multiplexors with a single data qubit, as below.

▶ **Corollary 5.** *Let U be an $(n,1)$-multiplexed rotation, and let $r = 2^{o(n/\log n)}$. Then there exists a quantum circuit with at most $(1+o(1))2^n$ CNOT gates to implement $U^{\otimes r}$.*

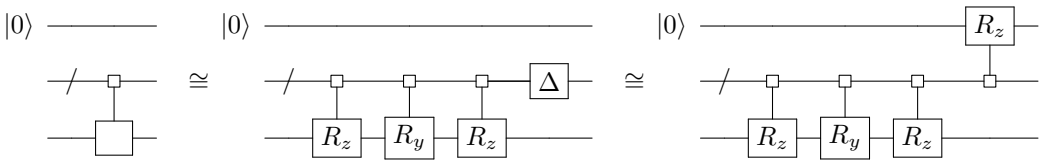
Proof. The R_z case follows by observing that $\bar{f}$ is exactly an $(n,1)$-multiplexed R_z in Theorem 4. This also extends to multiplexed R_x and R_y, because multiplexed R_x, R_y, and R_z are equivalent up to conjugation by single-qubit unitaries on the data qubit. That is, there exist single-qubit unitaries U and V such that:



Hence, the CNOT count is identical for multiplexed R_x and R_y as well. ◀

▶ **Corollary 6.** *Let U be an $(n,1)$-multiplexor, and let $r = 2^{o(n/\log n)}$. Then there exists a quantum circuit with at most $(4+o(1))2^n$ CNOT gates to implement $U^{\otimes r}$.*

Proof. By [16, Theorem 6], an arbitrary $(n,1)$-multiplexor may be implemented via a product of 4 $(n,1)$-multiplexed rotations, as below.



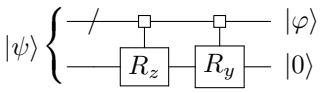
Applying Corollary 5 to each of the multiplexed rotations on the right side above completes the proof. ◀

4 States and General Unitaries

We now prove the main results of this work that generalize the mass production theorems above to state preparation and unitary compilation. The proofs proceed via the techniques of [16], by decomposing operators into multiplexors.

▶ **Theorem 1.** *Let $|\psi\rangle$ be an n-qubit quantum state, and let $r = 2^{o(n/\log n)}$. Then there exists a quantum circuit with at most $(1+o(1))2^n$ CNOT gates to prepare $|\psi\rangle^{\otimes r}$.*

Proof. By [16, Theorem 9], for any n-qubit quantum state $|\psi\rangle$, there exists an $(n-1)$-qubit state $|\varphi\rangle$ such that $|\psi\rangle$ has the following decomposition.



Applying this decomposition recursively, we conclude that $|\psi\rangle$ can be prepared by a circuit consisting of a pair of $(\ell,1)$-multiplexed rotations for each $1 \leq \ell \leq n-1$, and a pair of single-qubit gates.

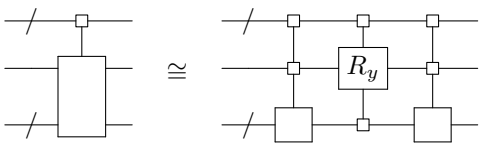
Apply Corollary 5 to the $(\ell,1)$-multiplexed rotations for each $\lceil n/2 \rceil \leq \ell \leq n-1$, and otherwise apply Proposition 3 r times for each $1 \leq \ell \leq \lceil n/2 \rceil - 1$. Then the total number of CNOT gates to prepare $|\psi\rangle^{\otimes r}$ is upper bounded by

$$\begin{aligned} r \cdot \sum_{\ell=1}^{\lceil n/2 \rceil - 1} 2^\ell + \sum_{\ell=\lceil n/2 \rceil}^{n-1} (1+o(1))2^\ell &\leq r2^{\lceil n/2 \rceil} + (1+o(1))2^n \\ &\leq 2^{\lceil n/2 \rceil + o(n/\log n)} + (1+o(1))2^n \\ &\leq (1+o(1))2^n \end{aligned}$$

◀

▶ **Theorem 2.** *Let U be an n-qubit unitary transformation, and let $r = 2^{o(n/\log n)}$. Then there exists a quantum circuit with at most $(5/2+o(1))4^n$ CNOT gates to implement $U^{\otimes r}$.*

Proof. By [16, Theorem 11], an arbitrary multiplexor can be expressed as below.



This decomposition is also valid when the multiplexor on the left side of the equivalence has 0 select bits. A recursive application of this decomposition implies that an arbitrary n-qubit unitary may be expressed as a product of $2^n - 1$ different $(n-1,1)$-multiplexors, of which $2^{n-1}-1$ are multiplexed R_y gates, and the remaining 2^{n-1} are arbitrary multiplexors. Applying Corollary 5 and Corollary 6 to these multiplexors gives the desired bound. ◀

5 Conclusion and Outlook

We have demonstrated that mass production phenomena are not unique to classical computation, and that they extend to quantum circuit complexity as well. As the message of this work is primarily conceptual in nature, we have not attempted to optimize every aspect of our results. Indeed, our mass production theorems could be extended further in a variety of ways; we outline a few such possibilities below.

If our results have any hope of being used in practice, then still more work needs to be done to optimize various constants. We suspect that the leading constant in Theorem 2 could be brought down from 5/2 to 1 with a more clever decomposition into multiplexors. The factors hidden in the $o(1)$ could probably be optimized further as well, especially those related to the constant factor d that appears in Theorem 4. Indeed, we believe that much of the redundancy in computing and uncomputing $\mathcal{B}_{n,k,\ell}$ for each $0 \leq \ell \leq 2^k$ could be reduced by more careful accounting.

It is also worth attempting to optimize other parameters of practical relevance, such as constraints on the gate set, locality, depth, and ancilla qubit count. In principle, our proof should allow for some tradeoff between depth and ancilla count, because the $\bar{g}_\ell$s in Figure 2 can either be evaluated sequentially or in parallel. Another particularly interesting question is whether ancilla qubits are necessary at all to achieve quantum mass production.

We leave open the circuit complexity of quantum mass production in other parameter regimes. As Theorem 1 and Theorem 2 only apply when $r = 2^{o(n/\log n)}$, it is natural to ask what happens when r is much larger. For Boolean functions, it is known that for any n-bit f, the "asymptotic complexity" of mass production $\lim_{r\to\infty} \frac{C(f^r)}{r}$ is bounded by $\mathrm{poly}(n)$ [13, 2], where $C(f^r)$ denotes the Boolean circuit complexity of implementing r copies of f. However, it is unclear whether the same approach would generalize to quantum circuits.

Lastly, we ask: are there any restricted examples of quantum circuits that exhibit a mass production phenomenon? What about Clifford circuits? We observe if one allows implementation by non-Clifford gates, then n copies of an arbitrary Clifford operation can be implemented by a circuit with at most $O(n^\omega)$ gates, where ω is the exponent of matrix multiplication. By the "canonical form theorem" of Aaronson and Gottesman [1], every Clifford circuit can be expressed in the form H-C-P-C-P-C-H-P-C-P-C, where each letter corresponds to a layer of Hadamard, CNOT, or phase gates. The Hadamard and phase layers contain at most $O(n)$ gates total, so it suffices to show how to implement n copies of a CNOT circuit using $O(n^\omega)$ gates. For any $M \in \mathbb{F}_2^{n\times n}$, define U_M as the unitary transformation that acts as $U_M |x\rangle |y\rangle = |x\rangle |y \oplus Mx\rangle$ on computational basis states. As every CNOT circuit implements an invertible linear transformation $|x\rangle \to |Mx\rangle$ for some $M \in \mathbb{F}_2^{n\times n}$, a CNOT circuit can be implemented using U_M and $U_{M^{-1}}$ and $O(n)$ additional gates via:

$$|x\rangle |0^n\rangle \xrightarrow{U_M} |x\rangle |Mx\rangle \xrightarrow{U_{M^{-1}}} |0^n\rangle |Mx\rangle \xrightarrow{\text{SWAP}} |Mx\rangle |0^n\rangle .$$

Then, as in Section 1, we can mass produce U_M and $U_{M^{-1}}$ using fast matrix multiplication.

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 Andreas Albrecht. On simultaneous realizations of Boolean functions, with applications. In Gottfried Wolf, Tamáas Legendi, and Udo Schendel, editors, *Parcella '88*, pages 51–56, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg. doi:10.1007/3-540-50647-0_102.

3 Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539, 2021. doi:10.1137/1.9781611976465.32.

4 Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, pages 67–76, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806701.

5 Matthias C. Caro. Learning quantum processes and Hamiltonians via the Pauli transfer matrix, 2022. arXiv:2212.04471.

6 Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li. Exponential separations between learning with and without quantum memory. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 574–585, 2022. doi:10.1109/FOCS52979.2021.00063.

7 Andrew Donald Drucker. *The complexity of joint computation*. PhD thesis, Massachusetts Institute of Technology, 2012. URL: http://dspace.mit.edu/handle/1721.1/7582.

8 Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing, 2022. arXiv:2210.10173.

9 Shuichi Hirahara. NP-hardness of learning programs and partial MCSP. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 968–979. IEEE, 2022. doi:10.1109/FOCS54457.2022.00095.

10 Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, and Jarrod R. McClean. Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186, 2022. doi:10.1126/science.abn7293.

11 Rahul Jain, Hartmut Klauck, and Miklos Santha. Optimal direct sum results for deterministic and randomized decision tree complexity. *Information Processing Letters*, 110(20):893–897, 2010. doi:10.1016/j.ipl.2010.07.020.

12 Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming*, pages 300–315, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. doi:10.1007/3-540-45061-0_26.

13 Wolfgang J. Paul. Realizing Boolean functions on disjoint sets of variables. *Theoretical Computer Science*, 2(3):383–396, 1976. doi:10.1016/0304-3975(76)90089-X.

14 Hanlin Ren and Rahul Santhanam. Hardness of KT Characterizes Parallel Cryptography. In Valentine Kabanets, editor, *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:58, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2021.35.

15 Claude. E. Shannon. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal*, 28(1):59–98, 1949. doi:10.1002/j.1538-7305.1949.tb03624.x.

16 Vivek V. Shende, Stephen S. Bullock, and Igor L. Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, 2006. doi:10.1109/TCAD.2005.855930.

17 Dietmar Uhlig. On the synthesis of self-correcting schemes from functional elements with a small number of reliable elements. *Matematicheskie Zametki*, 15(6):937–944, 1974. In Russian. URL: http://mi.mathnet.ru/mz7425.

18 Dietmar Uhlig. On the synthesis of self-correcting schemes from functional elements with a small number of reliable elements. *Mathematical notes of the Academy of Sciences of the USSR*, 15(6):558–562, 1974. Translated from Russian. doi:10.1007/BF01152835.

19 Dietmar Uhlig. *Networks Computing Boolean Functions for Multiple Input Values*, pages 165–173. London Mathematical Society Lecture Note Series. Cambridge University Press, 1992. doi:10.1017/CBO9780511526633.013.

On the Power of Nonstandard Quantum Oracles

Roozbeh Bassirian
University of Chicago, IL, USA

Bill Fefferman
University of Chicago, IL, USA

Kunal Marwaha
University of Chicago, IL, USA

Abstract

We study how the choices made when designing an oracle affect the complexity of quantum property testing problems defined relative to this oracle. We encode a regular graph of even degree as an invertible function f, and present f in different oracle models. We first give a one-query QMA protocol to test if a graph encoded in f has a small disconnected subset. We then use representation theory to show that no classical witness can help a quantum verifier efficiently decide this problem relative to an in-place oracle. Perhaps surprisingly, a simple modification to the standard oracle prevents a quantum verifier from efficiently deciding this problem, even with access to an unbounded witness.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases quantum complexity, QCMA, expander graphs, representation theory

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.11

Related Version *Full Version*: https://arxiv.org/abs/2212.00098

Funding *Roozbeh Bassirian*: AFOSR (YIP number FA9550-18-1-0148 and FA9550-21-1-0008); NSF under Grant CCF-2044923 (CAREER); U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers; DOE QuantISED grant DE-SC0020360.
Bill Fefferman: AFOSR (YIP number FA9550-18-1-0148 and FA9550-21-1-0008); NSF under Grant CCF-2044923 (CAREER); U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers; DOE QuantISED grant DE-SC0020360.
Kunal Marwaha: NSF Graduate Research Fellowship Program under Grant No. DGE-1746045

Acknowledgements Thanks to Casey Duckering, Juspreet Singh Sandhu, Peter Shor, and Justin Yirka for collaborating on early stages of this project. KM thanks many others for engaging discussions, including Adam Bouland, Antares Chen, Aram Harrow, Matt Hastings, Eric Hester, Neng Huang, Robin Kothari, Brian Lawrence, Yi-Kai Liu, Patrick Lutz, Tushant Mittal, Abhijit Mudigonda, Chinmay Nirkhe, and Aaron Potechin. Thanks to Chinmay Nirkhe for feedback on a previous version of this manuscript.

1 Introduction

Computational complexity is the study of the innate amount of resources required to complete some task. We assign *complexity classes* to sets of tasks that require similar amounts of resources; from here, the goal is to understand the relationship between complexity classes. There has been some success proving that two complexity classes are equal, for example IP = PSPACE [25], the PCP theorem [7], and $\text{MIP}^* = \text{RE}$ [17]; however, proving that two complexity classes are *unequal* has been much more elusive. For example, we cannot prove $\text{P} \neq \text{PSPACE}$, let alone $\text{P} \neq \text{NP}$.

One response to this difficulty is to equip a computational model with an *oracle*, which computes a fixed (but arbitrarily powerful) quantity in a single timestep. It is often easier to prove that a statement (e.g. P $\neq$ NP) is true relative to an oracle; furthermore, this restricts the kinds of proof techniques that can show the statement is false without an oracle. In addition to separating complexity classes, oracles and *query complexity* naturally arise in cryptography (e.g. [19]) and learning theory (e.g. [20]).

Even with respect to an oracle, proving that some complexity classes are unequal can be surprisingly difficult. Notably, Aharonov and Naveh define QCMA, a subset of QMA where the witness is a classical bitstring [3], and ask if QCMA $\subsetneq$ QMA. Aaronson and Kuperberg conjecture that an oracle separates these classes, but only prove a "quantum oracle" where this occurs [2]. Subsequent works [11, 22] remove the "quantumness" from the oracle model, but still use models with internal randomness or other nonstandard aspects.

We consider quantum property testing problems defined relative to oracles from various oracle models: encoding the edges of a graph in an invertible function f, we present f as either a *standard* oracle or *in-place* oracle, with or without internal randomness. With mild restrictions on the workspace of quantum verifiers, we find:

1. In several oracle models presenting f, a *quantum* witness can help a quantum verifier efficiently decide if the graph encoded in f has a small disconnected subset.
2. Where f is presented as a randomized in-place oracle, no *classical* witness can help a quantum verifier efficiently decide this problem.
3. Where f is presented as a randomized phase oracle, no witness *of any type or size* can help a quantum verifier efficiently decide this problem.

Our results highlight that the quantum complexity of a task defined relative to an oracle is influenced by the choice of oracle model.

1.1 Our techniques

We use a well-known fact of Petersen to encode the edges of any even-degree regular graph in an invertible function f. We then consider natural ways to install f within an oracle; we say that f is *presented* as a particular kind of oracle. For example, a standard oracle presents f through the map $|c, x\rangle \mapsto |c \oplus f(x), x\rangle$, while an in-place oracle presents f through the map $|x\rangle \mapsto |f(x)\rangle$. In general, we consider oracles that give access both to f and f^{-1}. An oracle may also have internal randomness: on every query to a *randomized* oracle, f is chosen uniformly at random from a fixed set of functions F.

Consider the Laplacian L_f of a graph encoded in f. We first provide a test such that for any input state $|\psi\rangle$, the test succeeds with probability expressible in terms of $\langle\psi| L_f |\psi\rangle$, independently of how an oracle presents f. We use this test to construct a QMA protocol verifying that the graph is not an *expander* graph. This problem is primarily motivated by the preimage-testing problem of Fefferman and Kimmel [11], which separates QMA and QCMA relative to a nonstandard oracle. They encode an invertible function π in an oracle *without efficient access to* π^{-1}, and test a property of π^{-1}; by design, this property can be verified but not easily computed. Crucially, we view a permutation and its inverse as the edges of an *undirected graph*; properties of undirected graphs are not sensitive to the ordering of $(x, \pi(x))$. We use multiple permutations to study graphs of higher degree, and notice that detecting if a graph has a non-expanding region is hard without traversing most of the graph. Some of these ideas are related to the *component mixer* concept of Lutomirski [21], and are simultaneously and independently explored by Natarajan and Nirkhe [22].

A randomized oracle presenting a set of functions F can be seen as a quantum channel, so small changes to F cause statistically indistinguishable changes to the oracle. We use this flexibility to modify non-expansion testing to a simple permutation problem: do the functions

$f \in F$ stabilize a small set $V \subseteq [N]$, or is F the set of all permutations on $[N]$? Notice that F is a group in both cases. When an oracle presenting F preserves the group structure of F, we can use representation theory. For this problem, this is satisfied by an *in-place* oracle; the oracle is then an orthogonal projector onto one of two symmetric subspaces of matrices. After finding an orthogonal basis for each subspace, we construct a hybrid argument to prove that only witnesses with knowledge of V can help a quantum verifier efficiently decide this problem. We also use representation theory to give a QCMA protocol for an analogous permutation problem in randomized standard oracles.

We finally study the permutation problem in a randomized phase oracle. We directly analyze the effect of the oracle on any input density matrix; with high probability, the oracle decreases the magnitude of every off-diagonal term by a $\frac{1}{2^{\mathrm{poly}(n)}}$ factor. We then construct a hybrid argument, bounding our measure of progress using an inequality relating the sizes of Schatten p-norms. When the state space is not too large, we prove that an exponential number of queries are required to distinguish most YES instances from the NO instance, *regardless of input state*. As a result, no witness can help a verifier distinguish YES from NO.

Note that our quantum verifiers are not fully general. Our lower bound techniques restrict the number of extra workspace qubits in the verifier; however, our upper bounds also work in this setting. In Section 3.2, we explain these restrictions in more detail and discuss prospects for generalizing our results.

1.2 Related work

Quantum oracle models

A fundamental constraint of quantum oracle models is that they must be unitary. We describe several nonstandard oracle models used in quantum computing:

- A *quantum* oracle is any unitary operation U in the full Hilbert space. Although the operation is unitary, the verifier doesn't necessarily have access to U^{-1}. Oracles like these are not typically classical because the unitary's action is not efficiently and classically representable.
- An *in-place* oracle maps $|x\rangle \to |\pi(x)\rangle$ for some classical invertible function π. Again, this computation is not efficiently reversible since the verifier may not have access to π^{-1}. When a standard oracle gives access to π^{-1}, an in-place oracle query can be simulated in two queries; otherwise, an exponential number of queries are required to construct one from the other [18].
- A *phase* oracle puts the output of a classical function f in the phase of a basis state. We consider the map $|x\rangle \to e^{f(x)\cdot 2\pi i/N}|x\rangle$. To contrast, note that the map $|c, x\rangle \to e^{cf(x)\cdot 2\pi i/N}|c, x\rangle$ is unitarily equivalent to the standard oracle.

All of these oracles can optionally have *internal randomness*, as considered by Harrow and Rosenbaum [16]; we call these *randomized* oracles. On every query to a randomized oracle, a unitary is chosen at random from a fixed set. This can be very powerful; for example, [16] gives examples of randomized oracles where problems *impossible* to decide with classical queries can be decided with a single quantum query.

QMA and QCMA

The *Merlin-Arthur* style of complexity classes considers a decision problem and two players. The magician (Merlin) has claimed the answer to the decision problem is YES, and gives the verifier a token (the *proof* or *witness*) to convince them. The verifier (Arthur) must then

ensure the answer is actually YES. Given a problem with size n, the verifier must accept a correct witness (i.e. when the answer is YES) with probability $1/q$ higher than a "lying" witness (i.e. when the answer is NO) for some $q = \text{poly}(n)$. The set of problems that can be decided this way in a classical setting is known as Merlin Arthur (MA). If the verifier is a quantum computer, this is QCMA; if the witness can be any quantum state, this is QMA.

Table 1 Complexity classes in the style of *Merlin-Arthur*. QCMA is a subset of QMA where the witness can be efficiently written as a classical bitstring.

	verifier is classical	**verifier is quantum**
witness is classical	MA	QCMA
witness is quantum	-	QMA

Since any classical bitstring can be efficiently written as a quantum state, QCMA $\subseteq$ QMA. But is the reverse true? Even the *oracle* version of this problem is open: at the top of a recent list of open problems, Aaronson asks for a standard oracle that separates the two classes [1]. All previous progress [2, 11, 22] relies on specifically chosen *nonstandardness* in the oracle.

Natarajan and Nirkhe [22] make progress on a standard oracle separation of QMA and QCMA by constructing an oracle with randomness. They simultaneously and independently provide a QMA protocol for testing non-expansion of a graph in an oracle. To prove their lower bound, they combine the adversary method, the polynomial method, and a reduction to a problem of Ambainis, Childs, and Liu [5]. However, their notion of randomness is different from ours and other works [16, 11, 6], and acts as follows: when an oracle is first queried, it chooses a function f from a distribution, but on subsequent queries, it uses the same function f. By contrast, our notion of randomness is memoryless: an oracle chooses f from a uniform distribution on F for *every* query. This allows one to make small changes to F without affecting the success of the QMA protocol; we use this flexibility to study a simpler permutation problem.

2 Our setup

Consider a d-regular graph on $N := 2^n$ vertices for any n and even d. We show that an invertible function can list the edges adjacent to each vertex in G.

▶ **Definition 2.1** (Graph-coded function)**.** *Consider a d-regular graph G (for even d) on N vertices. A G-coded function is a function $f : [N] \times [d/2] \to [N]$, such that $f_i(x) := f(x, i)$ is a bijection for each $i \in [d/2]$, and each edge is uniquely represented by a tuple $(x, f_i(x))$.*

▶ Remark 2.2 (Even-degree regular graphs have graph-coded functions). Every regular graph G of even degree has a G-coded function.

Proof. A d-regular graph G of even degree always has a 2-factorization [23]. This means that the edges of G can be partitioned into $d/2$ edge-disjoint subgraphs $[E_1, \ldots, E_{d/2}]$ where in each E_i, all vertices have degree two (i.e. a collection of cycles). Thus, we can represent each E_i with a permutation π_i, where the edge $(x, y) \in E_i$ if and only if $\pi_i(x) = y$ or $\pi_i(y) = x$. Then $f(x, i) := \pi_i(x)$ is a G-coded function. ◀

Graph-coded functions f are bijective, and therefore invertible. We now *present* f in various oracle models. Note that we define all oracles with access both to f and f^{-1}.

▶ **Definition 2.3** (An oracle model *presents* a function f)**.** *For each oracle model below (e.g. standard oracle), we say that this oracle model* presents *the function f.*

▶ Remark 2.4. For notational convenience, we refer to a qubit z that controls the inversion of a function f as taking on values in $\{\pm 1\}$, so that f^z is either $f^1 = f$ or f^{-1}.

▶ **Definition 2.5** (Standard oracle)**.** *For any $f : [N] \to [N]$, define $U_f : \mathbb{C}^{2N^2} \to \mathbb{C}^{2N^2}$ as*

$$U_f \sum_{c,x\in[N],z\in\{\pm1\}} \alpha_{c,x,z} |c,x,z\rangle := \sum_{c,x\in[N],z\in\{\pm1\}} \alpha_{c,x,z} |c \oplus f^z(x),x,z\rangle . \tag{1}$$

▶ **Definition 2.6** (In-place oracle [18])**.** *For any permutation $\pi : [N] \to [N]$, define $\widetilde{U}_\pi : \mathbb{C}^{2N} \to \mathbb{C}^{2N}$ as*

$$\widetilde{U}_\pi \sum_{x\in[N],z\in\{\pm1\}} \beta_{x,z} |x,z\rangle := \sum_{x\in[N],z\in\{\pm1\}} \beta_{x,z} |\pi^z(x),z\rangle . \tag{2}$$

▶ Remark 2.7 ([18]). A standard oracle U_f (with access to f^{-1}) can simulate an in-place oracle $\widetilde{U}_f$ in two queries:

$$(\mathbb{I}\otimes X)\circ U_f \circ (SWAP_{n,n}\otimes X)\circ U_f |0\rangle^{\otimes n} |x,z\rangle = |0\rangle^{\otimes n} |f^z(x),z\rangle . \tag{3}$$

▶ **Definition 2.8** (N^{th} root of unity)**.** *Define the N^{th} root of unity as $\omega_N := e^{2\pi i/N}$.*

▶ **Definition 2.9** (Phase oracle)**.** *For any function $f : [N] \to [N]$, define $\overline{U}_f : \mathbb{C}^{2N} \to \mathbb{C}^{2N}$ as*

$$\overline{U}_f \sum_{x\in[N],z\in\{\pm1\}} \alpha_{x,z} |x,z\rangle := \sum_{x\in[N],z\in\{\pm1\}} \alpha_{x,z}\omega_N^{f^z(x)} |x,z\rangle . \tag{4}$$

We describe how an oracle in our setup exhibits internal randomness. On each query, a *randomized* oracle chooses a function uniformly from a set F. We say that a randomized oracle *presents* F.

▶ Remark 2.10. Given a unitary U, we use the notation $\mathcal{U}$ to denote an operator on density matrices; that is,

$$\mathcal{U}[\rho] := U\rho U^\dagger . \tag{5}$$

▶ **Definition 2.11** (Randomized oracle (e.g. [16, 11]))**.** *For any set F of functions $f : [N] \to [N]$ corresponding to oracles $\{U_f \mid f \in F\}$, define the linear operator $\mathcal{O}_F$ as*

$$\mathcal{O}_F := \frac{1}{|F|}\sum_{f\in F} \mathcal{U}_f . \tag{6}$$

We match the notation of randomized oracle $\mathcal{O}_F$ with oracle U_f; e.g. $\widetilde{\mathcal{O}}_F$ is a randomized in-place *oracle.*

2.1 Problem statements

The problems below are not fully specified without the choice of oracle model. We prepend the names below with the choice of oracle model; for example, we denote Problem 2.12 in a standard oracle as STANDARD NON-EXPANSION(d,α,ε).

▶ **Problem 2.12** (NON-EXPANSION(d, α, ε))**.** *Consider an oracle U_f presenting a G-coded function f.*

1. *In a YES instance, we are promised that G is a union of two disconnected d-regular graphs, and that the smaller graph has N^α vertices.*
2. *In a NO instance, we are promised that G is d-regular and has spectral gap at least ε (for example, an* expander graph*).*

The problem is to decide whether U_f is a YES instance or NO instance.

We also consider a version of this problem with *randomized* oracles, where each randomized YES instance is specified by the set of vertices V of the smaller graph. On each query, an oracle chooses an graph-coded function f uniformly at random that corresponds to a graph where V and $[N]/V$ are disconnected.

▶ **Problem 2.13** (RANDOMIZED NON-EXPANSION(d, α, ε))**.** *Consider a randomized oracle $\mathcal{O}_F$ presenting a set of graph-coded functions F.*

1. *Each subset $V \subseteq [N]$ of size $|V| = N^\alpha$ specifies a YES instance $\mathcal{O}_{F_V}$. Let F_V be the set of all G-coded functions of d-regular graphs G with no edges between V and $[N]/V$.*
2. *There is a single NO instance $\mathcal{O}_{F_\varnothing}$. Let $F_\varnothing$ be the set of all G-coded functions of d-regular graphs G with spectral gap at least ε.*

The problem is to decide whether $\mathcal{O}$ is a YES instance or a NO instance.

In the configuration model of a random graph, F_V contains all functions $f(x, i)$ such that $f_i(x) := f(x, i)$ is the union of a permutation on $[N]/V$ and a permutation on V. In fact, we can use the oracle's internal randomness to adjust the underlying set F, and even consider graphs that are not typically expander graphs.

▶ **Definition 2.14** (Subset indicator)**.** *For a set $V \subseteq [N]$, define the function $i_V : [N] \to \{V, [N]/V\}$ as*

$$i_V(x) = \begin{cases} V & x \in V \\ [N]/V & x \notin V\,. \end{cases} \tag{7}$$

▶ **Definition 2.15** (Permutations that stabilize a subset)**.** *For a set $V \subseteq [N]$, define the set of permutations*

$$T_V := \{\pi : [N] \to [N] \,:\, i_V(x) = i_V(\pi(x))\, \forall x \in [N]\}\,. \tag{8}$$

We say that T_V stabilizes *the subset V.*

▶ **Problem 2.16** (RANDOMIZED HIDDEN SUBSET(α))**.** *Consider a randomized oracle $\mathcal{O}_F$ presenting a set of functions F.*

1. *Each subset $V \subseteq [N]$ of size $|V| = N^\alpha$ specifies a YES instance $\mathcal{O}_{T_V}$.*
2. *There is a single NO instance $\mathcal{O}_{T_\varnothing}$, where $T_\varnothing$ is the set of all permutations of $[N]$.*

The problem is to decide whether $\mathcal{O}$ is a YES instance or a NO instance.

Notice that RANDOMIZED HIDDEN SUBSET(α) is exactly RANDOMIZED NON-EXPANSION$(2, \alpha, 0)$.

Notice that T_V is a group under function composition. One can generalize this algebraic structure to a problem distinguishing oracles presenting subgroups of $T_\varnothing$ from an oracle presenting $T_\varnothing$:

▶ **Problem 2.17** (RANDOMIZED HIDDEN SUBGROUP$(\times, \{H_i\})$)**.** *Consider the set $T_\varnothing$ of all permutations on $[N]$ as a group with operation $\times$, such that each $H_i \subsetneq T_\varnothing$ is also a group. Suppose a randomized oracle $\mathcal{O}$ presents either $T_\varnothing$ or any H_i.*

1. *Each subgroup H_i specifies a YES instance $\mathcal{O}_{H_i}$.*
2. *There is a single NO instance $\mathcal{O}_{T_\varnothing}$, where $T_\varnothing$ is the set of all permutations of $[N]$.*

The problem is to decide whether $\mathcal{O}$ is a YES instance or a NO instance.

For example, RANDOMIZED HIDDEN SUBSET(α) is a special case of RANDOMIZED HIDDEN SUBGROUP$(\times, \{H_i\})$ using the group operation of function composition.

3 Our results

3.1 Non-expansion and quantum witness

Our first result shows that there is a one-query QMA protocol for NON-EXPANSION(d, α, ε) in many oracle models presenting a graph-coded function.

▶ **Theorem 3.1.** *There is a* QMA *protocol for* STANDARD NON-EXPANSION(d, α, ε) *and* IN-PLACE NON-EXPANSION(d, α, ε) *at every even $d \geq 4$, all $0 < \alpha < \frac{1}{2}$, and all constant $\varepsilon > 0$.*

Graphs with good *expansion* are well-connected despite their sparsity. For any graph G, let A_G be the adjacency matrix of G, and $L_G = d\mathbb{I} - A_G$ be the *graph Laplacian* of G. The smallest eigenvalue of L_G is $\lambda_1(L_G) = 0$, and the next-smallest eigenvalue $\lambda_2(L_G)$ measures the expansion of G. In this framework, NON-EXPANSION(d, α, ε) asks if an oracle presenting a G-coded function has $\lambda_2(L_G) = 0$ (YES), or if $\lambda_2(L_G) \geq \varepsilon$ (NO).

At the heart of our protocol is the *spectral test*, which takes an input state $|\psi\rangle$ and fails with probability proportional to $\langle\psi| L_G |\psi\rangle$. We describe the spectral test for both standard oracles and in-place oracles in Appendix A.1. A state that passes the spectral test is essentially supported on a subspace according to $\lambda(L_G) = o(\frac{1}{\mathrm{poly}(n)})$; in a NO instance, this is one-dimensional, and in a YES instance, this is at least two-dimensional. In fact, the uniform superposition over all inputs, $|+\rangle^{\otimes n}$, is always in this subspace. As a result, our protocol (Theorem 3.1) either runs the spectral test, or checks if the input state is close to $|+\rangle^{\otimes n}$.

Consider the randomized variant of NON-EXPANSION(d, α, ε). The graph of any graph-coded function presented in a YES instance is guaranteed to have a small set V (i.e. $|V| = N^\alpha$) disconnected from the rest of the graph. As a result, there is a state, defined only by the vertices of V, that is all-but-negligibly supported in the $\lambda(L_G) = 0$ subspace. This state is the *subset state* $|V\rangle$:

▶ **Definition 3.2** (Subset state)**.** *For any non-empty subset $S \subseteq [N]$, define the subset state of S as*

$$|S\rangle := \frac{1}{\sqrt{|S|}} \sum_{x \in S} |x\rangle \ . \tag{9}$$

Since $|V\rangle$ is a good witness for *every* graph encoded in a YES instance, the QMA protocol works just as well in the randomized setting (Theorem A.4).

Randomized oracles that present a set F of graph-coded functions are stable to small changes in the set F. In fact, an oracle presenting F encoding all d-regular expander graphs is indistinguishable from an oracle presenting F encoding all d-regular graphs. The latter oracle can be simulated with $d/2$ queries to the NO instance of RANDOMIZED HIDDEN SUBSET(α); we show in Theorem A.5 that the same QMA protocol can also decide this problem.

3.2 Randomized in-place oracles and classical witness

Our second result shows that a general class of verifiers cannot decide IN-PLACE RANDOMIZED HIDDEN SUBSET(α).

▶ **Theorem 3.3** (informal). *No quantum verifier with $O(\log(n))$ additional workspace qubits can efficiently decide* IN PLACE RANDOMIZED HIDDEN SUBSET(α) *given a polynomial sized classical witness.*

Recall that in this problem, a verifier has access to a quantum channel and a polynomial-sized classical witness, and must distinguish whether the oracle presents a uniformly random permutation or a permutation that stabilizes a hidden subset V. Let $\mathcal{Y}$ be the set of all YES instances; note that each instance is uniquely defined by a subset V.

Suppose there exists a QCMA algorithm for this problem. Since there are at most $O(2^{\mathrm{poly}(n)})$ different classical witnesses, there exists a set of YES instances $\mathcal{Y}'$ that share the same witness, such that $|\mathcal{Y}'| / |\mathcal{Y}| = \Omega(2^{-\mathrm{poly}(n)})$. We can refute the existence of such an algorithm by proving that the same verification "strategy" cannot distinguish all instances of $\mathcal{Y}'$ from the NO case with non-negligible probability. A "strategy" is exactly a quantum algorithm: a series of unitaries and oracle queries, followed by a POVM. Without loss of generality, a T-query algorithm alternates between unitaries and oracle queries on $\mathcal{H}_O \otimes \mathcal{H}_W$ followed by a measurement[1], where $\mathcal{H}_O$ is the Hilbert space of the "oracle" qubits and $\mathcal{H}_W$ is the extra workspace:

$$\mathcal{E}_O[\rho_0] = (\mathcal{O} \otimes \mathbb{I}) \circ \mathcal{U}_T \circ \ldots \circ \mathcal{U}_2 \circ (\mathcal{O} \otimes \mathbb{I}) \circ \mathcal{U}_1[\rho_0]\,. \tag{10}$$

One may try to use the hybrid argument of Bennett, Bernstein, Brassard, and Vazirani [8] and Ambainis [4] to prove that the diamond norm $\left|\mathcal{E}_{\widetilde{\mathcal{O}}_{T_V}} - \mathcal{E}_{\widetilde{\mathcal{O}}_{T_\varnothing}}\right|_\diamond$ is small in expectation over the choice of $\widetilde{\mathcal{O}}_{T_V} \in \mathcal{Y}'$. This would imply that the verifier cannot distinguish all instances of $\mathcal{Y}'$ with the same strategy. We can consider the optimal distinguishing probability in terms of $\left|\mathcal{E}_{\widetilde{\mathcal{O}}_{T_V}}[\rho_0] - \mathcal{E}_{\widetilde{\mathcal{O}}_{T_\varnothing}}[\rho_0]\right|_1$ for some fixed $\rho_0 \in \mathcal{H}_O \otimes \mathcal{H}_W$.

However, this statistical argument does not hold for some choices of $\mathcal{Y}'$. Consider the following simple example: $\mathcal{Y}'$ contains all V such that $1 \in V$. First, $\mathcal{Y}'$ satisfies the size implied by the pigeonhole principle. Second, for $\rho_0 = |1\rangle\langle 1| \otimes \mathbb{I}$, $\left|\widetilde{\mathcal{O}}_{T_V}[\rho_0] - \widetilde{\mathcal{O}}_{T_\varnothing}[\rho_0]\right|_1$ is large for *all* instances in $\mathcal{Y}'$, since $|1\rangle\langle 1|$ mixes only within a small subset. Note that this only implies the existence of an *instance-specific* POVM distinguishing each YES instance in $\mathcal{Y}'$ from the NO instance. By contrast, a verification strategy has a *fixed* POVM $\{E, \mathbb{I} - E\}$. This allows us to prove that the following value is small on average over the choice of V:

$$\left|\mathrm{Tr}\left[E\mathcal{E}_{\widetilde{\mathcal{O}}_{T_V}}[\rho_0]\right] - \mathrm{Tr}\left[E\mathcal{E}_{\widetilde{\mathcal{O}}_{T_\varnothing}}[\rho_0]\right]\right| \tag{11}$$

We must bound this value for arbitrary choices of E, ρ_0 and $\mathcal{U}_i$ fixed in the algorithm. In order to do this, we leverage tools from representation theory; this allows us to see randomized oracles in our problem as *orthogonal projectors* into a subspace of matrices with low dimension, and prove that the density matrix of good distinguishers is characterized

[1] Note that the last operation does not have to be a unitary – one can simply replace a unitary followed by a POVM with another equivalent POVM.

by the hidden subset. One caveat of our technique is that the verifier is only allowed to have $O(\log(n))$ extra workspace qubits. This restriction is necessary to reduce the subspace dimension to regimes we can handle.

Representation theory has been previously used to study symmetric operators on variables (in probability) or qubits (in quantum computing) using the language of de Finetti theorems (e.g. [15]); these operators project into subspaces of permutation-invariant sequences or quantum states. By contrast, we notice that some randomized oracles are symmetric operators on *density matrices*. This allows us to explicitly find an orthogonal basis for the associated symmetric subspaces. We match oracle models with problems with the same group structure: RANDOMIZED HIDDEN SUBSET(α) for in-place oracles in Appendix B.1, and an analogous special case of RANDOMIZED HIDDEN SUBGROUP$(\times, \{H_i\})$ for standard oracles in Appendix B.2.

3.3 Randomized phase oracles: no witness can help

Our third result shows that deciding RANDOMIZED HIDDEN SUBSET(α) in a phase oracle is much harder than other oracle models we consider. A random phase has *zero* expectation. We use this fact to show that queries to most YES instances and the NO instance reduce the magnitude of each off-diagonal of the density matrix by an exponential factor, regardless of the input state. We bound the Frobenius norm of the difference of query outputs to show that these instances are statistically indistinguishable when the state space is not too large. As a result, no untrustworthy witness can help decide this problem.

▶ **Theorem 3.4** (informal). *No quantum verifier with $o(n)$ additional workspace qubits can efficiently decide* PHASE RANDOMIZED HIDDEN SUBSET(α) *given* any *unbounded witness. Moreover, these verifiers require an exponential number of queries to statistically distinguish a YES instance from the NO instance, for* each *of asymptotically all YES instances.*

We defer the formal proof to Appendix C. Note that the query lower bound here is *statistical*. In the NO instance, a witness is designed to fool the verifier; in order to overcome this, the verifier must use the witness in tandem with the oracle. But this cannot be done efficiently: distinguishing the NO instance from nearly any YES instance requires an exponential number of queries, *regardless of input state*.

In fact, Theorem 3.4 holds for any oracle that sends $|c, x, z\rangle \to \omega_N^{c \cdot f^z(x)} |c, x, z\rangle$, where the c register has $k = o(n)$ qubits, while at $k = n$ qubits, the oracle is unitarily equivalent to a standard oracle, and thus has a QMA protocol for RANDOMIZED HIDDEN SUBSET(α) by Theorem 3.1.

References

1 Scott Aaronson. Open problems related to quantum query complexity, 2021. arXiv:2109.06917.

2 Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. *Theory Comput.*, 3(1):129–157, 2007. arXiv:quant-ph/0604056.

3 Dorit Aharonov and Tomer Naveh. Quantum np - a survey, 2002. arXiv:quant-ph/0210077.

4 Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing - STOC '00*. ACM Press, 2000. arXiv:quant-ph/0002066.

5 Andris Ambainis, Andrew M. Childs, and Yi-Kai Liu. Quantum property testing for bounded-degree graphs. *Lecture Notes in Computer Science*, pages 365–376, 2011. doi:10.1007/978-3-642-22935-0_31.

6 Atul Singh Arora, Alexandru Gheorghiu, and Uttam Singh. Oracle separations of hybrid quantum-classical circuits, 2022. arXiv:2201.01904.

7 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998. doi:10.1145/1236457.1236459.

8 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, October 1997. doi:10.1137/s0097539796300933.

9 Andrew Childs. Lecture Notes on Quantum Algorithms, 2022. URL: https://www.cs.umd.edu/~amchilds/qa/qa.pdf.

10 Paul Erdos. On the central limit theorem for samples from a finite population. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 4:49–61, 1959. URL: https://old.renyi.hu/~p_erdos/1959-13.pdf.

11 Bill Fefferman and Shelby Kimmel. Quantum vs classical proofs and subset verification, 2018. arXiv:1510.06750.

12 Joel Friedman. A proof of Alon's second eigenvalue conjecture and related problems. *CoRR*, 2004. arXiv:cs/0405020.

13 Chris Godsil and Hanmeng Zhan. Discrete-time quantum walks and graph structures. *Journal of Combinatorial Theory, Series A*, 167:181–212, 2019. doi:10.1016/j.jcta.2019.05.003.

14 Alex Bredariol Grilo, Iordanis Kerenidis, and Jamie Sikora. QMA with subset state witnesses. In *Mathematical Foundations of Computer Science 2015 - 40th International Symposium*. Springer, 2015. doi:10.1007/978-3-662-48054-0_14.

