ABSTRACT

Title of Dissertation: APPLICATIONS AND VERIFICATION OF QUANTUM COMPUTERS

Shih-Han Hung
Doctor of Philosophy, 2021

Dissertation Directed by: Professor Andrew M. Childs
Department of Computer Science

Quantum computing devices can solve problems that are infeasible for classical computers. While rigorously proving speedups over existing classical algorithms demonstrates the usefulness of quantum computers, analyzing the limits on efficient processes for computational tasks allows us to better understand the power of quantum computation. Indeed, hard problems for quantum computers also enable useful cryptographic applications.

In this dissertation, we aim to understand the limits on efficient quantum computation and base applications on hard problems for quantum computers. We consider models in which a classical machine can leverage the power of a quantum device, which may be affected by noise or behave adversarially. We present protocols and tools for detecting errors in a quantum machine and estimate how serious the deviation is. We construct a non-interactive protocol that enables a purely classical party to delegate any quantum computation to an untrusted quantum prover. In the setting of error-prone quantum hardware, we employ formal methods to construct a logical system for reasoning about the robustness of a quantum algorithm design.

We also study the limits of ideal quantum computers for computational tasks and give asymptotically optimal algorithms. In particular, we give quantum algorithms which provide speedups for the polynomial interpolation problem and show their optimality. Finally, we study the performance of quantum algorithms that learn properties of a matrix using queries that return its action on an input vector. In particular, we show that for various linear algebra problems, there is no quantum speedup, while for some problems, exponential speedups can be achieved.

APPLICATIONS AND VERIFICATION OF QUANTUM COMPUTERS

by

Shih-Han Hung

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:
Professor Andrew M. Childs, Chair/Advisor
Dr. Gorjan Alagic
Professor Dana Dachman-Soled, Dean's representative
Professor Michael Hicks
Professor Jonathan Katz
Professor Xiaodi Wu

Acknowledgments

I am indebted to my advisor, Andrew Childs, for his support and patience throughout my Ph.D. journey. Andrew has set an example of excellence as a researcher, teacher and mentor. His advice and courses have allowed me to gain a comprehensive view of quantum computing. I am grateful to Andrew for always being generous with his time and ideas on how to set up and approach a research problem, and for helping with improving my scientific writing skills. Furthermore, Andrew has given me a great amount of freedom to work on many different problems in collaboration with various groups. This allows me to expand my research horizons and learn from leading experts in many different areas.

I thank Gorjan Alagic for in-depth collaboration on quantum cryptography. Over the past four years, we had extensive discussions in various directions. His guidance has helped me to gain a foundation in the field.

I thank Michael Hicks and Xiaodi Wu for leading me into the research area of quantum programming language. I am grateful to Mike and Xiaodi for sharing their insights and experiences about how to choose and conduct research on impactful problems.

I thank Michael Coplan and Charles Clark for my time at the University of Maryland, and encouraging me to do research on quantum information science. I also thank Jonathan Katz for supporting me during the summer after my first year.

I thank the members of my dissertation committee, Andrew Childs, Gorjan Alagic,

Dana Dachman-Soled, Michael Hicks, Jonathan Katz and Xiaodi Wu for reading my dissertation and providing me valuable feedback during the various stages of the development.

I enjoyed many in-depth discussions with my collaborators, including Manuel Barbosa, Gilles Barthe, Shouvanik Chakrabarti, Jianxin Chen, Wim van Dam, Leo Fan, Benjamin Grégorie, Alex Grilo, Kesha Hietala, Jonathan Katz, Liyi Li, Tongyang Li, Yuxiang Peng, Robert Rand, Igor Shparlinski, Pierre-Yves Strub, Mingsheng Ying, Chunhao Wang, Xin Wang, Li Zhou, and Shaopeng Zhu. I also thank Alexey Gorshkov, Stephen Jordan, Brad Lackey, Yi-Kai Liu, and Carl Miller for many stimulating conversations. It has been a pleasure to work with and learn from you.

My life as a graduate student would not have been enjoyable without friends and colleagues in CS and QuICS: Aniruddha Bapat, Shouvanik Chakrabarti, Nai-Hui Chia, Jianxin Chen, Su-Kuan Chu, Abhinav Deshpande, Dhruv Devulapalli, Hong Hao Fu, James Garrison, Andrew Guo, Wenqi Han, Kesha Hietala, Shelby Kimmel, Tongyang Li, Cedric Lin, Jin-Peng Liu, Atul Mantri, Aaron Ostrander, Yuxiang Peng, Robert Rand, Julien Ross, Eddie Schoute, Troy Sewell, Manasi Shingane, Yuan Su, Pattara Sukprasert, Aarthi Sundaram, Minh Tran, Daochen Wang, Xin Wang, Xingyao Wu, Sheng Yang, Penghui Yao, Qi Zhao, and Shaopeng Zhu. I would also like to thank QuICS coordinators Javiera Caceres and Andrea Svejda. They were always there whenever I needed help.

Finally, I would like to thank my family for their love and encouragement over the years.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

It is widely believed that quantum devices can solve computational tasks that are intractable for classical computers. The advantage of quantum computers is determined by finding computational tasks for which substantial speedups are possible. Indeed, the existence of such tasks motivates scientific studies that advance theoretical and experimental progress in the research area of quantum information processing.

The implementation of quantum algorithms requires qubits to be protected from noise, but at the same time, must allow external control by users. While it is possible to reduce the effect of noise by developing quantum error-correction (QEC) protocols, the overhead introduced by QEC protocols is beyond the capabilities of current devices. In the near term, realizable quantum computers are so-called noisy intermediate-scale quantum (NISQ) devices with 50-100 qubits [1, 2]. An active research area in the NISQ era aims to extract the maximum computational power from such small quantum devices, and, meanwhile, the quality of computation still meets the demand by the users.