15 Aram W. Harrow. The church of the symmetric subspace, 2013. arXiv:1308.6595.

16 Aram W. Harrow and David J. Rosenbaum. Uselessness for an oracle model with internal randomness, 2011. arXiv:1111.1462.

17 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip*=re. *Communications of the ACM*, 64(11):131–138, 2021. arXiv:2001.04383.

18 Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Physical Review A*, 65(5), May 2002. doi:10.1103/physreva.65.050304.

19 Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020. URL: http://www.cs.umd.edu/~jkatz/imc.html.

20 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993. doi:10.1137/0222080.

21 Andrew Lutomirski. Component mixers and a hardness result for counterfeiting quantum money, 2011. arXiv:1107.0321.

22 Anand Natarajan and Chinmay Nirkhe. A classical oracle separation between qma and qcma, 2022. arXiv:2210.15380.

23 Julius Petersen. Die Theorie der regulären graphs. *Acta Mathematica*, 15:193–220, 1900. doi:10.1007/BF02392606.

24 Jason D. M. Rennie. Relating the trace and frobenius matrix norms, August 2005. URL: http://people.csail.mit.edu/jrennie/writing/traceFrobenius.pdf.

25 Adi Shamir. IP=PSPACE. *Journal of the ACM (JACM)*, 39(4):869–877, 1992. doi:10.1145/146585.146609.

26 Luca Trevisan. Max cut and the smallest eigenvalue, 2008. arXiv:0806.1978.

27 John Watrous. Quantum simulations of classical random walks and undirected graph connectivity, 1998. arXiv:cs/9812012.

A Verifying non-expansion with a quantum witness

A.1 The spectral test

We give a test that takes an input state $|\psi\rangle = \sum_{x\in[N]} a_x |x\rangle$ on n qubits, and fails with probability proportional to $\langle\psi| L_G |\psi\rangle$. This relies on a curious fact:

▶ **Lemma A.1.** *Consider a d-regular graph G (for even d) on 2^n vertices and a G-coded function f. Suppose we have a normalized quantum state $|\psi\rangle = \sum_{x\in[N]} a_x |x\rangle$ on n qubits. Then*

$$\sum_{i\in[d/2]} \sum_{x\in[N]} \|a_x \pm a_{f(x,i)}\|^2 = d \pm \langle\psi| A_G |\psi\rangle \ . \tag{12}$$

Proof.

$$\sum_{i\in[d/2]} \sum_{x\in[N]} \|a_x \pm a_{f(x,i)}\|^2 = \sum_{i\in[d/2]} \sum_{x\in[N]} \|a_x\|^2 + \|a_{f(x,i)}\|^2 \pm (a_x a^*_{f(x,i)} + a^*_x a_{f(x,i)}) \tag{13}$$

$$= d \pm \sum_{i\in[d/2]} \sum_{x\in[N]} (a_x a^*_{f(x,i)} + a^*_x a_{f(x,i)}) \tag{14}$$

$$= d \pm \langle\psi| A_G |\psi\rangle . \tag{15}$$

◀

We construct the spectral test with one query either to a standard oracle or in-place oracle presenting a graph-coded function f. The former (Procedure A.2) is a SWAP test but with an oracle query in the middle. The latter (Procedure A.3) relies on controlled access to the in-place oracle.

▶ **Procedure A.2** (Spectral test with a standard oracle). *Consider a d-regular graph G on $N = 2^n$ vertices where d is even, and normalized state $|\psi\rangle = \sum_{x\in[N]} a_x |x\rangle \in \mathbb{C}^N$. We assume access to a standard oracle $U_f : \mathbb{C}^{k\times k}$ for $k = N^2 2^{\lceil \log_2 d\rceil}$, which acts on a basis vector as*

$$U_f |c, x, i, z\rangle = |c \oplus f^z(x, i), x, i, z\rangle \ , \tag{16}$$

for $c, x \in [N]$, $i \in 2^{\lceil \log_2 d\rceil - 1}$, and $z \in \{\pm 1\}$.

1. *Pick $i \in [d/2]$ uniformly at random, and prepare the state $|i\rangle \in \mathbb{C}^{2^{\lceil \log_2 d\rceil - 1}}$.*
2. *Prepare a qubit in the state $|+\rangle = \frac{|1\rangle + |-1\rangle}{\sqrt{2}} \in \mathbb{C}^2$. (Recall that we label the values of this register in $\{\pm 1\}$.)*
3. *Combine n registers $|0\rangle^{\otimes n}$, the input state $|\psi\rangle$, and $|i\rangle$ and $|+\rangle$ to create $|0\rangle^{\otimes n} |\psi\rangle |i\rangle |+\rangle$.*
4. *Apply the oracle U_f, which creates the state*

$$\frac{1}{\sqrt{2}} \sum_{x\in[N]} a_x \left(|f(x,i)\rangle |x\rangle |i\rangle |1\rangle + \left|f^{-1}(x,i)\right\rangle |x\rangle |i\rangle |-1\rangle \right) . \tag{17}$$

5. *Swap the first two sets of n qubits, controlled by the last qubit. This creates the state*

$$\frac{1}{\sqrt{2}} \sum_{x\in[N]} a_x \left(|f(x,i)\rangle |x\rangle |i\rangle |1\rangle + |x\rangle \left|f^{-1}(x,i)\right\rangle |i\rangle |-1\rangle \right) \tag{18}$$

$$= \frac{1}{\sqrt{2}} \sum_{x\in[N]} a_x |f(x,i)\rangle |x\rangle |i\rangle |1\rangle + \frac{1}{\sqrt{2}} \sum_{x\in[N]} a_{f(x,i)} |f(x,i)\rangle |x\rangle |i\rangle |-1\rangle \ . \tag{19}$$

6. *Apply a Hadamard on the last qubit, which creates the state*

$$\frac{1}{2}\sum_{x\in[N]}(a_x + a_{f(x,i)})\,|f(x,i)\rangle\,|x\rangle\,|i\rangle\,|1\rangle + \frac{1}{2}\sum_{x\in[N]}(a_x - a_{f(x,i)})\,|f(x,i)\rangle\,|x\rangle\,|i\rangle\,|-1\rangle\ . \tag{20}$$

7. *Measure the last qubit and accept if it is* 1.
Moreover, by Lemma A.1, this procedure fails with probability

$$\frac{1}{d/2}\sum_{i\in[d/2]}\sum_{x\in[N]}\frac{\|a_x - a_{f(x,i)}\|^2}{4} = \frac{\langle\psi|\,L_G\,|\psi\rangle}{2d}\,. \tag{21}$$

▶ **Procedure A.3** (Spectral test with an in-place oracle)**.** *Consider a d-regular graph G on $N = 2^n$ vertices where d is even, and normalized state $|\psi\rangle = \sum_{x\in[N]} a_x\,|x\rangle \in \mathbb{C}^N$. We assume controlled access to an in-place oracle $\widetilde{U}_f : \mathbb{C}^{k\times k}$ for $k = N2^{\lceil\log_2 d\rceil+1}$, which acts on a basis vector as*

$$\widetilde{U}_f\,|a,x,i,z\rangle = |a, f^{a\cdot z}(x,i), i, z\rangle\ . \tag{22}$$

for control qubit $a \in \{0,1\}$, $x \in [N]$, $i \in 2^{\lceil\log_2 d\rceil-1}$, and $z \in \{\pm 1\}$.[2]

1. *Pick $i \in [d/2]$ uniformly at random, and prepare the state $|i\rangle \in \mathbb{C}^{2^{\lceil\log_2 d\rceil-1}}$.*
2. *Prepare a qubit in the state $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}} \in \mathbb{C}^2$.*
3. *Combine $|+\rangle$, the input state $|\psi\rangle$, $|i\rangle$, and a register $|1\rangle$ to create $|+\rangle\,|\psi\rangle\,|i\rangle\,|1\rangle$.*
4. *Apply the oracle $\widetilde{U}_f$, which creates the state*

$$\frac{1}{\sqrt{2}}\sum_{x\in[N]} a_x\,(|0\rangle\,|x\rangle + |1\rangle\,|f(x,i)\rangle)\,|i\rangle\,|1\rangle \tag{23}$$

$$=\frac{1}{\sqrt{2}}\sum_{x\in[N]}(a_x\,|0\rangle + a_{f^{-1}(x,i)}\,|1\rangle)\,|x\rangle\,|i\rangle\,|1\rangle\ . \tag{24}$$

5. *Apply a Hadamard on the first qubit, which creates the state*

$$\frac{1}{2}\sum_{x\in[N]}\big((a_x + a_{f^{-1}(x,i)})\,|0\rangle + (a_x - a_{f^{-1}(x,i)})\,|1\rangle\big)\,|x\rangle\,|i\rangle\,|1\rangle\ . \tag{25}$$

6. *Measure the first qubit and accept if it is* 0.
Moreover, by Lemma A.1, this procedure fails with probability

$$\frac{1}{d/2}\sum_{i\in[d/2]}\sum_{x\in[N]}\frac{\|a_x - a_{f^{-1}(x,i)}\|^2}{4} = \frac{\langle\psi|\,L_G\,|\psi\rangle}{2d}\,. \tag{26}$$

A.2 A one-query protocol

▶ **Theorem 3.1.** *There is a* QMA *protocol for* STANDARD NON-EXPANSION(d,α,ε) *and* IN-PLACE NON-EXPANSION(d,α,ε) *at every even $d \geq 4$, all $0 < \alpha < \frac{1}{2}$, and all constant $\varepsilon > 0$.*

[2] Note that this procedure does not actually need access to the inverse of f to conduct the spectral test.

Proof. Suppose the oracle presents a G-coded function. Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ be the eigenvalues of the graph Laplacian L_G. Note that the smallest eigenvalue of a regular graph G is $\lambda_1 = 0$. We always choose the eigenvector associated with λ_1 as a uniform superposition over vertices of the graph (i.e. $|\lambda_1\rangle := |[N]\rangle = |+\rangle^{\otimes n}$).

Suppose Arthur receives a state $|\psi\rangle = \sum_{i\in[N]} \alpha_i |\lambda_i\rangle$ from Merlin. Consider the following strategy:

- With probability $\frac{1}{2}$, measure $|\psi\rangle$ in the Hadamard basis. Fail if it is in the basis state according to $|+\rangle^{\otimes n}$, and pass otherwise.
- With probability $\frac{1}{2}$, use the spectral test (Procedure A.2 or Procedure A.3, respectively).

The probability of failure FAIL is

$$\text{FAIL} = \frac{1}{2}\| \langle\psi| |+\rangle^{\otimes n} \|^2 + \frac{1}{2}\frac{\langle\psi| L_G |\psi\rangle}{2d} \tag{27}$$

$$= \frac{1}{2}\left(\|\alpha_1\|^2 + \frac{1}{2d}\sum_{i=1}^{N} \lambda_i \|\alpha_i\|^2\right). \tag{28}$$

In a NO instance, $\lambda_k = \Omega(1)$ for all $k > 1$. So the probability of failure is always a positive constant:

$$\text{FAIL}_{\text{NO}} = \frac{1}{2}\|\alpha_1\|^2 + \frac{1}{2d}\sum_{i=2}^{N} \Omega(\|\alpha_i\|^2) = \Omega(\sum_{i=1}^{N} \|\alpha_i\|^2) = \Omega(1)\,. \tag{29}$$

In a YES instance, the spectrum of L_G is the combined spectrum of the two disconnected graphs. This means $\lambda_1 = \lambda_2 = 0$, and the associated eigenvectors are linear combinations of $|V\rangle$ and $|[N]/V\rangle$. Recall that $|\lambda_1\rangle := |+\rangle^{\otimes n}$. We find the orthogonal eigenvector $|\lambda_2\rangle$ in this subspace by inspection:

$$|\lambda_2\rangle = \sqrt{\frac{N-|V|}{N}}\,|V\rangle + \sqrt{\frac{|V|}{N}}\,|[N]/V\rangle\ . \tag{30}$$

Note that any vector with $\|\alpha_2\|^2 = 1 - o(\frac{1}{\text{poly}(n)})$ has negligible probability of failure:

$$\text{FAIL}_{\text{YES}} = \frac{1}{2}\|\alpha_1\|^2 + \frac{1}{2d}\sum_{i=1}^{N} \lambda_i\|\alpha_i\|^2 = O(\|\alpha_1\|^2 + \sum_{i=3}^{N} \|\alpha_i\|^2) = O(1 - \|\alpha_2\|^2)\,. \tag{31}$$

Suppose Merlin sends the subset state $|V\rangle$. Since $\| \langle V|\lambda_2\rangle \|^2 = 1 - \frac{|V|}{N} = 1 - O(\frac{1}{2^{\text{poly}(n)}})$, the strategy has probability of failure $O(\frac{1}{2^{\text{poly}(n)}})$. ◀

In general, the spectral test can be used in a QMA protocol to test the magnitude of the second-smallest or largest eigenvalue of a graph Laplacian to inverse polynomial precision. The former is a measure of the quality of a graph's expansion, and the latter is related to a measure of a graph's bipartiteness named the *bipartiteness ratio* [26].

Because this QMA protocol requires only one query of either a standard oracle or an in-place oracle, it works even when these oracles are randomized.

▶ **Theorem A.4.** *There is a* QMA *protocol for* STANDARD RANDOMIZED NON-EXPANSION(d, α, ε) *and* IN-PLACE RANDOMIZED NON-EXPANSION(d, α, ε) *at every even* $d \geq 4$*, all* $0 < \alpha < \frac{1}{2}$*, and all constant* $\varepsilon > 0$*.*

Proof. The strategy in Theorem 3.1 also works here. Consider any G-coded function presented in a YES instance; the same vertices V are exactly the vertices of the smaller component of G. So the witness $|V\rangle$ is close to the second eigenvector $|\lambda_2\rangle$, and the failure probability is negligible. Now consider any G-coded function presented in a NO instance. By definition, G is an expander graph, so the failure probability is always a positive constant. ◀

Because a randomized oracle chooses a function uniformly from a set F, it is statistically indistinguishable from an oracle with exponentially small changes to F. We use this fact to simplify the NO instance in RANDOMIZED NON-EXPANSION(d, α, ε). Suppose the NO instance instead presents graph-coded functions of *all* d-regular graphs. Since $1 - O(\frac{1}{\text{poly}(N)}) = 1 - O(\frac{1}{2^{\text{poly}(n)}})$ graphs have a constant spectral gap [12] when $d > 2$, the failure probability in the QMA protocol changes by at most $O(\frac{1}{2^{\text{poly}(n)}})$.

Notice that with this modification, the oracles are exactly $d/2$ copies of the oracles in RANDOMIZED HIDDEN SUBSET(α). One way to interpret this is that the randomization offers a substitute for expander graphs. An expander graph is sparse but well-mixing; a randomized oracle query instantaneously mixes across a graph's connected component. As a result, we can distinguish degree-2 graphs with this QMA protocol, even though they are not typically expander graphs:

▶ **Theorem A.5.** *There is a* QMA *protocol for* STANDARD RANDOMIZED HIDDEN SUBSET(α) *and* IN-PLACE RANDOMIZED HIDDEN SUBSET(α) *for all* $0 < \alpha < \frac{1}{2}$.

Proof. Perhaps surprisingly, the strategy in Theorem 3.1 also works here:

- Consider the graph G of any G-coded function presented in a YES instance. By definition, the vertices V are disconnected from all vertices in $[N]/V$. So the witness $|V\rangle$ is close to the second eigenvector $|\lambda_2\rangle$, and the failure probability is negligible.
- Consider the NO instance. Then f is chosen uniformly from the set $T_\varnothing$ of all permutations of $[N]$. Then the spectral test fails with probability

$$\mathop{\mathbb{E}}_{\pi\in T_\varnothing}\left[\frac{d - \langle\psi| A_\pi |\psi\rangle}{2d}\right]\Bigg|_{d=2} = \frac{1}{2} - \frac{1}{4}\mathop{\mathbb{E}}_{\pi\in T_\varnothing}\left[\langle\psi| A_\pi |\psi\rangle\right] \tag{32}$$

$$= \frac{1}{2} - \frac{1}{4}\left(\frac{1}{N!}\sum_{\pi\in T_\varnothing}\langle\psi| A_\pi |\psi\rangle\right) \tag{33}$$

$$= \frac{1}{2} - \frac{1}{8}\left(\frac{1}{(N!)^2}\sum_{\pi_1,\pi_2\in T_\varnothing}\langle\psi|\left(A_{\pi_1} + A_{\pi_2}\right)|\psi\rangle\right). \tag{34}$$

  The matrix $A_{\pi_1} + A_{\pi_2}$ determines the adjacency matrix of a random 4-regular graph in the configuration model; as a result,

$$\mathop{\mathbb{E}}_{\pi\in T_\varnothing}\left[\frac{d - \langle\psi| A_\pi |\psi\rangle}{2d}\right]\Bigg|_{d=2} = \mathop{\mathbb{E}}_{\pi_1,\pi_2\in T_\varnothing}\left[\frac{d - \langle\psi| A_{\pi_1,\pi_2} |\psi\rangle}{2d}\right]\Bigg|_{d=4}. \tag{35}$$

  Since a random 4-regular graph has constant spectral gap with probability $1 - O(\frac{1}{\text{poly}(N)}) = 1 - O(\frac{1}{2^{\text{poly}(n)}})$ [12], the failure probability is at least FAIL_{YES} from Theorem 3.1, less $O(\frac{1}{2^{\text{poly}(n)}})$. So the failure probability is $\Omega(1)$, just as before. ◀

B Randomized oracles and symmetric subspaces

We first formalize how randomized oracles are orthogonal projectors. We include the proofs in the arXiv version.

▶ **Definition B.1** (Representation of a group). *Consider a group G and a vector space V. A* representation *of G is a map R that sends each $g \in G$ to a linear operator $R(g) : \mathsf{V} \to \mathsf{V}$ such that $R(g_1 g_2) = R(g_1) \circ R(g_2)$ for all $g_1, g_2 \in G$.*

▶ **Theorem B.2** (Projecting onto the symmetric subspace [15, Proposition 2]). *Consider a finite group G, a vector space V, and a representation $R : G \to L(\mathsf{V})$. Then the operator*

$$\Pi_R := \frac{1}{|G|} \sum_{g \in G} R(g) \tag{36}$$

is an orthogonal projector onto $\mathsf{V}^G \subseteq \mathsf{V}$, where

$$\mathsf{V}^G := \{v \in \mathsf{V} \,:\, R(g)[v] = v \,\forall g \in G\}\,. \tag{37}$$

▶ **Theorem B.3** (Oracles on density matrices form a representation). *Consider a group G of functions $f : [N] \to [N]$ with bitwise $\oplus$ as the group operation. Then the map $f \mapsto \mathcal{U}_f$ is a representation over the vector space of $2N^2 \times 2N^2$ complex matrices.*

Similarly, consider a group $\widetilde{G}$ of permutations $\pi : [N] \to [N]$ with composition as the group operation. Then the map $\pi \mapsto \widetilde{\mathcal{U}}_\pi$ is a representation over the vector space of $2N \times 2N$ complex matrices.

▶ **Theorem B.4** (Some randomized oracles are orthogonal projectors). *Consider a group G of functions $f : [N] \to [N]$ with bitwise $\oplus$ as the group operation. Then $\mathcal{O}_G$ is an orthogonal projector, under the Frobenius inner product $(x|y) = \mathrm{Tr}\left[x^\dagger y\right]$ for $x, y \in \mathbb{C}^{2N^2 \times 2N^2}$, onto*

$$\mathsf{V}_G := \{\rho \in \mathbb{C}^{2N^2 \times 2N^2} \,:\, \mathcal{U}_f[\rho] = \rho \,\forall f \in G\}\,. \tag{38}$$

Similarly, consider a group $\widetilde{G}$ of permutations $\pi : [N] \to [N]$ with composition as the group operation. Then $\widetilde{\mathcal{O}}_{\widetilde{G}}$ is an orthogonal projector, under the Frobenius inner product $(x|y) = \mathrm{Tr}\left[x^\dagger y\right]$ for $x, y \in \mathbb{C}^{2N \times 2N}$, onto

$$\widetilde{\mathsf{V}}_{\widetilde{G}} := \{\rho \in \mathbb{C}^{2N \times 2N} \,:\, \widetilde{\mathcal{U}}_\pi[\rho] = \rho \,\forall \pi \in \widetilde{G}\}\,. \tag{39}$$

In IN-PLACE RANDOMIZED HIDDEN SUBSET(α), a quantum verifier is either given $\widetilde{\mathcal{O}}_{T_\varnothing}$ (NO) or $\widetilde{\mathcal{O}}_{T_V}$ for some $V \subseteq [N]$ where $|V| = N^\alpha$ (YES). Since $T_V \subseteq T_\varnothing$, the symmetric subspace according to $T_\varnothing$ is a subspace of that according to T_V, i.e. $\widetilde{\mathsf{V}}_{T_\varnothing} \subseteq \widetilde{\mathsf{V}}_{T_V}$. So we can exactly find the basis of the symmetric subspaces $\widetilde{\mathsf{V}}_{T_\varnothing}$ and $\widetilde{\mathsf{V}}_{T_V}$ (see the arXiv version for details). This key property is used throughout Appendix B.1.

B.1 In-place oracles: when classical witnesses are not enough

We interpret IN-PLACE RANDOMIZED HIDDEN SUBSET(α) as distinguishing the set of all permutations from a subgroup that stabilizes a small subset $V \subseteq [N]$. In Theorem 3.3, we prove that classical witnesses designed for the verifier to choose YES cannot help a quantum verifier efficiently decide this problem. This requires three main lemmas. First, we show in Lemma B.6 that input states distinguishing a YES instance or NO instance must have knowledge of the hidden subset V (either as a subset state $|V\rangle$ or a mixed state $\mathbb{I}_V$). However,

no density matrix can be close to too many subset states $|V\rangle$ (Lemma B.7), and no POVM can choose the right answer for too many mixed states $\mathbb{I}_V$ (Lemma B.8). We combine these facts in a hybrid argument; note that we must fix an algorithm by its unitaries *and* its POVM. We formally state the lemmas (proofs are in the arXiv version), and then prove Theorem 3.3.

We use the following measure of "progress" for the hybrid argument:

▶ **Definition B.5** (Difference of oracle queries)**.** *For any ρ, let $d_{V,\rho}$ be the difference of the two oracle queries*

$$d_{V,\rho} := \widetilde{\mathcal{O}}_{T_V}[\rho] - \widetilde{\mathcal{O}}_{T_\varnothing}[\rho]\,. \tag{40}$$

If the nuclear norm of $d_{V,\rho}$ is non-negligible, we say that ρ is a good distinguisher of $\widetilde{\mathcal{O}}_{T_V}$ and $\widetilde{\mathcal{O}}_{T_\varnothing}$. We show that every good distinguisher ρ has a certain form; we include the proof in the arXiv version.

▶ **Lemma B.6** (Good distinguishers have a certain form)**.** *Consider a density matrix ρ and up to $O(\log(n))$ extra workspace qubits. Suppose $\|d_{V,\rho}\|_1 = \Omega(\frac{1}{\mathrm{poly}(n)})$. Then among the quantities*

$$\langle V,z|\,\rho\,|V,z\rangle\,, \tag{41}$$

$$\mathrm{Tr}\left[\rho\left(\mathbb{I}_{V,z} - \frac{|V|}{N}\mathbb{I}_{[N],z}\right)\right], \tag{42}$$

for any $z \in \{\pm 1\}$, at least one has magnitude $\Omega(\frac{1}{\mathrm{poly}(n)})$.

We now state two lemmas about subsets and subset states. These help us prove that no quantum state can be a good distinguisher of too many YES instances. We include the proofs in the arXiv version.

▶ **Lemma B.7** (Can't approximate too many subset states)**.** *Consider a Hermitian $N \times N$ matrix ρ that is positive semidefinite and has trace at most 1. Consider the set of all subsets $V \subseteq [N]$, where $|V| = N^\alpha$ for a fixed $0 < \alpha < \frac{1}{2}$. Then the fraction of subsets V such that $\langle V|\,\rho\,|V\rangle = \Omega(\frac{1}{\mathrm{poly}(n)})$ decreases faster than any exponential in $\mathrm{poly}(n)$.*

▶ **Lemma B.8** (Not too many subsets can have elevated mean)**.** *Consider any $N \times N$ POVM $\{E, \mathbb{I} - E\}$, and the set of all subsets $V \subseteq [N]$, where $|V| = N^\alpha$ for a fixed $0 < \alpha < \frac{1}{2}$. Then the fraction of subsets V where*

$$|f(V)| := \left|\frac{1}{|V|}\,\mathrm{Tr}[\mathbb{I}_V E] - \frac{1}{N}\,\mathrm{Tr}[E]\right| = \Omega(\frac{1}{\mathrm{poly}(n)})\,, \tag{43}$$

decreases faster than any exponential in $\mathrm{poly}(n)$.

Intuitively, Lemma B.7 and Lemma B.8 hold because subset states can approximate *any* quantum state well. Grilo, Kerenidis, and Sikora [14] show that for any n-qubit quantum state $|\psi\rangle$, there exists a subset state $|S\rangle$ such that $|\,\langle S|\psi\rangle\,| \geq \frac{1}{8\sqrt{n+3}}$.

We now prove the main statement:

▶ **Theorem 3.3** (formal)**.** *No quantum verifier that entangles oracle queries with at most $O(\log(n))$ additional qubits can efficiently decide* IN PLACE RANDOMIZED HIDDEN SUBSET(α) *for any $0 < \alpha < \frac{1}{2}$, even with a polynomial-length classical witness designed for the verifier to choose YES.*

Proof. Let the set of YES instances be $\mathcal{Y}$; note that each YES instance corresponds to a set $V \subseteq [N]$ where $|V| = N^\alpha$, for some fixed $0 < \alpha < \frac{1}{2}$.

Suppose for contradiction that there is a protocol for this problem at some $\alpha < \frac{1}{2}$. Then the verifier can distinguish $\widetilde{\mathcal{O}}_{T_\varnothing}$ from any $\widetilde{\mathcal{O}}_{T_V}$ in a polynomial number of queries using a classical witness of size $O(\text{poly}(n))$. By the pigeonhole principle, there must exist a set of YES instances $\mathcal{Y}'$ such that $|\mathcal{Y}'|/|\mathcal{Y}| = \Omega(\frac{1}{2^{\text{poly}(n)}})$, where the verifier can use the *same algorithm* to distinguish $\widetilde{\mathcal{O}}_{T_\varnothing}$ from *every* YES instance in $\mathcal{Y}'$.

We then construct a *hybrid argument* in the style of Bennett, Bernstein, Brassard, and Vazirani [8] and Ambainis [4], which interpolates from queries of one oracle to queries of another oracle. For simplicity we write the proof without extra workspace qubits; however, we can have up to $O(\log(n))$ extra workspace qubits to satisfy Lemma B.6. Any polynomial query algorithm can be written as a set of unitaries $A = \{U^{(1)}, \ldots, U^{(k)}\}$ for some $k = O(\text{poly}(n))$ (alternating between unitary evolutions and oracle queries), and a POVM $\{E, \mathbb{I} - E\}$. Consider the following "hybrid" algorithms:

▶ **Definition B.9.** *Given any set of k unitaries $A = \{U^{(1)}, \ldots, U^{(k)}\}$, define the hybrid algorithm*

$$A_{V,\ell}\left[\rho_0\right] = \widetilde{\mathcal{O}}_{T_V}^{(k)} \circ \mathcal{U}^{(k)} \circ \cdots \circ \widetilde{\mathcal{O}}_{T_V}^{(\ell+1)} \circ \mathcal{U}^{(\ell)} \circ \widetilde{\mathcal{O}}_{T_\varnothing}^{(\ell)} \circ \mathcal{U}^{(\ell)} \circ \ldots \widetilde{\mathcal{O}}_{T_\varnothing}^{(1)} \circ \mathcal{U}^{(1)}\left[\rho_0\right], \tag{44}$$

which evolves ρ_0 under the oracle $\mathcal{O}_{T_\varnothing}$ for ℓ steps and under $\mathcal{O}_{T_V}$ for the other $k - \ell$ steps.

Then the following is true for each $\widetilde{\mathcal{O}}_{T_V} \in \mathcal{Y}'$:

$$\Omega(\frac{1}{\text{poly}(n)}) = |\text{Tr}[EA_{V,k}[\rho_0]] - \text{Tr}[EA_{V,0}[\rho_0]]| \leq \sum_{i=0}^{k-1} |\text{Tr}[EA_{V,i+1}[\rho_0]] - \text{Tr}[EA_{V,i}[\rho_0]]|\,, \tag{45}$$

which implies

$$\Omega(\frac{1}{\text{poly}(n)}) = \sum_{i=0}^{k-1} \left|\text{Tr}\left[E\left(\widetilde{\mathcal{O}}_{T_V}^{(k)} \circ \ldots \mathcal{U}^{(i)} \circ \left(\widetilde{\mathcal{O}}_{T_V}^{(i)} - \widetilde{\mathcal{O}}_{T_\varnothing}^{(i)}\right)[\rho^{(i)}]\right)\right]\right| = \sum_{i=0}^{k-1} \left|\text{Tr}\left[E^{V,(i)} d_{V,\rho^{(i)}}\right]\right|\,, \tag{46}$$

for the operator $E^{V,(i)}$ constructed by

$$E^{V,(i)} = \mathcal{U}^{\dagger(i)} \circ \widetilde{\mathcal{O}}_{T_V}^{(i)} \circ \cdots \circ \mathcal{U}^{\dagger(k)} \circ \widetilde{\mathcal{O}}_{T_V}^{(k)}\left[E\right]. \tag{47}$$

By Fact D.8, the operators $E^{V,(i)}$ and $\mathbb{I} - E^{V,(i)}$ are also Hermitian and positive semidefinite, so $\{E^{V,(i)}, \mathbb{I} - E^{V,(i)}\}$ is a POVM.

Using the pigeonhole principle, there must be a step ℓ in the summation with magnitude $\Omega(\frac{1}{\text{poly}(n)})$. Each $\widetilde{\mathcal{O}}_{T_V} \in \mathcal{Y}'$ has such a step. Again by the pigeonhole principle, there is a ℓ^* and set $\mathcal{Y}^* \subseteq \mathcal{Y}'$ where

$$\left|\text{Tr}\left[E^{V,(\ell^*)} d_{V,\rho^{(\ell^*)}}\right]\right| = \Omega(\frac{1}{\text{poly}(n)})\,, \tag{48}$$

and $|\mathcal{Y}^*|/|\mathcal{Y}'| \geq \frac{1}{k} = \Omega(\frac{1}{\text{poly}(n)})$. Notice that this implies $|\mathcal{Y}^*|/|\mathcal{Y}| = \Omega(\frac{1}{2^{\text{poly}(n)}})$.

Since the trace of M with a POVM operator is at most $\|M\|_1$ (Fact D.5), we have for all $\widetilde{\mathcal{O}}_{T_V} \in \mathcal{Y}^*$,

$$\Omega(\frac{1}{\text{poly}(n)}) = \left|\text{Tr}\left[E^{V,(\ell^*)} d_{V,\rho^{(\ell^*)}}\right]\right| \leq \left\|d_{V,\rho^{(\ell^*)}}\right\|_1. \tag{49}$$

When queries are entangled with at most $O(\log(n))$ additional qubits, the premise of Lemma B.6 holds; then one of the quantities in the theorem statement must be large. However, Lemma B.7 says that a given ρ can only satisfy either of the first two quantities for a smaller-than-exponential fraction of $\mathcal{Y}$. So for most choices of $\widetilde{\mathcal{O}}_{T_V} \in \mathcal{Y}^*$,

$$\mathrm{Tr}\left[\rho^{(\ell^*)}\left(\mathbb{I}_{V,z} - \frac{|V|}{N}\mathbb{I}_{[N],z}\right)\right] = \Omega(\frac{1}{\mathrm{poly}(n)})\,. \tag{50}$$

for at least one of $z \in \{\pm 1\}$.

Inspecting the proof of Lemma B.6, this implies $d_{V,\rho^{(\ell^*)}}$ can only have $\Omega(\frac{1}{\mathrm{poly}(n)})$ weight on $C_{4,z}$ for some $z \in \{\pm 1\}$ across all matrices in $\mathcal{C}$. In fact, for most choices of $\widetilde{\mathcal{O}}_{T_V} \in \mathcal{Y}^*$, we show that this is also true for

$$d_{V,\ell^*,j} := \widetilde{\mathcal{O}}_{T_V}^{(\ell^*+j)} \circ \mathcal{U}^{(\ell^*+j)} \circ \cdots \circ \widetilde{\mathcal{O}}_{T_V}^{(\ell^*+1)} \circ \mathcal{U}^{(\ell^*+1)} \circ d_{V,\rho^{(\ell^*)}}\,, \tag{51}$$

for all $0 \le j \le k - \ell^*$. We show this by induction. Note that by Fact D.5 and the fact that $d_{V,\ell^*,k-\ell^*}$ is the difference of two objects with nuclear norm 1, $\|d_{V,\ell^*,k-\ell^*}\|_1 = \Omega(\frac{1}{\mathrm{poly}(n)}) = O(1)$.

Consider $d_{V,\ell^*,i}$ for some $1 \le i \le k - \ell^*$, which can be represented with the basis $\mathcal{C}$. By Fact D.9, it has Frobenius norm at most $\|d_{V,\rho^{(\ell^*)}}\|_{Fr} = O(\frac{1}{\sqrt{|V|}})$. So it must have $o(\frac{1}{\mathrm{poly}(n)})$ weight on pure states. Inspecting the basis $\mathcal{C}$, this means $d_{V,\rho^{(\ell^*)}}$ can only have $\Omega(\frac{1}{\mathrm{poly}(n)})$ weight on $C_{4,z}$ or $\frac{1}{N}\mathbb{I}_{[N],z}$ for $z \in \{\pm 1\}$. By Fact D.9, $d_{V,\rho^{(\ell^*)}}$ has nuclear norm at least $\|d_{V,\ell^*,k-\ell^*}\|_1 = \Omega(\frac{1}{\mathrm{poly}(n)})$, so it must have $\Omega(\frac{1}{\mathrm{poly}(n)})$ weight on at least one such matrix. Suppose for contradiction that the matrix is $\frac{1}{N}\mathbb{I}_{[N],z}$ for $z \in \{\pm 1\}$. Then

$$\Omega(\frac{1}{\mathrm{poly}(n)}) = \mathrm{Tr}\left[\mathbb{I}_{[N],z}\widetilde{\mathcal{O}}_{T_V}\left[\mathcal{U}^{(\ell^*+i)}\left[d_{V,\ell^*,i-1}\right]\right]\right] = \mathrm{Tr}\left[\left(\mathcal{U}^{(\ell^*+i)\dagger} \circ \widetilde{\mathcal{O}}_{T_V}^{\dagger}[\mathbb{I}_{[N],z}]\right) d_{V,\ell^*,i-1}\right]. \tag{52}$$

By the inductive hypothesis, $d_{V,\ell^*,i-1}$ only has $\Omega(\frac{1}{\mathrm{poly}(n)})$ weight on some $C_{4,z}$ for $z \in \{\pm 1\}$. Then for some $z' \in \{\pm 1\}$,

$$\Omega(\frac{1}{\mathrm{poly}(n)}) = \mathrm{Tr}\left[\left(\mathcal{U}^{(\ell^*+i)\dagger} \circ \widetilde{\mathcal{O}}_{T_V}^{\dagger}[\mathbb{I}_{[N],z}]\right)\left(\frac{1}{|V|}\mathbb{I}_{V,z'} - \frac{1}{N}\mathbb{I}_{[N],z'}\right)\right]. \tag{53}$$

Notice that for any unitary U, the object $\{\mathcal{U}^{(\ell^*+i)\dagger} \circ \widetilde{\mathcal{O}}_{T_V}^{\dagger}[\mathbb{I}_{[N],z=+1}], \mathcal{U}^{(\ell^*+i)\dagger} \circ \widetilde{\mathcal{O}}_{T_V}^{\dagger}[\mathbb{I}_{[N],z=-1}]\}$ forms a POVM. By Lemma B.8, this can only be satisfied at either $z \in \{\pm 1\}$ for a smaller-than-exponential fraction of choices of V. So for most choices of $\widetilde{\mathcal{O}}_{T_V} \in \mathcal{Y}^*$ (i.e. a $\Omega(\frac{1}{2^{\mathrm{poly}(n)}})$ fraction of choices of V), $d_{V,\ell^*,i}$ has $\Omega(\frac{1}{\mathrm{poly}(n)})$ weight on $C_{4,z}$ for at least one of $z \in \{\pm 1\}$, and for no other matrices in $\mathcal{C}$.

Since $\Omega(\frac{1}{\mathrm{poly}(n)}) = |\mathrm{Tr}[E d_{V,\ell^*,k-\ell^*}]|$, our supposition then implies that for one of $z \in \{\pm 1\}$,

$$\Omega(\frac{1}{\mathrm{poly}(n)}) = |\mathrm{Tr}[EC_{4,z}]| = \left|\mathrm{Tr}\left[E\left(\frac{1}{|V|}\mathbb{I}_{V,z} - \frac{1}{N}\mathbb{I}_{[N],z}\right)\right] \pm O(\frac{1}{2^{\mathrm{poly}(n)}})\right|. \tag{54}$$

But by Lemma B.8, this can only be satisfied at either $z \in \{\pm 1\}$ for a smaller-than-exponential fraction of $\mathcal{Y}$. This is a contradiction. So there can be no efficient protocol for this problem. ◀

B.2 Standard oracles: when classical witnesses are enough

As shown in Theorem B.3, randomized standard oracles can also form a representation. But the preserved group structure is much different than for randomized in-place oracles. Consider the set $T_\varnothing$ of permutations on $[N]$. For any $f_1, f_2 \in T_\varnothing$, the element $f_1 f_2$ in this group structure acts for all $x \in [N]$ and $z \in \{\pm 1\}$ as

$$(f_1 f_2)^z(x) = f_1^z(x) \oplus f_2^z(x)\,. \tag{55}$$

Note that this operation is abelian; that is, $(f_1 f_2) = (f_2 f_1)$. Any finite abelian group can always be represented as the direct sum of cyclic groups. In fact, under this group operation, $T_\varnothing$ can be decomposed by the input $x \in [N]$ and function inverter $z \in \{\pm 1\}$:

$$T_\varnothing = \bigoplus_{x\in[2^n],z\in\{\pm1\}} \mathbb{Z}_{2^n}\,. \tag{56}$$

With this group operation, the only possible subgroups of $T_\varnothing$ have the form

$$\bigoplus_{x\in[2^n],z\in\{\pm1\}} \mathbb{Z}_{2^{k_{x,z}}}\,, \tag{57}$$

for $0 \le k_{x,z} \le n$. As a result, there is a QCMA protocol to distinguish any strict subgroup of $T_\varnothing$ from $T_\varnothing$.

▶ **Theorem B.10.** *There is a one-query* QCMA *protocol for* STANDARD RANDOMIZED HIDDEN SUBGROUP$(\times, \{H_i\})$ *when the group operation* $\times$ *is bitwise XOR, for any valid* $\{H_i\}$.

Proof. Suppose the classical witness is a bitstring of length at least $n+1$. The verifier can then:

1. Use the first n bits to construct x and the next bit to construct z.
2. Prepare the state $|0\rangle^{\otimes n}|x,z\rangle$.
3. Apply $\mathcal{O}_H$, creating the state $|f^z(x)\rangle|x,z\rangle$ for some $f \in H$.
4. Measure the first n qubits, and accept if the result is even.[3]

Consider a YES instance associated with a subgroup $H \subsetneq T_\varnothing$. Then H will have some $x \in [N], z \in \{\pm 1\}$ such that $k_{x,z} < n$. A witness can store x and z; since $k_{x,z} < n$, $f^z(x)$ will be even with probability 1.

In the NO instance, $H = T_\varnothing$. Then $f^z(x)$ is even with probability 0.5 for every $x \in [N], z \in \{\pm 1\}$. ◀

Note that Theorem B.10 holds even if the randomized standard oracle $\mathcal{O}_F$ does not have access to the function inverse.

C No witness is enough for phase oracles

▶ **Theorem 3.4** (formal)**.** *No quantum verifier that entangles oracle queries with at most* $o(n)$ *additional qubits can efficiently decide* PHASE RANDOMIZED HIDDEN SUBSET(α) *for any* $0 < \alpha < \frac{1}{2}$*, even with* any *witness designed for the verifier to choose YES. Moreover, these verifiers require an exponential number of queries to statistically distinguish a YES instance from the NO instance, for* each *of asymptotically all YES instances.*

[3] Depending on the encoding, one can simply measure the n^{th} qubit, and accept if the result is 0.

Proof. We now prove the query lower bound. Let k be the number of queries required to distinguish $\overline{\mathcal{O}}_{T_V}$ from $\overline{\mathcal{O}}_{T_\varnothing}$. Consider any algorithm that distinguishes the two instances, defined by a starting state ρ_0, k unitaries, k oracle queries, and a POVM $\{E, \mathbb{I} - E\}$. In the framework of hybrid algorithms (Definition B.9),

$$\Omega(\frac{1}{\text{poly}(n)}) = |\text{Tr}[EA_{V,k}[\rho_0]] - \text{Tr}[EA_{V,0}[\rho_0]]| \tag{58}$$

$$\leq \|A_{V,k}[\rho_0] - A_{V,0}[\rho_0]\|_1 \tag{59}$$

$$\leq k \max_{i\in\{0,\ldots,k-1\}} \|A_{V,i+1}[\rho_0] - A_{V,i}[\rho_0]\|_1 \tag{60}$$

$$\leq k \max_{i\in\{0,\ldots,k-1\}} \left\|\overline{\mathcal{O}}_{T_V}[\rho^{(i)}] - \overline{\mathcal{O}}_{T_\varnothing}[\rho^{(i)}]\right\|_1 , \tag{61}$$

where the last line follows because randomized oracles do not increase the nuclear norm (Fact D.9).