With devices that reach scales beyond achievable in the near term, direct classical simulation seems impossible for verifying the correctness of quantum computation. Depending on how the error is provided, the task of verification demands different techniques.

First, to ensure an implementation (e.g., a program or a circuit) or its optimization is

correct, one must check if the meaning of the implementation follows the intention. In the context of quantum computing, a quantum process is often denoted as a quantum channel which takes exponentially many parameters. Thus the naïve way of verifying correctness seems classically intractable. To this end, an appealing approach is to apply formal methods to prove an algorithm design works as intended [3, 4]. In a nutshell, first a syntax is defined to express an algorithm design in a machine-checkable format, and the semantics is defined to formally specify what it means by correctness. Then a sound logical system is constructed to enable a proof search with a computer.

Secondly, to test if a quantum device works as intended, an experimentalist needs to determine the quality of computation by sending signals to and collecting feedback from it. In theoretical computer science, such an interactive verification process can be modeled as a classical interactive proof system, which consists of a classical verifier and a quantum prover which may deviate from the protocol to an arbitrary degree. In the settings that the quantum device is efficient, one can rely on hard problems for quantum computers to establish a cryptographic leash. This restores the symmetry between the quantum prover and the less powerful classical verifier [5].

The rules of quantum mechanics define the limits on efficient quantum computation. In particular, showing significant resource requirements for a task rules out the possibility of developing high-performance quantum algorithms with reasonable resources. Interestingly, apart from verification of quantum computation [6, 7, 8], constructions based on tasks that are hard for quantum computers lead to various quantum-secure classical cryptosystems and quantum-unique applications, including proof of quantumness [9, 10, 11], certifiable randomness expansion [9], certified deletion [12], copy protection [13, 14, 15, 16], and quantum

money [13, 17, 18, 19]. While proving quantum hardness for a computational problem unconditionally, i.e., solely from the rules of quantum mechanics, requires a major breakthrough, a rigorous reasoning is often formalized in a query model or with respect to widely held hardness assumptions.

In a query model, an algorithm is given access to an oracle which encodes some information that is initially unknown, but can be partially extracted by making queries. The performance of an algorithm is defined as the probability of outputting the correct result, using a certain number of queries. While proving quantum speedups in a query model cannot be translated into an efficient quantum algorithm in general, proving a query lower bound implies (substantial) resource requirement with respect to a general instantiation of the oracle. To establish a quantum query lower bound, various methods have been proposed, including the polynomial method [20], the hybrid method [21], and the adversary methods [22, 23].

In the plain model, i.e., no oracle access is assumed, a hardness result often relies on a hardness assumption, which may not be proved but widely believed to hold. For example, learning with errors (LWE) is a computational problem which refers to solving an erroneous system of linear equations, and is conjectured to be hard to solve, even for quantum computers [24]. To show a problem P (e.g., breaking a cryptographic construction with respect to a security notion) is hard based on the hardness of P', one aims to give a reduction: assuming the existence of an algorithm A which solves P, there exists another algorithm B which simulates A and achieves "too good" performance for P', and thus break the assumption that P' is hard.

In this dissertation, we aim to understand the limits on efficients quantum computation

for computational tasks and explore the task of verification in different settings. Our results are briefly summarized as follows.

1.1 Classical verification of quantum computation

While rapid progress has been made toward building quantum devices, it remains challenging to verify that they work correctly, especially when they reach scales that rule out direct classical simulation. In a major breakthrough, Mahadev constructed the first interactive protocol which enables a purely classical party, called the verifier, to delegate any efficient quantum computation to an untrusted prover for any decision problem solvable in quantum polynomial time with bounded errors [6]. In Chapter 2, we show the same task can be achieved non-interactively, and in zero-knowledge.

Our protocols result from a sequence of improvements to the original four-message protocol due to Mahadev. We begin by making the first message instance-independent and moving it to an offline setup phase. We then establish a parallel repetition theorem for the resulting three-message protocol, with the soundness error decreasing at an optimal rate. This enables an application of the Fiat-Shamir transform, eliminating the public coin sent by the verifier and giving a non-interactive protocol.

Furthermore, we employ classical non-interactive zero-knowledge arguments for NP languages and classical fully homomorphic encryption (FHE) to give a zero-knowledge variant of our non-interactive protocol. This yields the first purely classical NIZK argument for QMA, a quantum analogue of NP.

We establish the security of our protocols under standard assumptions in quantum

secure cryptography. Specifically, the soundness is proved in the quantum random oracle model (QROM), under the assumption that LWE is hard against quantum algorithms. The NIZK construction also requires circuit-private FHE, an instantiation of which based on LWE is known [25].

Chapter 2 is based on the following paper:

[7] Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In *Theory of Cryptography Conference*, pages 153–180. Springer, 2020. arXiv:1911.08101.

1.2 Polynomial interpolation

Let $f \in \mathbb{K}[x_1, \ldots, x_n]$ be a degree-d polynomial of n variables over a field $\mathbb{K}$. In the polynomial interpolation problem, the algorithm is given access to an unknown polynomial as a black-box for evaluating it on any chosen input, and the task is to determine all the coefficients. The classical query complexity is well-known: to determine all the coefficients, $\binom{n+d}{d}$ queries are sufficient and necessary, regardless of the field $\mathbb{K}$. Shamir [26] used this fact to construct a protocol for secret sharing: a secret is encoded in a univariate polynomial f over a sufficiently large finite field and divided into $d+1$ shares $\{(x_i, f(x_i))\}_{i=1}^{d+1}$. Based on the classical query complexity, all the parts can be used to infer the secret, and learning only d parts gives no information.