We now bound $\left\|\overline{\mathcal{O}}_{T_V}[\rho] - \overline{\mathcal{O}}_{T_\varnothing}[\rho]\right\|_1$ for *any* ρ. Recall that a phase oracle $\overline{O}_F$ acts as

$$\overline{O}_F\left[|x_1, z_1\rangle\langle x_2, z_2|\right] = \frac{1}{|F|}\sum_{f\in F} \omega_N^{f^{z_1}(x_1) - f^{z_2}(x_2)} |x_1, z_1\rangle\langle x_2, z_2| , \tag{62}$$

for any $x_1, x_2 \in [N]$ and $z_1, z_2 \in \{\pm 1\}$. So every basis vector $|x_1, z_1\rangle\langle x_2, z_2|$ acquires a coefficient c_{x_1,z_1,x_2,z_2}.

We start with $\overline{O}_{T_\varnothing}$ (the NO instance). When $(x_1, z_1) = (x_2, z_2)$, the coefficient is 1. When $x_1 \neq x_2$, $f^z(x_1)$ and $f^{-z}(x_2)$ are uniformly likely to be any value, so the coefficient is

$$\frac{1}{N^2}\sum_{a\in[N],b\in[N]} \omega_N^{a-b} = \frac{1}{N^2}\left\|\sum_{a\in[N]} \omega_N^a\right\|^2 = 0 . \tag{63}$$

Similarly, when $x_1 \neq x_2$, $f^z(x_1)$ and $f^z(x_2)$ are uniformly likely to be any unequal values; the coefficient is

$$\frac{1}{N(N-1)}\sum_{a\in[N],b\in[N],a\neq b} \omega_N^{a-b} = \frac{1}{N(N-1)}\Big[\sum_{a\in[N],b\in[N]} \omega_N^{a-b} - \sum_{a\in[N]} \omega_N^{a-a}\Big] = -\frac{1}{N-1} . \tag{64}$$

We now consider $\overline{O}_{T_V}$ (a YES instance). When $(x_1, z_1) = (x_2, z_2)$, the coefficient is again 1. When $x_1 \neq x_2$, the values of $f^z(x_1)$ and $f^{-z}(x_2)$ are uniformly likely to be any value in $i_V(x_1)$ and $i_V(x_2)$, respectively, so the coefficient is

$$\frac{1}{|i_V(x_1)| \times |i_V(x_2)|}\sum_{a\in i_V(x_1),b\in i_V(x_2)} \omega_N^{a-b} . \tag{65}$$

Similarly, when $x_1 \neq x_2$, $f^z(x_1)$ and $f^z(x_2)$ are uniformly likely to be any unequal values in $i_V(x_1)$ and $i_V(x_2)$, respectively, so the coefficient is

$$\frac{1}{|i_V(x_1)| \times |i_V(x_2)|}\sum_{a\in i_V(x_1),b\in i_V(x_2),a\neq b} \omega_N^{a-b} = \frac{\left(\sum_{a\in i_V(x_1),b\in i_V(x_2)} \omega_N^{a-b}\right) - \delta|i_V(x_1)|}{|i_V(x_1)| \times |i_V(x_2) - \delta|} , \tag{66}$$

where δ is 1 if $i_V(x_1) = i_V(x_2)$ and 0 otherwise.

Consider the object $\overline{\mathcal{O}}_{T_V}[\rho] - \overline{\mathcal{O}}_{T_\varnothing}[\rho]$ as the sum of two matrices $A_V + B_V$. Let A_V contain the $(V,z) \times (V,z)$ submatrix for both $z \in \{\pm 1\}$, and B_V contain the rest of the entries. When the oracle query is entangled with $o(n)$ additional qubits, A_V has rank $O(|V|)$, and B_V has rank $O(N)$.

Since the roots of unity sum to zero, $\sum_{a\in V} \omega_N^a = -\sum_{a\in[N]/V} \omega_N^a$ for any $V \subseteq [N]$. Because of this,

$$\left\| \sum_{a,b\in i_V(x)} \omega_N^{a-b} \right\| \leq \left\| \sum_{a\in i_V(x)} \omega_N^a \right\|^2 \leq \left\| \sum_{a\in V} \omega_N^a \right\|^2 = O(|V|^2)\,. \tag{67}$$

As a result, all coefficients in B_V are $O(\frac{1}{N^{1-\alpha}})$.

In Lemma C.1, we show that for asymptotically all choices of V, all coefficients in A_V are $O(\frac{1}{N^{3\alpha/4}})$. This argument uses a Chernoff bound and a central limit argument on samples without replacement.

We bound the nuclear norm of $\overline{\mathcal{O}}_{T_V}[\rho] - \overline{\mathcal{O}}_{T_\varnothing}[\rho]$ with the rank and Frobenius norm of A_V and B_V (Fact D.3):

$$\begin{aligned}
\left\|\overline{\mathcal{O}}_{T_V}[\rho] - \overline{\mathcal{O}}_{T_\varnothing}[\rho]\right\|_1 &\leq \|A_V\|_1 + \|B_V\|_1 && (68)\\
&= O(\sqrt{V})\|A_V\|_{Fr} + O(\sqrt{N})\|B_V\|_{Fr} && (69)\\
&\leq \left(O(\sqrt{V})O(N^{-3\alpha/4}) + O(\sqrt{N})O(N^{\alpha-1})\right)\|\rho\|_{Fr} && (70)\\
&= O(N^{-\alpha/4} + N^{\alpha-1/2})\,. && (71)
\end{aligned}$$

Thus, for most choices of V, distinguishing $\overline{\mathcal{O}}_{T_V}$ and $\overline{\mathcal{O}}_{T_\varnothing}$ requires $k = \Omega(\min(N^{\alpha/4}, N^{1/2-\alpha}))$ queries. ◀

We now prove the Chernoff bound:

▶ **Lemma C.1.** *Fix any $0 < \alpha < \frac{1}{2}$, and consider all subsets $V \subseteq [N]$ such that $|V| = N^\alpha$. Then for all but a doubly exponentially small fraction of choices of V,*

$$\left\| \frac{1}{N^{2\alpha}} \sum_{a,b\in V} \omega_N^{a-b} \right\| = O(\frac{1}{N^{3\alpha/4}})\,. \tag{72}$$

Proof. Consider the distribution $X = \{\omega_N^k\}$ where k is chosen uniformly from N. Both $Re(X)$ and $Im(X)$ have mean zero and variance at most 1.

Take a size-N^α sample from the distribution X, *without replacement*. Denote Y as the distribution of the sample mean. Both $Re(Y)$ and $Im(Y)$ have expectation $Re(X) = Im(X) = 0$, and variance

$$\frac{\sigma_X^2}{N^\alpha}(1 - \frac{N^\alpha - 1}{N-1}) \leq \frac{1}{N^\alpha}\,. \tag{73}$$

Even when sampling without replacement, Y is asymptotically normally distributed [10]. So its moment generating function is

$$\mathrm{MGF}_Y[t] = e^{t\mu_Y + \sigma_Y^2 t^2/2} \leq e^{t^2/N^\alpha}\,. \tag{74}$$

We use a Chernoff bound to estimate when Y has magnitude at least $N^{-3\alpha/8}$. Notice that

$$\Pr[Y \geq a] = \Pr\left[e^{tY} \geq e^{ta}\right] \leq e^{-at}\mathrm{MGF}_Y[t] \leq e^{t^2/N^\alpha}e^{-at}\,, \tag{75}$$

so

$$\Pr\left[Y \geq \frac{0.5}{N^{3\alpha/8}}\right] \leq \inf_{t\geq 0} \exp\left(\frac{t^2}{N^\alpha} - \frac{0.5t}{N^{3\alpha/8}}\right) \underset{t=2N^{\alpha/2}}{\leq} \exp\left(4 - N^{\alpha/8}\right) = O(\frac{1}{\exp(\exp(n))}). \quad (76)$$

This implies that Y has magnitude at most $N^{-3\alpha/8}$ (and Y^2 at most $N^{-3\alpha/4}$) except in a doubly exponentially small fraction of choices of V. ◀

▶ Remark C.2. Consider an oracle that sends $|c, x, z\rangle \to \omega_N^{c\cdot f^z(x)} |c, x, z\rangle$, where the c register has k qubits. Note that Theorem 3.4 applies whenever $k = o(n)$. However, there must be a phase transition, since at $k = n$, this oracle is unitarily equivalent to a standard oracle, and thus has a QMA protocol for RANDOMIZED HIDDEN SUBSET(α) in Appendix A.

D Norms and inner products

Note that we work with arbitrary matrices, not just positive semidefinite ones.

▶ **Definition D.1** (Nuclear norm of a matrix). *The nuclear norm of a matrix M is the sum of its singular values; that is,*

$$\|M\|_1 = \sum_i \sigma_i(M) = \mathrm{Tr}\left[\sqrt{M^\dagger M}\right]. \quad (77)$$

▶ **Definition D.2** (Frobenius norm and inner product of a matrix). *The Frobenius inner product of $N \times N$ matrices A, B is*

$$(A|B) = \mathrm{Tr}\left[A^\dagger B\right] \quad (78)$$

This induces a norm, which is the square root of the sum of squares of the singular values:

$$\|A\|_{Fr} = \sqrt{\sum_i \sigma_i(A)^2} = \sqrt{\sum_{ij\in[N]} |A_{ij}|^2}. \quad (79)$$

▶ **Fact D.3.** *The nuclear norm of a matrix is at most the product of its Frobenius norm and the square root of its rank.*

Proof. See Rennie [24] for a proof with explanation. ◀

▶ **Fact D.4** (Nuclear norm of a positive semidefinite matrix). *The nuclear norm of a positive semidefinite Hermitian matrix is simply its trace; that is, if ρ is Hermitian and positive semidefinite, then*

$$\|\rho\|_1 = \mathrm{Tr}[\rho]. \quad (80)$$

Proof. For a Hermitian and positive semidefinite matrix, the eigenvalues are all real and nonnegative, so the singular values are exactly the eigenvalues. Alternatively, notice that $\rho = \sqrt{\rho^\dagger \rho}$ and use Definition D.1. ◀

▶ **Fact D.5** (POVM trace is at most the nuclear norm). *Consider any Hermitian matrix M and a POVM $\{E, \mathbb{I} - E\}$. Then*

$$\mathrm{Tr}[EM] \leq \|M\|_1. \quad (81)$$

Proof. Consider the singular value decomposition of a Hermitian $M = UDU^\dagger$. Then

$$\mathrm{Tr}[EM] = \mathrm{Tr}\big[(U^\dagger EU)D\big] = \mathrm{Tr}[E'D] \tag{82}$$

for some matrix E'. Note that $\{E', \mathbb{I} - E'\}$ make a POVM; they have the same eigenvalues of $\{E, \mathbb{I} - E\}$, respectively, and so are both positive semidefinite. Recall that the diagonal elements of a POVM are all nonnegative and at most 1. Then

$$\mathrm{Tr}[EM] = \mathrm{Tr}[E'D] = \sum_i E'_{ii} D_i \leq \sum_i |D_i| = \|D\|_1 = \|M\|_1 \,. \tag{83}$$

◀

▶ **Fact D.6** (Trace of outer product is inner product). *Consider vectors $|x\rangle, |y\rangle \in \mathbb{C}^m$ and a matrix $A \in \mathbb{C}^{m\times m}$. Then the inner product of $|y\rangle\langle x|$ and A is*

$$\mathrm{Tr}\big[(|y\rangle\langle x|)^\dagger A\big] = \mathrm{Tr}[A|x\rangle\langle y|] = \langle y|A|x\rangle \,. \tag{84}$$

Proof.

$$\mathrm{Tr}[A|x\rangle\langle y|] = \mathrm{Tr}\left[\sum_{i,k\in[m]} \left(\sum_{j\in[m]} A_{ij}x_j\right) y_k^\dagger\right]_i = \sum_{k,j\in[m]} A_{kj}x_j y_k^\dagger = \langle y|A|x\rangle \,. \tag{85}$$

◀

▶ Remark D.7 (Orthogonal basis for an input density matrix). We can decompose ρ into a basis $\mathcal{M}$ that is orthogonal under the Frobenius inner product $(a|b) = \mathrm{Tr}\big[a^\dagger b\big]$:

$$\rho = \sum_{M\in\mathcal{M}} c_M M \,. \tag{86}$$

Because the basis is orthogonal, for any $M \in \mathcal{M}$,

$$\mathrm{Tr}\big[M^\dagger \rho\big] = \sum_{M'\in\mathcal{M}} c_{M'} \mathrm{Tr}\big[M^\dagger M'\big] = c_M \|M\|_{Fr}^2 \,. \tag{87}$$

Moreover, by Cauchy-Schwarz, the inner product of M and ρ is at most the product of the norm of each, so

$$\|c_M M\|_{Fr} = \frac{\big|\mathrm{Tr}\big[M^\dagger\rho\big]\big|}{\|M\|_{Fr}} \leq \|\rho\|_{Fr} \,. \tag{88}$$

We also state two properties that hold for *any* randomized oracle:

▶ **Fact D.8** (Randomized oracles preserve trace, Hermiticity, and positive semidefiniteness). *Consider any randomized oracle $\mathcal{O}_F$ corresponding to a set of functions $f \in F$. Then $\mathcal{O}_F$ preserves the trace of its input. Moreover, if the input M is Hermitian, so is $\mathcal{O}_F[M]$; if M is also positive semidefinite, so is $\mathcal{O}_F[M]$.*

Proof. Consider any input matrix M. Then

$$\mathrm{Tr}[\mathcal{O}_F[M]] = \mathrm{Tr}\left[\frac{1}{|F|}\sum_{f\in F} \mathcal{U}_f[M]\right] = \frac{1}{|F|}\sum_{f\in F} \mathrm{Tr}\Big[U_f M U_f^\dagger\Big] = \frac{1}{|F|}\sum_{f\in F} \mathrm{Tr}[M] = \mathrm{Tr}[M] \,. \tag{89}$$

Now suppose M is Hermitian; that is, $M^\dagger = M$. Then

$$\mathcal{O}_F[M]^\dagger = \left(\frac{1}{|F|}\sum_{f\in F}\mathcal{U}_f[M]\right)^\dagger = \frac{1}{|F|}\sum_{f\in F}\left(U_f M U_f^\dagger\right)^\dagger = \frac{1}{|F|}\sum_{f\in F}U_f M^\dagger U_f = \mathcal{O}_F[M]\,. \quad (90)$$

Furthermore, suppose M is positive semidefinite; that is, there is a matrix B such that $M = B^\dagger B$. Then

$$\mathcal{O}_F[M] = \frac{1}{|F|}\sum_{f\in F}\mathcal{U}_f[M] = \frac{1}{|F|}\sum_{f\in F}U_f B^\dagger B U_f^\dagger = \sum_{f\in F}\left(BU_f\right)^\dagger\left(BU_f\right)\,, \quad (91)$$

which is a sum of positive semidefinite matrices. Thus, $\mathcal{O}_F[M]$ is positive semidefinite. ◀

▶ **Fact D.9** (Randomized oracles do not increase nuclear norm or Frobenius norm)**.** *Consider any randomized oracle $\mathcal{O}_F$ corresponding to a set of functions $f \in F$. Then $\mathcal{O}_F$ does not increase the nuclear norm nor the Frobenius norm of its input.*

Proof. Recall that both the nuclear norm and Frobenius norm are unitarily invariant. Now consider any input matrix M. Then the nuclear norm of $\mathcal{O}_F[M]$ is

$$\left\|\mathcal{O}_F[M]\right\|_1 = \left\|\frac{1}{|F|}\sum_{f\in F}U_f M U_f^\dagger\right\|_1 \le \frac{1}{|F|}\sum_{f\in F}\left\|U_f M U_f^\dagger\right\|_1 = \frac{1}{|F|}\sum_{f\in F}\|M\|_1 = \|M\|_1\,. \quad (92)$$

The Frobenius norm of $\mathcal{O}_F[M]$ follows in exactly the same way. ◀

We use one additional property of density matrices in the proof of Lemma B.6:

▶ **Fact D.10.** *Consider any $N \times N$ density matrix ρ and normalized states $|v\rangle, |w\rangle$. If $|\langle v|\rho|w\rangle| = \Omega(\frac{1}{\mathrm{poly}(n)})$, then both $\langle v|\rho|v\rangle$ and $\langle w|\rho|w\rangle$ are $\Omega(\frac{1}{\mathrm{poly}(n)})$.*

Proof. Recall that a density matrix is Hermitian and positive semidefinite, so it is diagonalizable and has real and nonnegative eigenvalues. As a result, it has a decomposition

$$\rho = S^\dagger \Lambda S = S^\dagger\sqrt{\Lambda}\sqrt{\Lambda}S = (\sqrt{\Lambda}S)^\dagger(\sqrt{\Lambda}S) = A^\dagger A\,, \quad (93)$$

for some diagonal Λ and $A := \sqrt{\Lambda}S$. Then by Cauchy-Schwarz,

$$|\langle v|\rho|w\rangle|^2 = \left|(A|v\rangle)^\dagger(A|w\rangle)\right|^2 \le \left|(A|v\rangle)^\dagger(A|v\rangle)\right| \cdot \left|(A|w\rangle)^\dagger(A|w\rangle)\right| = \langle v|\rho|v\rangle \cdot \langle w|\rho|w\rangle\,. \quad (94)$$

Since $\mathrm{Tr}[\rho] = 1$, $\langle\psi|\rho|\psi\rangle \le 1$ for all normalized states $|\psi\rangle$. Thus, both $\langle v|\rho|v\rangle$ and $\langle w|\rho|w\rangle$ are at least $|\langle v|\rho|w\rangle|^2 = \Omega(\frac{1}{\mathrm{poly}(n)})$. ◀

E Our setup contrasted with a discrete-time quantum walk

The way one stores a graph in an oracle drastically changes the difficulty of some problems. Consider a *discrete-time quantum walk* [27], which allows a vertex access to a superposition of its neighbors.[4] Given a d-regular graph $G(V, E)$, the operator $W : \mathbb{C}^{N^2 \times N^2}$ acts as

$$W = \left(\sum_{(j,k)\in E} |j,k\rangle \langle k,j| \right) C \tag{95}$$

$$C = \sum_{j\in V} |j\rangle \langle j| \otimes (2 |\partial_j\rangle \langle \partial_j| - \mathbb{I}) \tag{96}$$

$$|\partial_j\rangle = \frac{1}{\sqrt{d}} \sum_{(j,k)\in E} |k\rangle \ . \tag{97}$$

Using a discrete-time quantum walk, we can learn about the mixing properties of the associated graph; these are fundamentally related to the graph's spectral gap [13].

By contrast, we query each neighbor of a vertex $v \in G$ with the value of the registers encoding $i \in [d/2]$ (defined by a G-coded function). For example, [5] uses a similar oracle to show that deciding whether a graph is a single expander graph or two equal-sized disconnected expander graphs is outside of BQP. Intuitively, a lack of superposition access to neighbors of a vertex makes it harder for a quantum computer to "traverse" the graph.

[4] [9, Chapter 17] has a good introduction to this topic.

Efficient Tomography of Non-Interacting-Fermion States

Scott Aaronson
The University of Texas at Austin, TX, USA

Sabee Grewal
The University of Texas at Austin, TX, USA

Abstract

We give an efficient algorithm that learns a non-interacting-fermion state, given copies of the state. For a system of n non-interacting fermions and m modes, we show that $O(m^3n^2\log(1/\delta)/\epsilon^4)$ copies of the input state and $O(m^4n^2\log(1/\delta)/\epsilon^4)$ time are sufficient to learn the state to trace distance at most ϵ with probability at least $1-\delta$. Our algorithm empirically estimates one-mode correlations in $O(m)$ different measurement bases and uses them to reconstruct a succinct description of the entire state efficiently.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Theory of computation → Quantum information theory; Mathematics of computing → Probabilistic inference problems; Theory of computation → Quantum complexity theory

Keywords and phrases free-fermions, Gaussian fermions, non-interacting fermions, quantum state tomography, efficient tomography

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.12

Related Version *Full Version*: `https://arxiv.org/abs/2102.10458`

Funding *Scott Aaronson*: Supported by a Vannevar Bush Fellowship from the US Department of Defense, the Berkeley NSF-QLCI CIQC Center, a Simons Investigator Award, and the Simons "It from Qubit" collaboration.
Sabee Grewal: Supported by Scott Aaronson's Vannevar Bush Fellowship from the US Department of Defense, Berkeley NSF-QLCI CIQC Center, Simons Investigator Award, and Simons "It from Qubit" collaboration.

Acknowledgements We thank Andrew Zhao for notifying us that the previous version of this manuscript contained an error and providing other insightful comments. We also thank Yuxuan Zhang, Alex Kulesza, Ankur Moitra, William Kretschmer, Dax Enshan Koh, Andrea Rocchetto, and Patrick Rall for helpful discussions, and Alex Arkhipov, William Kretschmer, and Daniel Liang for helpful comments on a previous version of this manuscript.

1 Introduction

There are two types of particles in the universe: bosons and fermions. Bosons include force carriers, such as photons and gluons, and fermions include matter particles like quarks and electrons. Each particle can be in a certain mode (e.g., a position or state). For a system of n particles, a configuration of the system is described by specifying how many particles are in each of m modes. Bosons are particles where multiple occupancy of a mode is allowed, whereas fermions are particles where multiple occupancy is forbidden; that is, two or more fermions cannot occupy the same mode at once (this is the *Pauli exclusion principle*). It follows that a system of n fermions and m modes has $\binom{m}{n}$ possible configurations; we denote the set of possible configurations by $\Lambda_{m,n}$.

Our main result is an efficient algorithm (both in copy complexity and time complexity) for learning a non-interacting-fermion state (also called a free-fermion state or a Gaussian fermion state), which is a superposition over configurations in $\Lambda_{m,n}$. Even though a non-interacting-fermion state lives in an exponentially large Hilbert space, we show how to exploit its structure to output a succinct description of the state efficiently. A non-interacting-fermion state can be completely specified by an $m \times n$ column-orthonormal matrix A. Our algorithm measures copies of the input state in $O(m)$ different measurement bases, and uses the measurement data to reconstruct an $m \times n$ matrix $\hat{A}$ in polynomial time. We prove that a polynomial number of copies of the input state is enough for the output state to be ϵ-close to the original state in trace distance.

▶ **Theorem 1** (Main result). *Let $|\Psi\rangle$ be a state of n non-interacting fermions and m modes. There exists an algorithm that uses $O(m^3 n^2 \log(1/\delta)/\epsilon^4)$ copies of $|\Psi\rangle$, $O(m^4 n^2 \log(1/\delta)/\epsilon^4)$ classical time, and $O(m)$ measurement bases, and outputs a succinct description of a non-interacting-fermion state $|\hat{\Psi}\rangle$ that is ϵ-close in trace distance distance to $|\Psi\rangle$ with probability at least $1 - \delta$.*

Our algorithm can also be adapted to conventional quantum state tomography, which we explain in Section 5.

1.1 Main Ideas

Here and throughout, let $U \in \mathbb{C}^{m \times m}$ be the unitary that prepares the unknown non-interacting-fermion state $|\Psi\rangle$ from the standard initial state $|1_n\rangle$ (the state where the first n modes are occupied and the remaining are unoccupied), and let $A \in \mathbb{C}^{m \times n}$ be the column-orthonormal matrix corresponding to the first n columns of U. Define $K = (k_{ij}) \coloneqq AA^\dagger \in \mathbb{C}^{m \times m}$. We refer to K as the *kernel matrix* due to the connection between determinantal point processes and non-interacting fermions (which we discuss further in Section 1.2.1). In the physics literature, the kernel matrix is also called the one-body reduced density matrix (1-RDM) or the correlation matrix.

The elements of $\Lambda_{m,n}$ are the possible configurations of a system of n non-interacting fermions and m modes. Formally, $\Lambda_{m,n}$ is the set of all lists $S = (s_1, \ldots, s_m)$ such that $s_i \in \{0, 1\}$ and $\sum_{i \in [m]} s_i = n$. The set $\{|S\rangle_{S \in \Lambda_{m,n}}\}$ is a basis for n-fermion and m-mode systems, which we refer to as the standard basis. The $m \times n$ column-orthonormal matrix A describes the state

$$|\Psi\rangle = \sum_{S \in \Lambda_{m,n}} \det(A_S) \, |S\rangle \, ,$$

where, for $S = (s_1, \ldots, s_m) \in \Lambda_{m,n}$, A_S is the $n \times n$ submatrix obtained by removing row i of A if $s_i = 0$. Therefore, upon measurement, we observe the configuration $S \in \Lambda_{m,n}$ with probability

$$|\langle S|\Psi\rangle|^2 = |\det(A_S)|^2 = \det(K_S),$$

where, for $S = (s_1, \ldots, s_m)$, K_S is the $n \times n$ submatrix obtained by removing row and column i of K if $s_i = 0$. In other words, upon measurement, we observe a configuration $S \in \Lambda_{m,n}$ with probability equal to the corresponding principal minor of the kernel matrix K. The probability that any subset of k modes is occupied corresponds to a principal minor of order k, obtained as above (remove the rows and columns of K corresponding to unoccupied modes and compute the determinant of the resulting submatrix). For example, the diagonal

entries k_{ii} correspond to the one-mode correlations (i.e., k_{ii} is the probability that mode i is occupied). Passing $|\Psi\rangle$ through a unitary transformation $V \in \mathbb{C}^{m\times m}$ maps K to $VKV^\dagger$. Given copies of the unknown state $|\Psi\rangle$, our goal is to output a column-orthonormal matrix $\hat{A}$ such that $|\hat{\Psi}\rangle = \sum_S \det(\hat{A}_S)\,|S\rangle$ is ϵ-close to $|\Psi\rangle$ in trace distance.

At a high level, our algorithm constructs $\hat{K}$ (an approximation of the kernel matrix), computes a decomposition $\hat{K} = \hat{A}\hat{A}^\dagger$, and then outputs $\hat{A}$. Our algorithm begins by measuring $O(\log(1/\delta)/\gamma^2)$ copies of the input state in the standard basis to empirically estimate the one-mode correlations of the state to accuracy $\pm\gamma$. The estimates are obtained simply by computing the average number of times each mode was occupied and are the diagonal entries of $\hat{K}$. One can then estimate the (i,j) entry of K as follows. Apply the beamsplitter

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

on modes i and j, which maps the diagonal entry k_{ii} to $\frac{1}{2}(k_{ii}+k_{jj}+2\mathrm{Re}(k_{ij}))$, and measure the resulting state in the standard basis. Repeat this $O(\log(1/\delta)/\gamma^2)$ times. As we did before, average the number of times mode i is occupied to obtain an estimate for $\frac{1}{2}(k_{ii}+k_{jj}+2\mathrm{Re}(k_{ij}))$ to accuracy $\pm\gamma$. Finally, using the previously obtained estimates for k_{ii} and k_{jj}, solve for $\mathrm{Re}(k_{ij})$ (up to accuracy $\pm\gamma$). Repeat this process with the beamsplitter

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & i\\ 1 & -i\end{pmatrix}$$

to estimate the imaginary part of k_{ij} to accuracy $\pm\gamma$.

Our algorithm proceeds as follows. Simultaneously execute the process above on the pairs of modes $(1,2),(3,4),\ldots,(m-1,m)$, then $(1,3),(2,4),\ldots,(m-2,m)$, and so on, until all the off-diagonal entries are recovered. It is easy to check that $O(m)$ measurement bases are needed to recover all off-diagonal entries of K. Then, we compute $Q\Lambda Q^\dagger$, an eigendecomposition of $\hat{K}$. Finally, we set $\hat{A}$ to be the $m\times n$ matrix corresponding to the first n columns of Q, and return $\hat{A}$. Overall, the algorithm requires $O(m/\gamma^2)$ copies of the input state and $O(m^2/\gamma^2)$ time.

The technical part is to understand how far $|\hat{\Psi}\rangle$ is from $|\Psi\rangle$ in trace distance, given that our algorithm begins by learning the entries of K to within γ in magnitude. To do this, we give a new proof that learning the kernel matrix is enough to learn the state, despite the kernel matrix only consisting of one- and two-mode correlations.

▶ **Theorem 2** (Informal version of Theorem 5). *Let $|\Psi\rangle$ and $|\hat{\Psi}\rangle$ be n-fermion and m-mode non-interacting-fermion states with kernel matrices K and $\hat{K}$, respectively. Then*

$$d_{\mathrm{tr}}\left(|\hat{\Psi}\rangle, |\Psi\rangle\right) \leq \sqrt{n\|\hat{K}-K\|_2},$$

where $d_{\mathrm{tr}}(\cdot,\cdot)$ is the trace distance and $\|\cdot\|_2$ is the spectral norm.

While this may seem surprising, this has been known to physicists since the 1960s and is (in some sense) the content of the Hohenberg-Kohn theorems [13] and Kohn-Sham equations [17], which form the theoretical foundations of density functional theory. These results paved the way for computational methods in quantum chemistry, earning Walter Kohn and John Pople the 1998 Nobel Prize in Chemistry. Additionally, it is well-known that Wick's theorem [33] can be used to write higher-order correlations in terms of one- and two-mode correlations.

Although this topic has received intense study for decades, we claim that our error analysis offers two improvements. First, this is the first "physics-free" proof that kernel matrices suffice to learn the state: we make no mention of creation/annihilation operators, energy potentials, Hamiltonians, or the like. We believe our proof can be understood by any mathematician or theoretical computer scientist without a physics background and perhaps even undergraduates with a linear algebra background. Second, our theorem quantitatively relates the distance between the states and the distance between the kernel matrices, which (to our knowledge) has never been done.

In Section 4 we use this theorem to show that the trace distance between $|\hat{\Psi}\rangle$ and $|\Psi\rangle$ is at most $\sqrt{2nm\gamma}$. Therefore, the trace distance will be ϵ-close if we set γ to $\epsilon^2/2nm$.

1.2 Related Work

Previous work showed how to *simulate* non-interacting fermions efficiently. In 2002, Valiant [30] introduced a class of quantum circuits called matchgate circuits and showed that they can be simulated classically in polynomial time. Soon after, Terhal and DiVincenzo [28] (see also Knill [16]) showed that evolutions of non-interacting fermions give rise to unitary matchgate circuits. (See also [4][Appendix 13] for a simpler and faster simulation algorithm.) Thus, the contribution of this paper is to complement these classical simulation results with an efficient *learnability* result.

More broadly, *quantum state tomography* is the task of constructing a classical description of a d-dimensional quantum mixed state, given copies of the state. With entangled measurements, the optimal number of copies for quantum state tomography is known to be $\Theta(d^2)$ due to Haah et al. [12] and O'Donnell and Wright [22]. With unentangled measurements, the optimal number of copies is $\Theta(d^3)$ [18, 12, 11].

Quantum state tomography can be computationally efficient in restricted settings. Montanaro [21] showed that *stabilizer states* are efficiently learnable using measurements in the Bell basis. Cramer et al. [9] showed that states approximated by *matrix product states* are efficiently learnable. Arunachalam et al. [6] showed that some classes of *phase states* are efficiently learnable. With this work, non-interacting-fermion states is an additional class of quantum states for which we know computationally efficient learning algorithms.

Different models for learning properties of mixed states ρ have been studied. For example, Aaronson [1] showed that such states are learnable under the Probably Approximately Correct (PAC) model, using training sequences of length only logarithmic in the Hilbert space dimension. Since our goal is simply to reconstruct a distribution, we have no need for the PAC framework. Aaronson also introduced *shadow tomography* [2, 7], where, given a list of known two-outcome observables and copies of an unknown state, the goal is to estimate the expectation value of each observable with respect to the unknown state to additive accuracy. Although computationally inefficient, Aaronson showed that the number of copies of the input state scales polylogarithmically with both the number of observables and the Hilbert space dimension. Soon after, Huang, Kueng, and Preskill [15] introduced *classical shadows*, a shadow tomography algorithm that is computationally efficient for certain problem instances. For example, with random Clifford measurements, the classical time cost in classical shadows is dominated by computing quantities of the form $\langle s|O|s\rangle$, where O is an observable and $|s\rangle$ is some stabilizer state, which is computationally efficient for certain observables.

Recently, there have been several results that extend the classical shadows protocol to fermionic states and circuits [34, 31, 19, 23]. In particular, after the original version of this paper appeared but before the current version, O'Gorman [23] gave an algorithm for learning non-interacting-fermion states, which is based on classical shadows. His learning

algorithm uses $O(m^7 n^2 \log(m/\delta)/\epsilon^4)$ samples and $O(m^9 n^2 \log(m/\delta)/\epsilon^4)$ time to learn a non-interacting-fermion state to trace distance at most ϵ with probability at least $1-\delta$. Our work has substantially better sample and time complexities, and makes no use of randomized measurements.

Finally, [25, Appendix C] proposes an algorithm for reconstructing a kernel matrix that involves iterating over $O(m)$ perfect matchings, just as our algorithm does, which we were unaware of until the final stages of our work. However, we note that their circuits/measurements differ from ours, and they do not provide an error analysis for their algorithm.

1.2.1 Determinantal Point Processes

For reasons having nothing to do with non-interacting fermions, problems extremely close to ours have already been studied in classical machine learning, in the field of *Determinantal Point Processes* (DPPs). A DPP is a model specified by an $m \times m$ matrix K (typically symmetric or Hermitian), such that the probabilities of various events are given by various principal minors of K, exactly as for non-interacting fermions. The connection between DPPs and fermions has been known for decades [20].

Two results in particular are directly relevant to us: Rising et al. [27] and Urschel et al. [29]. Rising et al. give an efficient algorithm for the *symmetric principal minor assignment problem*: given a list of all 2^m principal minors of an unknown $m \times m$ symmetric matrix K, reconstruct K. Their algorithm is based on constructing an m-vertex graph with an edge from i to j whenever $K_{ij} \neq 0$, and then analyzing the minimum spanning trees and chordless cycles in that graph. Rising et al., however, do not do an error analysis (they assume exact knowledge of the principal minors), and they solve the problem for real and complex *symmetric* matrices, whereas in our problem, the matrix is complex and Hermitian. This difference turns out to be surprisingly important, as the determinants of Hermitian matrices are always real, so much of the phase information vanishes – making the Hermitian case much harder.

Urschel et al. [29] further exploit connections between DPPs and graph theory to give an algorithm that recovers the entries of K, given samples from an unknown DPP. Their focus is on parameter learning (i.e., approximately recovering the entries of K), rather than learning the distribution induced by K in, say, total variation distance. They again assume that the DPP is described by a real symmetric matrix.

Our work provides insight on the Hermitian versions of these problems. Since there are many non-interacting-fermion states with the same distribution over the standard basis, there must be many kernel matrices that are consistent with the same list of principal minors. The goal for the Hermitian principal minor assignment problem is to output any such matrix.

It is also clear that the Hermitian version of Urschel et al.'s problem is *impossible*. Samples from an unknown DPP correspond to standard basis measurement outcomes, and learning the entries of a DPP corresponds to learning the kernel matrix. However, in Theorem 5, we show that learning the kernel matrix suffices to learn the entire state, which is impossible when restricted to standard basis measurements. (Even distinguishing $|+\rangle$ from $|-\rangle$ is information-theoretically impossible when given only standard basis measurements.) What one could hope for, and which we leave open, is to learn *some* kernel matrix that gives rise to a distribution close in variation distance to the observed one.

1.2.2 Errors in Previous Version

A previous version of this manuscript [5] – where we claimed to recover a non-interacting fermion distribution using only standard basis measurements – had serious errors, which we explain below.[1]

In the previous manuscript, we sought to recover the rows $v_1, \ldots, v_m \in \mathbb{C}^n$ of the $m \times n$ column-orthonormal matrix A up to isometry (see Section 1.1 for the definition of A). By estimating the two-mode correlations (i.e., the probability of finding a fermion in both mode i and mode j), one can deduce the approximate value of $|\langle v_i, v_j \rangle|$, i.e., the absolute value of the inner product, for any $i \neq j$. From that information, our goal was to recover $v_1, \ldots, v_m$ (or, more precisely, their relative configuration in n-dimensional space up to isometry).

The approach was as follows: if we knew $\langle v_i, v_j \rangle$ for all $i \neq j$, then we would get linear equations that iteratively constrained each v_i in terms of $\langle v_i, v_j \rangle$ for $j < i$, so all that would be required would be to solve those linear systems, and then show that the solution is robust with respect to small errors in our estimates of each $\langle v_i, v_j \rangle$. While it is true that the measurements only reveal $|\langle v_i, v_j \rangle|$ rather than $\langle v_i, v_j \rangle$ itself, the "phase information" in $\langle v_i, v_j \rangle$ seemed manifestly irrelevant, since it in any case depended on the irrelevant global phases of v_i and v_j themselves.

Alas, it turns out that the phase information *does* matter. As an example, suppose we only knew the following about three unit vectors $u, v, w \in \mathbb{R}^3$:

$$|\langle u, v \rangle| = |\langle u, w \rangle| = |\langle w, v \rangle| = \frac{1}{2}.$$

This is not enough to determine these vectors up to isometry! In one class of solution, all three vectors belong to the same plane, like so:

$$u = (1, 0, 0), \quad v = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0 \right), \quad w = \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0 \right).$$

In a completely different class of solution, the three vectors do not belong to the same plane, and instead look like three edges of a tetrahedron meeting at a vertex:

$$u = (1, 0, 0), \quad v = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0 \right), \quad w = \left(\frac{1}{2}, \frac{\sqrt{3}}{6}, \sqrt{\frac{2}{3}} \right).$$

Both classes of solutions are shown in Figure 1. These solutions correspond to different sign choices for $|\langle u, v \rangle|, |\langle u, w \rangle|$, and $|\langle w, v \rangle|$ – choices that collectively matter, even though each one is individually irrelevant.

It follows that, even in the special case where the vectors are all real, the two-mode correlations are not enough to determine the vectors' relative positions. And alas, the situation is even worse when, as for us, the vectors can be complex. Indeed, even for systems of 2 fermions and 4 modes, it is possible to exhibit distributions that require complex vectors. For example, let

$$A = \begin{pmatrix} \frac{1}{2} & 0 \\ \frac{1}{2\sqrt{2}} & \sqrt{\frac{2}{5}} \\ \frac{1}{2\sqrt{2}} & \sqrt{\frac{2}{5}} i \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{10}} - \frac{1}{\sqrt{10}} i \end{pmatrix}.$$

[1] These errors were previously explained in https://scottaaronson.blog/?p=5706.

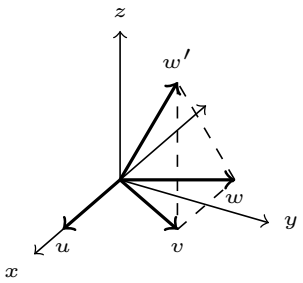


Figure 1 The vectors u, v, and w belong to the same plane, while v, w, and w' are edges of a tetrahedron that meet at a vertex. Both sets of vectors satisfy the same inner product constraints *up to phase information*, yet belong to two distinct classes of solutions.

Denote the one-mode correlations by p_i and the two-mode correlations by p_{ij}. Then by explicit calculation (e.g., compute K, then compute the appropriate principal minors of K), one can verify that

$$p_1 = \frac{1}{4}, \quad p_2 = p_3 = \frac{21}{40}, \quad p_{12} = p_{13} = p_{23} = \frac{1}{10}.^2$$

Hence, to represent the corresponding distribution with real vectors only, one must find three vectors in $\mathbb{R}^2$ with squared lengths $\frac{1}{4}$, $\frac{21}{40}$, and $\frac{21}{40}$, respectively, such that the squared area of the parallelogram created by any pair of vectors is $\frac{1}{10}$. One can verify that this is not possible.

We conclude that any possible algorithm for learning fermionic distributions from standard-basis measurements will have to solve a system of nonlinear equations (albeit, a massively overconstrained system that is guaranteed to have a solution); it will have to use three-mode correlations (i.e., statistics of triples of fermions), and indeed (one can show) in some exceptional cases four-mode correlations and above; it will sometimes have to output complex solutions even when all the input data is real (which rules out a purely linear-algebraic approach); and it will have to learn the phase information relevant to the *distribution* (rather than the entire state).

Finally, separate from the issues above, we are grateful to Andrew Zhao for identifying an error in the most recent version of this manuscript (v3 on the arXiv). In our error analysis, we assumed that the $m \times m$ matrix output by our algorithm is rank-n, when, in fact, it is full-rank. In short, we handle this issue by computing an eigendecomposition of said matrix and discarding the smallest $m - n$ eigenvalues and eigenvectors.

2 Preliminaries

Throughout this work, we use the following notation. $[n] \coloneqq \{1, \ldots, n\}$. Let $X \in \mathbb{C}^{n \times n}$ and $v \in \mathbb{C}^n$. Then $\|v\|_p \coloneqq (\sum_{i \in [n]} |v_i|^p)^{1/p}$ is the ℓ_p-norm, and $\|X\|_2 \coloneqq \sup_{\|v\|_2 = 1} \|Xv\|_2$ is the spectral norm. Let ρ and σ be two quantum mixed states. Then $d_{\mathrm{tr}}(\rho, \sigma) \coloneqq \frac{1}{2}\mathrm{tr}\left(\sqrt{(\rho - \sigma)^2}\right) = \frac{1}{2}\sum_i |\lambda_i|$ is the trace distance, where the λ_i's are eigenvalues of the error matrix $\rho - \sigma$.

[2] We do not need to look at the fourth mode to show that complex numbers are necessary.

2.1 Non-Interacting Fermions

We briefly review non-interacting fermions. For more detail, see e.g. [3, Chapters 7-10].