In Chapter 3, we study the quantum query complexity of the polynomial interpolation problem over finite fields, the real numbers and the complex numbers. Previously, for univariate polynomial interpolation over a finite field, Kane and Kutin [27] and Meyer and

Pommersheim [28] independently showed that $d/2+1$ queries are necessary. We show that the lower bound is indeed optimal—$d/2+1$ queries are sufficient to learn all the coefficients with probability $1-O(1/q)$ over a field of q elements. Furthermore, the algorithm can be implemented with gate complexity $\mathrm{poly}(\log q)$ with negligible decrease in the success probability.

The optimality of our algorithm is based on the simple fact that any algorithm successfully distinguishes n quantum states in an m-dimensional space with probability at most m/n. Later, Copeland and Pommersheim [29] generalized the idea and gave a representation-theoretic characterization of the optimal success probability for the *coset identification problem*, of which the polynomial interpolation can be viewed as a special case.

For multivariate polynomial interpolation, we show that over the field $\mathbb{C}$ of complex numbers, $k = \lceil \frac{1}{n+1}\binom{n+d}{d} \rceil$ is sufficient to succeed with probability 1, except for a few special cases of d and n. Furthermore, over the field $\mathbb{R}$ of real numbers and a finite field $\mathbb{F}_q$, we show that $2k$ and $\lceil \frac{d}{n+d}\binom{n+d}{d} \rceil$ queries are sufficient to succeed with probability approaching 1. Thus we present a speedup of factor $n+1$, $\frac{n+1}{2}$, and $\frac{n+d}{d}$ over $\mathbb{C}$, $\mathbb{R}$ and $\mathbb{F}_q$ respectively.

Chapter 3 is partly based on the following papers:

[30] Andrew M. Childs, Wim van Dam, Shih-Han Hung, and Igor E. Shparlinski. Optimal quantum algorithm for polynomial interpolation. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics*, pages 16:1–16:13, 2016. arXiv:1509.09271.

[31] Jianxin Chen, Andrew M. Childs, and Shih-Han Hung. Quantum algorithm for multivariate polynomial interpolation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209):20170480, 2018. arXiv:1701.03990.

1.3 Query complexity with matrix-vector products

Linear algebra problems including solving linear systems and determining basic properties of matrices such as rank, determinant, trace, and eigenvalues constitute a fundamental area of research in applied mathematics and computer science. Algorithmic linear algebra also provides a fundamental toolbox that can inspire the design of algorithms in general.

Algorithms for solving linear algebra problems can depend significantly on the model of access to a matrix. One natural model which has received attention recently [32, 33] is the matrix-vector product (Mv) oracle: for a matrix M over a field $\mathbb{K}$, on input a vector x, the oracle outputs Mx. Another related model is the vector-matrix-vector (vMv) model, which returns $y^\top Mx$ for given input vectors x, y.

In Chapter 4, we study quantum algorithms which learn properties of a matrix in these models. We show that for computing trace, determinant, rank of a matrix or solving a linear system that it specifies, quantum computers do not provide an asymptotic speedup over classical computers. On the other hand, we show that for some properties about the matrix, such as computing the parities of the rows or coumns, or deciding if there are two identical rows or columns, quantum computers provide exponential speedups. We demonstrate this by showing equivalence between models that provide matrix-vector products, vector-matrix products and vector-matrix-vector products. In contrast, the power of these models can vary significantly for classical computation.

Chapter 4 is based on the following paper:

[34] Andrew M. Childs, Shih-Han Hung, and Tongyang Li. Quantum Query Complexity with Matrix-Vector Products. In *48th International Colloquium on Automata, Languages,*

and Programming (ICALP 2021), volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 55:1–55:19, 2021. arXiv:2102.11349.

1.4 Quantitative robustness analysis

A major challenge to leverage the power of quantum speedups, especially in the near future, is to deal with errors during execution of quantum algorithms on a quantum device. Most existing work on the study of quantum algorithms and their implementations assumes the problem will be solved by the hardware while ignoring the possibility of errors. Unfortunately, providing such a general-purpose, fault-tolerant quantum computing abstraction appears to be impractical for near-term quantum devices, for which precisely controllable qubits are scarce and error-prone. As such, research on practical quantum computation must focus on noisy intermediate-scale quantum (NISQ) computers [1], which will lack general-purpose fault tolerance. To guide the design of practical applications, it demands a principled method to estimate the error-affected performance of implementations.

A naïve approach is to directly calculate the distance between the noisy output state and its ideal equivalent with respect to an appropriate metric, e.g., trace distance. However, direct calculation is computationally intractable when the programs reach scales that rule out classical simulation.

In Chapter 5, we apply formal methods to reason about the performance of quantum programs in the presence of quantum noise. In the classical setting, Carbin, Misalovic, Rinard [35] gave a logic for reasoning about classical programming on unreliable classical hardware. Inspired by the previous works, we provide the syntax and semantics of the *quantum while*

language, extended to include noisy operations. We define the notion of quantum robustness, which describes the closeness between a noisy output and its ideal correspondence when the input satisfies a property to at least a certain degree. Furthermore, we define a proof system for reasoning about quantum robustness.

Chapter 5 is based on the following paper:

[36] Shih-Han Hung, Kesha Hietala, Shaopeng Zhu, Mingsheng Ying, Michael Hicks, and Xiaodi Wu. Quantitative robustness analysis of quantum programs. *Proceedings of the ACM Symposium on Programming Languages*, 3(POPL):1–29, 2019. arXiv:1811.03585.