As mentioned above, $\Lambda_{m,n}$ is the set of all lists $S = (s_1, \ldots, s_m)$ such that $s_i \in \{0, 1\}$ and $\sum_{i\in[m]} s_i = n$. One can easily verify that the total number of configurations $|\Lambda_{m,n}| = \binom{m}{n}$. Define $M \coloneqq \binom{m}{n}$. The set $\{|S\rangle\}_{S\in\Lambda_{m,n}}$ is a basis for systems of n non-interacting fermions in m modes, which we refer to as the standard basis (also called the Fock basis or occupation number basis). Hence, a non-interacting-fermion state is a unit vector in an M-dimensional complex Hilbert space with the form

$$|\Psi\rangle = \sum_{S\in\Lambda_{m,n}} \alpha_S |S\rangle, \quad \text{where} \sum_{S\in\Lambda_{m,n}} |\alpha_S|^2 = 1,$$

and, upon measurement, we observe an element $S \in \Lambda_{m,n}$ with probability

$$\mathbf{Pr}[S] = |\langle\Psi|S\rangle|^2 = |\alpha_S|^2.$$

The transformations on non-interacting-fermion states can be described by $m \times m$ unitary matrices, which can always be constructed with $O(m^2)$ elementary operations called *beamsplitters* and *phaseshifters* (see [26] for a proof of this statement). A beamsplitter acts on two modes and has the form

$$\begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & \cos\theta & -\sin\theta & & & \\ & & \sin\theta & \cos\theta & & & \\ & & & & \ddots & & \\ & & & & & 1 & \end{pmatrix},$$

while a phaseshifter applies a complex phase to a single mode and has the form

$$\begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & e^{i\theta} & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}.$$

An $m \times m$ unitary U describes how a single fermion would be scattered from one mode to the others. This induces an $M \times M$ unitary transformation $\varphi(U)$ on the Hilbert space of n-fermion states, where φ is the homomorphism that lifts $m \times m$ unitary transformations to n-fermion unitary transformations. For an $m \times m$ unitary matrix U, one way to define φ is

$$\langle S|\,\varphi(U)\,|T\rangle = \det(U_{S,T}), \tag{1}$$

for all $S = (s_1, \ldots, s_m), T = (t_1, \ldots, t_m) \in \Lambda_{m,n}$, where $U_{S,T}$ is the $n \times n$ submatrix obtained by removing row i of U if $s_i = 0$ and removing column j of U if $t_j = 0$. One can also view $\varphi(U)$ as a unitary transformation on m-qubit states. In this case, $\varphi(U)$ is a Hamming-weight-preserving matchgate unitary [30]. This follows directly from Terhal and DiVincenzo's work [28], which shows the equivalence between evolutions of non-interacting fermions and matchgate circuits.

Intuitively, the reason the determinant arises in Equation (1) is that $\langle S|\,\varphi(U)\,|T\rangle$ is the sum over the $n!$ permutations that take n fermions in configuration S to configuration T, each permutation contributing to the overall amplitude. When a permutation from S to T is odd, its contribution to the overall amplitude has a phase factor of -1, while when the permutation is even the phase factor is 1. This is the antisymmetry property of fermions: swapping two fermions picks up a -1 phase in the amplitude.[3]

2.2 Problem Setup

We use the following notation: $|1_n\rangle \coloneqq |1,\ldots,1,0,\ldots,0\rangle$ is the standard initial state (where n fermions occupy the first n modes), $U \in \mathbb{C}^{m\times m}$ is the unitary that prepares the unknown non-interacting-fermion state (i.e., $|\Psi\rangle = \varphi(U)\,|1_n\rangle$), and $A \in \mathbb{C}^{m\times n}$ is the $m \times n$ column-orthonormal matrix corresponding to the first n columns of U. Define $K \coloneqq AA^\dagger$.

For each element $S = (s_1,\ldots,s_m) \in \Lambda_{m,n}$, let A_S be the $n \times n$ submatrix obtained by removing row i of A if $s_i = 0$, and let K_S be the $n \times n$ submatrix obtained by removing row and column i of K if $s_i = 0$. Then from Equation (1), it follows that

$$|\Psi\rangle = \sum_{S\in\Lambda_{m,n}} \det(A_S)\,|S\rangle\,,$$

and $\mathcal{D}_K$, the probability distribution over $S \in \Lambda_{m,n}$ obtained by measuring the state $|\Psi\rangle = \varphi(U)\,|1_n\rangle$ in the standard basis, is given by

$$|\langle 1_n|\,\varphi(U)\,|S\rangle|^2 = |\det(A_S)|^2 = \det(K_S),$$

where the last equality uses the fact that, for any square matrices X and Y, $\det(X)^* = \det(X^\dagger)$ and $\det(X)\det(Y) = \det(XY)$. Further, for any list $S = (s_1,\ldots,s_m)$ where $\sum_i s_i = k < n$, the marginal probability that those k modes are occupied is $\det(K_S)$.

It is easy to verify that K is Hermitian, positive semi-definite, and a projector ($K^2 = K$); and that $\mathrm{tr}(K) = n$. Additionally, observe that the (i,j) entry is the inner product between the ith and jth rows of A (i.e., K is a *Gram matrix*). Finally, note that A is a *highly non-unique* description of $|\Psi\rangle$ and K. Let R be any $n \times n$ unitary matrix. Then A and AR describe the same state:

$$|\Psi\rangle = \sum_{S\in\Lambda_{m,n}} \det(A_S R)\,|S\rangle = \sum_{S\in\Lambda_{m,n}} \det(A_S)\det(R)\,|S\rangle = \sum_{S\in\Lambda_{m,n}} \det(A_S)\,|S\rangle\,,$$

where, in the last equality, we use the fact that the determinant of a unitary matrix is a complex unit, which only adds an irrelevant global phase. Meanwhile, the kernel matrix of $|\Psi\rangle$ is unchanged: $K = (AR)(AR)^\dagger = AA^\dagger$.

Applying a unitary V maps $|\Psi\rangle = \varphi(U)\,|1_n\rangle$ to $\varphi(VU)\,|1_n\rangle$. It is easy to check that the probability that we observe S upon measuring $\varphi(VU)\,|1_n\rangle$ in the standard basis is

$$|\langle 1_n|\,\varphi(VU)\,|S\rangle|^2 = \det((VKV^\dagger)_S),$$

and in general, applying V has the following effect on the matrix K: $K \mapsto VKV^\dagger$.

Given copies of $|\Psi\rangle$ and the ability to apply beamsplitter networks before measurement, our goal is to output a matrix $\hat{A} \in \mathbb{C}^{m\times n}$ such that $|\hat{\Psi}\rangle = \sum_S \det(\hat{A}_S)$ is ϵ-close to $|\Psi\rangle$ in trace distance.

[3] Meanwhile, bosons are symmetric under transpositions, so no minus signs show up. This is precisely why permanents arise when computing amplitudes for non-interacting bosons, while determinants show up for non-interacting fermions.

3 Learning Algorithm

In this section, we present our learning algorithm, which is given copies of an unknown non-interacting-fermion state, and outputs an $m \times n$ matrix $\hat{A}$ such that the corresponding state $|\hat{\Psi}\rangle = \sum_S \det(\hat{A}_S)\,|S\rangle$ is close to the original in trace distance. The algorithm has three phases: first, we learn the diagonal entries of the kernel matrix K with standard basis measurements, then we learn the off-diagonal entries by measuring the unknown state in $O(m)$ different bases. Finally, we decompose the reconstructed kernel matrix into the $m \times n$ output matrix $\hat{A}$.

Algorithm 1 Efficient tomography of non-interacting-fermion states.

Input: Black-box access to copies of $|\Psi\rangle$ (the input state), $\gamma \in (0,1)$ (accuracy parameter), and $\delta \in (0,1)$ (confidence parameter).
Output: an $m \times n$ matrix $\hat{A} \in \mathbb{C}^{m\times n}$.

1: Measure $O(\log(1/\delta)/\gamma^2)$ copies of $|\Psi\rangle$ in the standard basis and estimate the one-mode correlations for all m modes. Set $\hat{k}_{ii}$ to the empirical estimate that mode i is occupied.
2: Choose $O(m)$ different perfect matchings among the m modes, which together cover all possible (i,j) pairs.
3: **for** each perfect matching **do**
4: Apply the beamsplitter

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

to each pair of modes in the perfect matching and measure in the standard basis. Repeat this $O(\log(1/\delta)/\gamma^2)$ times, and use the measurement data to estimate the one-mode correlations.
5: Repeat the previous step with the beamsplitter

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & i\\ 1 & -i\end{pmatrix}.$$

6: For each pair (i,j) in the perfect matching, the beamsplitter network in step 4 maps the one-mode correlation k_{ii} to $k'_{ii} := \frac{1}{2}(k_{ii} + k_{jj} + 2\mathrm{Re}(k_{ij}))$. Denote the ith estimate obtained in step 4 by $\hat{k}'_{ii}$. Let $\mathrm{Re}(\hat{k}_{ij})$ be the estimate of $\mathrm{Re}(k_{ij})$ obtained by solving the following equation:

$$\hat{k}'_{ii} = \frac{1}{2}(\hat{k}_{ii} + \hat{k}_{jj} + 2\mathrm{Re}(\hat{k}_{ij})),$$

where $\hat{k}_{ii}$ and $\hat{k}_{jj}$ are the estimates from step 1.
7: Repeat the previous step with the estimates from step 5 to obtain estimate $\mathrm{Im}(\hat{k}_{ij})$ for each pair (i,j). (Note that the beamsplitter network in step 5 maps k_{ii} to $\frac{1}{2}(k_{ii} + k_{jj} + 2\mathrm{Im}(k_{ij}))$, for each pair (i,j) in the perfect matching.)
8: For each pair (i,j) in the perfect matching, set $\hat{k}_{ij} = \mathrm{Re}(\hat{k}_{ij}) + i\mathrm{Im}(\hat{k}_{ij})$ and $\hat{k}_{ji} = \hat{k}^*_{ij}$.
9: Let $\hat{K} = (\hat{k}_{ij}) \in \mathbb{C}^{m\times m}$, and let $Q\Lambda Q^\dagger$ be an eigendecomposition of $\hat{K}$. Set $\hat{A}$ to be the $m \times n$ matrix corresponding to the first n columns of Q.
10: **return** $\hat{A}$.

For each measurement basis, we estimate $O(m)$ entries of K to within γ in magnitude, which can be accomplished with $O(\log(1/\delta)/\gamma^2)$ copies [8, Theorem 9] and $O(m\log(1/\delta)/\gamma^2)$ time. To estimate the off-diagonal entries to the target accuracy, an additional constant factor appears in the sample complexity, which is absorbed into the $O(\log(1/\delta)/\gamma^2)$. We use $O(m)$ measurement bases in total, so the overall copy and time complexities are $O(m\log(1/\delta)/\gamma^2)$ and $O(m^2\log(1/\delta)/\gamma^2)$ respectively. Our algorithm then computes an eigendecomposition of an $m \times m$ matrix, which requires $O(m^3)$ time, but computing this decomposition is not the bottleneck in our algorithm.

We note that $\hat{K}$ is clearly Hermitian by construction, so the eigendecomposition of $\hat{K}$ exists. Following convention, it is assumed that the eigenvalues are ordered from largest to smallest (i.e., the first column of Q corresponds to the largest eigenvalue of $\hat{K}$, and so on). The output of our algorithm $\hat{A}$ is column-orthonormal because the columns of the unitary Q are orthonormal. Therefore, $\hat{A}$ describes a non-interacting-fermion state $|\hat{\Psi}\rangle = \sum_S \det(\hat{A}_S)\,|S\rangle$. In the next section, we show that if $\gamma = \frac{\epsilon^2}{2nm}$, then the trace distance between $|\Psi\rangle$ and $|\hat{\Psi}\rangle$ is at most ϵ. Hence, for the two states to be ϵ-close in trace distance, $O(m^3n^2\log(1/\delta)/\epsilon^4)$ copies and $O(m^4n^2\log(1/\delta)/\epsilon^4)$ time suffice.

4 Error Analysis

In this section, we show that the trace distance between $|\Psi\rangle$ and $|\hat{\Psi}\rangle$ is at most $\sqrt{2nm\gamma}$. Therefore, if $\gamma = \frac{\epsilon^2}{2nm}$, then the trace distance between $|\Psi\rangle$ and $|\hat{\Psi}\rangle$ is at most ϵ.

The error analysis is presented in two parts. First, we show that the trace distance between any two non-interacting-fermion states is bounded above the spectral distance between their kernel matrices. Then we show that the output of our algorithm is close to the original state in trace distance.

4.1 The Kernel Matrix Suffices

We prove that the trace distance between two non-interacting-fermion states is upper bounded by the spectral difference between their kernel matrices. To show this, we need the following two lemmas.

▶ **Lemma 3.** *Let $a_1, a_2, \ldots, a_n \in [0,1]$. Then*

$$1 - \prod_i a_i \leq n \max_i 1 - a_i.$$

Proof. For any $x, y \in [0,1]$,

$$1 - xy = 1 - x + x - xy = (1-x) + x(1-y) \leq (1-x) + (1-y).$$

We can inductively apply this to get

$$1 - \prod_i a_i \leq \sum_i 1 - a_i \leq n \max_i 1 - a_i.$$ ◀

▶ **Lemma 4.** *Let $A, \hat{A} \in \mathbb{C}^{m\times n}$ $(m \geq n)$ be $m \times n$ matrices, and let A be column-orthonormal. Define $K \coloneqq AA^\dagger$ and $\hat{K} \coloneqq \hat{A}\hat{A}^\dagger$. Let $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ where σ_i are the singular values of $\hat{A}^\dagger A$. Then*

$$\|I - \Sigma^2\|_2 \leq \|\hat{K} - K\|_2.$$

Proof. Since A is column-orthonormal, $A^\dagger A = I$ (I is the identity matrix). Let $Q\Sigma V^\dagger$ be a singular value decomposition of $\hat{A}^\dagger A$. First, note that

$$V\Sigma^2 V^\dagger = (Q\Sigma V^\dagger)^\dagger Q\Sigma V^\dagger = (\hat{A}^\dagger A)^\dagger \hat{A}^\dagger A = A^\dagger \hat{A}\hat{A}^\dagger A.$$

Therefore,

$$\|I - \Sigma^2\|_2 = \|I - A^\dagger \hat{A}\hat{A}^\dagger A\|_2 = \|A^\dagger A - A^\dagger \hat{A}\hat{A}^\dagger A\|_2 = \|A^\dagger (A - \hat{A}\hat{A}^\dagger A)\|_2 \leq \|A - \hat{A}\hat{A}^\dagger A\|_2,$$

where the first step uses the fact that the spectral norm is unitarily invariant and the final step follows from the submultiplicativity of matrix norms and that $\|A^\dagger\|_2 = \|A\|_2 = 1$ since A is column-orthonormal. Finally,

$$\|A - \hat{A}\hat{A}^\dagger A\|_2 = \|AA^\dagger A - \hat{A}\hat{A}^\dagger A\|_2 \leq \|AA^\dagger - \hat{A}\hat{A}^\dagger\|_2 = \|K - \hat{K}\|_2. \quad \blacktriangleleft$$

We are now ready to show that if two kernel matrices are close, then the corresponding states will also be close.

▶ **Theorem 5.** *Let $|\Psi\rangle$ and $|\hat{\Psi}\rangle$ be non-interacting-fermion states of n fermions and m modes described by the $m \times n$ column-orthonormal matrices $A, \hat{A} \in \mathbb{C}^{m\times n}$, respectively. Define $K := AA^\dagger$ and $\hat{K} := \hat{A}\hat{A}^\dagger$. Then*

$$d_{\mathrm{tr}}\left(|\hat{\Psi}\rangle, |\Psi\rangle\right) \leq \sqrt{n\|\hat{K} - K\|_2}.$$

Proof. Recall that $|\Psi\rangle$ and $|\hat{\Psi}\rangle$ can be written as

$$|\Psi\rangle = \sum_{S\in\Lambda_{m,n}} \det(A_S)\,|S\rangle \qquad \text{and} \qquad |\hat{\Psi}\rangle = \sum_{S\in\Lambda_{m,n}} \det(\hat{A}_S)\,|S\rangle$$

for the column-orthonormal matrices $A, \hat{A} \in \mathbb{C}^{m\times n}$. Then

$$\begin{aligned} d_{\mathrm{tr}}\left(|\hat{\Psi}\rangle, |\Psi\rangle\right) &= \sqrt{1 - |\langle\hat{\Psi}|\Psi\rangle|^2} \\ &= \sqrt{1 - \Big|\sum_{S\in\Lambda_{m,n}} \det(\hat{A}_S)^* \det(A_S)\Big|^2} \\ &= \sqrt{1 - \Big|\sum_{S\in\Lambda_{m,n}} \det(\hat{A}_S^\dagger) \det(A_S)\Big|^2} \\ &= \sqrt{1 - |\det(\hat{A}^\dagger A)|^2}, \end{aligned}$$

where the second-to-last step follows because, for any square matrix X, $\det(X^T) = \det(X)$ and $\det(X)^* = \det(X^*)$; and the final step follows from the Cauchy-Binet formula.

Let $Q\Sigma V^\dagger$ be a singular value decomposition of $\hat{A}^\dagger A$, where $\sigma_1, \ldots, \sigma_n$ are the singular values on the diagonal of Σ. Then

$$|\det(\hat{A}^\dagger A)|^2 = |\det(Q\Sigma V^\dagger)|^2 = |\det(Q)\det(\Sigma)\det(V^\dagger)|^2 = \det(\Sigma)^2 = \prod_i \sigma_i^2.$$

Note that, for all $i \in [n]$, $\sigma_i \leq \sigma_i(\hat{A}^\dagger)\|A\|_2 = 1$ [14, Chapter 3], where $\sigma_i(\hat{A}^\dagger)$ is the ith singular value of $\hat{A}^\dagger$. Plugging this into our bound on the trace distance, we get

$$\begin{aligned} d_{\mathrm{tr}}\left(|\hat{\Psi}\rangle, |\Psi\rangle\right) &= \sqrt{1 - |\det(\hat{A}^\dagger A)|^2} \\ &= \sqrt{1 - \prod_i \sigma_i^2} \\ &\leq \sqrt{n\left(\max_i 1 - \sigma_i^2\right)} && \text{(By Lemma 3).} \\ &= \sqrt{n\|I - \Sigma^2\|_2} \\ &\leq \sqrt{n\|\hat{K} - K\|_2} && \text{(By Lemma 4).} \end{aligned}$$

◀

4.2 Completing the Analysis

We prove that the fermionic state output by Section 3 is close to the input state in trace distance. To do so, we make use of Weyl's inequality, which implies that the spectrum of a Hermitian matrix is stable under small perturbations.

▶ **Theorem 6** (A Consequence of Weyl's Inequality [32]). *Let $M, N, R \in \mathbb{C}^{n\times n}$ be $n \times n$ Hermitian matrices such that $M = N + R$. Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of M, and let $\mu_1, \ldots, \mu_n$ be the eigenvalues of N. Then, for all $i \in [n]$,*

$$|\lambda_i - \mu_i| \leq \|R\|_2.$$

We are now ready to prove that Section 3 successfully learns a non-interacting-fermion state.

▶ **Theorem 7.** *Let $\hat{A}$ be the output of Section 3 when given $|\Psi\rangle$ as input, and let $|\hat{\Psi}\rangle$ be the non-interacting-fermion state described by $\hat{A}$. Then*

$$d_{\mathrm{tr}}\left(|\hat{\Psi}\rangle, |\Psi\rangle\right) \leq \sqrt{2nm\gamma}.$$

Proof. It is convenient to recall the last steps of Section 3:

In our algorithm, once the quantum measurements are complete, we have a matrix $\hat{K}$ whose entries are within γ in magnitude of K. We then compute the eigendecomposition $Q\Lambda Q^\dagger$ of $\hat{K}$, where the first column of Q is the eigenvector corresponding to the largest eigenvalue of $\hat{K}$ and so on. Finally, we set $\hat{A}$ to be the $m \times n$ matrix corresponding to the first n columns of Q, and output $\hat{A}$. Therefore, the kernel matrix of $|\hat{\Psi}\rangle$ is $\hat{A}\hat{A}^\dagger$, and, by Theorem 5, the trace distance between $|\Psi\rangle$ and $|\hat{\Psi}\rangle$ is bounded above by $\sqrt{n\|\hat{A}\hat{A}^\dagger - K\|_2}$, where $K = AA^\dagger$ is the kernel matrix corresponding to the input state $|\Psi\rangle$. To complete the proof, we must bound $\|\hat{A}\hat{A}^\dagger - K\|_2$. To that end, by the triangle inequality,

$$\|\hat{A}\hat{A}^\dagger - K\|_2 \leq \|\hat{A}\hat{A}^\dagger - \hat{K}\|_2 + \|\hat{K} - K\|_2. \tag{2}$$

Observe that $\hat{K} = K + E$, where E is a perturbation of K whose entries have magnitude at most γ. Therefore, $\|\hat{K} - K\|_2 = \|K + E - K\|_2 = \|E\|_2$. The error matrix E is Hermitian because K and $\hat{K}$ are Hermitian, and Hermitian matrices are closed under addition/subtraction. Therefore, since $\hat{K}$, K, and E are all Hermitian, we can use Theorem 6 to upper bound the absolute difference between the eigenvalues of K and $\hat{K}$. In particular, the absolute difference between the ith eigenvalues of K and $\hat{K}$ is at most $\|E\|_2$, for all $i \in [m]$.

Let $\mathbb{1}_n = \mathrm{diag}(1, \ldots, 1, 0, \ldots, 0) \in \mathbb{R}^{m \times m}$ be the diagonal matrix whose first n diagonal entries are 1 and the rest 0. Observe that $\hat{A}\hat{A}^\dagger = Q\mathbb{1}_n Q^\dagger$, since $\hat{A}$ is the first n columns of Q, and recall that $\hat{K} = Q\Lambda Q^\dagger$. Therefore,

$$\|\hat{A}\hat{A}^\dagger - \hat{K}\|_2 = \|Q\mathbb{1}_n Q^\dagger - Q\Lambda Q^\dagger\|_2 = \|\mathbb{1}_n - \Lambda\|_2,$$

where, in the last equality, we use the fact that the spectral norm is unitarily invariant. Note that Λ contains the eigenvalues of $\hat{K}$, and $\mathbb{1}_n$ contains the eigenvalues of K (since K is a trace-n, rank-n projector). Therefore, $\|\mathbb{1}_n - \Lambda\|_2$ is the maximum absolute difference between the eigenvalues of K and $\hat{K}$, which is at most $\|E\|_2$, as we argued in the previous paragraph.

The trivial bound on the spectral norm of E is the Frobenius norm of E, which is maximum when all entries of E have magnitude γ. Specifically, $\|E\|_2 \leq \|E\|_F \leq m\gamma$. Note $\|E\|_2 = \|E\|_F$ when the entries of E are all γ, so we cannot hope for a tighter bound on $\|E\|_2$.

Plugging this into Equation (2),

$$\begin{aligned}\|\hat{A}\hat{A}^\dagger - K\|_2 &\leq \|\hat{A}\hat{A}^\dagger - \hat{K}\|_2 + \|\hat{K} - K\|_2 \\ &\leq \|E\|_2 + \|E\|_2 \\ &\leq 2m\gamma.\end{aligned}$$

◀

5 Connections to Quantum State Tomography

Although physically different, our problem is closely related to the quantum state tomography problem. In quantum state tomography, we want to recover an unknown Hermitian matrix, namely a d-dimensional mixed state $\rho \in \mathbb{C}^{d \times d}$, and applying a quantum circuit V to ρ maps ρ to $V\rho V^\dagger$. In our problem, applying a unitary V to $|\Psi\rangle$ maps K to $VKV^\dagger$, where K is an unknown Hermitian matrix, and our algorithm is able to recover the entries of K to within γ in magnitude. Therefore, our algorithm can also be viewed as a state tomography algorithm: measure copies of ρ in the $O(d)$ measurement bases obtained by choosing perfect matchings that cover all (i, j) pairs and output the resulting matrix $\hat{\rho}$ (skipping the last few steps of the algorithm that involve computing an eigendecomposition). As before, the algorithm requires $O(d \log(1/\delta)/\gamma^2)$ copies and $O(d^2 \log(1/\delta)/\gamma^2)$ time.

The error analysis is slightly different than for learning a fermionic state. For the state tomography problem, we want the output matrix $\hat{\rho}$ to be close to ρ in trace distance, which is proportional to $\|\hat{\rho} - \rho\|_1$, whereas, for fermionic tomography, our trace distance upper bound is proportional to $\|\hat{K} - K\|_2$ (see Theorem 5). Hence, to analyze the performance of our algorithm for state tomography, we must upper bound $\|\hat{\rho} - \rho\|_1$.

Recall that the error matrix $E = \hat{\rho} - \rho$ is Hermitian and has entries with magnitude at most γ. For $i \in [m]$, let λ_i denote the eigenvalues of E. Then

$$d_{\mathrm{tr}}(\hat{\rho}, \rho) = \frac{1}{2}\|\hat{\rho} - \rho\|_1 = \frac{1}{2}\|E\|_1 = \frac{1}{2}\sum_i |\lambda_i| \leq \frac{\sqrt{d}}{2}\sqrt{\sum_i |\lambda_i|^2} \leq \frac{1}{2}d^{3/2}\gamma.$$

The first inequality follows from the fact that the arithmetic mean is bounded above by the quadratic mean, and the second inequality follows from the fact that $\sqrt{\sum_i |\lambda_i|^2} = \|E\|_F \leq d\gamma$. For $d_{\mathrm{tr}}(\hat{\rho}, \rho) \leq \epsilon$, we must set $\gamma = 2d^{-3/2}\epsilon$. The resulting copy complexity is $O(d^4 \log(1/\delta)/\epsilon^2)$ and time complexity is $O(d^5 \log(1/\delta)/\epsilon^2)$. Note that the optimal copy complexity for quantum state tomography *with unentangled measurements* is $\Theta(d^3/\epsilon^2)$ [18, 12, 11], compared to $\Theta(d^2/\epsilon^2)$ with entangled measurements [22, 12].

Finally, we show that our upper bound on $\|\hat{\rho} - \rho\|_1$ is tight. Let F be the $d \times d$ Fourier transform whose (i, j) entry is $\exp(2\pi ij/d)/\sqrt{d}$. Then F scaled by a factor of $\sqrt{d}\gamma$ is a valid error matrix whose 1-norm is equal to $d^{3/2}\gamma$. Indeed, any $d \times d$ unitary matrix scaled by a factor of $\gamma\sqrt{d}$ will match our upper bound.

6 Open Problems

Perhaps the most interesting open problem is to give an algorithm to learn fermionic distributions using only standard basis measurements. Specifically, the following problems remain open:

1. **Learn real DPPs in variation distance.** Given sample access to a distribution induced by an $m \times m$ symmetric matrix $K \in \mathbb{R}^{m \times m}$, output an $m \times m$ matrix $\hat{K}$ such that the induced distribution is close in variation distance. (See Section 1.2.1 for detail on DPPs.)
2. **Hermitian principal minor assignment problem.** Given a list of all 2^m principal minors of an unknown Hermitian matrix $K \in \mathbb{C}^{m \times m}$, reconstruct any Hermitian matrix that is consistent with that list.
3. **Learn non-interacting fermion distributions with standard basis measurements.** Given sample access to a non-interacting fermion distribution, efficiently learn the distribution in total variation distance.

The third problem is in some sense a combination of the first and second. To solve the first problem, we believe that the connections between DPPs and graph theory used in Rising et al. [27] and Urschel et al. [29] should be enough to develop an efficient algorithm. As discussed in Section 1.2.1, our work shows that there are many kernel matrices that have the same principal minors (indeed, any set non-interacting states that have the same distribution over the standard basis will give rise to a set of kernel matrices that are consistent with the same list of principal minors). The goal for the second and third problems is to output any one of the valid matrices.

For the second problem, however, the following example shows that some combinatorial information about the kernel matrix K, above and beyond the obvious complex conjugation ambiguities, is *not* determined even in principle by K's principal minors. Consider the following 4×4 Hermitian matrix:

$$K = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & a & b^* \\ 1 & a^* & 1 & c \\ 1 & b & c^* & 1 \end{pmatrix}.$$

Suppose we have learned, by looking at the 2×2 and 3×3 principal minors, that

$$a = e^{ix}, \quad b = e^{iy}, \quad \text{and} \quad c = e^{iz},$$

where $|x| = |y| = |z| = w$ for some w that is known. That is, we have determined a, b, and c up to complex conjugation, and up to complex conjugation they are all equal. By looking at the bottom-most 3×3 principal minor, we can learn $\mathrm{Re}(abc)$ and hence $|x + y + z|$. Suppose that this is also w. From the 4×4 minor, combined with the 2×2 and 3×3 minors, we get one additional piece of information, namely:

$$\mathrm{Re}(ab) + \mathrm{Re}(ac) + \mathrm{Re}(bc).$$

Suppose that, as expected, this is $2 + \cos(2w)$. Then even though we have extracted all information from the principal minors, there are *still* three essentially different solutions possible. Namely,

(1) $x = y = w$ and $z = -w$, (2) $x = z = w$ and $y = -w$, (3) $y = z = w$ and $x = -w$.

Of course, for fermionic distributions, the matrix K must be Hermitian, positive semi-definite, and a projector, and $\mathrm{rank}(K) = n$. The example above is neither positive semi-definite nor a projector. However, we conjecture that this example can be embedded into a larger matrix that does satisfy these constraints.

Other directions for future work include improving the copy and time complexities of our algorithm, or giving conditional or unconditional lower bounds. Currently, the best lower bound we know is that $\Omega(m/\log m)$ measurements are needed, just from an information-theoretic argument (each measurement gives at most $n \log m$ bits of information and the state is characterized by $2nm$ real parameters). Similarly, $\Omega(mn)$ time is needed just to write down the output.

Since $\mathrm{rank}(K) = n$, it should be possible to reduce the number of measurement bases from $O(m)$ to $O(n)$ (perhaps with the low-rank matrix recovery techniques used in [18]). Doing so would yield an immediate improvement in the copy and time complexities of our algorithm.

Also, just as our algorithm can be adapted to quantum state tomography, it is possible that the converse holds. Can quantum state tomography algorithms (in the entangled or unentangled measurement setting) be adapted to non-interacting-fermion state tomography? Also, do quantum state tomography lower bounds imply lower bounds for learning non-interacting-fermion state? In analogy with quantum state tomography, perhaps $\Theta(mn)$ copies are optimal to learn non-interacting-fermion states with entangled measurements and $\Theta(mn^2)$ copies are optimal with unentangled measurements.

It would be interesting to generalize our algorithm – for example, to superpositions over different numbers of fermions, or fermionic circuits that take inputs – and to find other classes of quantum states that admit efficient learning algorithms (for example, perhaps low-entanglement states or the outputs of small-depth circuits or low-stabilizer-complexity states [10]). We remark that [24] gives evidence that generalizing our algorithm to superpositions over different numbers of fermions may not be possible unless one limits the number of terms in the superposition. On the other hand, [24, Theorem 8] might be useful in developing learning algorithms for matchgate circuits.

Finally, what can be said about learning non-interacting *boson* states? The goal would be to reconstruct an $m \times n$ column-orthonormal matrix A given copies of a non-interacting boson state. However, the boson case is even trickier than the fermion case. In particular, boson statistics no longer depend only on the inner products between the rows of A, the way fermion statistics do. Indeed, even if we collected enough information to reconstruct a bosonic state, it seems that any algorithm would have to solve a quite complicated set of nonlinear equations.

References

1 Scott Aaronson. The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3089–3114, 2007. doi:10.1098/rspa.2007.0113.

2 Scott Aaronson. Shadow Tomography of Quantum States. *SIAM Journal on Computing*, 49(5):STOC18–368–STOC18–394, 2020. doi:10.1145/3188745.3188802.

3 Scott Aaronson. Introduction to Quantum Information Science II Lecture Notes, 2022. scottaaronson.com/qisii.pdf.

4 Scott Aaronson and Alex Arkhipov. BosonSampling is Far From Uniform. *Quantum Information and Computation*, 14(15-16):1383–1423, 2014.

5 Scott Aaronson and Sabee Grewal. Efficient Learning of Non-Interacting Fermion Distributions. *arXiv preprint arXiv:2102.10458v2*, 2021.

6 Srinivasan Arunachalam, Sergey Bravyi, Arkopal Dutt, and Theodore J. Yoder. Optimal algorithms for learning quantum phase states, 2022. doi:10.48550/arxiv.2208.07851.

7 Costin Bădescu and Ryan O'Donnell. Improved Quantum Data Analysis. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 1398–1411, 2021. doi:10.1145/3406325.3451109.

8 Clément L. Canonne. A short note on learning discrete distributions. *arXiv preprint arXiv:2002.11457*, 2020. doi:10.48550/arXiv.2002.11457.

9 Marcus Cramer, Martin B. Plenio, Steven T. Flammia, Rolando Somma, David Gross, Stephen D. Bartlett, Olivier Landon-Cardinal, David Poulin, and Yi-Kai Liu. Efficient quantum state tomography. *Nature communications*, 1(1):1–7, 2010. doi:10.1038/ncomms1147.

10 Sabee Grewal, Vishnu Iyer, William Kretschmer, and Daniel Liang. Low-Stabilizer-Complexity Quantum States Are Not Pseudorandom. *arXiv preprint arXiv:2209.14530*, 2022. doi:10.48550/arXiv.2209.14530.

11 Madalin Guţă, Jonas Kahn, Richard Kueng, and Joel A. Tropp. Fast state tomography with optimal error bounds. *Journal of Physics A: Mathematical and Theoretical*, 53(20):204001, 2020. doi:10.1088/1751-8121/ab8111.

12 Jeongwan Haah, Aram W. Harrow, Zhengfeng Ji, Xiaodi Wu, and Nengkun Yu. Sample-Optimal Tomography of Quantum States. *IEEE Transactions on Information Theory*, 63(9):5628–5641, 2017. doi:10.1109/TIT.2017.2719044.

13 Pierre Hohenberg and Walter Kohn. Inhomogeneous Electron Gas. *Physical Review*, 136(3B):B864, 1964. doi:10.1103/PhysRev.136.B864.

14 Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1994. doi:10.1017/CBO9780511840371.

15 Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020. doi:10.1038/s41567-020-0932-7.

16 Emanuel Knill. Fermionic Linear Optics and Matchgates. *arXiv preprint quant-ph/0108033*, 2001. doi:10.48550/arXiv.quant-ph/0108033.

17 Walter Kohn and Lu Jeu Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Physical Review*, 140(4A):A1133, 1965. doi:10.1103/PhysRev.140.A1133.

18 Richard Kueng, Holger Rauhut, and Ulrich Terstiege. Low Rank Matrix Recovery From Rank One Measurements. *Applied and Computational Harmonic Analysis*, 42(1):88–116, 2017. doi:10.1016/j.acha.2015.07.007.

19 Guang Hao Low. Classical shadows of fermions with particle number symmetry, 2022. doi:10.48550/arxiv.2208.08964.

20 Odile Macchi. The Coincidence Approach to Stochastic Point Processes. *Advances in Applied Probability*, 7(1):83–122, 1975. doi:10.2307/1425855.

21 Ashley Montanaro. Learning stabilizer states by Bell sampling. *arXiv:1707.04012*, 2017. doi:10.48550/arXiv.1707.04012.

22 Ryan O'Donnell and John Wright. Efficient Quantum Tomography. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 899–912, 2016. doi:10.1145/2897518.2897544.

23 Bryan O'Gorman. Fermionic tomography and learning, 2022. doi:10.48550/arxiv.2207.14787.

24 Michał Oszmaniec, Ninnat Dangniam, Mauro E.S. Morales, and Zoltán Zimborás. Fermion Sampling: A Robust Quantum Computational Advantage Scheme Using Fermionic Linear Optics and Magic Input States. *PRX Quantum*, 3:020328, 2022. doi:10.1103/PRXQuantum.3.020328.

25 Google AI Quantum and Collaborators. Hartree-Fock on a superconducting qubit quantum computer. *Science*, 369(6507):1084–1089, 2020. doi:10.1126/science.abb9811.

26 Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73:58–61, 1994. doi:10.1103/PhysRevLett.73.58.

27 Justin Rising, Alex Kulesza, and Ben Taskar. An Efficient Algorithm for the Symmetric Principal Minor Assignment Problem. *Linear Algebra and its Applications*, 473:126–144, 2015. doi:10.1016/j.laa.2014.04.019.

28 Barbara M. Terhal and David P. DiVincenzo. Classical simulation of noninteracting-fermion quantum circuits. *Physical Review A*, 65(3):032325, 2002. doi:10.1103/PhysRevA.65.032325.

29 John Urschel, Victor-Emmanuel Brunel, Ankur Moitra, and Philippe Rigollet. Learning Determinantal Point Processes with Moments and Cycles. In *International Conference on Machine Learning*, pages 3511–3520. PMLR, 2017.

30 Leslie G. Valiant. Quantum Circuits That Can Be Simulated Classically in Polynomial Time. *SIAM Journal on Computing*, 31(4):1229–1254, 2002. doi:10.1137/S0097539700377025.

31 Kianna Wan, William J. Huggins, Joonho Lee, and Ryan Babbush. Matchgate Shadows for Fermionic Quantum Simulation, 2022. doi:10.48550/arxiv.2207.13723.

32 Hermann Weyl. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912. doi:10.1007/BF01456804.

33 Gian-Carlo Wick. The Evaluation of the Collision Matrix. *Physical Review*, 80(2):268, 1950. doi:10.1103/PhysRev.80.268.

34 Andrew Zhao, Nicholas C. Rubin, and Akimasa Miyake. Fermionic Partial Tomography via Classical Shadows. *Phys. Rev. Lett.*, 127:110504, 2021. doi:10.1103/PhysRevLett.127.110504.

Quantum Policy Gradient Algorithms

Sofiene Jerbi
Institute for Theoretical Physics, Universität Innsbruck, Austria

Arjan Cornelissen
QuSoft and University of Amsterdam, The Netherlands

Maris Ozols
QuSoft and University of Amsterdam, The Netherlands

Vedran Dunjko
applied Quantum algorithms (aQa), Leiden University, The Netherlands

Abstract

Understanding the power and limitations of quantum access to data in machine learning tasks is primordial to assess the potential of quantum computing in artificial intelligence. Previous works have already shown that speed-ups in learning are possible when given quantum access to reinforcement learning environments. Yet, the applicability of quantum algorithms in this setting remains very limited, notably in environments with large state and action spaces. In this work, we design quantum algorithms to train state-of-the-art reinforcement learning policies by exploiting quantum interactions with an environment. However, these algorithms only offer full quadratic speed-ups in sample complexity over their classical analogs when the trained policies satisfy some regularity conditions. Interestingly, we find that reinforcement learning policies derived from parametrized quantum circuits are well-behaved with respect to these conditions, which showcases the benefit of a fully-quantum reinforcement learning framework.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Theory of computation → Design and analysis of algorithms; Theory of computation → Reinforcement learning

Keywords and phrases quantum reinforcement learning, policy gradient methods, parametrized quantum circuits

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.13

Related Version *arXiv Version*: `https://arxiv.org/abs/2212.09328`

Funding *Sofiene Jerbi*: SJ acknowledges support from the Austrian Science Fund (FWF) through the projects DK-ALM:W1259-N27 and SFB BeyondC F7102. SJ also acknowledges the Austrian Academy of Sciences as a recipient of the DOC Fellowship.
Maris Ozols: MO was supported by an NWO Vidi grant (Project No. VI.Vidi.192.109).
Vedran Dunjko: This work was in part supported by the Dutch Research Council (NWO/OCW), as part of the Quantum Software Consortium programme (project number 024.003.037).

1 Introduction

When studying the potential advantages of quantum computing in machine learning, a natural question that arises is whether quantum algorithms that exploit *quantum access* to data can speed up learning. In the context of supervised learning, this led to the development of algorithms based on quantum RAMs, which can achieve high-degree polynomial improvements over their classical analogs [7]. In reinforcement learning, where we consider learning agents interacting with task environments, the question becomes: can quantum interactions with an environment, and in particular the ability to explore several trajectories in superposition, be beneficial for a learning agent. In recent years, several works have approached this

question from a variety of angles [13, 28]: based on Grover's algorithm [16], some works have for instance shown that searching for an optimal sequence of actions in an environment can be done using quadratically fewer interactions given the appropriate oracular access to the environment [12, 33, 18]. Other works have considered the more general problem of finding the optimal policy in a Markov Decision Process (MDP), and have found that up to quadratic speed-ups in the number of interactions are also possible, again given the proper oracular access [42, 41, 32, 6, 43]. Finally, tailored MDP environments (based, e.g., on Simon's problem) have also been introduced, which allow for exponential quantum speed-ups in learning times compared to the best classical agents [11].

Yet, all the quantum algorithms that have been proposed in this quantum-accessible setting remain inefficient in the most well-publicised use cases of reinforcement learning, such as Go [37], city navigation [29], and computer games [30]: environments with large state-action spaces. Indeed, aside from the task-specific algorithms of Ref. [11], the proposed algorithms scale at best as the square root of the size of the state-action space, which is intractable in most modern-day applications that deal for instance with image-based inputs. In the classical literature, modern approaches to reinforcement learning in large spaces commonly replace the explicit storage of a policy (and/or a value function) in a table of values by a parametrized model (e.g., a deep neural network), whose parameters $\boldsymbol{\theta}$ have a much smaller size than the state-action space. One of the earliest approaches based on such parametrized models is that of *policy gradient algorithms* [44, 40]. This approach frames reinforcement learning as a direct optimization problem, where the expected rewards (or value function) $V_{\pi_{\boldsymbol{\theta}}}(s_0)$ of a given policy $\pi_{\boldsymbol{\theta}}$ starting its interactions in a state s_0 is optimized via gradient ascent on the policy parameters $\boldsymbol{\theta}$. Therefore, the core task in this approach is to estimate the gradient $\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}(s_0)$ to a certain error ε in the ℓ_∞-norm. For this task, two approaches are common: *numerical* gradient estimation [24], where the value function is evaluated at different parameter settings $\boldsymbol{\theta}'$ centered around $\boldsymbol{\theta}$, that are combined to estimate the gradient at $\boldsymbol{\theta}$ (using, e.g., a central difference method), and *analytical* gradient estimation [40], using a formulation of this gradient as a function of the rewards and the gradients of the policy $\pi_{\boldsymbol{\theta}}$, averaged over trajectories generated by $\pi_{\boldsymbol{\theta}}$ (i.e., a Monte Carlo method).

Concurrently in the last few years, several works have introduced quantum parametrized models, known most commonly as parametrized or variational quantum circuits, that could take the place of deep neural networks in both policy-based [21, 35, 4, 27] and value-based [5, 25, 45, 38] reinforcement learning. While evaluated on a quantum computer, these models are however trained via classical interaction with the environment using, e.g., a classical policy gradient method.

In this work, we present quantum algorithms that speed up both the numerical and analytical gradient estimation approaches to policy gradient methods. These algorithms exploit an appropriately defined oracular access to the environment that allows to explore several trajectories in superposition, combined with subroutines for numerical gradient estimation [14, 8] and multivariate Monte Carlo estimation [10, 9]. Both these subroutines are however known to offer full quadratic speed-ups only in certain regimes, that depend in our setting on the smoothness of the value function $V_{\pi_{\boldsymbol{\theta}}}(s_0)$ and on the ℓ_p-norm of its gradient $\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}(s_0)$, respectively. Conveniently, we also identify families of parametrized quantum policies $\pi_{\boldsymbol{\theta}}$ previously studied in the literature [21] that satisfy the conditions of these regimes. We therefore end up with quantum policy gradient algorithms to train quantum policies, i.e., a fully quantum approach to reinforcement learning in large spaces.