Chapter 2: Non-interactive classical verification of quantum computation

2.1 Introduction

Quantum computing devices are expected to solve problems that are infeasible for classical computers. However, as significant progress is made toward constructing quantum computers, it is challenging to verify that they work correctly. This becomes particularly difficult when devices reach scales that rule out direct classical simulation.

This problem has been considered in various models, such as with multiple entangled quantum provers [37, 38, 39, 40, 41, 42, 43, 44] or with verifiers who have limited quantum resources [45, 46, 47, 48]. Such solutions are not ideal since they require assumptions about the ability of the provers to communicate or require the verifier to have some quantum abilities.

In a major breakthrough, Mahadev recently described the first secure protocol enabling a purely classical verifier to certify the quantum computations of a single untrusted quantum prover [6]. The Mahadev protocol uses a quantum-secure cryptographic assumption to give the classical verifier leverage over the quantum prover. Specifically, the protocol is sound under the assumption that the Learning with Errors (LWE) problem does not admit a polynomial-time quantum algorithm. This assumption is widely accepted, and underlies some of the most promising candidates for quantum-secure cryptography [49].

The Mahadev protocol. Mahadev's result settled a major open question concerning the power of *quantum-prover interactive arguments* (QPIAs). In a QPIA, two computationally-bounded parties (a quantum prover $\mathcal{P}$ and a classical verifier $\mathcal{V}$) interact with the goal of solving a decision problem. Mahadev's result showed that there is a four-round[1] QPIA for BQP with negligible completeness error and constant soundness error $\delta \approx 3/4$. The goal of the protocol is for the verifier to decide whether an input Hamiltonian H from a certain class (which is BQP-complete) has a ground state energy that is low (YES) or high (NO).

The protocol has a high-level structure analogous to classical Σ-protocols [50]:

1. $\mathcal{V}$ generates a private-public key pair (pk, sk) and sends pk to $\mathcal{P}$;

2. $\mathcal{P}$ prepares the ground state of H and then coherently evaluates a certain classical function f_{pk}. This yields a state of the form

$$\sum_x \alpha_x |x\rangle_X |f_{pk}(x)\rangle_Y \,, \tag{2.1}$$

   where the ground state is in a subregister of register X. $\mathcal{P}$ measures the output register Y and sends the result y to $\mathcal{V}$. Note that $\mathcal{P}$ now holds a superposition over the preimages of y.

3. $\mathcal{V}$ replies with a uniformly random *challenge* bit $c \in \{0, 1\}$.

4. If $c = 0$ ("test round"), $\mathcal{P}$ measures the X register in the computational basis and sends the outcome. If $c = 1$ ("Hadamard round"), $\mathcal{P}$ measures X in the Hadamard basis and sends the outcome.

[1] We take one round to mean a single one-way message from the prover to the verifier, or vice-versa. The Mahadev protocol involves four such messages.

After the four message rounds above are completed, the verifier uses their knowledge of H and the secret key sk to either accept or reject the instance H.

Our results. In this work, we show that the Mahadev protocol can be transformed into protocols with significantly more favorable parameters, and with additional properties of interest. Specifically, we show how to build non-interactive protocols (with setup) for the same task, with negligible completeness and soundness errors. One of our protocols enables a verifier to publish a single public "setup" string and then receive arbitrarily many proofs from different provers, each for a different instance. We also construct a non-interactive protocol that satisfies the zero-knowledge property [51].

In principle, one could ask for a slightly less interactive protocol: one where the prover and the verifier both receive the instance from some third party, and then the prover simply sends a proof to the verifier, with no setup required. While we cannot rule such a protocol out, constructing it seems like a major challenge (and may even be impossible). In such a setting, the proof must be independent of the secret randomness of the verifier, making it difficult to apply the "cryptographic leash" technique of Mahadev. On the other hand, without cryptographic assumptions, such a protocol would result in the unlikely inclusion BQP $\subseteq$ MA [52].

All of our results are conditioned on the hardness of the LWE problem for quantum computers; we call this *the* LWE *assumption*. This assumption is inherited from the Mahadev protocol. For the zero-knowledge protocol, we also require fully-homomorphic encryption (FHE) with circuit privacy [25]. Our security proofs hold in the Quantum Random Oracle Model (QROM) [53]. For simplicity, in our exposition we assume that the relevant security

parameters are polynomial in the input BQP instance size n, so that efficient algorithms run in time poly(n) and errors are (ideally) negligible in n.

Warmup: A non-interactive test of quantumness. To explain our approach, we first briefly describe how to make the "cryptographic test of quantumness" (CTQ) of [9] into a non-interactive protocol (with setup.) This is a significantly simplified version of the Mahadev protocol: there is no ground state, and the initial state (2.1) is simply in uniform superposition over X. The soundness error is 1/2, meaning that a *classical* prover can convince the verifier to accept with probability at most 1/2 [9]. A quantum prover can easily answer both challenges, so the completeness is 1.

To reduce the interaction in this protocol, we perform two transformations. First, we repeat the protocol independently in parallel k times, with the verifier accepting if and only if all k copies accept. We then remove Round 3 via the Fiat-Shamir transform [54]: the prover computes the challenges $\mathbf{c} = (c_1, c_2, \ldots, c_k) := \mathcal{H}(y_1, \ldots, y_k)$ themselves via a public hash function $\mathcal{H}$. This allows the prover to go directly to Round 4, i.e., measuring the X registers. The verifier then performs the k-fold accept/reject verdict calculations, using coins $\mathbf{c}$ computed in the same manner. The result is a two-message protocol. Moreover, since the keys are drawn from a fixed distribution, we can give the pk_j to the prover and the sk_j to the verifier in an offline setup phase, so that the protocol only requires one message in the online phase. We refer to this protocol as NI-CTQ.