2 Preliminaries

In this section, we present the main tools and concepts that we need to design our quantum policy gradient algorithms. We start by introducing policy gradient methods in Sec. 2.1. We then define the general oracle types that we consider in this work in Sec. 2.2, which allows us to properly define the notion of quantum access to a reinforcement learning environment in Sec. 2.3. We define the parametrized quantum policies that we apply our quantum policy gradient algorithms to in Sec. 2.4. And finally, we present the core subroutines used in our quantum algorithms in Sec. 2.5.

2.1 Policy gradient methods

At the core of policy gradient methods are two ingredients: a parametrized policy $\pi_{\boldsymbol{\theta}}$, that governs an agent's actions in an environment, and its associated value function $V_{\pi_{\boldsymbol{\theta}}}$, which evaluates the long-term performance of this policy in the environment. The policy $\pi_{\boldsymbol{\theta}}(\cdot|s)$ is a conditional probability distribution over actions given a state s, parametrized by a vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. When acting with a given policy in the environment, the agent experiences sampled trajectories (or episodes) $\tau = (s_0, a_0, r_0, s_1, \ldots)$ composed of states, actions and rewards that depend both on the policy of the agent and the environment dynamics (see Sec. 2.3 for more details). The standard figure of merit used to assess the performance of a policy $\pi_{\boldsymbol{\theta}}$ is called the value function $V_{\pi_{\boldsymbol{\theta}}}(s_0)$ and is given by the expected sum of rewards (or return) $R(\tau)$ collected in a trajectory:

$$V_{\pi_{\boldsymbol{\theta}}}(s_0) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}, P_E}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right] = \mathbb{E}_{\pi_{\boldsymbol{\theta}}, P_E}\left[R(\tau)\right] \tag{1}$$

where s_0 is the initial state of the agent's interaction τ with the environment and P_E a description of the environment dynamics (e.g., in the form of an MDP, see Def. 4). Each episode of interaction has a horizon (or length) $T \in \mathbb{N} \cup \{\infty\}$ and the returns $R(\tau)$ involve a discount factor $\gamma \in [0, 1]$ that allows, when $\gamma < 1$, to avoid diverging value functions for an infinite horizon, i.e., $T = \infty$.

Policy gradient methods take a direct optimization approach to RL: starting from an initial policy $\pi_{\boldsymbol{\theta}}$, its parameters are iteratively updated such as to maximize its associated value function $V_{\pi_{\boldsymbol{\theta}}}(s_0)$, via gradient ascent. For this method to be applicable, one needs to evaluate the gradient of the value function $\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}$, up to some error ε in ℓ_∞-norm to be specified.

2.1.1 Numerical gradient estimation

The most straightforward approach to estimate the value function of a policy is via a Monte Carlo approach: by collecting N episodes $\tau_i = (s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, s_1^{(i)} \ldots)$ governed by $\pi_{\boldsymbol{\theta}}$, one can compute for each of these the discounted return $R(\tau)$ appearing in Eq. (1) and average the results. The resulting value

$$\widetilde{V}_{\pi_{\boldsymbol{\theta}}}(s) = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T-1} \gamma^t r_t^{(i)} \tag{2}$$

is a Monte Carlo estimate of the value function.

With the capacity to estimate the value function, we can also estimate its gradient using numerical methods. In its simplest form, a finite-difference method simply evaluates $\widetilde{V}_{\pi_{\boldsymbol{\theta}}}(s_0)$ and $\widetilde{V}_{\pi_{\boldsymbol{\theta}+\delta\boldsymbol{e}_i}}(s_0)$ for $\delta > 0$ and $e_i = (0, \ldots, 0, 1_i, 0 \ldots, 0)$ a unit vector with support on the i-th parameter in $\boldsymbol{\theta}$, and returns the estimate:

$$\partial_i V_{\pi_{\boldsymbol{\theta}}}(s_0) \approx \frac{\widetilde{V}_{\pi_{\boldsymbol{\theta}+\delta\boldsymbol{e}_i}}(s_0) - \widetilde{V}_{\pi_{\boldsymbol{\theta}}}(s_0)}{\delta}. \tag{3}$$

Even though more elaborate finite difference methods exist (that we will use in Sec. 3), they inherently have a sample complexity (in terms of the number of interactions with the environment) that scales linearly in the dimension of $\boldsymbol{\theta}$.

2.1.2 Analytical gradient estimation

Perhaps one of the most appealing aspects of policy gradient methods is that the gradients of value functions also have an analytical formulation whose evaluation has a sample complexity only logarithmic in the dimension of $\boldsymbol{\theta}$ [23]. This analytical formulation is known as the policy gradient theorem:

▶ **Theorem 1** (Policy gradient theorem [40]). *Given a policy $\pi_{\boldsymbol{\theta}}$ that generates trajectories $\tau = (s_0, a_0, r_0, s_1, \ldots)$ in a reinforcement learning environment with time horizon $T \in \mathbb{N} \cup \{\infty\}$, the gradient of the value function $V_{\pi_{\boldsymbol{\theta}}}$ with respect to $\boldsymbol{\theta}$ is given by*

$$\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}(s_0) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} \right]. \tag{4}$$

A simple derivation of this Theorem can be found in Appendix A. Essentially, due to the so-called "log-likelihood trick" [36], the differentiation with respect to the policy parameters can be made to act solely on the random variables "inside" the expected value, while leaving the probability distribution behind this expected value unchanged. This means that the gradient of the value function can, similarly to the value function itself, be estimated via Monte Carlo sampling of trajectories governed by a fixed $\pi_{\boldsymbol{\theta}}$ and environment-independent computations (i.e., the evaluation of $\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t)$).

2.2 Input models

To design our quantum algorithms, we need to define access models to the environment as well as the policy $\pi_{\boldsymbol{\theta}}$ to be trained. We do this in terms of oracles that can be queried in superposition. Throughout this manuscript, we will be dealing with several types of such oracles, all defined in this section.

▶ **Definition 2** (Oracle types). *Let $\mathcal{X}$ be a finite set whose elements $x \in \mathcal{X}$ can be encoded as mutually orthogonal states $|x\rangle$, and let $f : \mathcal{X} \mapsto [0, B]$ be a function acting on this set, whose output is bounded by some $B \in \mathbb{R}$. We define different types of oracle access to f:*

1. ***Binary oracle**: $f(x)$ is encoded in an additional register using a binary representation of a desired precision:*

$$\mathcal{B}_f : |x\rangle |0\rangle \mapsto |x\rangle |f(x)\rangle , \tag{5}$$

2. ***Phase oracle**: $f(x)$ is encoded in the phase of the input register:*

$$O_f : |x\rangle \mapsto e^{i \frac{f(x)}{B}} |x\rangle , \tag{6}$$

3. ***Probability oracle***: *$f(x)$ is encoded in the amplitude of an additional qubit (possibly entangled to arbitrary states $|\psi_0(x)\rangle$ and $|\psi_1(x)\rangle$ of an additional register):*

$$\widetilde{O}_f : |x\rangle |0\rangle |0\rangle \mapsto |x\rangle \left(\sqrt{\frac{f(x)}{B}} |0\rangle |\psi_0(x)\rangle + \sqrt{1 - \frac{f(x)}{B}} |1\rangle |\psi_1(x)\rangle \right). \tag{7}$$

Clearly, having access to a binary oracle $\mathcal{B}_f$, we can easily convert it into a phase or probability oracle O_f or $\widetilde{O}_f$, using one call to $\mathcal{B}_f$ first, then a single-qubit rotation or a phase gate controlled on $|f(x)\rangle$, and finally a call to $\mathcal{B}_f^\dagger$ to uncompute $|f(x)\rangle$.

We will also need a subroutine to convert probability oracles into phase oracles:

▶ **Lemma 3** (Probability to phase oracle (Corollary 4.1 in [14])). *Suppose that we are given a probability oracle $\widetilde{O}_f$ for $f : \mathcal{X} \to [0, B]$. We can implement a phase oracle O_f up to operator norm error ε, with query complexity $\mathcal{O}(\log(1/\varepsilon))$, i.e., this many calls to $\widetilde{O}_f$ and its inverse.*

2.3 Quantum-accessible environments

Inspired by previous work that considered the quantum-accessible reinforcement learning setting [11, 42, 41, 32, 6], we define oracular access to a specific type of reinforcement learning environments called Markov Decision Processes (MDPs) [39], defined as follows:

▶ **Definition 4** (Markov Decision Process (MDP)). *A Markov Decision Process is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition probability matrix with entries $P(s'|s, a)$ that govern the transition to a state $s' \in \mathcal{S}$ after performing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, $R : \mathcal{S} \times \mathcal{A} \to [-|R|_{max}, |R|_{max}]$ is a reward function bounded by $|R|_{max} \in \mathbb{R}_+$ that assigns a reward $R(s, a)$ to every state-action pair, $T \in \mathbb{N} \cup \{\infty\}$ is a (possibly infinite) time horizon for each episode of interaction, and $\gamma \in [0, 1]$ is a discount factor, with the restriction that $\gamma < 1$ for $T = \infty$.*

Our oracular access to the environment takes the form of two oracles that coherently implement the MDP dynamics:

▶ **Definition 5** (Quantum access to an MDP). *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$ be an MDP as defined in Def. 4. We say that we have quantum access to the MDP if we can call the following oracles:*

1. *An oracle $\mathcal{P}$ that coherently samples a column of the transition probability matrix P:*

$$\mathcal{P} : |s, a\rangle |0\rangle \mapsto |s, a\rangle \sum_{s' \in \mathcal{S}} \sqrt{P(s'|s, a)} |s'\rangle . \tag{8}$$

2. *An oracle $\mathcal{R}$ that returns a binary representation of the output of the reward function R:*

$$\mathcal{R} : |s, a\rangle |0\rangle \mapsto |s, a\rangle |R(s, a)\rangle . \tag{9}$$

We also assume the ability to construct a unitary Π that coherently implements a policy $\pi_{\boldsymbol{\theta}}$:

▶ **Definition 6** (Quantum evaluation of a policy). *Let $\pi_{\boldsymbol{\theta}} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ be a reinforcement learning policy acting in a state-action space $\mathcal{S} \times \mathcal{A}$ and parametrized by a vector $\boldsymbol{\theta} \in \mathbb{R}^d$ (that can be encoded with finite precision as $|\boldsymbol{\theta}\rangle$). We say that the policy is quantum-evaluatable if we can construct a unitary satisfying:*

$$\Pi : |\boldsymbol{\theta}\rangle |s\rangle |0\rangle \mapsto |\boldsymbol{\theta}\rangle |s\rangle \sum_{a \in \mathcal{A}} \sqrt{\pi_{\boldsymbol{\theta}}(a|s)} |a\rangle . \tag{10}$$

Such a construction would be very natural for some quantum policies (such as the RAW-PQC defined in the next subsection). But any policy that can be computed classically could also be turned into such a unitary via quantum simulation of the classical computation of $(\pi_{\boldsymbol{\theta}}(a|s) : a \in \mathcal{A})$ and known subroutines to encode this probability vector into the amplitudes of a quantum state [15].

Equipped with the proper quantum access to the environment and the policy, we can construct simple subroutines that create superpositions of trajectories in the environment and evaluate the returns of these trajectories.

▶ **Lemma 7** (Superposition of trajectories). *Let $\mathcal{M}$ be a quantum-accessible MDP with oracles $\mathcal{P}, \mathcal{R}$ as defined in Def. 5, and let $\pi_{\boldsymbol{\theta}}$ be a quantum-evaluatable policy with its unitary implementation Π as defined in Def. 6. A unitary that prepares a coherent superposition of all trajectories $\tau = (s_0, a_0, \ldots, s_{T-1}, a_{T-1})$ of length T (without their rewards), i.e.,*

$$U_{P(\tau)} : |\boldsymbol{\theta}\rangle |s_0\rangle |0\rangle \mapsto |\boldsymbol{\theta}\rangle \sum_{\tau} \sqrt{P_{\boldsymbol{\theta}}(\tau)} |s_0, a_0, \ldots, s_{T-1}, a_{T-1}\rangle \tag{11}$$

for $P_{\boldsymbol{\theta}}(\tau) = \prod_{t=0}^{T-1} \pi_{\boldsymbol{\theta}}(a_t|s_t) P(s_{t+1}|s_t, a_t)$, can be implemented using $\mathcal{O}(T)$ calls to $\mathcal{P}$ and Π.

Proof. We apply sequentially Π and $\mathcal{P}$ on the registers indexed $\{0, 2i+1, 2i+2\}$ and $\{2i+1, 2i+2, 2i+3\}$ respectively, for $i = 0, \ldots, T-1$. This amounts to T calls to each oracle. ◀

▶ **Lemma 8** (Return). *Let $\mathcal{M}$ be a quantum-accessible MDP with oracles $\mathcal{P}, \mathcal{R}$ as defined in Def. 5, and let $\tau = (s_0, a_0, \ldots, s_{T-1}, a_{T-1})$ be a trajectory of length T in this MDP (without its rewards). A unitary that computes the return $R(\tau) = \sum_{t=0}^{T-1} \gamma^t r_t$ associated to this trajectory, i.e.,*

$$U_{R(\tau)} : |\tau\rangle |0\rangle \mapsto |\tau\rangle |R(\tau)\rangle \tag{12}$$

can be implemented using $\mathcal{O}(T)$ calls to $\mathcal{R}$.

Proof. Using T calls to $\mathcal{R}$, we simply collect all the rewards of the trajectory in an additional register. Then we simulate a classical circuit that computes the discounted sum of these rewards $R(\tau)$ (then uncompute the rewards using T calls to $\mathcal{R}$ on the same register). ◀

2.4 Quantum policies

The efficiency of our quantum policy gradient algorithms depends on regularity conditions on the policies $\pi_{\boldsymbol{\theta}}$ to be trained. Particularly well-behaved policies are policies defined out of parametrized quantum circuits (PQC) [2] that have been previously studied in classical reinforcement learning environments [21]. For each of our numerical and analytical gradient estimation algorithms, we will be interested more specifically in a certain type of PQC-policies, depicted in Fig. 1, and defined below.

▶ **Definition 9** (RAW-PQC). *Given a PQC acting on n qubits, taking as input a state $s \in \mathcal{S}$ and parameters $\boldsymbol{\phi} \in \mathbb{R}^d$, such that its corresponding unitary $U(s, \boldsymbol{\phi})$ produces the quantum state $|\psi_{s,\boldsymbol{\phi}}\rangle = U(s, \boldsymbol{\phi}) |0^{\otimes n}\rangle$, we define its associated RAW-PQC policy as:*

$$\pi_{\boldsymbol{\theta}}(a|s) = \langle P_a \rangle_{s,\boldsymbol{\theta}} \tag{13}$$

where $\langle P_a \rangle_{s,\boldsymbol{\theta}} = \langle \psi_{s,\boldsymbol{\phi}} | P_a | \psi_{s,\boldsymbol{\phi}} \rangle$ is the expectation value of a projection P_a associated to action a, such that $\sum_a P_a = I$ and $P_a P_{a'} = \delta_{a,a'} P_a$. $\boldsymbol{\theta} = \boldsymbol{\phi}$ constitutes all of its trainable parameters.

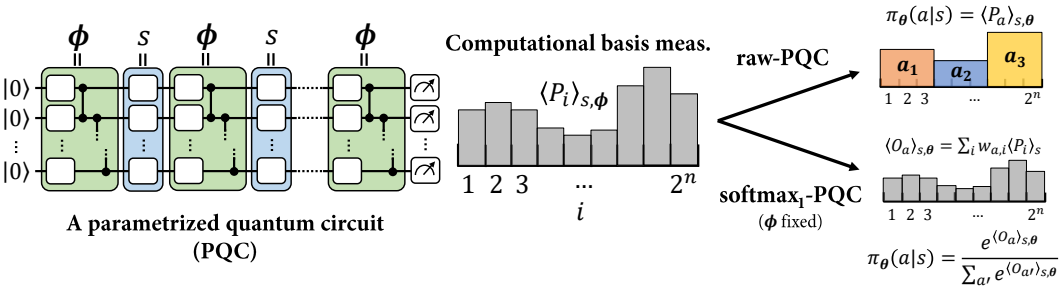


Figure 1 The parametrized quantum policies considered in this work. A parametrized quantum circuit (PQC) taking as input the agent's state s and parameters $\boldsymbol{\phi}$ produces a quantum state which has probability $\langle P_i \rangle_{s,\boldsymbol{\phi}}$ of being projected onto the (computational) basis state $|i\rangle$. The RAW-PQC policy simply assigns a subset of these basis states to each action $a \in \mathcal{A}$, and its parameters are $\boldsymbol{\theta} = \boldsymbol{\phi}$. The SOFTMAX$_1$-PQC policy uses instead a fixed assignment of $\boldsymbol{\phi}$, and computes the weighted expectation values $\langle O_a \rangle_{s,\boldsymbol{\theta}} = \sum_i w_{a,i} \langle P_i \rangle_s$.[1] The softmax of these expectation values gives the policy $\pi_{\boldsymbol{\theta}}$, whose parameters are $\boldsymbol{\theta} = \boldsymbol{w}$.

▶ **Definition 10** (SOFTMAX-PQC). *Given a PQC acting on n qubits, taking as input a state $s \in \mathcal{S}$ and parameters $\boldsymbol{\phi} \in \mathbb{R}^{d'}$, such that its corresponding unitary $U(s, \boldsymbol{\phi})$ produces the quantum state $|\psi_{s,\boldsymbol{\phi}}\rangle = U(s, \boldsymbol{\phi}) |0^{\otimes n}\rangle$, we define its associated SOFTMAX-PQC policy as:*

$$\pi_{\boldsymbol{\theta}}(a|s) = \frac{e^{\langle O_a \rangle_{s,\boldsymbol{\theta}}}}{\sum_{a'} e^{\langle O_{a'} \rangle_{s,\boldsymbol{\theta}}}} \tag{14}$$

where $\langle O_a \rangle_{s,\boldsymbol{\theta}} = \langle \psi_{s,\boldsymbol{\phi}} | \sum_i w_{a,i} H_{a,i} | \psi_{s,\boldsymbol{\phi}} \rangle$ is the expectation value of the weighted Hermitian operators $H_{a,i}$ associated to action a with weights $w_{a,i} \in \mathbb{R}$. $\boldsymbol{\theta} = (\boldsymbol{\phi}, \boldsymbol{w})$ constitutes all of its trainable parameters.

More specifically, we are interested in a restricted family of SOFTMAX-PQC policies:

▶ **Definition 11** (SOFTMAX$_1$-PQC). *We define a SOFTMAX$_1$-PQC policy as a SOFTMAX-PQC where $\boldsymbol{\phi} = \varnothing$ and, for all $a \in \mathcal{A}$, $H_{a,i} = P_{a,i}$ is a projection on a subspace indexed by i, such that $\sum_i P_{a,i} = I$ and $P_{a,i} P_{a,i'} = \delta_{i,i'} P_{a,i}$.*[2]

We call the resulting policy a SOFTMAX$_1$-PQC, as its log-policy gradient is always bounded in ℓ_1-norm, i.e., $\|\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s)\|_1 \leq 2, \forall s, a, \boldsymbol{\theta}$ (see Lemma 20).

2.5 Core subroutines

The core methods behind numerical and analytical policy gradient algorithms both have their quantum analogs, that offer up to quadratic speed-ups in certain regimes. In this section, we present these quantum subroutines and explain the conditions that govern the speed-up regimes.

[1] Note that the choice of basis for the measurement, i.e., the P_i's, could also depend on a.
[2] This constraint includes the degenerate case where $P_{a,i} = P_{a',i} = P_i$, for all a, a', illustrated in Fig. 1.

2.5.1 Quantum gradient estimation

Quantum algorithms for gradient estimation have been studied since early works in quantum computing. Notably, Jordan's algorithm [22] manages to estimate gradients $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$ with a query complexity that is independent of their dimension $d = |\boldsymbol{\theta}|$. However, this algorithm assumes a very powerful binary oracle access to the input function f (see Def. 2). And for functions that cannot be evaluated to arbitrary precision ε with a negligible cost in ε^{-1} (e.g., $\mathcal{O}(1)$ or $\mathcal{O}(\log(\varepsilon^{-1}))$), which is the case of value functions, the construction of this oracle introduces non-negligible costs [14]. More precisely, these costs depend on the dimension d, but also on the smoothness of the derivatives of f, as smoother functions are more amenable to efficient evaluation of their gradient. Notably, a measure of smoothness that has been studied for quantum gradient estimation is the Gevrey condition [14, 8]:

▶ **Definition 12** (Gevrey functions). *Let $d \in \mathbb{N}$, $\sigma \in [0,1]$, $M > 0$, $c > 0$, $\Omega \subseteq \mathbb{R}^d$ an open subset and $f : \mathbb{R}^d \to \mathbb{R}$. We say that f is a Gevrey function on Ω with parameters M, c and σ, and denote $f \in \mathcal{G}_{d,M,c,\sigma,\Omega}$ when all (higher order) partial derivatives of f exist, and the following upper bound on its partial derivatives is satisfied for all $\boldsymbol{x} \in \Omega$, $k \in \mathbb{N}_0$ and $\boldsymbol{\alpha} \in [d]^k$:*

$$|\partial_{\boldsymbol{\alpha}} f(\boldsymbol{x})| \leq \frac{M}{2} c^k (k!)^\sigma. \tag{15}$$

The query complexity of the quantum gradient estimation algorithm is summarized in the following theorem:

▶ **Theorem 13** (Numerical gradient estimation (Theorem 3.8 in [8])). *Given phase oracle access O_f to a function $f \in \mathcal{G}_{d,M,c,\sigma,\Omega}$, an $\varepsilon \in (0,c)$, and an $\boldsymbol{x} \in \Omega$ (such that a hypercube of edge length $\mathcal{O}(\log(cd^\sigma/\varepsilon)/\varepsilon)$ centered around $\boldsymbol{x}$ is still in Ω), there exists an algorithm that returns an ε-precise estimate of $\nabla f(\boldsymbol{x})$ in ℓ_∞-norm with success probability at least $2/3$ using*

$$\widetilde{\mathcal{O}}\left(\frac{Mcd^{\max\{\sigma,1/2\}}}{\varepsilon}\right) \tag{16}$$

queries to O_f.

Notably, in this case the dependence on the dimension of the gradient can only be reduced to $\sqrt{d}$ when the Gevrey condition of f satisfies $\sigma \leq 1/2$.

2.5.2 Quantum multivariate Monte Carlo

Quantum algorithms for estimating the mean $\mathbb{E}[X]$ of a *univariate* random variable X taking values in $\mathbb{R}$ [31] have been studied since early works by Grover [17], and culminated to a near-optimal algorithm that outperforms any classical estimator [19]. However, the case of *multivariate* random variables X taking values in $\mathbb{R}^d$ has been studied only more recently [10, 9, 20], and exhibits a dependence on the dimension d that can be up to exponentially worse than for classical estimators (which is $\mathcal{O}(\log(d))$, see Lemma 25). Before presenting explicitly this dependence on d, we first define the input model we consider for this problem:

▶ **Definition 14** (Quantum samples). *Consider a finite random variable $X : \Omega \to E$ on a probability space $(\Omega, 2^\Omega, P)$. Let $\mathcal{H}_\Omega$ and $\mathcal{H}_E$ be two Hilbert spaces with basis states $\{|\omega\rangle\}_{\omega\in\Omega}$ and $\{|x\rangle\}_{x\in E}$ respectively. We say that we have quantum-sample access to X when we can call the two following oracles:*

1. *A unitary U_P acting on $\mathcal{H}_\Omega$ as:*

$$U_P : |0\rangle \mapsto \sum_{\omega\in\Omega} \sqrt{P(\omega)}\,|\omega\rangle \tag{17}$$

and its inverse U_P^{-1}.

2. *A binary oracle $\mathcal{B}_X$ acting on $\mathcal{H}_\Omega \otimes \mathcal{H}_E$ such that:*

$$\mathcal{B}_X : |\omega\rangle\,|0\rangle \mapsto |\omega\rangle\,|X(\omega)\rangle\,. \tag{18}$$

▶ **Theorem 15** (Multivariate Monte Carlo estimation (Theorem 3.3 in [9])). *Let X be a d-dimensional bounded random variable such that $\|X\|_p \leq B$ for some $p \geq 1$. Given quantum-sample access to X, for any $\varepsilon, \delta > 0$, there exists a quantum multivariate mean estimator that returns an ε-precise estimate of $\mathbb{E}[X]$ in ℓ_∞-norm with success probability at least $1-\delta$ using*

$$\widetilde{\mathcal{O}}\left(\frac{Bd^{\xi(p)}}{\varepsilon}\right) \tag{19}$$

queries to X, where $\xi(p) = \max\{0, \frac{1}{2} - \frac{1}{p}\}$.

In contrast to the exposition of Theorem 3.3 in [9], we have used Hölder's inequality $\|X\|_2 \leq d^{\xi(p)}\|X\|_p$ to make use of a bound on X in any ℓ_p-norm, renormalized X by $d^{\xi(p)}B$ (a factor which reappears linearly in the number of oracle calls needed, as it impacts linearly the precision needed), and trivially upper bounded $\mathbb{E}[\|X\|_2]$ by $L_2 = 1$.

3 Numerical gradient estimation

We obtain our numerical policy gradient algorithm from the quantum gradient estimation subroutine introduced in Sec. 2.5.1. For this, we need to construct a phase oracle to the value function $V_{\pi_{\boldsymbol{\theta}}}(s_0)$, which can easily be obtained from the unitaries $U_{P(\tau)}$ and $U_{R(\tau)}$ constructed in Lemma 7 and 8 (see below). But we also need to show that the value function satisfies a Gevrey condition $\sigma \leq 1/2$ in order to get a full quadratic speed-up in sample complexity. For this, we identify the quantity:

$$D = \max_{k\in\mathbb{N}^*} (D_k)^{1/k} \tag{20}$$

where $\mathbb{N}^* = \mathbb{N}\backslash\{0\} \cup \{\infty\}$ and

$$D_k = \max_{s\in\mathcal{S}, \boldsymbol{\alpha}\in[d]^k} \sum_{a\in\mathcal{A}} |\partial_{\boldsymbol{\alpha}} \pi_{\boldsymbol{\theta}}(a|s)|. \tag{21}$$

which we show governs the Gevrey condition of the value function. More precisely, we find in Lemma 26 that it satisfies $\sigma = 0, M = \frac{4|R|_{\max}}{1-\gamma}$ and $c = DT^2$ in Def. 12. This allows us to show the following Theorem:

▶ **Theorem 16** (Numerical policy gradient algorithm). *Let $\pi_{\boldsymbol{\theta}}$ be a policy parametrized by a vector $\boldsymbol{\theta} \in \mathbb{R}^d$, that can be used to interact with a quantum-accessible MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$ with $\gamma T \geq 2$, and such that $\pi_{\boldsymbol{\theta}}$ has a bounded smoothness parameter D, defined in Eq. (20). The gradient of the value function corresponding to this policy, $\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}(s_0)$, can be evaluated to error ε in ℓ_∞-norm, using*

$$\widetilde{\mathcal{O}}\left(\sqrt{d}\frac{DT^2|R|_{max}}{\varepsilon(1-\gamma)}\right) \tag{22}$$

length-T episodes of interaction with the environment using a quantum numerical gradient estimator, while a classical numerical gradient estimator needs

$$\widetilde{\mathcal{O}}\left(d\left(\frac{DT^2|R|_{max}}{\varepsilon(1-\gamma)}\right)^2\right) \tag{23}$$

length-T episodes of interaction with the environment.

Proof. We apply Theorem 13 for $f = V_{\pi_{\boldsymbol{\theta}}}(s_0)$ as a function of $\boldsymbol{\theta}$. To construct the phase oracle O_f, we first construct a probability oracle $\widetilde{O}_f$ to f. For this we apply on the state $|s_0\rangle|0\rangle$ the unitaries $U_{P(\tau)}$ and $U_{R(\tau)}$ from Lemmas 7 and 8 respectively, to get

$$|\boldsymbol{\theta}\rangle|s_0\rangle|0\rangle|0\rangle \mapsto |\boldsymbol{\theta}\rangle\sum_\tau \sqrt{P_{\boldsymbol{\theta}}(\tau)}\,|\tau\rangle|R(\tau)\rangle|0\rangle\,. \tag{24}$$

Then we rotate the last qubit proportionally to the return $R(\tau)$, such that the probability of this qubit being $|0\rangle$ encodes the value function:

$$\mapsto |\boldsymbol{\theta}\rangle\sum_\tau \sqrt{P_{\boldsymbol{\theta}}(\tau)}\,|\tau\rangle|R(\tau)\rangle\left(\sqrt{\widetilde{R}(\tau)}\,|0\rangle + \sqrt{1-\widetilde{R}(\tau)}\,|1\rangle\right) \tag{25}$$

$$= |\boldsymbol{\theta}\rangle\sqrt{\widetilde{V}_{\pi_{\boldsymbol{\theta}}}(s_0)}\,|\psi_0\rangle|0\rangle + \sqrt{1-\widetilde{V}_{\pi_{\boldsymbol{\theta}}}(s_0)}\,|\psi_1\rangle|1\rangle \tag{26}$$

where $\widetilde{R}(\tau) = \frac{R(\tau)(1-\gamma)}{|R|_{\max}}$ and $\widetilde{V}_{\pi_{\boldsymbol{\theta}}}(s_0) = \frac{V_{\pi_{\boldsymbol{\theta}}}(s_0)(1-\gamma)}{|R|_{\max}}$. This probability oracle $\widetilde{O}_f$ can be converted into a phase oracle O_f using Lemma 3, which only comes with a logarithmic overhead in the query complexity.
From Lemma 26, we know that the value function satisfies the Gevrey conditions for $\sigma = 0, M = \frac{4|R|_{\max}}{1-\gamma}$ and $c = DT^2$, in Theorem 13, resulting in the stated quantum query complexity.

The classical query complexity is proven in Lemma 30. ◀

Note that the total query complexity of the quantum and classical numerical gradient estimators, in terms of the number of calls to $\mathcal{P}$ and $\mathcal{R}$, is $\widetilde{\mathcal{O}}\left(\sqrt{d}\frac{DT^3|R|_{\max}}{\varepsilon(1-\gamma)}\right)$ and $\widetilde{\mathcal{O}}\left(d\frac{D^2T^5|R|^2_{\max}}{\varepsilon^2(1-\gamma)^2}\right)$, respectively.
The RAW-PQC policies are then a perfect fit for these algorithms as we can show that:

▶ **Lemma 17.** *Any RAW-PQC policy as defined in Def. 9 satisfies $D \leq 1$.*

See Appendix D for a proof.

▶ **Corollary 18.** *Any RAW-PQC policy as defined in Def. 9 can benefit from a full quadratic speed-up from quantum numerical gradient estimation.*

4 Analytical gradient estimation

We obtain our analytical policy gradient algorithm by applying the quantum multivariate Monte Carlo algorithm of Sec. 2.5.2 to the formulation of the gradient given by the policy gradient theorem (see Sec. 2.1.2). The random variable in this formulation

$$X(\tau) = \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) R(\tau) \tag{27}$$

can easily be bounded in ℓ_p-norm given an upper bound on the return $R(\tau)$ and the ℓ_p-norm of the gradient of the log-policy:

$$B_p = \max_{s\in\mathcal{S}, a\in\mathcal{A}} \left\|\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s)\right\|_p. \tag{28}$$

With this notation we can show the following Theorem:

▶ **Theorem 19** (Analytical policy gradient algorithm). *Let $\pi_{\boldsymbol{\theta}}$ be a policy parametrized by a vector $\boldsymbol{\theta} \in \mathbb{R}^d$, that can be used to interact with a quantum-accessible MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$, and such that $\pi_{\boldsymbol{\theta}}$ has a bounded smoothness parameter B_p for some $p \geq 1$, defined in Eq. (28). Call $\xi(p) = \max\{0, \frac{1}{2} - \frac{1}{p}\}$. The gradient of the value function corresponding to this policy, $\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}(s_0)$, can be evaluated to error ε in ℓ_∞-norm, using*

$$\widetilde{\mathcal{O}}\left(d^{\xi(p)} \frac{B_p T |R|_{max}}{\varepsilon(1-\gamma)}\right) \tag{29}$$

length-T episodes of interaction with the environment using a quantum analytical gradient estimator, while a classical analytical gradient estimator needs

$$\widetilde{\mathcal{O}}\left(\left(\frac{B_p T |R|_{max}}{\varepsilon(1-\gamma)}\right)^2\right) \tag{30}$$

length-T episodes of interaction with the environment.[3] *Notably, for $p \in [1, 2]$, we get a full quadratic speed-up in the quantum setting.*

Proof. We apply Theorem 15 for the random variable $X(\tau) = \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \sum_{t'=0}^{T-1} \gamma^{t'} r_{t'}$ distributed according to $P_{\boldsymbol{\theta}}(\tau) = \prod_{t=0}^{T-1} \pi_{\boldsymbol{\theta}}(a_t|s_t) P(s_{t+1}|s_t, a_t)$.

To construct the appropriate quantum access to $X(\tau)$ (see Def. 14), we use the unitary $U_{P(\tau)}$ defined in Lemma 7 to implement U_P, and implement the binary oracle $\mathcal{B}_X$ using the unitary $U_{R(\tau)}$ defined in Lemma 8 along with a simulated classical circuit that multiplies the returns $R(\tau) = \sum_{t'=0}^{T-1} \gamma^{t'} r_{t'}$ with $\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t)$.

From Lemma 23, we get the bound $\|X(\tau)\|_p \leq \frac{T B_p |R|_{\max}}{1-\gamma}$, which we use as the bound B in Theorem 15, resulting in the stated quantum query complexity.

The classical complexity derives directly from Lemma 25 by noting that $\|X(\tau)\|_\infty \leq \|X(\tau)\|_p$ for any $p \geq 1$, and that sampling a trajectory τ (to compute a sample of $X(\tau)$) requires 1 episode of interaction with the environment. ◀

[3] Note that the classical estimator still has a logarithmic dependence in d, hidden in the $\widetilde{\mathcal{O}}$ notation.

Note that the total query complexity of the quantum and classical analytical gradient estimators, in terms of the number of calls to $\mathcal{P}$ and $\mathcal{R}$, is $\widetilde{\mathcal{O}}\left(d^{\xi(p)}\frac{B_p T^2 |R|_{\max}}{\varepsilon(1-\gamma)}\right)$ and $\widetilde{\mathcal{O}}\left(\frac{B_p^2 T^3 |R|_{\max}^2}{\varepsilon^2(1-\gamma)^2}\right)$, respectively.
The SOFTMAX$_1$-PQC policies are then a perfect fit for these algorithms as we can show that:

▶ **Lemma 20.** *Any SOFTMAX$_1$-PQC policy as defined in Def. 11 satisfies $B_1 \leq 2$.*

See Appendix E for a proof.

▶ **Corollary 21.** *Any SOFTMAX$_1$-PQC policy as defined in Def. 11 can benefit from a full quadratic speed-up from quantum analytical gradient estimation.*

5 Discussion

In this work, we design quantum algorithms to train parametrized policies in quantum-accessible environments. These algorithms can provide up to quadratic speed-ups in the number of interactions needed to evaluate the parameter updates of these policies, provided the environments allow for the appropriate quantum access. Their sample complexity is mostly governed by the number of parameters d of the policy, as well as the smoothness parameters D and B_p, depending on whether the numerical or analytical gradient estimation is used. These two smoothness parameters are hard to relate to each other in general, making the performances of these two algorithms hard to compare. Nonetheless, we show that quantum policies previously studied in the literature are smooth with respect to each of these parameters (i.e., with D or B_1 in $\mathcal{O}(1)$), which allows them to benefit from a full quadratic speed-up in sample complexity.

We note that in our results we only obtain quadratic speed-ups over specific classical algorithms that exploit the same smoothness conditions as our quantum algorithms. In order to strengthen these results, one would ideally prove matching lower bounds for the classical complexity of this task. We leave as an open question whether known classical lower bounds [1, 26] can be adapted to policy gradient evaluation.

In the analysis of the smoothness of the value function in Appendix F (specifically around Eq. (68)), we end up bounding its derivatives $\partial_{\boldsymbol{\alpha}} V_{\pi_{\boldsymbol{\theta}}}^{(t)}(s)$ using a loose upper bound, especially in the regime where the order $k = |\boldsymbol{\alpha}|$ of the derivation is small. The reason for this loose bound is that we need to cast it as a Gevrey condition in order to apply the numerical gradient algorithms of Refs. [14, 8]. We conjecture that a modification of the construction in [14, 8] may be possible such as to gain an improvement by a factor of T in the sample complexity of our numerical gradient algorithm, and such that the resulting scaling in T would match that of our analytical gradient estimation algorithm. Side-stepping the Gevrey-formulation of the bound would also remove the need for the condition $\gamma T \geq 2$ that we enforce in the MDP (which is in any case not a very limiting condition, as MDPs of interest usually have a large horizon T and a discount factor γ close to 1 – typically $T \approx 10\,000$ and $\gamma \approx 0.99$ for Atari games [30]).

References

1 Abdulrahman Alabdulkareem and Jean Honorio. Information-theoretic lower bounds for zero-order stochastic gradient estimation. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2316–2321. IEEE, 2021.

2 Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.

3 Marco Cerezo and Patrick J Coles. Higher order derivatives of quantum neural networks with barren plateaus. *Quantum Science and Technology*, 6(3):035006, 2021.

4 Samuel Yen-Chi Chen, Chih-Min Huang, Chia-Wei Hsing, Hsi-Sheng Goan, and Ying-Jer Kao. Variational quantum reinforcement learning via evolutionary optimization. *Machine Learning: Science and Technology*, 3(1):015025, 2022.

5 Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan. Variational quantum circuits for deep reinforcement learning. *IEEE Access*, 8:141007–141024, 2020.

6 El Amine Cherrat, Iordanis Kerenidis, and Anupam Prakash. Quantum reinforcement learning via policy iteration. *arXiv:2203.01889*, 2022.

7 Nai-Hui Chia, András Pal Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. *Journal of the ACM*, 69(5):1–72, 2022.

8 Arjan Cornelissen. Quantum gradient estimation of gevrey functions. *arXiv:1909.13528*, 2019.

9 Arjan Cornelissen, Yassine Hamoudi, and Sofiene Jerbi. Near-optimal quantum algorithms for multivariate mean estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 33–43, 2022.

10 Arjan Cornelissen and Sofiene Jerbi. Quantum algorithms for multivariate monte carlo estimation. *arXiv:2107.03410*, 2021.

11 Vedran Dunjko, Yi-Kai Liu, Xingyao Wu, and Jacob M Taylor. Exponential improvements for quantum-accessible reinforcement learning. *arXiv:1710.11160*, 2017.

12 Vedran Dunjko, Jacob M Taylor, and Hans J Briegel. Quantum-enhanced machine learning. *Physical review letters*, 117(13):130501, 2016.

13 Vedran Dunjko, Jacob M Taylor, and Hans J Briegel. Advances in quantum reinforcement learning. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 282–287. IEEE, 2017.

14 András Gilyén, Srinivasan Arunachalam, and Nathan Wiebe. Optimizing quantum optimization algorithms via faster quantum gradient computation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1425–1444. SIAM, 2019.

15 Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *quant-ph/0208112*, 2002.

16 Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.

17 Lov K Grover. A framework for fast quantum mechanical algorithms. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 53–62, 1998.

18 Arne Hamann, Vedran Dunjko, and Sabine Wölk. Quantum-accessible reinforcement learning beyond strictly epochal environments. *Quantum Machine Intelligence*, 3(2):1–18, 2021.

19 Yassine Hamoudi. Quantum sub-gaussian mean estimator. In *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

20 William J Huggins, Kianna Wan, Jarrod McClean, Thomas E O'Brien, Nathan Wiebe, and Ryan Babbush. Nearly optimal quantum algorithm for estimating multiple expectation values. *Physical Review Letters*, 129(24):240501, 2022.

21 Sofiene Jerbi, Casper Gyurik, Simon Marshall, Hans Briegel, and Vedran Dunjko. Parametrized quantum policies for reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021. URL: `https://proceedings.neurips.cc/paper/2021/hash/eec96a7f788e88184c0e713456026f3f-Abstract.html`.

22 Stephen P Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical review letters*, 95(5):050501, 2005.

23 Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, UCL (University College London), 2003.

24 Nate Kohl and Peter Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 2619–2624. IEEE, 2004.

25 Owen Lockwood and Mei Si. Reinforcement learning with quantum variational circuit. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pages 245–251, 2020.

26 G. Lugosi and S. Mendelson. Mean estimation and regression under heavy-tailed distributions: A survey. *Foundations of Computational Mathematics*, 19(5):1145–1190, 2019. doi:10.1007/s10208-019-09427-x.

27 Nico Meyer, Daniel D Scherer, Axel Plinge, Christopher Mutschler, and Michael J Hartmann. Quantum policy gradient algorithm with optimized action decoding. *arXiv preprint arXiv:2212.06663*, 2022.

28 Nico Meyer, Christian Ufrecht, Maniraman Periyasamy, Daniel D Scherer, Axel Plinge, and Christopher Mutschler. A survey on quantum reinforcement learning. *arXiv preprint arXiv:2211.03464*, 2022.

29 Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. Learning to navigate in cities without a map. *Advances in Neural Information Processing Systems*, 31:2419–2430, 2018.

30 Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

31 Ashley Montanaro. Quantum speedup of monte carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015.

32 Pooya Ronagh. The problem of dynamic programming on a quantum computer. *arXiv:1906.02229*, 2019.

33 Valeria Saggio, Beate E Asenbeck, Arne Hamann, Teodor Strömberg, Peter Schiansky, Vedran Dunjko, Nicolai Friis, Nicholas C Harris, Michael Hochberg, Dirk Englund, et al. Experimental quantum speed-up in reinforcement learning agents. *Nature*, 591(7849):229–233, 2021.