Since the soundness experiment of NI-CTQ only involves classical provers, and the verifier is also classical, soundness can be deduced from existing classical results. Specifi-

cally, standard parallel repetition[2] theorems [55, 56, 57] combined with soundness of Fiat-Shamir [54, 58, 59] yield the fact that NI-CTQ has negligible soundness and completeness errors, in the Random Oracle Model (ROM).

Transforming the Mahadev protocol. Similar to NI-CTQ above, we will apply various transformations to the Mahadev verification protocol itself:

1. making the first message instance-independent (i.e., moving it to an offline setup phase);
2. applying parallel repetition, via a new parallel repetition theorem;
3. adding zero-knowledge, by means of classical NIZKs and classical FHE; and
4. applying Fiat-Shamir (in the QROM [53]).

Unlike with NI-CTQ, however, establishing that these transformations satisfy desirable properties is much more challenging. For instance, since cheating provers can now be quantum, classical parallel repetition theorems do not apply.

Instance-independent setup. To improve over the instance-dependent key generation in the original Mahadev protocol, our first transformation is relatively simple to describe, at a high level. In the Mahadev protocol, the initial message depends on a sequence of basis choices (X or Z) for measuring the ground state of a ZX Hamiltonian. These choices need to be consistent with a particular two-local term drawn from some distribution $\mathcal{D}$. Clearly, a *random* choice is correct with probability $1/4$. Now, if we consider multiple copies of the

[2] A subtlety is that this is a private-coin protocol. However, the $c = 0$ branch is publicly simulable so [55] applies. Alternatively, one can apply the techniques of [56].

ground state, and each copy is assigned both a random choice of bases and a random term from $\mathcal{D}$, then about 1/4 of the copies get a consistent assignment. We can then make the initial message instance-independent by increasing the number of copies of the ground state in the Mahadev protocol by a constant factor. We establish this fact[3] in Lemma 2.4.1 below. We refer to the result as "the three-round Mahadev protocol," and denote it by $\mathfrak{M}$.

Parallel repetition. The k-fold sequential repetition of a protocol is a simple way of decreasing the original soundness error δ to δ^k, at the cost of multiplying the number of interaction rounds by k. Parallel repetition is much more desirable because it does not increase the number of rounds. However, even in the case of purely classical protocols, proving that parallel repetition reduces soundness error is often quite difficult, and may require adapting the protocol itself [56, 60].

Does parallel repetition work for quantum-prover interactive arguments? The Mahadev protocol is a natural case to consider since it already exhibits the full decisional power of QPIAs, namely BQP. However, several complications arise when attempting to establish parallel repetition using classical techniques. First, the Mahadev protocol is clearly private-coin, precisely the category that is challenging even classically [56, 60]. Second, classical proofs of parallel repetition typically involve constructing a prover (for the single-copy protocol) that uses many rounds of nested rejection sampling. The quantum analogue of such a procedure is quantum rewinding, which can only be applied in special circumstances [61, 62] and seems difficult to apply to parallel repetition.

In this work, we establish a new parallel repetition theorem with alternative techniques,

[3]More precisely, we apply this transformation at the level of the Morimae-Fitzsimons protocol [47], an important building block of the Mahadev protocol.

suited specifically for the Mahadev protocol. We show that, for NO instances, the accepting paths of the verifier for the two different challenges ($c = 0$ and $c = 1$) correspond to two nearly (computationally) orthogonal projectors. We also establish that this persists in k-fold parallel repetition, meaning that each pair of distinct challenge strings $\mathbf{c}, \mathbf{c}' \in \{0,1\}^k$ corresponds to nearly orthogonal projectors. From there, a straightforward argument shows that the prover cannot succeed for more than a non-negligible fraction of challenge strings. Our result shows that k-fold parallel repetition yields the same optimal soundness error δ^k as sequential repetition.

Taken together with the first transformation, the result is a three-round QPIA (with offline setup) for verifying BQP, with negligible completeness and soundness errors. We denote this protocol by $\mathfrak{M}^k$.

Theorem 2.1.1. *Under the* LWE *assumption, the k-fold parallel repetition $\mathfrak{M}^k$ of the three-round Mahadev protocol $\mathfrak{M}$ is a three-round protocol (with offline setup) for verifying* BQP *with completeness $1 - \mathrm{negl}(n)$ and soundness error $2^{-k} + \mathrm{negl}(n)$.*

Zero-knowledge. Zero-knowledge is a very useful cryptographic property of proof systems. Roughly, a protocol is zero-knowledge if the verifier "learns nothing" from the interaction with the honest prover, besides the fact that the relevant instance is indeed a "yes" instance. This notion is formalized by requiring an efficient simulator whose output distribution is indistinguishable from the distribution of the outcomes of the protocol.

In our next result, we show how to modify the protocol $\mathfrak{M}^k$ of Theorem 2.1.1 to achieve zero-knowledge against arbitrary classical verifiers. Our approach is similar to that of [63], but uses a purely classical verifier. Instead of the prover providing the outcomes of the

measurements to be checked by the verifier (as in $\mathfrak{M}^k$), a classical non-interactive zero-knowledge proof (NIZK) is provided. However, the NP statement "the measurements will pass verification" depends on the inversion trapdoor of the verifier, which must remain secret from the prover. To overcome this obstacle, we use classical fully homomorphic encryption (FHE). In the setup phase, an encryption of the verifier's secret keys is provided to the prover, enabling the prover to later compute the NIZK homomorphically. To establish the zero-knowledge property, we require the FHE scheme to have circuit privacy, which means that the verifier cannot learn the evaluated circuit *from the ciphertext* provided by the prover. To prove the zero-knowledge property, we also need the extra assumption that the setup phase is performed by a trusted third party, since we cannot rely on the verifier to perform it honestly anymore.

In classical zero-knowledge arguments, it is common to consider efficient provers who are provided an NP-witness of the statement to prove. In the quantum setting, if we assume that the quantum polynomial-time prover has access to a quantum proof of a QMA statement,[4] we achieve the following.

Theorem 2.1.2 (informal)**.** *Under the* LWE *assumption, if circuit-private FHE exists, then there exists a three-round zero-knowledge argument for* QMA *(with trusted setup) with negligible completeness and soundness error.*

Fiat-Shamir transformation. Note that in the protocols discussed above (both $\mathfrak{M}^k$ and its ZK-variant), the second message of the verifier to the prover is a uniformly random $\mathbf{c} \in \{0,1\}^k$. In the final transformation, we eliminate this "challenge" round. This is done

[4]QMA is a decision problem class which is a quantum analogue of NP. In QMA, an untrusted quantum proof is provided to a quantum poly-time verifier.

via the well-known Fiat-Shamir transform [54]: we ask the prover to generate the challenge bits $\mathbf{c} \in \{0,1\}^k$ themselves by evaluating a public hash function $\mathcal{H}$ on the transcript of the protocol thus far. In our case, recalling (2.1), this means that the prover selects $\mathbf{c} := \mathcal{H}(H, pk, y)$. Note that pk and y are now both k-tuples, since we are transforming k-fold parallel-repeated protocols. Of course, the verifier also needs to adapt their actions at the verdict stage, using $\mathbf{c} = \mathcal{H}(H, pk, y)$ when deciding whether to accept or reject. The resulting protocols now only have a setup phase and a single message from the prover to the verifier.

A standard approach with Fiat-Shamir (FS) is to establish security in the Random Oracle Model, in the sense that FS preserves soundness up to a loss that is negligible provided $\mathcal{H}$ has a superpolynomially-large range [58, 59]. It is straightforward to see that this last condition is required; it is also the reason that we applied parallel repetition prior to FS. A well-known complication in the quantum setting is that quantum computers can evaluate any public classical function $\mathcal{H}$ in superposition via the unitary operator $U_{\mathcal{H}} \colon |x\rangle|y\rangle \mapsto |x\rangle|y \oplus \mathcal{H}(x)\rangle$. This means that we must work in the Quantum Random Oracle Model (QROM) [53], which grants all parties oracle access to $U_{\mathcal{H}}$. Proving the security of transformations like FS in the QROM is the subject of recent research, and newly developed techniques have largely shown that FS in the QROM preserves soundness for so-called Σ-protocols [64, 65]. Extending those results to our protocols is relatively straightforward. Applying FS to the three-round verification protocol from Theorem 2.1.1 then yields the following.

Theorem 2.1.3. *Let $k = \omega(\log n)$, and let $\mathrm{FS}(\mathfrak{M}^k)$ denote the protocol resulting from applying Fiat-Shamir to the k-fold parallel repetition of the three-round Mahadev protocol. Under the* LWE *assumption, in the QROM, $\mathrm{FS}(\mathfrak{M}^k)$ is a non-interactive protocol (with offline setup)*

for verifying BQP *with negligible completeness and soundness errors.*

If we instead apply the Fiat-Shamir transform to the zero-knowledge protocol from Theorem 2.1.2, we achieve the following.[5]

Theorem 2.1.4 (informal)**.** *Under the* LWE *assumption, in the QROM, there exists a classical non-interactive zero-knowledge argument (with trusted offline setup) for* QMA*, with negligible completeness and soundness errors.*

Related results. Broadbent, Ji, Song, and Watrous [66] presented the first quantum zero-knowledge proofs for QMA with efficient provers. Vidick and Zhang [67] combined this protocol with the Mahadev protocol [6] to make the communication classical. Broadbent and Grilo [68] showed a "quantum Σ" zero-knowledge proof for QMA (with a quantum verifier). In the non-interactive setting, Coladangelo, Vidick, and Zhang [63] constructed a non-interactive zero-knowledge argument with quantum setup and Broadbent and Grilo [68] showed a quantum statistical zero-knowledge proof in the secret parameter model.

We remark that Radian and Sattath [69] recently established what they call "a parallel repetition theorem for NTCFs," which are the functions f_{pk} in the Mahadev protocol. However, the context of [69] is very different from that of our Theorem 2.1.1. They work with 1-of-2 puzzles, not BQP verification; in particular, their soundness experiment is quite different. Moreover, their parallel repetition theorem follows from a purely classical result.

After an initial version of our work was made public, showing how the Mahadev protocol can be reduced to four rounds using parallel repetition and the Fiat-Shamir transform,

[5]Note that $\mathrm{FS}(\mathfrak{M}^k)$ in Theorem 2.1.3 is also a protocol for verifying QMA with negligible error if the prover is given a quantum witness.

Chia, Chung, and Yamakawa posted a preprint [8] describing the same result, with an alternative proof of parallel repetition. They also showed how to make the verifier run in time polylogarithmic in the instance size using indistinguishability obfuscation. Our work was performed independently, and we subsequently improved our result to make the protocol non-interactive with setup and zero-knowledge.

Open questions. This work raises several natural open questions. First, is it possible to prove the soundness of our protocol when the oracle $\mathcal{H}$ is instantiated with a concrete (e.g., correlation-intractable [70]) hash function? Our current analysis only applies in an idealized model.