34 Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

35 André Sequeira, Luis Paulo Santos, and Luis Soares Barbosa. Policy gradients using variational quantum circuits. *Quantum Machine Intelligence*, 5(1):18, 2023.

36 David Silver. Lectures on reinforcement learning. URL: https://www.davidsilver.uk/teaching/, 2015.

37 David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

38 Andrea Skolik, Sofiene Jerbi, and Vedran Dunjko. Quantum agents in the gym: a variational quantum algorithm for deep q-learning. *Quantum*, 6:720, 2022. doi:10.22331/q-2022-05-24-720.

39 Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT Press, 1998.

40 Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

41 Daochen Wang, Aarthi Sundaram, Robin Kothari, Ashish Kapoor, and Martin Roetteler. Quantum algorithms for reinforcement learning with a generative model. In *International Conference on Machine Learning*, pages 10916–10926. PMLR, 2021.

42 Daochen Wang, Xuchen You, Tongyang Li, and Andrew M Childs. Quantum exploration algorithms for multi-armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10102–10110, 2021.

43 Simon Wiedemann, Daniel Hein, Steffen Udluft, and Christian Mendl. Quantum policy iteration via amplitude estimation and grover search–towards quantum advantage for reinforcement learning. *arXiv preprint arXiv:2206.04741*, 2022.

44 Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

45 Shaojun Wu, Shan Jin, Dingding Wen, and Xiaoting Wang. Quantum reinforcement learning in continuous action space. *arXiv:2012.10711*, 2020.

A Simple derivation of the policy gradient theorem

▶ **Theorem 22** (Policy gradient theorem [40]). *Given a policy $\pi_{\boldsymbol{\theta}}$ that generates trajectories $\tau = (s_0, a_0, r_0, s_1, \ldots)$ in a reinforcement learning environment with time horizon $T \in \mathbb{N} \cup \{\infty\}$, the gradient of the value function $V_{\pi_{\boldsymbol{\theta}}}$ with respect to $\boldsymbol{\theta}$ is given by*

$$\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}(s_0) = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \sum_{t'=0}^{T-1} \gamma^{t'} r_{t'} \right]. \tag{31}$$

Proof. Call $R(\tau) = \sum_{t=0}^{T-1} \gamma^t r_t$ the return of a trajectory τ, and $P_{\boldsymbol{\theta}}(\tau) = \prod_{t=0}^{T-1} \pi_{\boldsymbol{\theta}}(a_t|s_t) P_E(s_{t+1}|s_t, a_t)$ the probability of this trajectory, where P_E describes the unknown dynamics of the environment.

Then, we can write the value function as

$$V_{\pi_{\boldsymbol{\theta}}}(s_0) = \sum_\tau P_{\boldsymbol{\theta}}(\tau) R(\tau) \tag{32}$$

and its gradient as

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}(s_0) &= \sum_\tau \nabla_{\boldsymbol{\theta}} P_{\boldsymbol{\theta}}(\tau) R(\tau) && (33)\\
&= \sum_\tau P_{\boldsymbol{\theta}}(\tau) \frac{\nabla_{\boldsymbol{\theta}} P_{\boldsymbol{\theta}}(\tau)}{P_{\boldsymbol{\theta}}(\tau)} R(\tau) && (34)\\
&= \sum_\tau P_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{\theta}} \log(P_{\boldsymbol{\theta}}(\tau)) R(\tau) && (35)\\
&= \sum_\tau P_{\boldsymbol{\theta}}(\tau) \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) R(\tau) && (36)\\
&= \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log(\pi_{\boldsymbol{\theta}}(a_t|s_t)) R(\tau) \right] && (37)
\end{aligned}$$

where we have artificially divided and multiplied each term by $P_{\boldsymbol{\theta}}(\tau)$ in the second line, and used the independence on $\boldsymbol{\theta}$ of the environment dynamics $P_E(s_{t+1}|s_t, a_t)$ in the fourth line. ◀

B Lemmas concerning properties of MDPs

B.1 An upper bound on the value function

▶ **Lemma 23.** *Consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$ as defined in Def. 4. The value function $V_{\pi_{\boldsymbol{\theta}}}(s_0) = \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t r_t\right]$ of any policy $\pi_{\boldsymbol{\theta}}$, evaluated on any initial state $s_0 \in \mathcal{S}$ is upper bounded by*

$$|V_{\pi_{\boldsymbol{\theta}}}(s_0)| \leq \min\left\{T, \frac{1}{1-\gamma}\right\} |R|_{max}. \tag{38}$$

Proof. We have, by definition of the MDP, $r_t \leq |R|_{\max}$, which implies:

$$\left|\sum_{t=0}^{T-1}\gamma^t r_t\right| \leq \sum_{t=0}^{T-1}\gamma^t|r_t| \leq \sum_{t=0}^{T-1}\gamma^t|R|_{\max} \leq \begin{cases}\frac{|R|_{\max}}{1-\gamma} & \text{if } \gamma<1\\ T|R|_{\max} & \text{always}\end{cases} \tag{39}$$

which also holds in expectation value over all trajectories of length T. ◀

B.2 The effective time horizon of an MDP

▶ **Lemma 24.** *Consider an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$ as defined in Def. 4, with an infinite horizon $T = \infty, \gamma < 1$ and a value function $V_{\pi_{\boldsymbol{\theta}}}$. The finite-horizon MDP $\mathcal{M}' = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T^*, \gamma)$, where*

$$T^* = \left\lceil \frac{\log\left(\frac{\varepsilon(1-\gamma)}{|R|_{max}}\right)}{\log(\gamma)} \right\rceil = \widetilde{\mathcal{O}}\left(\frac{1}{1-\gamma}\right) \tag{40}$$

has a value function $V'_{\pi_{\boldsymbol{\theta}}}$ that satisfies

$$\left|V_{\pi_{\boldsymbol{\theta}}}(s_0) - V'_{\pi_{\boldsymbol{\theta}}}(s_0)\right| \leq \varepsilon \tag{41}$$

for any initial state $s_0 \in \mathcal{S}$ and any policy $\pi_{\boldsymbol{\theta}}$.

Proof.

$$\begin{aligned}\left|V_{\pi_{\boldsymbol{\theta}}}(s_0) - V'_{\pi_{\boldsymbol{\theta}}}(s_0)\right| &= \left|\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right] - \mathbb{E}\left[\sum_{t=0}^{T^*-1}\gamma^t r_t\right]\right| && (42)\\ &= \left|\mathbb{E}\left[\sum_{t=T^*}^{\infty}\gamma^t r_t\right]\right| && (43)\\ &\leq \gamma^{T^*}\frac{|R|_{\max}}{1-\gamma} && (44)\\ &\leq \frac{\varepsilon(1-\gamma)}{|R|_{\max}}\frac{|R|_{\max}}{1-\gamma} = \varepsilon. && (45)\end{aligned}$$

◀

Because of this lemma, we always assume the time horizon T of an MDP to be in $\widetilde{\mathcal{O}}\left(\frac{1}{1-\gamma}\right)$.

C Complexity of a classical MVMC algorithm

▶ **Lemma 25** (Classical multivariate Monte Carlo estimation)**.** *Let X be a d-dimensional bounded random variable such that $\|X\|_\infty \leq B$. Given sampling access to X, $\varepsilon, \delta > 0$, there exists a classical multivariate mean estimator that returns an ε-precise estimate of $\mathbb{E}[X]$ in ℓ_∞-norm with success probability at least $1-\delta$ using*

$$\widetilde{\mathcal{O}}\left(\left(\frac{B}{\varepsilon}\right)^2\right) \tag{46}$$

samples of X.

Proof. Consider the following algorithm:

1. Collect $N = \left\lceil \frac{2B^2}{\varepsilon^2} \log\left(\frac{2d}{\delta}\right) \right\rceil$ samples of X: $\left\{ \boldsymbol{x}^{(i)} = (x_1^{(i)}, \ldots, x_d^{(i)}) \right\}_{1 \leq i \leq N}$.
2. Compute the d coordinate-wise averages $\widehat{x}_j = \frac{1}{N} \sum_{i=1}^{N} x_j^{(i)}$ and use $\widehat{\boldsymbol{x}} = (\widehat{x}_1, \ldots, \widehat{x}_d)$ as an estimate.

Now consider the probability of failure of this algorithm, i.e., that at least one of the estimates is more than ε away from its expected value:

$$\begin{aligned}
\mathbb{P}\left(\bigvee_{j \in [d]} |\widehat{x}_j - \mathbb{E}[x_j]| \geq \varepsilon \right) &\leq \sum_{j=1}^{d} \mathbb{P}\left(|\widehat{x}_j - \mathbb{E}[x_j]| \geq \varepsilon \right) && \text{\# union bound} \\
&\leq d \times \max_{j \in [d]} \mathcal{P}\left(|\widehat{x}_j - \mathbb{E}[x_j]| \geq \varepsilon \right) \\
&\leq 2d \exp\left(-\frac{2N^2\varepsilon^2}{4NB^2} \right) && \text{\# Hoeffding's bound and bound on } x_j \\
&\leq \delta. && \text{\# definition of } N
\end{aligned}$$

Hence, for arbitrary ε and δ, the d expectations can be estimated to error ε in the ℓ_∞-norm with success probability $1 - \delta$ using $N = \mathcal{O}\left(\frac{B^2}{\varepsilon^2} \log\left(\frac{d}{\delta}\right) \right)$ samples of X. ◀

D Proof of Lemma 17

▶ **Lemma 17.** *Any* RAW*-PQC policy as defined in Def. 9 satisfies* $D \leq 1$.

Proof. Given a RAW-PQC policy $\pi_{\boldsymbol{\theta}}$ as defined in Def. 9, we seek to bound the following quantity:

$$D = \max_{k \in \mathbb{N}^*} (D_k)^{1/k} \tag{47}$$

where

$$D_k = \max_{s \in \mathcal{S}, \boldsymbol{\alpha} \in [d]^k} \sum_{a \in \mathcal{A}} |\partial_{\boldsymbol{\alpha}} \pi_{\boldsymbol{\theta}}(a|s)|. \tag{48}$$

Gradients of this PQC policy can be evaluated using the parameter-shift rule [34]:

$$\partial_i \pi_{\boldsymbol{\theta}}(a|s) = \partial_i \langle P_a \rangle_{s,\boldsymbol{\theta}} = \frac{\langle P_a \rangle_{s,\boldsymbol{\theta} + \frac{\pi}{2} e_i} - \langle P_a \rangle_{s,\boldsymbol{\theta} - \frac{\pi}{2} e_i}}{2} \tag{49}$$

which can easily be generalized to higher-order derivatives [3]:

$$\partial_{\boldsymbol{\alpha}} \pi_{\boldsymbol{\theta}}(a|s) = \frac{1}{2^k} \sum_{\boldsymbol{\omega}} c_{\boldsymbol{\omega}} \langle P_a \rangle_{s,\boldsymbol{\theta}+\boldsymbol{\omega}} \tag{50}$$

for $\boldsymbol{\alpha} \in [d]^k, \boldsymbol{\omega} \in \{0, \pm\frac{\pi}{2}, \pm\pi, \pm\frac{3\pi}{2}\}^d$, and $c_{\boldsymbol{\omega}} \in \mathbb{Z}$ such that $\sum_{\boldsymbol{\omega}} |c_{\boldsymbol{\omega}}| = 2^k$.

Now, by combining Eq. (48) and (50), we get:

$$\begin{aligned}
D_k &= \max_{s \in \mathcal{S}, \boldsymbol{\alpha} \in [d]^k} \sum_{a \in \mathcal{A}} \left| \frac{1}{2^k} \sum_{\boldsymbol{\omega}} c_{\boldsymbol{\omega}} \langle P_a \rangle_{s,\boldsymbol{\theta}+\boldsymbol{\omega}} \right| && (51) \\
&\leq \max_{s \in \mathcal{S}, \boldsymbol{\alpha} \in [d]^k} \frac{1}{2^k} \sum_{a \in \mathcal{A}} \sum_{\boldsymbol{\omega}} |c_{\boldsymbol{\omega}}| \left| \langle P_a \rangle_{s,\boldsymbol{\theta}+\boldsymbol{\omega}} \right| && (52) \\
&= \max_{s \in \mathcal{S}, \boldsymbol{\alpha} \in [d]^k} \frac{1}{2^k} \sum_{\boldsymbol{\omega}} |c_{\boldsymbol{\omega}}| \sum_{a \in \mathcal{A}} \left| \langle P_a \rangle_{s,\boldsymbol{\theta}+\boldsymbol{\omega}} \right| = 1. && (53)
\end{aligned}$$

where in the last line we used $\sum_a P_a = I$ in the definition of the RAW-PQC policy and $\sum_{\boldsymbol{\omega}} |c_{\boldsymbol{\omega}}| = 2^k$.

Since this bound is valid for all $k \in \mathbb{N}^*$, then also $D \leq 1$. ◀

E Proof of Lemma 20

▶ **Lemma 20.** *Any* SOFTMAX$_1$*-PQC policy as defined in Def. 11 satisfies* $B_1 \leq 2$.

Proof. Given a SOFTMAX$_1$-PQC policy $\pi_{\boldsymbol{\theta}}$ as defined in Def. 11, we seek to bound the following quantity:

$$B_1 = \max_{s\in\mathcal{S},a\in\mathcal{A}} \|\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s)\|_1. \tag{54}$$

From the definition of this policy, we have:

$$\langle O_a \rangle_{s,\boldsymbol{\theta}} = \langle \psi_s | \sum_i w_{a,i} P_{a,i} | \psi_s \rangle \tag{55}$$

such that $\sum_i P_{a,i} = I$ and $P_{a,i}P_{a,i'} = \delta_{i,i'}P_{a,i}$, $\forall a \in \mathcal{A}$. This implies that

$$\partial_{w_{a',i}} \langle O_a \rangle_{s,\boldsymbol{\theta}} = \delta_{a,a'} \langle \psi_s | P_{a',i} | \psi_s \rangle = \delta_{a,a'} \langle P_{a',i} \rangle_s . \tag{56}$$

Since this is a SOFTMAX-PQC, it follows from Lemma 1 in [21] that:

$$\partial_{w_{a',i}} \log \pi_{\boldsymbol{\theta}}(a|s) = \partial_{w_{a',i}} \langle O_a \rangle_{s,\boldsymbol{\theta}} - \sum_{a''\in\mathcal{A}} \pi_{\boldsymbol{\theta}}(a''|s) \partial_{w_{a',i}} \langle O_{a''} \rangle_{s,\boldsymbol{\theta}} \tag{57}$$

$$= \delta_{a,a'} \langle P_{a',i} \rangle_s - \pi_{\boldsymbol{\theta}}(a'|s) \langle P_{a',i} \rangle_s . \tag{58}$$

Therefore,

$$\|\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s)\|_1 = \sum_{a',i} \left| \partial_{w_{a',i}} \log \pi_{\boldsymbol{\theta}}(a|s) \right| \tag{59}$$

$$\leq \sum_{a',i} \left[\left| \delta_{a,a'} \langle P_{a',i} \rangle_s \right| + \left| \pi_{\boldsymbol{\theta}}(a'|s) \langle P_{a',i} \rangle_s \right| \right] \tag{60}$$

$$\leq \sum_i \langle P_{a,i} \rangle_s + \sum_{a',i} \pi_{\boldsymbol{\theta}}(a'|s) \langle P_{a',i} \rangle_s \tag{61}$$

$$\leq 1 + \max_{a'} \sum_i \langle P_{a',i} \rangle_s \tag{62}$$

$$\leq 2 \tag{63}$$

where we made use of the triangle inequality in the first inequality, the positivity of $\langle P_{a,i} \rangle_s$ and $\pi_{\boldsymbol{\theta}}(a'|s)$ in the second inequality, and the normalization constraint of $\{P_{a,i}\}_i$ in the third and fourth inequalities. ◀

F Gevrey condition of value functions

In this section, we investigate the smoothness of the value function, in terms of the smoothness of the policy. More precisely, we prove the following lemma:

▶ **Lemma 26.** *Let $\pi_{\boldsymbol{\theta}}$ be a parametrized policy with a bounded smoothness parameter D, defined in Eq. (20). Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$ be an MDP as defined in Def. 4 with $T\gamma \geq 2$. Then the value function $V_{\pi_{\boldsymbol{\theta}}}(s_0)$ associated to the policy $\pi_{\boldsymbol{\theta}}$ in $\mathcal{M}$, as a function of the policy parameters $\boldsymbol{\theta}$, satisfies the Gevrey conditions of Def. 12 for $\sigma = 0, M = \frac{4|R|_{max}}{1-\gamma}$ and $c = DT^2$.*

As a first step, we observe that we can use the Markovian nature of an MDP to describe the value function as the limit of a sequence of improving approximations, by iteratively increasing the time horizon at which we evaluate the MDP. More precisely, we define inductively, for all states $s \in \mathcal{S}$ and time horizons $t \geq 0$,

$$V_{\pi_{\boldsymbol{\theta}}}^{(t+1)}(s) = \sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s) \left[R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V_{\pi_{\boldsymbol{\theta}}}^{(t)}(s') \right],$$

where for the induction basis, we use $V_{\pi_{\boldsymbol{\theta}}}^{(0)}(s) = 0$, for all states $s \in \mathcal{S}$. We easily check that the value function at time horizon $T \in \mathbb{N} \cup \{\infty\}$ of an MDP, $V_{\pi_{\boldsymbol{\theta}}}(s)$, is indeed given by letting t go to T in the above definition.

This recursive definition of approximations to the value function provides us with a convenient handle on its derivatives. In particular, for all integers $k, t > 0$ and sequences $\boldsymbol{\alpha} \in [d]^k$, where d is the number of parameters of $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta} \in \mathbb{R}^d$, we obtain that

$$\partial_{\boldsymbol{\alpha}} \left[V_{\pi_{\boldsymbol{\theta}}}^{(t+1)}(s) - V_{\pi_{\boldsymbol{\theta}}}^{(t)}(s) \right] = \gamma \partial_{\boldsymbol{\alpha}} \left[\sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\theta}}(a|s) \sum_{s' \in \mathcal{S}} P(s'|s,a) (V_{\pi_{\boldsymbol{\theta}}}^{(t)}(s') - V_{\pi_{\boldsymbol{\theta}}}^{(t-1)}(s')) \right]. \quad (64)$$

Since the value function with time horizon $t = 0$ vanishes, we can express the partial derivatives at any given time horizon t as the telescoping sum

$$\partial_{\boldsymbol{\alpha}} V_{\pi_{\boldsymbol{\theta}}}^{(t)}(s) = \sum_{t'=0}^{t-1} \partial_{\boldsymbol{\alpha}} \left[V_{\pi_{\boldsymbol{\theta}}}^{(t'+1)}(s) - V_{\pi_{\boldsymbol{\theta}}}^{(t')}(s) \right].$$

The main idea of this section is to expand the expression on the right-hand side in the above equation, using the recursive characterization provided in Eq. (64).

We start by defining some shorthand notation:

▶ **Definition 27.** *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$ be an MDP, and $\pi_{\boldsymbol{\theta}}$ be a policy parametrized by $\boldsymbol{\theta} \in \mathbb{R}^d$. Let $V_{\pi_{\boldsymbol{\theta}}}^{(t)}$ be its value function with horizon $t > 0$, and for all $k, t > 0$, let*

$$g(k,t) = \max_{s \in \mathcal{S}, \boldsymbol{\alpha} \in [d]^k} \left| \partial_{\boldsymbol{\alpha}} \left[V_{\pi_{\boldsymbol{\theta}}}^{(t+1)}(s) - V_{\pi_{\boldsymbol{\theta}}}^{(t)}(s) \right] \right|, \qquad \text{and} \qquad U(k,t) = \sum_{t'=0}^{t-1} g(k,t').$$

We observe that

$$|\partial_{\boldsymbol{\alpha}} V_{\pi_{\boldsymbol{\theta}}}^{(t)}(s)| \leq \sum_{t'=0}^{t-1} \left| \partial_{\boldsymbol{\alpha}} \left[V_{\pi_{\boldsymbol{\theta}}}^{(t'+1)}(s) - \partial_{\boldsymbol{\alpha}} V_{\pi_{\boldsymbol{\theta}}}^{(t')}(s) \right] \right| \leq \sum_{t'=0}^{t-1} g(k,t') = U(k,t), \quad (65)$$

and hence to bound the smoothness of (the approximations to) the value function, it suffices to find a good upper bound on $U(k,t)$. The previous definition already foreshadows that the resulting expression explicitly depends on the smoothness of the policy through the parameter D.

In order to upper bound $U(k,t)$, we first find an expression that upper bounds $g(k,t)$, which is the objective of the following lemma.

▶ **Lemma 28.** *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$ be an MDP, and $\pi_{\boldsymbol{\theta}}$ be a policy parametrized by $\boldsymbol{\theta} \in \mathbb{R}^d$. Let $V_{\pi_{\boldsymbol{\theta}}}^{(t)}$ be its value function with horizon $t > 0$. For all $k \in \mathbb{N}$, let Λ_k be the set of all partitions of k, where every partition $\lambda \in \Lambda_k$ is a multiset of positive integers that*

sums to k. By $\{\lambda\}$, we denote the set of elements in λ, i.e., without repetition. We let $\#\ell(\lambda)$ be the number of occurrences of ℓ in the multiset λ, and let $\#\lambda = \{\#\ell(\lambda) : \ell \in \{\lambda\}\}$ be the multiset of occurrences in λ. For all non-negative integers k, t, we have

$$g(k,t) \leq \gamma^t |R|_{max} \cdot \sum_{\lambda \in \Lambda_k} \binom{k}{\lambda} \binom{|\lambda|}{\#\lambda} \binom{t+1}{|\lambda|} \prod_{\ell \in \lambda} D_\ell.$$

Proof. We give a combinatorial argument. To that end, let $k, t \geq 0$ be integers, and let $\boldsymbol{\alpha} \in [d]^k$ be a finite sequence of indices with respect to which we want to compute the partial derivative of $V_{\pi_{\boldsymbol{\theta}}}^{(t)}$. The main idea is to apply the product rule to the expression on the right-hand side of Eq. (64).

In particular, by repeatedly substituting the right-hand side of Eq. (64) into itself, we obtain that there are $t+1$ probabilities $\pi_{\boldsymbol{\theta}}(a|s)$ to which we can associate any given index of $\boldsymbol{\alpha}$. Thus, we count the number of occurrences where the distribution of indices in $\boldsymbol{\alpha}$ across the $t+1$ different factors forms the partition $\lambda \in \Lambda_k$. We call this number c_λ, and we indeed observe that all these terms are upper bounded by $\prod_{\ell \in \lambda} D_\ell$, which means that it remains to prove that

$$c_\lambda = \binom{k}{\lambda} \binom{|\lambda|}{\#\lambda} \binom{t+1}{|\lambda|}.$$

Observe that we must first choose which factors to assign any derivative to at all, which can be done in $\binom{t+1}{|\lambda|}$ ways. Then, we must decide how many derivatives we are going to assign to each of the selected factors, which can be done in $\binom{|\lambda|}{\#\lambda}$ ways. Finally, we must distribute the k derivatives among the groups, which can be done in $\binom{k}{\lambda}$ ways. This completes the proof. ◀

Now that we have found an expression that upper bounds $g(k,t)$, we can use it to upper bound $U(k,t)$ as well. This is the objective of the following lemma.

▶ **Lemma 29.** *Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, |R|_{max}, T, \gamma)$ be an MDP, and $\pi_{\boldsymbol{\theta}}$ be a policy parametrized by $\boldsymbol{\theta} \in \mathbb{R}^d$. Let $V_{\pi_{\boldsymbol{\theta}}}^{(t)}$ be its value function with horizon $t > 0$. For all non-negative integers k, t such that $\gamma \geq 2/t$, we have*

$$U(k,t) \leq \frac{2|R|_{\max}}{1-\gamma} \cdot (\gamma D t^2)^k.$$

Proof. By plugging in the bound derived in Lemma 28, we obtain directly that

$$U(k,t) = \sum_{t'=0}^{t-1} g(k,t) \leq \sum_{t'=0}^{t-1} \gamma^{t'} |R|_{\max} \sum_{\lambda \in \Lambda_k} \binom{k}{\lambda} \binom{|\lambda|}{\#\lambda} \binom{t'+1}{|\lambda|} \prod_{\ell \in \lambda} D_\ell. \tag{66}$$

First, for all $\lambda \in \Lambda_k$, we observe that the final product can be upper bounded as

$$\prod_{\ell \in \lambda} D_\ell = \prod_{\ell \in \lambda} (D_\ell^{1/\ell})^\ell \leq \prod_{\ell \in \lambda} D^\ell = D^k.$$

Next, we can swap the summations in Eq. (66), and after rewriting we obtain

$$U(k,t) \leq |R|_{\max} D^k \cdot \sum_{r=1}^{k} \sum_{\substack{k_1,\ldots,k_r \in \mathbb{N} \\ k_1+\cdots+k_r=k}} \binom{k}{k_1,\ldots,k_r} r! \cdot \sum_{t'=0}^{t-1} \gamma^{t'} \binom{t'+1}{r}. \tag{67}$$

We now focus on the final summation in the above expression. First, we observe that if $t < r$, then all the binomial coefficients evaluate to 0, and therefore the summation as a whole vanishes as well. Thus, the only terms in the above expression that are non-zero are those where $r \leq t$, which means that we can change the upper limit of summation in the outermost summation to $\min(k, t)$. We can take at least r factors of γ out, and as such obtain

$$\sum_{t'=0}^{t-1} \gamma^{t'} \binom{t'+1}{r} = \gamma^r \sum_{t'=0}^{t-r-1} \gamma^{t'} \binom{t'+r+1}{r} \leq \gamma^r \binom{t}{r} \sum_{t'=0}^{t-r-1} \gamma^{t'} \leq \frac{\gamma^r t^r}{(1-\gamma) r!}.$$

Plugging this expression back into Eq. (67) yields

$$U(k,t) \leq \frac{|R|_{\max} D^k}{1-\gamma} \cdot \sum_{r=1}^{\min(k,t)} (\gamma t)^r \sum_{\substack{k_1,\ldots,k_r \in \mathbb{N} \\ k_1+\cdots+k_r=k}} \binom{k}{k_1,\ldots,k_r} = \frac{|R|_{\max} D^k}{1-\gamma} \cdot \sum_{r=1}^{\min(k,t)} (\gamma t)^r r^k. \quad (68)$$

In the summation on the right-hand side, the last term is by far the biggest. We can show this crudely by observing that for all $a \geq 2$,

$$\frac{1}{n^k a^n} \sum_{r=1}^{n} r^k a^r = \sum_{r=1}^{n} \left(\frac{r}{n}\right)^k a^{r-n} \leq \sum_{r=1}^{n} \left(\frac{1}{a}\right)^{n-r} \leq \sum_{r=0}^{n-1} \left(\frac{1}{2}\right)^r \leq 2.$$

Thus, by setting $n = \min(k, t)$, and $a = \gamma t$, we obtain that

$$U(k,t) \leq \frac{2|R|_{\max}}{1-\gamma} \cdot (\gamma D t^2)^k.$$

This completes the proof. ◀

Lemma 26 then follows immediately from this lemma and Eq. (65) for $t = T$.

G Classical complexity of numerical gradient estimation

In this Appendix, we analyze the complexity of a classical numerical gradient estimation algorithm that relies on the same smoothness conditions of the value function as the quantum algorithm. More precisely, we show the following lemma:

▶ **Lemma 30.** *Let $\pi_{\boldsymbol{\theta}}$ be a parametrized policy that can be used to interact with an MDP, and that has a bounded smoothness parameter D, defined in Eq. (20). The gradient of the value function corresponding to this policy $\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}(s_0)$ can be evaluated to error ε in the ℓ_∞-norm, using*

$$\widetilde{\mathcal{O}}\left(d\left(\frac{DT^2|R|_{max}}{\varepsilon(1-\gamma)}\right)^2\right) \quad (69)$$

length-T episodes of interaction with the environment using a classical numerical gradient estimator.

To prove this lemma, we consider a central-difference method that, compared to a simple finite-difference method, can exploit more evaluations of a function f and bounds on its higher-order derivatives to evaluate $f'(x)$ with higher precision. We perform an error analysis of this method and calculate its query complexity for functions f that cannot be evaluated exactly but only through Monte Carlo estimation (such as value functions).

G.1 Central difference numerical differentiation

Suppose that we can evaluate a function $f : \mathbb{R} \to \mathbb{R}$ that is k times differentiable at some point $x \in \mathbb{R}$, with $f^{(k-1)}$ continuous on some interval around x. For $\delta \in \mathbb{R}$ such that $x + \delta$ is in this interval, Taylor's theorem (with the Lagrange formulation of the remainder) gives us:

$$f(x+\delta) = f(x) + f'(x)\delta + \frac{f''(x)}{2!}\delta^2 + \ldots + \frac{f^{(k-1)}(x)}{(k-1)!}\delta^{k-1} + \frac{f^{(k)}(\xi)}{k!}\delta^k \tag{70}$$

for a $\xi \in [x, x+\delta]$.
For $k = 2$ specifically, this expression becomes:

$$\begin{cases} f(x+\delta) = f(x) + f'(x)\delta + \frac{f''(\xi_+)}{2!}\delta^2, \\ f(x-\delta) = f(x) - f'(x)\delta + \frac{f''(\xi_-)}{2!}\delta^2, \end{cases} \tag{71}$$

for some $\xi_+, \xi_- \in [x, x+\delta]$.

The central difference method for numerical differentiation uses the following formula, derived from the expressions above:

$$f'(x) = \frac{f(x+\delta) - f(x-\delta)}{2\delta} + \frac{f''(\xi_+) - f''(\xi_-)}{4}\delta. \tag{72}$$

When a bound C_2 for f'' is known on the interval $[x-\delta, x+\delta]$, the remainder term can be bounded by

$$\left| \frac{f''(\xi_+) - f''(\xi_-)}{4}\delta \right| \leq \frac{C_2}{2}\delta. \tag{73}$$

The method can be generalized to use higher order derivatives (up to some $k \in \mathbb{N}$), such that $f'(x)$ is now of the form

$$f'(x) = \sum_{l=-m}^{m} \underbrace{\frac{a_l^{(2m)} f(x+l\delta)}{\delta}}_{f_l} + \underbrace{\sum_{l=-m}^{m} a_l^{(2m)} \frac{f^{(k)}(\xi_l)}{k!} l^k \delta^{k-1}}_{R_k} \tag{74}$$

for $m = \lfloor \frac{k-1}{2} \rfloor$ and where

$$a_l^{(2m)} = \begin{cases} 1 & \text{if } l = 0, \\ \frac{(-1)^{l+1}(m!)^2}{l(m+l)!(m-l)!} & \text{otherwise.} \end{cases} \tag{75}$$

G.2 Bounding the errors

When a bound C_k for $f^{(k)}$ is known on the interval $[x - m\delta, x + m\delta]$, the remainder term R_k can be bounded by

$$|R_k| \leq \left| \sum_{l=-m}^{m} a_l^{(2m)} l^k \right| \frac{C_k}{k!}\delta^{k-1} \leq 2m^k \frac{C_k}{k!}\delta^{k-1} \tag{76}$$

where the last inequality comes from Theorem 3.4 in [8].

In order for $|R_k| \leq \frac{\varepsilon}{2}$, we then need

$$\delta \leq \sqrt[k-1]{\frac{k!\varepsilon}{4m^k C_k}}. \tag{77}$$

We take

$$\delta = \frac{2}{e}\left(\frac{\varepsilon}{4C_k}\right)^{1/k} \tag{78}$$

$$\leq (2\pi k)^{\frac{1}{2k}}\frac{k}{me}\left(\frac{\varepsilon}{4C_k}\right)^{1/k} \tag{79}$$

$$\leq \left(\frac{\sqrt{2\pi k}(k/e)^k\varepsilon}{4m^kC_k}\right)^{1/k} \tag{80}$$

$$\leq \sqrt[k]{\frac{k!\varepsilon}{4m^kC_k}} \tag{81}$$

$$\leq \sqrt[k-1]{\frac{k!\varepsilon}{4m^kC_k}}. \tag{82}$$

Moreover, we are interested in the case where f cannot be evaluated exactly, but rather when we have access to random samples whose expectation value is $f(x)$ (and are bounded by C_0). If we want to estimate each f_l, $l = -m, \ldots, m$, to precision $\frac{\varepsilon}{2k}$ (such that we get their sum to precision $\frac{\varepsilon}{2}$), it is sufficient to estimate each $f(x + l\delta)$ to precision $\frac{\varepsilon\delta}{a_l^{(2m)}2k}$. From Lemma 25, we have that this requires a total number of queries (or samples) that scales as

$$\widetilde{\mathcal{O}}\left(\sum_{l=-m}^{m}\left(\frac{C_0ka_l^{(2m)}}{\varepsilon\delta}\right)^2\right) \leq \widetilde{\mathcal{O}}\left(\left(\frac{C_0k}{\varepsilon\delta}\right)^2\sum_{l=-m}^{m}\left|a_l^{(2m)}\right|\right) \tag{83}$$

$$\leq \widetilde{\mathcal{O}}\left(\left(\frac{C_0k}{\varepsilon\delta}\right)^2\left(1+2\sum_{l=1}^{m}\frac{1}{l}\right)\right) \tag{84}$$

$$\leq \widetilde{\mathcal{O}}\left(\left(\frac{C_0k}{\varepsilon\delta}\right)^2(3+2\log(m))\right) \tag{85}$$

$$\leq \widetilde{\mathcal{O}}\left(\left(\frac{C_0k}{\varepsilon\delta}\right)^2\right) \tag{86}$$

where the first two inequalities follow from $a_0^{(2m)} = 1$ (Eq. (75)) and $\left|a_l^{(2m)}\right| \leq \frac{1}{|l|}, l \in \{-m, \ldots, m\}\backslash\{0\}$ (Theorem 3.4 in [8]), and the third inequality follows from a simple upper bound on harmonic numbers.

Combining Eqs. (78) and (86), we find that a total of

$$\widetilde{\mathcal{O}}\left(\left(\frac{C_0k}{\varepsilon}\left(\frac{C_k}{\varepsilon}\right)^{1/k}\right)^2\right) \tag{87}$$

queries are sufficient to estimate $f'(x)$ to precision ε.

G.3 Application to value functions

In the case of value functions, we have $C_k = \frac{2|R|_{\max}}{1-\gamma}\left(DT^2\right)^k \forall k \in \mathbb{N}$ (see Lemma 26). Therefore, we can choose

$$k = \log\left(\frac{2|R|_{\max}}{\varepsilon(1-\gamma)}\right) \tag{88}$$

and use the identity $x^{1/\log(x)} = e^{\log(x)/\log(x)} = e$, such that, from Eq. (87):

$$\widetilde{\mathcal{O}}\left(d\left(\frac{DT^2|R|_{\max}}{\varepsilon(1-\gamma)}\right)^2\right) \tag{89}$$

queries are sufficient to estimate the gradient $\nabla_{\boldsymbol{\theta}} V_{\pi_{\boldsymbol{\theta}}}$ to ε precision in the ℓ_∞-norm. The multiplicative factor d comes from the fact that we need to estimate each of the d coordinates of the gradient independently.

Local Hamiltonians with No Low-Energy Stabilizer States

Nolan J. Coble
Joint Center for Quantum Information and Computer Science (QuICS), Department of Computer Science, University of Maryland, College Park, MD, USA

Matthew Coudron
Joint Center for Quantum Information and Computer Science (QuICS), Department of Computer Science, University of Maryland, College Park, MD, USA
National Institute of Standards and Technology, Gaithersburg, MD, USA

Jon Nelson
Joint Center for Quantum Information and Computer Science (QuICS), Department of Computer Science, University of Maryland, College Park, MD, USA

Seyed Sajjad Nezhadi
Joint Center for Quantum Information and Computer Science (QuICS), Department of Computer Science, University of Maryland, College Park, MD, USA

Abstract

The recently-defined No Low-energy Sampleable States (NLSS) conjecture of Gharibian and Le Gall [16] posits the existence of a family of local Hamiltonians where all states of low-enough constant energy do not have succinct representations allowing perfect sampling access. States that can be prepared using only Clifford gates (i.e. stabilizer states) are an example of sampleable states, so the NLSS conjecture implies the existence of local Hamiltonians whose low-energy space contains no stabilizer states. We describe families that exhibit this requisite property via a simple alteration to local Hamiltonians corresponding to CSS codes. Our method can also be applied to the recent NLTS Hamiltonians of Anshu, Breuckmann, and Nirkhe [4], resulting in a family of local Hamiltonians whose low-energy space contains neither stabilizer states nor trivial states. We hope that our techniques will eventually be helpful for constructing Hamiltonians which simultaneously satisfy NLSS and NLTS.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Hamiltonian complexity, Stabilizer codes, Low-energy states

Digital Object Identifier 10.4230/LIPIcs.TQC.2023.14

Related Version *Full Version*: https://arxiv.org/abs/2302.14755

Acknowledgements This paper is a contribution of NIST, an agency of the US government, and is not subject to US copyright. We thank Alexander Barg and Chinmay Nirkhe for helpful discussions.

1 Introduction

Local Hamiltonians are ubiquitous in quantum physics and quantum computation. From the physical perspective, Hamiltonians describe the dynamics and energy spectra of closed quantum systems, with "local" Hamiltonians corresponding to models where only a small number of particles can directly interact with each other. From the computational perspective, local Hamiltonians naturally generalize well-studied constraint satisfaction problems through the "local Hamiltonian problem", which asks about the complexity of approximating the ground-state energy of local Hamiltonians.

▶ **Definition** (LH-$\delta(n)$)**.** *A k-local Hamiltonian, $\mathcal{H} = \frac{1}{m}\sum_{i=1}^{m}\mathcal{H}_i$, is a sum of $m = \mathsf{poly}(n)$ Hermitian matrices, $\mathcal{H}_i \in \mathbb{C}^{2^n\times 2^n}$, where each $\mathcal{H}_i$ acts non-trivially on at most $k = \mathcal{O}(1)$ qubits*[1] *and has bounded spectral norm, $\|\mathcal{H}_i\| \leq 1$.*

Given a local Hamiltonian, $\mathcal{H}$, and two real numbers $a < b$ with $b - a > \delta(n)$, the ***local Hamiltonian problem with promise gap $\delta(n)$*** *is to decide if (1) there is a state with energy $\langle\psi_0|\mathcal{H}|\psi_0\rangle \leq a$ or (2) all states have energy $\langle\psi|\mathcal{H}|\psi\rangle \geq b$, given that one of these cases is true.*[2] *The value $\delta(n)$ is called the promise gap of the problem.*

LH is a natural quantum analogue of the NP-complete constraint satisfaction problem (CSP):[3] the local terms serve as *quantum* constraints on an n-qubit state, and the energy of a local term corresponds to how well the state satisfies that local constraint. The lowest energy state – or ground-state – of $\mathcal{H}$ is the state that optimally satisfies all of the local constraints.

It is straightforward to show that CSP is NP-complete for a promise gap $\delta(n) = 1/\mathsf{poly}(n)$, and the celebrated classical PCP Theorem [7, 8] shows that [surprisingly] CSP is still NP-complete when $\delta(n) = \Omega(1)$, a constant. Since LH is the quantum generalization of a CSP we can similarly ask whether it is complete for the class QMA, the quantum version of NP. Kitaev showed that LH is QMA-complete for $\delta(n) = 1/\mathsf{poly}(n)$ when he originally defined the class of QMA problems [19]. Perhaps the most important open question in quantum complexity theory is whether or not a quantum version of the PCP theorem holds. The "quantum PCP conjecture" [3, 1] states that LH with a constant promise gap is QMA-hard; the conjecture has thus far eluded proof.

As a possible step towards proving quantum PCP, Freedman and Hastings suggested the No Low-energy Trivial States (NLTS) conjecture which is implied by the quantum PCP conjecture (assuming NP $\neq$ QMA). A local Hamiltonian has the NLTS property if there is a constant strictly larger than the ground-state energy which lower bounds the energy of any state preparable in constant-depth ("trivial states"). The NLTS conjecture posits the existence of an NLTS Hamiltonian. This seemingly simpler problem remained open for nearly a decade until Anshu and Breuckmann solved the combinatorial version [5], followed shortly after by a complete proof by Anshu, Breuckmann, and Nirkhe [4]. They explicitly constructed an NLTS Hamiltonian using recently developed asymptotically-good quantum LDPC codes [20].

While the NLTS Theorem makes significant progress, there are still many other properties that a candidate Hamiltonian must satisfy in order to be QMA-hard with a constant promise gap. For instance, Gharibian and Le Gall defined the No Low-energy Sampleable States (NLSS) conjecture [16]. A state, $|\psi\rangle$ is "sampleable" if a classical computer can efficiently draw an $x \in \{0,1\}^n$ from the distribution defined by $p(x) = |\langle x|\psi\rangle|^2$ and can calculate all of the amplitudes, $\langle x|\psi\rangle$.[4] A local Hamiltonian has the NLSS property if there is a constant which lower-bounds the energy of every sampleable state. The NLSS conjecture posits the existence of an NLSS Hamiltonian, and Gharibian and Le Gall showed that unless MA = QMA the quantum PCP conjecture implies the NLSS conjecture.

---

[1] $\mathcal{H}_i = h_i \otimes \mathbb{I}_{2^{n-k}}$ where h_i is a $2^k \times 2^k$ Hermitian matrix and $\mathbb{I}_{2^{n-k}}$ is the $2^{n-k} \times 2^{n-k}$ identity matrix

[2] This is equivalent to deciding if $\mathcal{H}$ has an eigenvalue less than a or if all of the eigenvalues of $\mathcal{H}$ are larger than b, which is the more typical formulation of the problem.

[3] Technically LH is a generalization of the decision problem MAX-k-CSP.

[4] The more proper terminology, as in [16], would be that $|\psi\rangle$ has a succinct representation allowing perfect sampling access. We will not be directly addressing the NLSS conjecture, so we will use the term "sampleable" for brevity.