It is also natural to study parallel repetition for general QPIAs. Natural examples include the protocols of [9, 67, 71]. It is known that parallel repetition does not reduce the soundness error even for all classical private-coin protocols [60]. However, if the classical protocol is slightly modified to include "random terminations" with low probability, then parallel repetition is in fact possible [55, 56, 57]. It is an open question whether similar mild relaxations can enable a parallel repetition theorem for any QPIA.

Finally, we remark that a classical NIZK protocol (in the Random Oracle Model with setup) could also be achieved using the techniques of locally simulatable codes/proofs [68, 72]. We leave as an open problem understanding whether such a protocol could give us useful properties that are not achieved with our current approach.

Organization. The remainder of the paper is organized as follows. In Section 2.3, we introduce QPIAs, the local Hamiltonian problem as it relates to BQP verification, the Mahadev

protocol, and non-interactive zero knowledge. In Section 2.4, we explain how to make the initial step of the protocol instance-independent. In Section 2.5, we show that parallel repetition reduces the soundness error of the Mahadev protocol at the optimal rate. In Section 2.6, we describe how to make the protocol zero-knowledge. Finally, in Section 2.7, we show that under the Fiat-Shamir transformation, a generalization of Σ-protocols (which includes the protocols of interest to us) remains secure in the QROM, and we use this to establish non-interactive protocols (with offline setup) for verifying BQP and for zero-knowledge verification of QMA.

2.2 Standard cryptographic primitives

2.2.1 Commitment schemes

The following definition is taken from [63] (with modification). A trapdoor commitment scheme is a tuple of algorithms $\mathsf{Com} = (\mathsf{gen}, \mathsf{commit}, \mathsf{verify})$, described as follows.

1. $\mathsf{gen}(1^\lambda)$, on input the security parameter, outputs a public key pk and a secret key sk.[6]

2. $\mathsf{commit}_{pk}(b, s)$, on input the public key pk, a bit $b \in \{0, 1\}$ (to commit to) and a string s, outputs the commitment z.

3. $\mathsf{verify}_{pk}(b, z, s)$, on input the public key pk, a bit $b \in \{0, 1\}$, a string s, and the commitment z, outputs "accept" or "reject."

The scheme based on LWE [63] satisfies the following properties.

1. Perfectly binding: if $\mathsf{commit}_{pk}(b, s) = \mathsf{commit}_{pk}(b', s')$, then $b = b'$.

[6]The secret key is never used.

2. Quantum computational concealing: for any quantum adversary $\mathcal{A}$,

$$\Pr\left[\mathcal{A}(pk,z)=b \;\middle|\; \begin{array}{c}(pk,sk)\leftarrow \mathsf{gen}(1^\lambda)\\ s\leftarrow\{0,1\}^\ell\\ z\leftarrow \mathsf{commit}_{pk}(b,s)\end{array}\right]\leq \frac{1}{2}+\mathrm{negl}(\lambda). \tag{2.2}$$

2.2.2 Fully homomorphic encryption with circuit privacy

A fully homomorphic encryption scheme $\mathsf{FHE}=(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec},\mathsf{Eval})$ with malicious circuit privacy [63, 73] consists of the following algoithms.

1. $\mathsf{Gen}(1^\lambda)$: a probabilistic algorithm that, on input 1^λ, outputs a secret key sk and a public key pk.

2. $\mathsf{Enc}_{pk}(x)$: a probabilistic algorithm that, on input the public key pk and message x, outputs a ciphertext.

3. $\mathsf{Dec}_{sk}(c)$: a deterministic algorithm that, on input the secret key sk and ciphertext c, outputs a message x.

4. $\mathsf{Eval}_{pk}(C,c_1,\ldots,c_m)$: a probabilistic algorithm that, on input the public key pk, a circuit description $C\in\mathcal{C}_\lambda$, and ciphertexts $c_1,\ldots,c_m$, outputs another ciphertext c'.

The scheme satisfies the following properties for any polynomial-sized classical circuits $\{\mathcal{C}_\lambda\}_\lambda$:

1. Correctness: for any $\lambda \in \mathbb{N}$, $m \in \{0,1\}^*$ and $C \in \mathcal{C}_\lambda$,

$$\Pr\left[\mathsf{Dec}_{sk}(c') = C(x) \;\middle|\; \begin{array}{c}(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda) \\ c \leftarrow \mathsf{Enc}_{pk}(m) \\ c' \leftarrow \mathsf{Eval}_{pk}(C, c)\end{array}\right] = 1. \tag{2.3}$$

2. Semantic security against quantum adversaries: for every $\lambda \in \mathbb{N}$, there exists a negligible function μ such that for any pair of messages m_0, m_1 of polynomial size and any quantum adversary $\mathcal{A}$,

$$\Pr\left[\mathcal{A}(pk, c) = b \;\middle|\; \begin{array}{c}b \leftarrow \{0,1\} \\ (pk, sk) \leftarrow \mathsf{Gen}(1^\lambda) \\ c \leftarrow \mathsf{Enc}_{pk}(m_b)\end{array}\right] \leq \frac{1}{2} + \mu(\lambda). \tag{2.4}$$

3. Malicious[7] circuit privacy: there exist unbounded algorithms Sim, Ext such that for any $x \in \{0,1\}^*$, possibly malformed pk^*, and $ct \leftarrow \mathsf{Enc}_{pk^*}(x)$, we have $\mathsf{Ext}_{pk^*}(1^\lambda, ct) = x$. Furthermore, for any C and possibly malformed pk^*, ct^*,

$$\mathsf{Eval}_{pk^*}(C, ct^*) \approx_s \mathsf{Sim}_{pk^*}(C(\mathsf{Ext}_{pk^*}(ct^*; 1^\lambda)); 1^\lambda), \tag{2.5}$$

   where $\approx_s$ denotes that the two distributions are statistically indistinguishable.