In this paper we examine a simplified version of the NLSS conjecture, where instead of sampleable states we consider stabilizer states. A stabilizer state is the unique state stabilized by a commuting subgroup of the Pauli group with size 2^n. Equivalently, stabilizer states are those states that can be prepared using only Clifford gates, i.e. Hadamard, Phase, and CNOT gates. We say that a local Hamiltonian has the No Low-energy Stabilizer States (NLCS)[5] property if there is a constant which lower-bounds the energy of any stabilizer state.[6] The Gottesman-Knill Theorem [18] shows that any stabilizer state can be efficiently sampled, so any NLSS Hamiltonian must also be an NLCS Hamiltonian. We show that a generic construction can be used to produce many NLCS Hamiltonians.

To prove the NLCS property for a particular local Hamiltonian one must show an explicit lower bound on the energy of all stabilizer states. Let $\mathcal{H} = \frac{1}{m}\sum \mathcal{H}_i$ be a local Hamiltonian and let $|\psi\rangle$ be an n-qubit state. The energy of any particular Hamiltonian term can be expressed as $\langle\psi|\mathcal{H}_i|\psi\rangle = \mathrm{Tr}\left[\psi_{A_i} h_i\right]$, where A_i is the set of qubits where $\mathcal{H}_i$ acts non-trivially, ψ_{A_i} is the reduced state of $|\psi\rangle$ on A_i, and h_i is the non-trivial part of $\mathcal{H}_i$. Suppose for simplicity that $|A_i| = k$ for all i. One particularly strong way to lower-bound the energy of $|\psi\rangle$ would be to "locally" bound each energy term. That is, prove that each $\mathrm{Tr}\left[\psi_{A_i} h_i\right]$ is lower-bounded by a constant. In general this is not an easy task. However, stabilizer states have a rather convenient property: we show in Claim 3 that if $|\psi\rangle$ is a stabilizer state, then every ψ_{A_i} is a convex combination of stabilizer states on k qubits. Thus, to lower-bound $\mathrm{Tr}\left[\psi_{A_i} h_i\right]$ for every n-qubit stabilizer state, $|\psi\rangle$, it is sufficient to lower-bound the quantity $\langle\zeta| h_i |\zeta\rangle$ for every *k-qubit* stabilizer state $|\zeta\rangle$.

This observation leads to a rather simple NLCS Hamiltonian. First, consider the Hamiltonian $\mathcal{H}_0 = \frac{1}{n}\sum |1\rangle\langle 1|_i$ where $|1\rangle\langle 1|_i$ is the projector to $|1\rangle$ on the i-th qubit and identity elsewhere. All of the local terms are the single-qubit projector $|1\rangle\langle 1|$. Clearly, we cannot lower-bound the energy of stabilizer states since $|0\rangle$ has energy 0. We can fix this, however, by instead considering a "conjugated" version of $\mathcal{H}_0$:

$$\tilde{\mathcal{H}}_0 \equiv \frac{1}{n}\sum_{i=1}^{n}\left(e^{i\frac{\pi}{8}Y}|1\rangle\langle 1| e^{-i\frac{\pi}{8}Y}\right)|_i,$$

which can alternatively be expressed as $\tilde{\mathcal{H}}_0 = (e^{i\frac{\pi}{8}Y})^{\otimes n}\mathcal{H}_0(e^{-i\frac{\pi}{8}Y})^{\otimes n}$. Each local term is the single-qubit projector $e^{i\frac{\pi}{8}Y}|1\rangle\langle 1| e^{-i\frac{\pi}{8}Y}$, and it is straightforward to calculate that every single-qubit stabilizer state has high energy under this local term. We give a self-contained proof that $\tilde{\mathcal{H}}_0$ is NLCS in Appendix B.

The quantum PCP conjecture not only implies the existence of NLTS/NLCS/NLSS Hamiltonians, but also the existence of *simultaneous* NLTS/NLCS/NLSS Hamiltonians. The process of conjugating a local Hamiltonian by a low-depth circuit conveniently preserves the NLTS property. That is, if $\mathcal{H}$ is NLTS and C is a constant-depth circuit, then $C^\dagger\mathcal{H}C$ is also NLTS (see Lemma 4).

We note that since $|1\rangle\langle 1| = \frac{1}{2}(\mathbb{I} - Z)$ the Hamiltonian $\mathcal{H}_0$ is an example of a *CSS Hamiltonian*, i.e. the local Hamiltonian terms are of the form $\frac{1}{2}(\mathbb{I} - P_i)$ where the P_i's are commuting X and Z type Pauli operators. As the Hamiltonian $\tilde{\mathcal{H}}_0$ is simply $\mathcal{H}_0$ conjugated

[5] The "C" in NLCS stands for Clifford, since states prepared by Clifford circuits and stabilizer states are equivalent.

[6] The existence of NLCS Hamiltonians has been suggested before as a direct consequence of the quantum PCP conjecture, for instance in [6]. We discuss the relationship of NLCS and more to the quantum PCP conjecture in Section 1.1.

by a depth-1 circuit $(e^{-i\frac{\pi}{8}Y})^{\otimes n}$ it may be natural to ask whether the same procedure can be done to the NLTS Hamiltonians from [4] as they are also CSS Hamiltonians. The main result of our paper is the following:

▶ **Theorem 1** (Informal version of Theorem 12). *Let $\mathcal{H}_{NLTS}$ be the NLTS local Hamiltonian from [4]. The local Hamiltonian given by $\tilde{\mathcal{H}}_{NLTS} \equiv (e^{i\frac{\pi}{8}Y})^{\otimes n}\mathcal{H}_{NLTS}(e^{-i\frac{\pi}{8}Y})^{\otimes n}$ satisfies both NLTS and NLCS.*

We prove Theorem 12 by exhibiting local lower bounds on the individual Hamiltonian terms. In particular, we show that if $h = \frac{1}{2}(\mathbb{I} - P^{\otimes k})$ is a k-local term where $P \in \{X, Z\}$, then

$$\langle\zeta| (e^{i\frac{\pi}{8}Y})^{\otimes k} h (e^{-i\frac{\pi}{8}Y})^{\otimes k} |\zeta\rangle \geq \sin^2(\pi/8)$$

for *every* k-qubit stabilizer state $|\zeta\rangle$, as long as k is odd. Combining this lower bound with the fact that the reduced state of a stabilizer state is a convex combination of stabilizer states, we have that conjugating a CSS Hamiltonian by $(e^{-i\frac{\pi}{8}Y})^{\otimes n}$ results in an NLCS Hamiltonian, at least in the case that many of the Hamiltonian terms act on an odd number of qubits.

The condition of odd weight is unfortunately a necessary condition of our local techniques: if k is even then there is always a k-qubit stabilizer state with $\langle\zeta_0| (e^{i\frac{\pi}{8}Y})^{\otimes k} h (e^{-i\frac{\pi}{8}Y})^{\otimes k} |\zeta_0\rangle = 0$. Nonetheless, we prove in Section 4 of the Full Version that there is an explicit NLTS Hamiltonian from [4] where every local term acts on an odd number of qubits. Since conjugating by a constant-depth circuit preserves NLTS, we ultimately have that $\tilde{\mathcal{H}}_{NLTS}$ satisfies both NLTS and NLCS.

1.1 Implications of the quantum PCP conjecture

We turn now to the question of what Hamiltonians are guaranteed to exist by the quantum PCP conjecture. The quantum PCP conjecture has two main formulations; we focus here on the gap amplification version. See [2] for a great survey on the conjecture.

▶ **Conjecture** (Conjecture 1.3 of [2]). *Let $\epsilon > 0$ be a constant.* LH-ϵ *is* QMA-hard *under quantum polynomial-time reductions.*

In other words, the conjecture says there is a worst-case local Hamiltonian whose ground state energy is QMA-hard to approximate within a constant. Approximating ground-state energies and finding ground states of local-Hamiltonians are of central importance to condensed matter theory and quantum simulation algorithms. If true, the quantum PCP conjecture says that there are some Hamiltonians whose ground-state energies we could never hope to approximate, let alone find their ground states.[7]

The key insight of [14] when they defined the NLTS conjecture was that some states have properties which allow their ground state energies to be calculated in a smaller complexity class than QMA. For a constant, k, we say that an n-qubit state, ρ, is ***k*-locally-approximable** if it has a polynomial-sized classical description from which every k-local reduced state, $\rho_A \equiv \mathrm{Tr}_{-A}[\rho]$ where $|A| \leq k$, can be approximated to inverse-polynomial precision in polynomial-time. Consider the following simple result:

▶ **Fact 2.** *Suppose $\mathcal{H} = \frac{1}{m}\sum_{i=1}^{m}\mathcal{H}_i$ is a k-local Hamiltonian and ρ is a k-locally approximable state. The energy of ρ under $\mathcal{H}$ can be approximated to inverse-polynomial precision in* NP.

[7] Unless, of course, one believes QMA ⊆ P or some other weakening of QMA.

Proof. Each $\mathcal{H}_i$ acts non-trivially on at most k qubits, $A_i \subset [n]$, so the energy of ρ for $\mathcal{H}_i$ is $\mathrm{Tr}[\rho\mathcal{H}_i] = \mathrm{Tr}\left[\rho_{A_i} h_i\right]$, where $h_i \equiv \mathrm{Tr}_{-A_i}[\mathcal{H}_i]$ is the non-trivial part of $\mathcal{H}_i$. Since $h_i \in \mathbb{C}^{2^k \times 2^k}$ and by assumption we can efficiently compute ρ_{A_i} to inverse-polynomial precision from the classical description of ρ, each $\mathrm{Tr}[\rho\mathcal{H}_i]$ can be brute-force approximated in polynomial-time. ◀

Trivial states are locally approximable. If $|\psi\rangle$ is a trivial state then there is a constant-depth circuit such that $|\psi\rangle = C|0\rangle^{\otimes n}$. For a set of k qubits, A, the only gates that contribute to ψ_A are those in the reverse-lightcone[8] of A. As the reverse-lightcone has size at most $k2^d$, a constant, only a constant number of gates from C are needed to brute-force approximate ψ_A. Thus, we can approximate local reduced states of $|\psi\rangle\langle\psi|$ from the classical description of C.

The assumption of being able to compute local reduced states also holds for stabilizer states. Suppose $|\psi\rangle$ is an n-qubit stabilizer state. Since $|\psi\rangle$ is a stabilizer state there are n independent and commuting Pauli operators $\{P_1, \ldots, P_n\}$ that stabilize $|\psi\rangle$. The list of these Pauli operators will serve as the classical description of $|\psi\rangle\langle\psi|$ from which local reduced states can be computed. The reduced state ψ_A can be written as

$$\psi_A = \frac{1}{2^k} \sum_{P \in G_A} P, \tag{1}$$

where G_A is the subgroup of the stabilizers of $|\psi\rangle$ which act non-trivially only on qubits in A. There are 4^k such Pauli group elements (ignoring phases) which we denote by $\mathcal{P}_A$. For $P \in \mathcal{P}_A$, one of $\pm P$ is in the stabilizer group of $|\psi\rangle$ if and only if P commutes with every stabilizer generator. So, we can determine the elements of G_A by brute-force checking which elements of $\mathcal{P}_A$ commute with every generator.[9] This computation can be done in polynomial-time since there are only a constant number of Pauli operators to check, so using Equation (1) we can compute ψ_A efficiently.

Thus, in addition to being an implication of NLSS, NLCS Hamiltonians are also implied by the quantum PCP conjecture assuming NP $\neq$ QMA: if every local Hamiltonian has a low-energy stabilizer state then the ground state energy could be computed in NP via Fact 2.

2 Preliminaries

For a natural number, n, we denote $[n] \equiv \{1, \ldots, n\}$. For a subset, $A \subseteq [n]$, we denote the set complement by $-A \equiv [n] \setminus A$ and the partial trace over the qubits in A by Tr_A. In particular, $\mathrm{Tr}_{-A}[|\psi\rangle\langle\psi|]$ denotes the local density matrix of $|\psi\rangle$ on the qubits in A.

2.1 States

Let $C = \{C_n\}$ be a countable family of quantum circuits consisting of one and two-qubits gates where each C_n acts on n qubits. If the depth of C_n is upper bounded by a function $d(n)$ for all n, then we say C is a **depth-$d(n)$** family of quantum circuits. If $d(n) = \mathcal{O}(1)$ then we say C is a depth-$\mathcal{O}(1)$ (or constant-depth) family of quantum circuits. Similarly, if $d(n) = \mathsf{poly}(n)$ then we say C is a depth-$\mathsf{poly}(n)$ (or polynomial-sized) family of quantum circuits.

[8] See Figure 1(a).

[9] It remains to determine whether $+P$ or $-P$ is in the stabilizer group. Although slightly more complicated, this can be done in polynomial-time *independent* of the weight of P.

The single-qubit **Pauli group** is the set $\mathcal{P}_1 \equiv \{i^\ell P \mid P \in \{\mathbb{I}, X, Y, Z\}, \ell \in \{0,1,2,3\}\}$, and the n-qubit Pauli group is its n-fold tensor-power, $\mathcal{P}_n = \bigotimes_{i\in[n]} \mathcal{P}_1$. For an element $S = P_1 \otimes \cdots \otimes P_n \in \mathcal{P}_n$, the **weight** of S is defined to be the number of qubits where P_i is not identity, i.e. $\mathrm{wt}(S) = \left|\{P_i \mid P_i \neq i^\ell \mathbb{I}\}\right|$. We denote the set of these qubits where S acts non-trivially by $\mathrm{N}(S) \subseteq [n]$.

The n-qubit **Clifford group**, $\mathcal{C}_n$, is the set of unitary operators which stabilize the Pauli group.It is well-known that $\mathcal{C}_n$ is generated by the set $\{\mathrm{H}, \mathrm{P}, \mathrm{CNOT}\}$, where H is the single-qubit Hadamard gate, P is the single-qubit phase gate, and CNOT is the two-qubit controlled-NOT gate. A **Clifford circuit** is defined to be any element of the Clifford group.

Let ψ be a [possibly mixed] state on n qubits and let $N \geq n$. If there is a quantum circuit, C, acting on N qubits such that $\psi = \mathrm{Tr}_N[C \left|0^N\right\rangle\!\left\langle 0^N\right| C^\dagger]$ then we say that C **prepares** ψ. ψ is said to be: a **trivial state** if there is a constant-depth quantum circuit preparing it, an [efficiently] **preparable state** if there is a polynomial-sized circuit preparing it, a **Clifford state** if there is a polynomial-sized Clifford circuit preparing it, and an **almost Clifford state** if there is a polynomial-sized quantum circuit containing Clifford + $\mathcal{O}(\log(n))$ T-gates preparing it. A pure state, $|\psi\rangle$ is said to be a **sampleable state** if (1) there is a classical algorithm exactly computing $\langle x|\psi\rangle$ for every $x \in \{0,1\}^n$ and (2) there is a classical algorithm that exactly samples $x \in \{0,1\}^n$ from the distribution $p(x) = |\langle x|\psi\rangle|^2$.

A **stabilizer group** is an abelian subgroup, G, of $\mathcal{P}_n$ not containing $-\mathbb{I}$. As a finite group, we can always find a list of mutually independent and commuting generators, $\mathcal{S} = \{S_1, \ldots, S_k\}$, of G. We will refer simply to the subgroup $\langle\mathcal{S}\rangle = G$ when this generating set is clear. Note that given a stabilizer group, there is a well-defined **stabilizer code** [17, 12, 13], $\mathcal{C}_\mathcal{S}$, which is the common $+1$ eigenspace of the operators in $\langle\mathcal{S}\rangle$.

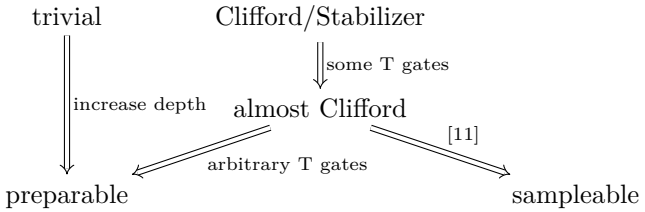
If a given stabilizer group has a generating set, $\mathcal{S}$, consisting of tensor products of only Pauli X and $\mathbb{I}$ or only Pauli Z and $\mathbb{I}$, then we say $\mathcal{C}_\mathcal{S}$ is a **CSS code** and that $\mathcal{S}$ generates a CSS code.

The **stabilizer group** of a pure state, $|\psi\rangle$, is the subgroup of the Pauli group defined by $\mathrm{Stab}(|\psi\rangle) \equiv \{P \in \mathcal{P}_n \mid P|\psi\rangle = |\psi\rangle\}$. We say that a $P \in \mathrm{Stab}(|\psi\rangle)$ **stabilizes** $|\psi\rangle$. Note that $\mathrm{Stab}(|\psi\rangle)$ is an abelian subgroup of the Pauli group not containing $-\mathbb{I}$, and so it is a valid stabilizer group as before.

A pure state, $|\psi\rangle$, is said to be a **stabilizer state** if $|\mathrm{Stab}(|\psi\rangle)| = 2^n$, or equivalently, if there are n independent Pauli operators that stabilize $|\psi\rangle$. We note that $|\psi\rangle\!\langle\psi| = \frac{1}{2^n}\sum_{g\in G} g$ where $G = \mathrm{Stab}(|\psi\rangle)$.

A mixed state, ψ, is said to be a stabilizer state if ψ is a convex combination of pure stabilizer states, i.e. $\psi = \sum_j p_j |\varphi_j\rangle\!\langle\varphi_j|$ where each $|\varphi_j\rangle$ is a pure stabilizer state on n qubits, $\sum_j p_j = 1$, and $p_j \geq 0$.

All of the states defined here are related to one another via the following:



(2)

By definition of the Clifford group, stabilizer states and Clifford states are equivalent for pure states. We will interchangeably use the terms "stabilizer state" and "Clifford state" even for mixed states, which is motivated by the following result:

▷ Claim 3. If ψ is a Clifford state, then it is also a stabilizer state.

A proof can be found in Appendix A.1. Claim 3 says that the reduced state of a pure stabilizer state is a convex combination of pure stabilizer states on the subsystem. This is essential in our energy lower bound arguments: To lower-bound the energy of all n-qubit stabilizer states for a k-local term of the Hamiltonian, $\mathcal{H}_i$, it is sufficient to lower-bound the energy of all k-qubit stabilizer states for the non-trivial part of $\mathcal{H}_i$.

2.2 Hamiltonians

A ***k*-local Hamiltonian**, $\mathcal{H}^{(n)}$, is a Hermitian operator on the space of n qubits, $(\mathbb{C}^2)^{\otimes n}$, which can be written as a sum $\mathcal{H}^{(n)} = \frac{1}{m}\sum_{i=1}^{m}\mathcal{H}_i$, where each $\mathcal{H}_i$ is a Hermitian matrix acting non-trivially on only k qubits and with spectral norm $\|\mathcal{H}_i\| \leq 1$. A **family of *k*-local Hamiltonians**, $\{\mathcal{H}^{(n)}\}$, is a countable set of k-local Hamiltonians indexed by system size, n, where $k = \mathcal{O}(1)$ and $m = \mathsf{poly}(n)$. We will often use the term "local Hamiltonian" to mean a family of k-local Hamiltonians.

The **ground-state energy** of $\mathcal{H}$ is $E_0 \equiv \min_\rho \mathrm{Tr}[\rho\mathcal{H}]$, where the minimization is taken over all n-qubit mixed states. $\mathcal{H}$ is said to be **frustration-free** if $E_0 = 0$. A state, ψ, is said to be a **ground state** of $\mathcal{H}$ if $\mathrm{Tr}[\psi\mathcal{H}] = E_0$. A state, ψ, is said to be an **ϵ-low-energy state** of $\mathcal{H}$ if $\mathrm{Tr}[\psi\mathcal{H}] < E_0 + \epsilon$. If $\psi = |\psi\rangle\langle\psi|$ is a pure state, this condition simplifies to $\langle\psi|\,\mathcal{H}\,|\psi\rangle < \lambda_{\min}(\mathcal{H}) + \epsilon$, where $\lambda_{\min}(\mathcal{H})$ is the smallest eigenvalue of $\mathcal{H}$. For frustration-free Hamiltonians this is equivalent to $\langle\psi|\,\mathcal{H}\,|\psi\rangle < \epsilon$. All of the Hamiltonians we consider will be frustration-free.

For $S \in \mathcal{P}_n$, we denote the orthogonal projector to the $+1$ eigenspace of S by Π_S, i.e. $\Pi_S \equiv \frac{\mathbb{I}-S}{2}$. Since Π_S acts non-trivially on only $\mathrm{wt}(S)$ qubits, we can write $\Pi_S = \Pi_S\,|_{\mathrm{N}(S)} \otimes \mathbb{I}_{[n]\setminus\mathrm{N}(S)}$.

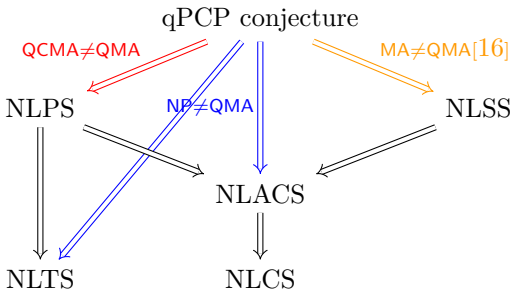
Given a stabilizer group, $\langle\mathcal{S}\rangle$, with generating set $\mathcal{S}$, the **stabilizer Hamiltonian** associated to $\mathcal{S}$ is $\mathcal{H}_\mathcal{S} \equiv \frac{1}{|\mathcal{S}|}\sum_{S\in\mathcal{S}}\Pi_S$. If each qubit is acted on non-trivially by at most $\mathrm{wt}(\mathcal{S})$ elements of $\mathcal{S}$, then $\mathcal{H}_\mathcal{S}$ is a $\mathrm{wt}(\mathcal{S})$-local Hamiltonian. If $\mathcal{C}$ is the Stabilizer code associated with $\mathcal{S}$, then every $|\psi\rangle \in \mathcal{C}$ is a zero-energy state of $\mathcal{H}_\mathcal{S}$. In particular, $\mathcal{H}_\mathcal{S}$ is frustration-free with ground-state space $\mathcal{C}$. If $\mathcal{S}$ generates a CSS code then we say $\mathcal{H}_\mathcal{S}$ is a **CSS Hamiltonian**.

If $\{\langle\mathcal{S}_n\rangle \mid \langle\mathcal{S}_n\rangle \leq \mathcal{P}_n\}$ is a countable family of stabilizer groups then the **family of stabilizer (or CSS) Hamiltonians** associated with $\{\mathcal{S}_n\}$ is $\{\mathcal{H}_{\mathcal{S}_n}\}$. This will be a family of local Hamiltonians when: (1) each qubit is acted on non-trivially by at most $\mathrm{wt}(\mathcal{S}_n)$ elements of $\mathcal{S}_n$, (2) $\mathrm{wt}(\mathcal{S}_n) = \mathcal{O}(1)$, and (3) $|\mathcal{S}_n| = \Theta(n)$. Such families, $\{\langle\mathcal{S}_n\rangle\}$, of stabilizer groups correspond to quantum LDPC code families.

For each of the states in the previous section we can consider an analogue of NLTS.

▶ **Definition.** *A family of k-local Hamiltonians, $\{\mathcal{H}^{(n)}\}$, is said to have the **ϵ-NLXS** property if for all sufficiently large n, $\mathcal{H}^{(n)}$ has no ϵ-low-energy states of type X. The family, $\{\mathcal{H}^{(n)}\}$, is said to have the **NLXS** property if it is ϵ-NLXS for some constant ϵ.*

The following implications between the NLXS theorems/conjectures and quantum PCP conjecture hold. A complexity inequality next to an arrow denotes an implication that holds if the separation is true, e.g. if the quantum PCP conjecture is true and $\mathsf{MA} \neq \mathsf{QMA}$, then NLSS is true.



(3)

The relationships between each of the NLXS results are implicitly given by Diagram 2. Trivial states, stabilizer states, and almost Clifford states are all examples of locally-approximable states, so they following from the quantum PCP conjecture via Fact 2. The implication of NLSS was given by Gharibian and Le Gall when they originally defined NLSS [16]. The implication of NLPS is well-known: if every local Hamiltonian has a low-energy preparable state, $C\left|0\right\rangle^{\otimes n}$, then given the classical description of C a quantum prover could simply prepare the state and measure its energy. This would put LH-$\epsilon \in \mathsf{QCMA}$, implying $\mathsf{QMA} = \mathsf{QCMA}$ if the quantum PCP conjecture is true.

For a family of k-local Hamiltonians, $\{\mathcal{H}^{(n)}\}$, and a family, $C = \{C_n\}$, of depth-$\mathcal{O}(1)$ quantum circuits, we define the ***C*-rotated version** of $\{\mathcal{H}^{(n)}\}$ as $\{\mathcal{H}^{(n)}\}^C \equiv \{{C_n}^\dagger \mathcal{H}^{(n)} C_n\}$. This is still a family of local Hamiltonians, albeit with a possibly different k than the original Hamiltonian. This is because the only qubits that interact non-trivially with a single Hamiltonian term, $C^\dagger \mathcal{H}_i C$, are those qubits in the reverse-lightcone of the qubits acted on by $\mathcal{H}_i$. The number of qubits in the reverse-lightcone of a single qubit grows exponentially in the depth of a circuit, which is still constant since C is constant-depth. See Figure 1 for an example of this. When $C = V^{\otimes n}$ is the tensor-product of a single-qubit gate, V, we will use the term "V-rotated" as opposed to "$V^{\otimes n}$-rotated".

The utility of considering a C-rotated Hamiltonian is that in addition to preserving locality, the NLTS property is also preserved.

▶ **Lemma 4.** *If $\{\mathcal{H}^{(n)}\}$ is a family of ϵ_0-NLTS local Hamiltonians and $C = \{C_n\}$ is a family of constant-depth circuits, then $\{\mathcal{H}^{(n)}\}^C$ is also ϵ_0-NLTS.*

Proof. Suppose that $\{\mathcal{H}^{(n)}\}^C$ is not NLTS. By definition, for every $\epsilon > 0$ there is an n and constant-depth circuit $U_{\epsilon,n}$ such that $U_{\epsilon,n}\left|0\right\rangle^{\otimes n}$ is an ϵ-low-energy state of $C_n^\dagger \mathcal{H}^{(n)} C_n$, i.e.

$$\left\langle 0\right|^{\otimes n} U_{\epsilon,n}^\dagger C_n^\dagger \mathcal{H}^{(n)} C_n U_{\epsilon,n} \left|0\right\rangle^{\otimes n} < \lambda_{\min}(C_n^\dagger \mathcal{H}^{(n)} C_n) + \epsilon.$$

Since C_n is a unitary operator the minimum eigenvalues of $\mathcal{H}^{(n)}$ and $C_n^\dagger \mathcal{H}^{(n)} C_n$ are equal. Defining $\left|\psi_{\epsilon_0,n}\right\rangle \equiv C_n U_{\epsilon_0,n}\left|0\right\rangle^{\otimes n}$ we have

$$\left\langle \psi_{\epsilon_0,n}\right| \mathcal{H}^{(n)} \left|\psi_{\epsilon_0,n}\right\rangle < \lambda_{\min}(\mathcal{H}^{(n)}) + \epsilon_0,$$

i.e. $\left|\psi_{\epsilon_0,n}\right\rangle$ is an ϵ_0-low-energy state of $\mathcal{H}^{(n)}$. Since $C_n U_{\epsilon_0,n}$ is a constant-depth circuit this implies that $\mathcal{H}^{(n)}$ has a low-energy trivial state, contradicting the assumption of ϵ_0-NLTS. ◀

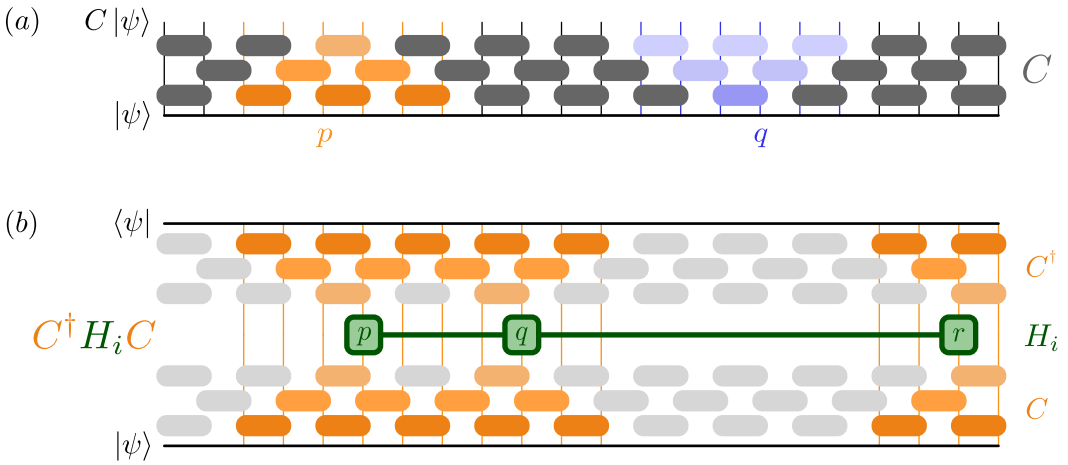


Figure 1 (a) Consider a constant-depth circuit, C. The [blue] highlighted gates on the right of the figure represent the "lightcone" of qubit q, i.e. the set of gates that can be traced back to q. The [orange] highlighted gates on the left of the figure represent the gates in the "reverse-lightcone" of qubit p, i.e. the gates that will ultimately affect p.
(b) Consider a single k-local Hamiltonian term, $\mathcal{H}_i$, that acts only on qubits p, q, and r. When conjugating $\mathcal{H}_i$ with C, any gate not in the reverse-lightcone of one of p, q, or r will cancel with its inverse. The number of qubits in the reverse-lightcone of any one qubit is $\leq 2^d$ where d is the depth of C, so $C^\dagger \mathcal{H}_i C$ will be at most $k2^d$-local. Note that we have only drawn a 2D geometrically-local circuit here, whereas this upper bound holds for a constant-depth circuit with arbitrary connectivity.

3 NLCS from CSS codes

We will show that rotating by the tensor product of a single-qubit gate is sufficient to turn most CSS Hamiltonians into NLCS Hamiltonians, including the quantum Tanner codes used in [4]. In particular, we consider the single-qubit gate $D \equiv e^{-i\frac{\pi}{8}Y}$ and rotate a CSS Hamiltonian by $D^{\otimes n}$. For a local Hamiltonian, $\mathcal{H}^{(n)}$, we will denote its ***D*-rotated version** by $\tilde{\mathcal{H}}^{(n)} \equiv D^{\dagger\otimes n}\mathcal{H}^{(n)}D^{\otimes n}$. We denote the ***D*-rotated projector** associated with a Pauli element $S \in \mathcal{P}_n$ by $\tilde{\Pi}_S \equiv D^{\otimes n}\Pi_S D^{\dagger\otimes n}$. By definition, we have $\tilde{\Pi}_S = \tilde{\Pi}_S|_{\mathrm{N}(S)} \otimes \mathbb{I}_{[n]\setminus \mathrm{N}(S)}$, where $\tilde{\Pi}_S|_{\mathrm{N}(S)} = D^{\otimes \mathrm{wt}(S)}\Pi_S|_{\mathrm{N}(S)}\, D^{\dagger\otimes \mathrm{wt}(S)}$. Note that we have not explicitly included D in the above notations since D will refer exclusively to $e^{-i\frac{\pi}{8}Y}$, throughout.

We have the following result:

▶ **Theorem 5.** *Let $\{\mathcal{H}_{\mathcal{S}_n}\}$ be a family of CSS Hamiltonians associated with a family of quantum (CSS) LDPC codes, $\{\langle \mathcal{S}_n\rangle\}$. Suppose for every n a constant fraction, $\alpha > 0$, of the generators $S \in \mathcal{S}_n$ have odd weight. Then $\{\tilde{\mathcal{H}}_{\mathcal{S}_n}\}$ is a family of NLCS Hamiltonians.*

We prove this by giving local lower bounds on the energies of D-rotated projectors associated with CSS generators. As a technical requirement, these lower bounds only hold when the weight of a generator is odd.

Recall that, up to a permutation of the qubits, the generators of a CSS code can be written as either $\bar{X} \otimes \mathbb{I}$ or $\bar{Z} \otimes \mathbb{I}$, where $\bar{X} \equiv X^{\otimes k}$ and $\bar{Z} \equiv Z^{\otimes k}$. First consider what happens to the projectors $\Pi_{\bar{X}}$ and $\Pi_{\bar{Z}}$ when rotating by D:

▷ Claim 6.

$$\tilde{\Pi}_{\bar{X}} = \frac{\mathbb{I} - \mathrm{H}^{\otimes k}}{2}, \qquad \tilde{\Pi}_{\bar{Z}} = \frac{\mathbb{I} - (-X\,\mathrm{H}\,X)^{\otimes k}}{2}.$$

These identities are derived in Appendix A.2. The local lower bounds will be a result of the following:

▶ **Lemma 7.** *If k is odd, then for every k-qubit stabilizer state, $|\eta\rangle$, we have $\left|\langle\eta|\,\mathrm{H}^{\otimes k}\,|\eta\rangle\right| \leq \frac{1}{\sqrt{2}}$. On the other hand, if k is even then there exists a k-qubit stabilizer state, $|\eta_0\rangle$, with $\langle\eta_0|\,\mathrm{H}^{\otimes k}\,|\eta_0\rangle = 1$.*

The proof will use the following result on the geometry of stabilizer states:

▶ **Fact 8** (Corollary 3 of [15])**.** *Let $|\zeta\rangle, |\xi\rangle$ be two stabilizer states. If $|\langle\zeta|\xi\rangle| \neq 1$, then $|\langle\zeta|\xi\rangle| \leq \frac{1}{\sqrt{2}}$.*

Proof of Lemma 7. Since H is a Clifford gate, $\mathrm{H}^{\otimes k}\,|\eta\rangle$ is a stabilizer state. We will show that $\left|\langle\eta|\,\mathrm{H}^{\otimes k}\,|\eta\rangle\right| \neq 1$ in the case of odd k, which by Fact 8 will imply the bound.

Recall that $|\eta\rangle\langle\eta| = \frac{1}{|G|}\sum_{g\in G} g$, where $G \equiv \mathrm{Stab}(|\eta\rangle)$. We have two cases:

(1) (Every $S \in G$ contains an $\mathbb{I}$ or a Y in some position) In this case, we calculate

$$\begin{aligned}\langle\eta|\,\mathrm{H}^{\otimes k}\,|\eta\rangle &= \mathrm{Tr}\Big[|\eta\rangle\langle\eta|\,\mathrm{H}^{\otimes k}\Big],\\ &= \frac{1}{|G|}\sum_{g\in G}\mathrm{Tr}\Big[g\,\mathrm{H}^{\otimes k}\Big],\\ &= \frac{1}{|G|}\sum_{g\in G}\prod_{i\in[k]}\mathrm{Tr}[g_i\,\mathrm{H}],\\ &= 0,\end{aligned}$$

where the last line follows since $g_j \in \{\mathbb{I}, Y\}$ for some j, and $\mathrm{Tr}[\mathrm{H}] = \mathrm{Tr}[Y\,\mathrm{H}] = 0$.

(2) (There is an $S \in G$ which consists of only X's and Z's) Consider the case when k is odd. Since $\mathrm{wt}(S) = k$, S contains either (1) an odd number of X's and an even number of Z's or (2) an even number of X's and an odd number of Z's. We focus on the former situation; the latter is similar.

Note that $\left|\langle\eta|\,\mathrm{H}^{\otimes k}\,|\eta\rangle\right| = 1$ if and only if $\mathrm{H}^{\otimes k}\,|\eta\rangle$ and $|\eta\rangle$ have the same stabilizer group. Since S stabilizes $|\eta\rangle$, $\mathrm{H}^{\otimes k}\,S\,\mathrm{H}^{\otimes k}$ stabilizes $\mathrm{H}^{\otimes k}\,|\eta\rangle$. We know how H conjugates Pauli operators: $X \mapsto Z$, $Z \mapsto X$, and $Y \mapsto -Y$. By assumption, S has an odd number of X's and an even number of Z's, so $\mathrm{H}^{\otimes k}\,S\,\mathrm{H}^{\otimes k}$ will have an even number of X's and an odd number of Z's. Therefore, we have that $S\cdot(\mathrm{H}^{\otimes k}\,S\,\mathrm{H}^{\otimes k}) = -(\mathrm{H}^{\otimes k}\,S\,\mathrm{H}^{\otimes k})\cdot S$, which implies S and $\mathrm{H}^{\otimes k}\,S\,\mathrm{H}^{\otimes k}$ cannot both be elements of the same stabilizer group. Hence, $\mathrm{Stab}(|\eta\rangle) \neq \mathrm{Stab}(\mathrm{H}^{\otimes k}\,|\eta\rangle)$ and $\left|\langle\eta|\,\mathrm{H}^{\otimes k}\,|\eta\rangle\right| \neq 1$.

Since in both cases $\left|\langle\eta|\,\mathrm{H}^{\otimes k}\,|\eta\rangle\right| \neq 1$, by Fact 8 we must have that $\left|\langle\eta|\,\mathrm{H}^{\otimes k}\,|\eta\rangle\right| \leq \frac{1}{\sqrt{2}}$ when k is odd. We note that the above proof will not work for even k, since it can be the case that all stabilizers have an even number of X's and Z's (or both odd). In this case $\mathrm{H}^{\otimes k}$ will be in the normalizer of G, and the two stabilizer groups may be equal.

We can easily find an example with even k where no non-trivial upper bound can be found. Note that $|\Phi^+\rangle \equiv \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is a $+1$ eigenstate of $\mathrm{H}^{\otimes 2}$, so for even k define $|\eta_0\rangle \equiv |\Phi^+\rangle^{\otimes k/2}$. ◀

We can now prove the local lower bound on odd-weight CSS generators.

▶ **Lemma 9.** *For every k-qubit stabilizer state, $|\eta\rangle$, $\langle\eta|\,\tilde{\Pi}_{\bar{X}}\,|\eta\rangle \geq c_k$ and $\langle\eta|\,\tilde{\Pi}_{\bar{Z}}\,|\eta\rangle \geq c_k$, where $c_k = 0$ if k is even and $c_k = \sin^2(\frac{\pi}{8})$ if k is odd.*

Proof. Let $|\eta\rangle$ be a k-qubit stabilizer state. We first consider $\langle\eta|\,\tilde{\Pi}_{\bar{X}}\,|\eta\rangle$:

$$\langle\eta|\,\tilde{\Pi}_{\bar{X}}\,|\eta\rangle \equiv \langle\eta|\,D^{\dagger\otimes k}\left(\frac{\mathbb{I}-\bar{X}}{2}\right)D^{\otimes k}\,|\eta\rangle\,, \tag{4}$$

(By Claim 6)
$$= \langle\eta|\,\frac{\mathbb{I}-\mathrm{H}^{\otimes k}}{2}\,|\eta\rangle \tag{5}$$

$$= \frac{1}{2}\left(1-\langle\eta|\,\mathrm{H}^{\otimes k}\,|\eta\rangle\right). \tag{6}$$

The bound follows from Lemma 7, since $\sin^2(\frac{\pi}{8}) = \frac{1}{2}(1-\frac{1}{\sqrt{2}})$.

For $\langle\eta|\,\tilde{\Pi}_{\bar{Z}}\,|\eta\rangle$, we have:

$$\langle\eta|\,\tilde{\Pi}_{\bar{Z}}\,|\eta\rangle \equiv \langle\eta|\,D^{\dagger\otimes k}\left(\frac{\mathbb{I}-\bar{Z}}{2}\right)D^{\otimes k}\,|\eta\rangle\,, \tag{7}$$

(By Claim 6)
$$= \langle\eta|\,\frac{\mathbb{I}-(-X\,\mathrm{H}\,X)^{\otimes k}}{2}\,|\eta\rangle \tag{8}$$

$$= \frac{1}{2}\left(1-\langle\eta|\,(-X\,\mathrm{H}\,X)^{\otimes k}\,|\eta\rangle\right), \tag{9}$$

$$= \frac{1}{2}\left(1-(-1)^k\,\langle\zeta|\,\mathrm{H}^{\otimes k}\,|\zeta\rangle\right) \tag{10}$$

where $|\zeta\rangle \equiv X^{\otimes k}\,|\eta\rangle$ is another stabilizer state since $X = X^\dagger$ is in the Clifford group. The bound follows again from Lemma 7. ◀

Lemma 9 implies the following lower bound for n-qubit stabilizer states.

▶ **Lemma 10.** *Let $S \in \mathcal{P}_n$ be a tensor product of only Pauli X and $\mathbb{I}$ or only Pauli Z and $\mathbb{I}$. Denote $k = \mathrm{wt}(S)$. For every n-qubit stabilizer state, $|\eta\rangle$, $\langle\eta|\,\tilde{\Pi}_S\,|\eta\rangle \geq c_k$.*

Proof. Recall that $\tilde{\Pi}_S = \tilde{\Pi}_S\,|_{\mathrm{N}(S)} \otimes \mathbb{I}_{[n]\setminus\mathrm{N}(S)}$, so

$$\langle\eta|\,\tilde{\Pi}_S\,|\eta\rangle = \mathrm{Tr}\left[\eta_{\mathrm{N}(S)}\tilde{\Pi}_S\,|_{\mathrm{N}(S)}\right], \tag{11}$$

where $\eta_{\mathrm{N}(S)} \equiv \mathrm{Tr}_{-\mathrm{N}(S)}[|\eta\rangle\langle\eta|]$ is the reduced state of $|\eta\rangle$ on $\mathrm{N}(S) \subset [n]$. Since $\eta_{\mathrm{N}(S)}$ is the reduced state of a Clifford state, by Claim 3 there are pure stabilizer states on k qubits, $\{|\eta_j\rangle\}$ such that $\eta_{\mathrm{N}(S)} = \sum_j p_j\,|\eta_j\rangle\langle\eta_j|$. The lower bound follows by applying Lemma 9 to each $\langle\eta_j|\,\tilde{\Pi}_S\,|_{\mathrm{N}(S)}\,|\eta_j\rangle$. ◀

We can now prove Theorem 5.

▶ **Theorem 5.** *Let $\{\mathcal{H}_{\mathcal{S}_n}\}$ be a family of CSS Hamiltonians associated with a family of quantum (CSS) LDPC codes, $\{\langle\mathcal{S}_n\rangle\}$. Suppose for every n a constant fraction, $\alpha > 0$, of the generators $S \in \mathcal{S}_n$ have odd weight. Then $\{\tilde{\mathcal{H}}_{\mathcal{S}_n}\}$ is a family of NLCS Hamiltonians.*

Proof. By definition, $\tilde{\mathcal{H}}_{\mathcal{S}_n} = \frac{1}{|\mathcal{S}_n|}\sum_{S\in\mathcal{S}_n}\tilde{\Pi}_S$ where $\tilde{\Pi}_S$ is the D-rotated projector associated with $S \in \mathcal{S}_n$. Let ψ be a stabilizer state on n qubits. We will directly lower-bound the energy of ψ.

By definition, $\psi = \sum_j p_j\,|\varphi_j\rangle\langle\varphi_j|$, where each $|\varphi_j\rangle$ is a pure stabilizer state on n qubits. We have:

$$\mathrm{Tr}\left[\psi\tilde{\mathcal{H}}_{\mathcal{S}_n}\right] = \sum_j p_j \left\langle\varphi_j\right| \tilde{\mathcal{H}}_{\mathcal{S}_n} \left|\varphi_j\right\rangle, \tag{12}$$

$$= \frac{1}{|\mathcal{S}_n|} \sum_{S\in\mathcal{S}_n} \sum_j p_j \left\langle\varphi_j\right| \tilde{\Pi}_S \left|\varphi_j\right\rangle, \tag{13}$$

(By Lemma 10) $$\geq \frac{1}{|\mathcal{S}_n|} \sum_{S\in\mathcal{S}_n} c_{\mathrm{wt}(S)} \sum_j p_j, \tag{14}$$

(Definition of c_k) $$= \frac{1}{|\mathcal{S}_n|} \sum_{S\in\mathcal{S}_n:\mathrm{wt}(S),\ \mathrm{odd}} \sin^2\left(\frac{\pi}{8}\right), \tag{15}$$

$$= \alpha \sin^2\left(\frac{\pi}{8}\right), \tag{16}$$

where the last line follows by assumption $\alpha|\mathcal{S}_n|$ terms of $\mathcal{S}_n$ have odd weight. Since this holds for all stabilizer states, ψ, we have that $\{\tilde{\mathcal{H}}_{\mathcal{S}_n}\}$ is ϵ-NLCS with $\epsilon = \alpha\sin^2(\frac{\pi}{8}) = \Omega(1)$. ◀

We now turn to our main result, the existence of a simultaneous NLTS and NLCS family of local Hamiltonians. Recall the NLTS result of [4]:

▶ **Theorem** (Theorem 5 of [4], simplified)**.** *There exists a constant $\epsilon_0 > 0$ and an explicit family of CSS Hamiltonians associated with a family of quantum LDPC codes, $\{\langle\mathcal{S}_n\rangle\}$, which is ϵ_0-NLTS.*

In order to use our Theorem 5, we require that a constant fraction of the stabilizer generators in $\mathcal{S}_n$ have an odd weight. It is not immediately clear that this would be true for the quantum Tanner codes from [20]. However, we have the following result:

▷ Claim 11. There exists an explicit family of CSS codes satisfying the conditions of Theorem 5 of [4] such that every stabilizer-generator has odd weight.

Section 4 of the Full Version is dedicated to proving Claim 11. The proof is rather straightforward and relies on the random choice of local codes in the construction of quantum Tanner codes. Essentially, we show that the local codes of the two component *classical* Tanner codes of a quantum Tanner code can be chosen such that all of the parity-checks of the global codes have odd weight. This implies that all of the stabilizer-generators of the quantum Tanner code also have odd weight.

With Claim 11, we are now prepared to prove the main result of our paper.

▶ **Theorem 12.** *Let $\{\mathcal{H}^{(n)}\}$ be the family of CSS Hamiltonians from Claim 11. The D-rotated version, $\{\tilde{\mathcal{H}}^{(n)}\}$, is a family of simultaneous NLTS and NLCS local Hamiltonians.*

Proof. Since $\{\mathcal{H}^{(n)}\}$ satisfies the conditions of Theorem 5 of [4] it is a valid local Hamiltonian, and it is ϵ_0-NLTS for some constant $\epsilon_0 > 0$. Since $D^{\otimes n}$ is a depth-$\mathcal{O}(1)$ circuit by Lemma 4 the D-rotated family $\{\tilde{\mathcal{H}}^{(n)}\}$ is also ϵ_0-NLTS.

By Claim 11, all of the stabilizer terms of $\mathcal{H}^{(n)}$ have odd weight for every n. Thus, by Theorem 5 $\{\tilde{\mathcal{H}}^{(n)}\}$ is ϵ_1-NLCS for $\epsilon_1 \equiv \sin^2(\frac{\pi}{8})$. Letting $\epsilon' \equiv \min\{\epsilon_0, \epsilon_1\}$, we have that $\{\tilde{\mathcal{H}}^{(n)}\}$ is both ϵ'-NLTS and ϵ'-NLCS. ◀

4 Future work

(1) The most immediate problem raised by this work is to show that rotating arbitrary CSS Hamiltonians by $(e^{-i\frac{\pi}{8}Y})^{\otimes n}$ yields NLCS Hamiltonians. We have shown this when a constant fraction of the stabilizer generators have odd weight, which is a technical requirement of our proof technique. Nonetheless, we believe all $e^{-i\frac{\pi}{8}Y}$-rotated CSS Hamiltonians are NLCS. A first step would be to show this for $\mathcal{H} \equiv \frac{1}{n}\sum |11\rangle\langle 11|_{i,i+1} = \frac{1}{n}\sum \frac{1}{2}(\mathbb{I} - Z_iZ_{i+1})$, which has only even weight stabilizer generators.

(2) NLACS Hamiltonians are an implication of either NLSS or the quantum PCP conjecture together with NP $\neq$ QMA (see Diagram 3), so we believe they exist. In Appendix B we give a self-contained proof that the simple D-rotated zero Hamiltonian, $\tilde{\mathcal{H}}_0 = \frac{1}{n}\sum (e^{i\frac{\pi}{8}Y}|1\rangle\langle 1|e^{-i\frac{\pi}{8}Y})_i$, is NLCS, and in Appendix B.1, we give a sharp lower-bound on the energy of states produced by Clifford + 1 T gate under $\tilde{\mathcal{H}}_0$. We also conjecture a sharp lower-bound on the energy for states prepared by Clifford + t T gates, for any $t \leq n$.

(3) We hope that our techniques may lead to local Hamiltonians which satisfy NLSS. Consider the zero Hamiltonian, $\mathcal{H}_0 = \frac{1}{n}\sum |1\rangle\langle 1|_i$, and a family of Haar-random low-depth circuits, $C = \{C_n\}$. The unique ground-state of the local Hamiltonian $C\mathcal{H}_0C^\dagger$ is exactly $C|0^n\rangle$,[10] which is not sampleable (as defined in Section 2) unless P = #P [9, 21]. We hope that the same is true for states of low-enough constant energy, but new techniques would be necessary to show this. If true, $C\mathcal{H}_0C^\dagger$ would be an NLSS Hamiltonian unless P = #P. Analogously to our result for simultaneous NLTS and NLCS, one may hope that rotating arbitrary CSS Hamiltonians by random low-depth circuits could also yield simultaneous NLTS and NLSS. However, there are many unresolved prerequisites needed to show this. For example, for a CSS Hamiltonian, $\mathcal{H}$, every ground-state of $C\mathcal{H}C^\dagger$ has the form $C|\psi\rangle$ for a codestate $|\psi\rangle$. It is not a fortiori true that applying a random low-depth circuit to codestates of a CSS code will result in a state that is not sampleable, so it is not clear that even the ground-space of such a Hamiltonian is not sampleable.

(4) It is important to note that the technique of rotating Hamiltonians by a constant-depth circuit, while potentially useful for establishing NLSS, seemingly cannot provide certain other prerequisites of the quantum PCP conjecture. For example, Fact 2 says that the energies of locally-approximable states can be computed in NP, and so the quantum PCP conjecture implies the following (assuming NP $\neq$ QMA):

▶ **Conjecture 13** (No Low-energy Locally-approximable States (NLLS)). *There exists a family of local Hamiltonians, $\mathcal{H}^{(n)}$, and a constant $\epsilon > 0$ such that all ϵ-low-energy states of $\mathcal{H}^{(n)}$ are not locally-approximable.*

A closely-related conjecture ("no low-lying classically-evaluatable states" conjecture) was very recently stated in [22].[11] Rotating by a constant-depth circuit preserves the NLLS property in the same way that it preserves the NLTS property, thus ruling out the use of rotating Hamiltonians in solving the NLLS conjecture.

[10] Note that we typically denote rotating by C as $C^\dagger\mathcal{H}C$, not $C\mathcal{H}C^\dagger$. We have swapped the order here so that the ground state is $C|0^n\rangle$, as opposed to $C^\dagger|0^n\rangle$.

[11] Note that these conjectures would not imply LH-$\epsilon \notin$ NP as it would not rule out Hamiltonians whose ground-state energies have indirect NP-witnesses. [10] constructs such witnesses for certain commuting Hamiltonians.

Furthermore, for any CSS Hamiltonian rotated by a constant-depth circuit, which includes every construction considered in this paper, the local Hamiltonian problem is contained in NP. To see this, note that every C-rotated CSS Hamiltonian has a ground state of the form $C^\dagger |\varphi\rangle$ for some stabilizer state $|\varphi\rangle$. Such states are locally-approximable since the local density matrices can be efficiently calculated by using a combination of the local density matrix calculation for trivial states and stabilizer states.

References

1 Dorit Aharonov, Itai Arad, Zeph Landau, and Umesh Vazirani. The Detectability Lemma and Quantum Gap Amplification, November 2008. arXiv:0811.3412 [cond-mat, physics:quant-ph]. URL: http://arxiv.org/abs/0811.3412.

2 Dorit Aharonov, Itai Arad, and Thomas Vidick. The Quantum PCP Conjecture, September 2013. arXiv: 1309.7495. URL: http://arxiv.org/abs/1309.7495.

3 Dorit Aharonov and Tomer Naveh. Quantum NP - A Survey, October 2002. arXiv:quant-ph/0210077. URL: http://arxiv.org/abs/quant-ph/0210077.

4 Anurag Anshu, Nikolas Breuckmann, and Chinmay Nirkhe. NLTS Hamiltonians from good quantum codes, June 2022. number: arXiv:2206.13228 arXiv:2206.13228 [cond-mat, physics:quant-ph]. URL: http://arxiv.org/abs/2206.13228.

5 Anurag Anshu and Nikolas P. Breuckmann. A construction of Combinatorial NLTS, June 2022. number: arXiv:2206.02741 arXiv:2206.02741 [quant-ph]. URL: http://arxiv.org/abs/2206.02741.

6 Anurag Anshu and Chinmay Nirkhe. Circuit Lower Bounds for Low-Energy States of Quantum Code Hamiltonians. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:22, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2022.6.

7 S. Arora and S. Safra. Probabilistic checking of proofs; a new characterization of NP. In *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*, pages 2–13, Pittsburgh, PA, USA, 1992. IEEE. doi:10.1109/SFCS.1992.267824.

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

9 Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15(2):159–163, October 2018. doi:10.1038/s41567-018-0318-2.

10 S. Bravyi and M. Vyalyi. Commutative version of the k-local Hamiltonian problem and common eigenspace problem, December 2004. arXiv:quant-ph/0308021. URL: http://arxiv.org/abs/quant-ph/0308021.

11 Sergey Bravyi and David Gosset. Improved Classical Simulation of Quantum Circuits Dominated by Clifford Gates. *Physical Review Letters*, 116(25):250501, June 2016. doi:10.1103/PhysRevLett.116.250501.

12 A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane. Quantum Error Correction and Orthogonal Geometry. *Physical Review Letters*, 78(3):405–408, January 1997. doi:10.1103/PhysRevLett.78.405.

13 A.R. Calderbank, E.M. Rains, P.M. Shor, and N.J.A. Sloane. Quantum error correction via codes over GF(4). *IEEE Transactions on Information Theory*, 44(4):1369–1387, July 1998. doi:10.1109/18.681315.

14 M. H. Freedman and M. B. Hastings. Quantum Systems on Non-k-Hyperfinite Complexes: A Generalization of Classical Statistical Mechanics on Expander Graphs, July 2013. arXiv:1301.1363 [quant-ph]. URL: http://arxiv.org/abs/1301.1363.

15 Héctor J. García, Igor L. Markov, and Andrew W. Cross. On the Geometry of Stabilizer States, November 2017. arXiv:1711.07848 [quant-ph]. URL: http://arxiv.org/abs/1711.07848.

16 Sevag Gharibian and François Le Gall. Dequantizing the Quantum Singular Value Transformation: Hardness and Applications to Quantum Chemistry and the Quantum PCP Conjecture. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 19–32, June 2022. arXiv:2111.09079 [quant-ph]. doi:10.1145/3519935.3519991.

17 Daniel Gottesman. Class of quantum error-correcting codes saturating the quantum Hamming bound. *Physical Review A*, 54(3):1862–1868, September 1996. doi:10.1103/PhysRevA.54.1862.

18 Daniel Gottesman. The Heisenberg Representation of Quantum Computers, July 1998. arXiv:quant-ph/9807006. URL: http://arxiv.org/abs/quant-ph/9807006.

19 A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, Rhode Island, May 2002. doi:10.1090/gsm/047.

20 Anthony Leverrier and Gilles Zémor. Quantum Tanner codes, April 2022. number: arXiv:2202.13641 arXiv:2202.13641 [quant-ph]. URL: http://arxiv.org/abs/2202.13641.

21 Ramis Movassagh. Quantum supremacy and random circuits, 2019. doi:10.48550/ARXIV.1909.06210.

22 Jordi Weggemans, Marten Folkertsma, and Chris Cade. Guidable Local Hamiltonian Problems with Implications to Heuristic Ansätze State Preparation and the Quantum PCP Conjecture, 2023. doi:10.48550/ARXIV.2302.11578.

A Omitted proofs

A.1 Mixed Clifford states

▶ **Definition 14.** *Let G be a stabilizer group, $P = P_1 \otimes \cdots \otimes P_n \in \mathcal{P}_n$ be any Pauli operator, and $A \subseteq [n]$ be any subset of n qubits. We define the set $G_{A,P}$ to be*

$$G_{A,P} \equiv \left\{ g_A \;\middle|\; g \in G,\ g_j = P_j \text{ for all } j \notin A \right\},$$

where g_A denote the restriction of g to A (note that g_A acts on $|A|$ qubits, not n qubits).

$G_{A,P}$ can be thought of as all of the elements of G which are equal to P outside of the subset A, though we consider the restriction of these elements to A only (including overall phases). By abuse of notation we will denote $G_{i,P} \equiv G_{\{i\},P}$ and $G_{-A,P} \equiv G_{[n]\setminus A,P}$ for $i \in [n]$. We denote the special case of $G_{A,\mathbb{I}}$ by G_A. $G_A \equiv \{g_A \mid g \in G \text{ and } \mathrm{N}(g) \subseteq A\} \cup \{\mathbb{I}_A\}$ is the set of all elements in G which act non-trivially only on qubits in A.

Claim 3 is immediate from the following two well-known facts.

▶ **Fact 15.** *Let $G \leq \mathcal{P}_n$ be a stabilizer group and $\mathcal{C}$ the associated codespace. $\frac{1}{|G|}\sum_{g\in G} g$ is the projector onto $\mathcal{C}$. If $|G| = 2^n$, then $\frac{1}{2^n}\sum_{g\in G} g = |\psi\rangle\langle\psi|$, where $|\psi\rangle$ is the stabilizer state associated with G. Otherwise, $|G| = 2^{n-r}$ for $r > 0$ and there are 2^r logical basis states of $\mathcal{C}$. Let $\{|\bar{x}\rangle\}$ denote the logical computational basis states for $\mathcal{C}$. Then*

$$\frac{1}{2^{n-r}} \sum_{g\in G} g = \sum_{x\in\mathbb{F}_2^r} |\bar{x}\rangle\langle\bar{x}| \,.$$

▶ **Fact 16.** *Suppose $|\psi\rangle$ is a stabilizer state on N qubits with stabilizer group G and let A be a subset of the qubits of size n. By Fact 15 we can write $|\psi\rangle\langle\psi| = \frac{1}{2^N}\sum_{g\in G} g$. The local state on A, $\psi \equiv \mathrm{Tr}_{-A}[|\psi\rangle\langle\psi|]$, is equal to*

$$\psi = \frac{1}{2^n} \sum_{\hat{g}\in G_A} \hat{g}.$$

▷ Claim 3. If ψ is a Clifford state, then it is also a stabilizer state.

Proof. By definition, there is a pure Clifford state $|\psi\rangle$ on $N \geq n$ qubits and a subset A of n qubits such that $\psi = \mathrm{Tr}_{-A}[|\psi\rangle\langle\psi|]$. Let $G \equiv \mathrm{Stab}(|\psi\rangle)$, and let G_A be defined as in Fact 16. By definition, G_A is an abelian subgroup of $\mathcal{P}_n$ not containing $-\mathbb{I}$, and so it is a valid stabilizer group. Let $|G_A| = 2^{n-r}$. We have

$$\text{(By Fact 16)} \qquad \psi = \frac{1}{2^r 2^{n-r}} \sum_{\hat{g} \in G_A} \hat{g}, \tag{17}$$

$$\text{(By Fact 15)} \qquad = \frac{1}{2^r} \sum_{x \in \mathbb{F}_2^r} |\bar{x}\rangle\langle\bar{x}| \,. \tag{18}$$

Since each $|\bar{x}\rangle\langle\bar{x}|$ is a stabilizer state on n qubits and $\sum_{x \in \mathbb{F}_2^r} \frac{1}{2^r} = 1$, the statement is proven. ◁

A.2 Rotated projectors

Return to Claim 6.

▷ Claim 6.

$$\tilde{\Pi}_{\bar{X}} = \frac{\mathbb{I} - \mathrm{H}^{\otimes k}}{2}, \qquad \tilde{\Pi}_{\bar{Z}} = \frac{\mathbb{I} - (-X\,\mathrm{H}\,X)^{\otimes k}}{2}.$$

Proof. We will show that $D^\dagger X D = \mathrm{H}$ and $D^\dagger Z D = -X\,\mathrm{H}\,X$. As $\Pi_{\bar{X}} \equiv (\frac{1}{2})(\mathbb{I} - X)$ and $\Pi_{\bar{Z}} \equiv (\frac{1}{2})(\mathbb{I} - Z)$, the result follows.

$$\begin{aligned}
D^\dagger X D &= \left(\cos\left(\frac{\pi}{8}\right)\mathbb{I} + \sin\left(\frac{\pi}{8}\right) ZX\right) X \left(\cos\left(\frac{\pi}{8}\right)\mathbb{I} + \sin\left(\frac{\pi}{8}\right) XZ\right), \\
&= \left(\cos\left(\frac{\pi}{8}\right) X + \sin\left(\frac{\pi}{8}\right) Z\right) \left(\cos\left(\frac{\pi}{8}\right)\mathbb{I} + \sin\left(\frac{\pi}{8}\right) XZ\right), \\
&= \cos^2\left(\frac{\pi}{8}\right) X + \sin\left(\frac{\pi}{8}\right)\cos\left(\frac{\pi}{8}\right) Z + \sin\left(\frac{\pi}{8}\right)\cos\left(\frac{\pi}{8}\right) Z - \sin^2\left(\frac{\pi}{8}\right) X, \\
&= \cos\left(\frac{\pi}{4}\right) Z + \sin\left(\frac{\pi}{4}\right) X, \\
&= \frac{1}{\sqrt{2}}(Z + X), \\
&= \mathrm{H}\,.
\end{aligned}$$

$$\begin{aligned}
D^\dagger Z D &= \left(\cos\left(\frac{\pi}{8}\right)\mathbb{I} + \sin\left(\frac{\pi}{8}\right) ZX\right) Z \left(\cos\left(\frac{\pi}{8}\right)\mathbb{I} + \sin\left(\frac{\pi}{8}\right) XZ\right), \\
&= \left(\cos\left(\frac{\pi}{8}\right) Z - \sin\left(\frac{\pi}{8}\right) X\right) \left(\cos\left(\frac{\pi}{8}\right)\mathbb{I} + \sin\left(\frac{\pi}{8}\right) XZ\right), \\
&= \cos^2\left(\frac{\pi}{8}\right) Z - \sin\left(\frac{\pi}{8}\right)\cos\left(\frac{\pi}{8}\right) X - \sin\left(\frac{\pi}{8}\right)\cos\left(\frac{\pi}{8}\right) X - \sin^2\left(\frac{\pi}{8}\right) Z, \\
&= \cos\left(\frac{\pi}{4}\right) Z - \sin\left(\frac{\pi}{4}\right) X, \\
&= \frac{1}{\sqrt{2}}(Z - X), \\
&= -X\,\mathrm{H}\,X.
\end{aligned}$$

◁

B A simple NLCS Hamiltonian

The goal of this section is to demonstrate the existence of a simple family of NLCS Hamiltonians.

▶ **Definition 17.** *The* ***zero Hamiltonian****,* $\mathcal{H}_0^{(n)}$ *is defined as*

$$\mathcal{H}_0^{(n)} \equiv \frac{1}{n}\sum_{i=1}^{n} |1\rangle\langle 1|_i \otimes \mathbb{I}_{[n]\setminus\{i\}}\,.$$

Note that $\mathcal{H}_0^{(n)}|x\rangle = \frac{|x|}{n}|x\rangle$ *for all* $x \in \mathbb{F}_2^n$*. In particular, the unique ground state of* $\mathcal{H}_0^{(n)}$ *is* $|0\rangle^{\otimes n}$ *with energy* 0*. For* $n = 1$ *we have* $\mathcal{H}_0^{(1)} = |1\rangle\langle 1|$*, so we can write the zero Hamiltonian on* n *qubits as*

$$\mathcal{H}_0^{(n)} \equiv \frac{1}{n}\sum_{i=1}^{n} \mathcal{H}_0^{(1)} \otimes \mathbb{I}_{[n]\setminus\{i\}}\,.$$

▶ Remark. Define a set of stabilizer generators, $\mathcal{S}_n \equiv \{Z_1, \ldots, Z_n\}$ where Z_i is a Pauli Z on qubit i and identity elsewhere. The zero Hamiltonian is the CSS Hamiltonian associated with $\langle \mathcal{S}_n \rangle$, since $|1\rangle\langle 1| = \frac{\mathbb{I}-Z}{2}$. The results of this section are a direct corollary of the results in Section 3. ⌟

Let $D \equiv e^{-i\frac{\pi}{8}Y}$. We define the D-rotated zero Hamiltonian as

$$\tilde{\mathcal{H}}_0^{(n)} \equiv \frac{1}{n}\sum_{i=1}^{n} \tilde{\mathcal{H}}_0^{(1)} \otimes \mathbb{I}_{[n]\setminus\{i\}},$$

where $\tilde{\mathcal{H}}_0^{(1)} = D^\dagger |1\rangle\langle 1| D$. We will prove the D-rotated zero Hamiltonian is NLCS by demonstrating a simple lower bound on the energy of stabilizer states for each local term. Since the reduced state of every stabilizer state is a convex combination of stabilizer states by Claim 3, these "local" lower bounds imply a global lower bound for all stabilizer states. We have the following local energy bound. Note that

▶ **Lemma 18.** *If* $|\eta\rangle$ *is a single-qubit stabilizer state, then* $\langle\eta|\tilde{\mathcal{H}}_0^{(1)}|\eta\rangle \geq \sin^2(\frac{\pi}{8})$.

Proof. By definition $\tilde{H}_0^{(1)} = D^\dagger|1\rangle\langle 1|D$, so $\langle\eta|\tilde{H}_0^{(1)}|\eta\rangle = |\langle 1|D|\eta\rangle|^2$. As

$$D = \cos\left(\frac{\pi}{8}\right)\mathbb{I} - i\sin\left(\frac{\pi}{8}\right)Y = \cos\left(\frac{\pi}{8}\right)\mathbb{I} + \sin\left(\frac{\pi}{8}\right)XZ,$$

we have

$$D = \begin{bmatrix} \cos\left(\frac{\pi}{8}\right) & -\sin\left(\frac{\pi}{8}\right) \\ \sin\left(\frac{\pi}{8}\right) & \cos\left(\frac{\pi}{8}\right) \end{bmatrix}.$$

There are only six single-qubit stabilizer states and it is easy to verify that the minimum value of $|\langle 1|D|\eta\rangle|^2$ is $\sin^2(\frac{\pi}{8})$. ◀

▶ **Corollary 19.** *If* η *is a single-qubit mixed stabilizer state, then* $\mathrm{Tr}\left[\eta\tilde{\mathcal{H}}_0^{(1)}\right] \geq \sin^2(\frac{\pi}{8})$.

Proof. By definition, $\eta = \sum_j p_j |\varphi_j\rangle\langle\varphi_j|$, where each $|\varphi_j\rangle$ is a pure stabilizer state on a single qubit. The lower bound follows by applying Lemma 18 to each $|\varphi_j\rangle$. ◀

We now have the following global lower bound.

▶ **Lemma 20.** *If $|\eta\rangle$ is an n-qubit stabilizer state, then $\langle\eta|\,\tilde{\mathcal{H}}_0^{(n)}\,|\eta\rangle \geq \sin^2(\frac{\pi}{8})$.*

Proof. By definition, $\tilde{\mathcal{H}}_0^{(n)} = \frac{1}{n}\sum_{i=1}^{n} \tilde{\mathcal{H}}_0^{(1)}\,|_i \otimes \mathbb{I}_{[n]\setminus\{i\}}$, so

$$\langle\eta|\,\tilde{\mathcal{H}}_0^{(n)}\,|\eta\rangle = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Tr}\Big[\eta_i \tilde{\mathcal{H}}_0^{(1)}\Big],$$

where $\eta_i \equiv \mathrm{Tr}_{-i}[|\eta\rangle\langle\eta|]$ is the reduced state of $|\eta\rangle$ on qubit i. Since η_i is the reduced density matrix of a Clifford state, by Claim 3 it is also a stabilizer state. The bound follows by applying Corollary 19 to each term in the summation. ◀

▶ **Proposition 21.** *$\{\tilde{\mathcal{H}}_0^{(n)}\}$ is a family of NLCS Hamiltonians.*

Proof. By definition, $\psi = \sum_j p_j\,|\varphi_j\rangle\langle\varphi_j|$, where each $|\varphi_j\rangle$ is a pure stabilizer state on n qubits. The lower bound follows by applying Lemma 20 to each $|\varphi_j\rangle$. Thus, every n-qubit stabilizer state has energy at least $\sin^2(\frac{\pi}{8})$ with respect to $\tilde{\mathcal{H}}_0^{(n)}$, which implies $\tilde{\mathcal{H}}_0^{(n)}$ is ϵ-NLCS with $\epsilon = \sin^2(\frac{\pi}{8})$. ◀

B.1 Towards NLACS

There are several notions of how "non-Clifford" a state is, the number of T gates being a common one. The notion we consider here is the number of arbitrary Pauli-rotation gates, $e^{i\theta P}$ for $\theta \in [0, 2\pi)$ and $P \in \mathcal{P}_n$, as it encapsulates the T gate count.[12]

▶ **Lemma 22.** *Let C be a quantum circuit on n-qubits containing polynomially many Clifford gates and at most t arbitrary Pauli-rotation gates, $e^{i\theta_j P'_j}$. There exist t Pauli operators, $\{P_j\} \subset \mathcal{P}_n$ and a stabilizer state $|\varphi\rangle$ such that*

$$C\,|0\rangle^{\otimes n} = \prod_{j\in[t]} \left[e^{i\theta_j P_j}\right] |\varphi\rangle\,, \tag{19}$$

where by convention $C\,|0\rangle^{\otimes n} = |\varphi\rangle$ if $t = 0$.

Proof. By definition we can decompose C as

$$C = C_t e^{i\theta_t P'_t} C_{t-1} \ldots e^{i\theta_2 P'_2} C_1 e^{i\theta_1 P'_1} C_0, \tag{20}$$

where each C_ℓ is a Clifford circuit.

For every $j \in [t]$ we have $e^{i\theta_j P'_j} = \cos(\theta_j)\,\mathbb{I} + i\sin(\theta_j) P'_j$. Since Clifford gates normalize the Pauli group, for every Clifford circuit, C', and every Pauli operator, $P' \in \mathcal{P}_n$, there is another Pauli operator, $P'' \in \mathcal{P}_n$, such that $C'(\cos\theta\,\mathbb{I} + i\sin\theta P') = (\cos\theta\,\mathbb{I} + i\sin\theta P'')C'$. Thus, we can move each Clifford circuit, C_ℓ, past all of the Pauli-rotation gates by changing only the individual Pauli operators via the conjugation relations of C_ℓ.

Ultimately, we can rewrite C as

$$C = e^{i\theta_t P_t} \ldots e^{i\theta_2 P_2} e^{i\theta_1 P_1} C_t \ldots C_1 C_0, \tag{21}$$

for t Pauli operators, $\{P_t\}$, as desired. ◀

[12] The T gate is equal to $\mathrm{T} = \cos\left(\frac{\pi}{8}\right)\mathbb{I} + i\sin\left(\frac{\pi}{8}\right) Z = e^{i\frac{\pi}{8}Z}$.

Proposition 21 shows that the D-rotated zero Hamiltonian, $\tilde{\mathcal{H}}_0 = \frac{1}{n}\sum \left(D^\dagger |1\rangle\langle 1| D\right)_i$, is $\sin^2\left(\frac{\pi}{8}\right)$-NLCS. It is natural to question if $\tilde{\mathcal{H}}_0$ is also ϵ-NLACS for some appropriate constant ϵ. In this section we will prove an explicit lower-bound on all states prepared by Clifford gates + at most 1 Pauli-rotation gate:

$$\langle\psi| \tilde{\mathcal{H}}_0^{(n)} |\psi\rangle \geq \left(1 - \frac{1}{n}\right) \sin^2\left(\frac{\pi}{8}\right). \tag{22}$$

In fact, there is numerical evidence suggesting the following lower bound for an arbitrary number of Pauli-rotation gates, though we have been unable to prove it analytically:

▶ **Conjecture 23.** *Let $|\psi\rangle$ be an n-qubit state prepared by a Clifford circuit plus at most t Pauli-rotation gates. For the D-rotated zero-Hamiltonian, $\tilde{\mathcal{H}}_0^{(n)}$, the energy of $|\psi\rangle$ is lower-bounded as*

$$\langle\psi| \tilde{\mathcal{H}}_0^{(n)} |\psi\rangle \geq \left(1 - \frac{t}{n}\right) \sin^2\left(\frac{\pi}{8}\right). \tag{23}$$

In particular, if there is a constant $\beta \in [0,1)$ such that $t \leq \beta n$ for all sufficiently large n, then the energy of $|\psi\rangle$ is lower-bounded by $(1-\beta)\sin^2\left(\frac{\pi}{8}\right) > 0$, a constant.

By Lemma 22, the most general such state is a stabilizer state with t Pauli-rotation gates applied to it and no intermediate circuits between them. The intuition behind Conjecture 23 is that the only way to reduce the energy of a stabilizer state is to "undo" one of the D gates conjugating the Hamiltonian. For instance, to produce a state with sub-constant energy one could apply $n - o(n)$ D gates to $|0\rangle^{\otimes n}$.

We note also that is in unclear what, if any, similar lower bound could be shown for an arbitrary D-rotated CSS Hamiltonian (as considered in Theorem 5). We leave this as an open problem, as well. For now, we consider the case of $t = 1$ for the D-rotated zero Hamiltonian.

First, recall the following definition.

▶ **Definition 14.** *Let G be a stabilizer group, $P = P_1 \otimes \cdots \otimes P_n \in \mathcal{P}_n$ be any Pauli operator, and $A \subseteq [n]$ be any subset of n qubits. We define the set $G_{A,P}$ to be*

$$G_{A,P} \equiv \left\{ g_A \;\middle|\; g \in G,\ g_j = P_j \text{ for all } j \notin A \right\},$$

where g_A denote the restriction of g to A (note that g_A acts on $|A|$ qubits, not n qubits).

The following lemma gives an explicit description of the local density matrices of states with at most 1 Pauli-rotation gate.

▶ **Lemma 24.** *Let $|\psi\rangle = e^{i\theta P}|\varphi\rangle$ for $P \in \mathcal{P}_n$, $\theta \in [0, 2\pi)$, and let $|\varphi\rangle$ be a stabilizer state with $G \equiv \mathrm{Stab}(|\varphi\rangle)$. For $A \subset [n]$ we can write $\psi_A \equiv \mathrm{Tr}_{-A}[|\psi\rangle\langle\psi|]$ as*

$$\psi_A = \frac{1}{2^{|A|}} \sum_{\hat{g} \in G_A} \left(\cos^2(\theta)\hat{g} + \sin^2(\theta) P_A \hat{g} P_A \right) + \frac{1}{2^{|A|}} \sum_{g' \in G_{A,P}} i \sin(\theta)\cos(\theta)[P_A, g']. \tag{24}$$

The left part of this expression can be thought of as the stabilizer part of ψ_A, as it is the convex combination of two stabilizer states, and the right hand part can be thought of as the non-stabilizer part, as it equals zero if $P \in G$ or if $P_A = \mathbb{I}$.

Proof. Since $|\varphi\rangle$ is a stabilizer state there is a stabilizer group G with $|G| = 2^n$ such that $|\varphi\rangle\langle\varphi| = \frac{1}{2^n}\sum_{g\in G} g$. Using the exponential of Pauli matrices we have

$$\psi = \frac{1}{2^n}\sum_{g\in G}(\cos(\theta)\,\mathbb{I} + i\sin(\theta)P)g(\cos(\theta)\,\mathbb{I} - i\sin(\theta)P), \tag{25}$$

$$= \frac{1}{2^n}\sum_{g\in G}\cos^2(\theta)g + \sin^2(\theta)PgP + i\sin(\theta)\cos(\theta)Pg - i\sin(\theta)\cos(\theta)gP, \tag{26}$$

$$= \frac{1}{2^n}\sum_{g\in G}\Big(\cos^2(\theta)g + \sin^2(\theta)PgP\Big) + \frac{1}{2^n}\sum_{g\in G}\Big(i\sin(\theta)\cos(\theta)(Pg - gP)\Big). \tag{27}$$

Consider tracing out all qubits outside of the set A. The only Pauli group element with nonzero trace is $\mathbb{I}$, which has trace 2. For the left term in Equation (27), we have

$$\frac{1}{2^n}\sum_{g\in G}\Big(\cos^2(\theta)\,\mathrm{Tr}_{-A}[g] + \sin^2(\theta)\,\mathrm{Tr}_{-A}[PgP]\Big) \tag{28}$$

$$= \frac{1}{2^n}\sum_{g\in G}\Big(\cos^2(\theta)g_A\prod_{j\in[n]\setminus A}\mathrm{Tr}[g_j] + \sin^2(\theta)P_Ag_AP_A\prod_{j\in[n]\setminus A}\mathrm{Tr}[P_jg_jP_j]\Big), \tag{29}$$

$$= \frac{1}{2^n}\sum_{g\in G}\Big(\cos^2(\theta)g_A + \sin^2(\theta)P_Ag_AP_A\Big)\Big(\prod_{j\in[n]\setminus A}\mathrm{Tr}[g_j]\Big), \tag{30}$$

$$= \frac{1}{2^{|A|}}\sum_{\hat{g}\in G_A}\Big(\cos^2(\theta)\hat{g} + \sin^2(\theta)P_A\hat{g}P_A\Big), \tag{31}$$

where the last line follows since only those $g \in G$ which are identity outside of A will have nonzero trace, and the product of the individual traces when non-zero is $2^{n-|A|}$.

Similarly, for the right term in Equation (27) we have

$$\frac{1}{2^n}\sum_{g\in G}\Big(i\sin(\theta)\cos(\theta)\,\mathrm{Tr}_{-A}[Pg - gP]\Big), \tag{32}$$

$$= \frac{1}{2^n}\sum_{g\in G}\Big(i\sin(\theta)\cos(\theta)[P_A, g_A]\Big)\Big(\prod_{j\in[n]\setminus A}\mathrm{Tr}[P_jg_j]\Big), \tag{33}$$

$$= \frac{1}{2^{|A|}}\sum_{g'\in G_{A,P}} i\sin(\theta)\cos(\theta)[P_A, g'], \tag{34}$$

where the last line follows again since the trace will be non-zero only if $g_j = P_j$ for all $j \notin A$. ◀

▶ **Lemma 25.**

$$\langle\psi|\,\tilde{\mathcal{H}}_0^{(n)}\,|\psi\rangle \geq \left(1 - \frac{1}{n}\right)\sin^2\left(\frac{\pi}{8}\right). \tag{35}$$

Proof. By Lemma 22 there is a Pauli operator, P, and an n-qubit Clifford state $|\varphi\rangle$ such that $|\psi\rangle = e^{i\theta P}|\varphi\rangle$. Let $G \equiv \mathrm{Stab}(|\varphi\rangle)$.

Recall that by definition $\tilde{\mathcal{H}}_0^{(n)} = \frac{1}{n}\sum_{i=1}^n \tilde{\mathcal{H}}_0^{(1)}\,|_i \otimes \mathbb{I}_{[n]\setminus\{i\}}$, so

$$\langle\psi|\,\tilde{\mathcal{H}}_0^{(n)}\,|\psi\rangle = \frac{1}{n}\sum_{i=1}^n \mathrm{Tr}\Big[\psi_i\tilde{\mathcal{H}}_0^{(1)}\Big], \tag{36}$$

where $\psi_i \equiv \mathrm{Tr}_{-i}[|\psi\rangle\langle\psi|]$ is the reduced state of $|\psi\rangle$ on qubit i. We will show that at most one of the terms in this summation can be 0, and that the remainder of the terms are lower-bounded by $\sin^2\left(\frac{\pi}{8}\right)$.

By Lemma 24 we can write the reduced state as

$$\psi_i = \frac{1}{2}\sum_{\hat{g}\in G_i}\left(\cos^2(\theta)\hat{g} + \sin^2(\theta)P_i\hat{g}P_i\right) + \frac{1}{2}\sum_{g'\in G_{i,P}} i\sin(\theta)\cos(\theta)[P_i, g']. \tag{37}$$

We proceed in cases:

(1) If $P \in G$, $P_i = \mathbb{I}$, $G_{i,P} = \emptyset$, or $G_{i,P} = \{\mathbb{I}\}$ then ψ_i is a stabilizer state, so $\mathrm{Tr}\left[\psi_i\tilde{\mathcal{H}}_0^{(1)}\right] \geq \sin^2\left(\frac{\pi}{8}\right)$.

(2) Suppose the four conditions from Case **I.** do not hold. It must be that $G_{i,P} = \{\mathbb{I}, P^\star\}$ for some $P^\star \in \mathcal{P}_1 \setminus \{\mathbb{I}, P_i\}$; $P^\star$ cannot be P_i as this would imply $P \in G$. Note that $G_{i,P}$ cannot be any larger as this would contradict the fact G is a stabilizer group. We now consider cases for G_i.

(a) If $G_i = \{\mathbb{I}\}$, then ψ_i can be written as

$$\psi_i = \frac{1}{2}\mathbb{I} + \frac{1}{2}i\sin(\theta)\cos(\theta)[P_i, P^\star], \tag{38}$$

$$= \frac{1}{2}\mathbb{I} + \frac{1}{4}\sin(2\theta)\sigma, \tag{39}$$

since $P_i \neq P^\star$ and $2i[P_i, P^\star] = \sigma$ for some non-identity Pauli. The desired bound holds by direct computation over $\sigma \in \mathcal{P} \setminus \{\pm\mathbb{I}\}$.

(b) If G_i is non-trivial then $G_i = \{\mathbb{I}, P^\star\}$ since it must commute with the $g \in G$ which satisfies $g_i = P^\star$ and $g_{-i} = P_{-i}$ (which exists since we are in Case **II.**) Since $P^\star \notin \{\mathbb{I}, P_i\}$ we can write ψ_i as

$$\psi_i = \frac{1}{2}\mathbb{I} + \frac{1}{2}\left(\cos^2(\theta) - \sin^2(\theta)\right)P^\star + \frac{1}{2}i\sin(\theta)\cos(\theta)[P_i, P^\star], \tag{40}$$

$$= \frac{1}{2}\mathbb{I} + \frac{1}{2}\cos(2\theta)P^\star + \frac{1}{4}\sin(2\theta)i[P_i, P^\star]. \tag{41}$$

By direct computation we have the following:

(i) If $P_i \neq Y$ then $\mathrm{Tr}\left[\psi_i\tilde{\mathcal{H}}_0^{(1)}\right] \geq \sin^2\left(\frac{\pi}{8}\right)$ regardless of θ.

(ii) If $P_i = Y$ and $P^\star \neq Z$ then $\mathrm{Tr}\left[\psi_i\tilde{\mathcal{H}}_0^{(1)}\right] \geq \sin^2\left(\frac{\pi}{8}\right)$ regardless of θ.

(iii) If $P_i = Y$ and $P^\star = Z$ then $\mathrm{Tr}\left[\psi_i\tilde{\mathcal{H}}_0^{(1)}\right] \geq 0$ with possible equality.

To recap the cases, ψ_i can have energy less than $\sin^2\left(\frac{\pi}{8}\right)$ only if (1) $P_i = Y$, (2) $Z_i \in G$, and (3) there is a $g \in G$ such that $g_i = Z$ and $g_{-i} = P_{-i}$, i.e. g and P agree on every qubit except i.

We must show that at most one qubit can satisfy all three of these condition for a given $P \in \mathcal{P}_n$ and stabilizer group G. Suppose there are two such qubits, i and j, which satisfy (1) $P_i = P_j = Y$, (2) $Z_i, Z_j \in G$, and (3) there exist $g, h \in G$ such that $g_i = h_j = Z$, $g_{-i} = P_{-i}$, and $h_{-j} = P_{-j}$. By condition (3) $g_i = Z$ and $g_j = Y$ and by condition (2) $Z_j \in G$, but this implies that $gZ_j = -Z_jg$, which contradicts the fact that G is abelian. Thus, at most a single qubit can satisfy the conditions required for the reduced state ψ_i to have energy less than $\sin^2\left(\frac{\pi}{8}\right)$, which implies the desired lower bound. ◀