An FHE scheme with malicious circuit privacy [25] is known to exist, assuming that LWE is (quantum) computationally intractable.

[7] In contrast to the semi-honest counterpart, it is not required that the public key and the ciphertext to Eval are well-formed.

2.2.3 NIZK for NP

We will use NIZK protocols $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V}, \mathsf{S})$ for NP. The protocol is described in terms of the following algorithms.

1. (Setup) $\mathsf{Setup}(1^\lambda)$ outputs a common reference string crs on input 1^λ.

2. (Prove) $\mathsf{P}(\mathsf{crs}, x, u)$ outputs a string e on input crs, instance x, and witness u.

3. (Verify) $\mathsf{V}(\mathsf{crs}, x, e)$ outputs a bit $b \in \{0,1\}$ on input crs, instance x, and proof e.

4. (Simulate) $\mathsf{S}(x)$ outputs a transcript (crs, x, e).

We use the Peikert-Shiehian construction based on LWE [70]. For any NP language $\mathcal{L}$, the protocol satisfies the following:

1. Completeness: for $(x,u) \in \mathcal{R}_\mathcal{L}$,

$$\Pr\left[\mathsf{V}(\mathsf{crs}, x, e) = 1 \,\middle|\, \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ e \leftarrow \mathsf{P}(\mathsf{crs}, x, w) \end{array}\right] \geq 1 - \mathrm{negl}(n). \tag{2.6}$$

2. Adaptive soundness: for any quantum adversary $\mathcal{A}$,

$$\Pr\left[x \notin \mathcal{L} \wedge \mathsf{V}(\mathsf{crs}, x, e) = 1 \,\middle|\, \begin{array}{c} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x, e) \leftarrow \mathcal{A}(\mathsf{crs}) \end{array}\right] \leq \mathrm{negl}(n). \tag{2.7}$$

3. Zero-knowledge: for $(x,u) \in \mathcal{R}_\mathcal{L}$, the distributions $\{(\mathsf{crs}, \mathsf{P}(\mathsf{crs}, x, u))\}$ and $\{\mathsf{S}(x)\}$ are computationally indistinguishable.

Remark 2.2.1. *In a* classical *NIZK for* QMA*, there are several differences from the above: (i.) the witness provided to the prover is a quantum state $|\psi\rangle$ instead of the classical value u, (ii.)* P *is a quantum polynomial-time algorithm, with classical output, and (iii.) our NIZK protocol for* QMA *works in the random oracle model with setup, instead of the (desired) common reference string model.*

2.3 Preliminaries

2.3.1 Quantum-prover interactive arguments

A quantum-prover interactive argument (QPIA) is an interactive protocol between two polynomially-bounded parties, a quantum prover and a classical verifier, interacting over a classical channel. A QPIA is described by a pair of algorithms: the PPT algorithm of the honest verifier $\mathcal{V}$, and the QPT algorithm of the honest prover $\mathcal{P}$.

Definition 2.3.1. *Fix a language $L \subseteq \{0,1\}^*$ and a* QPIA *$(\mathcal{P}, \mathcal{V})$. We say that $(\mathcal{P}, \mathcal{V})$ is a* QPIA *for $\mathcal{L}$ with* completeness *c and* soundness error *s if*

- *for all $x \in L$, $\Pr[\mathcal{P}$ and $\mathcal{V}$ accept $x] \geq c$.*
- *for all $x \notin L$ and for all QPT algorithms $\mathcal{P}'$, $\Pr[\mathcal{P}'$ and $\mathcal{V}$ accept $x] \leq s$.*

The completeness error *is $1 - c$ and the* soundness *is $1 - s$.*

As discussed above, the soundness of a QPIA can be amplified via sequential repetition, and in some cases also with parallel repetition.

2.3.2 The local Hamiltonian problem and verification for BQP

Any promise problem $L = (L_{\text{yes}}, L_{\text{no}}) \in \text{QMA}$ can be reduced to the local Hamiltonian problem such that for $x \in L_{\text{yes}}$, the Hamiltonian H_x has a low-energy ground state $|\psi_x\rangle$, and for $x \in L_{\text{no}}$, all quantum states have large energy [74]. While the quantum witness $|\psi_x\rangle$ may be hard to prepare for general $L \in \text{QMA}$, it can be prepared efficiently if $L \in \text{BQP}$. Furthermore, the problem remains QMA-complete even with a Hamiltonian that can be measured by performing standard (Z) and Hadamard (X) basis measurements [75, 76].

Problem 1 (The 2-local ZX-Hamiltonian problem [47, 75, 76])**.** *The 2-local ZX-Hamiltonian promise problem* $\text{ZX} = (\text{ZX}_{\text{yes}}, \text{ZX}_{\text{no}})$*, with parameters* $a, b \in \mathbb{R}$*,* $b > a$ *and gap* $(b - a) > \text{poly}(n)^{-1}$*, is defined as follows. An instance is a local Hamiltonian*

$$H = \sum_{i<j} J_{ij}(X_i X_j + Z_i Z_j) \tag{2.8}$$

where each J_{ij} *is a real number such that* $2\sum_{i<j} |J_{ij}| = 1$ *and each* X_i *(resp.* Z_i*) is a Pauli* X *(resp. Pauli* Z*) gate acting on the* i*th qubit. For* $H \in \text{ZX}_{\text{yes}}$*, the smallest eigenvalue of* H *is at most* a*, while if* $H \in \text{ZX}_{\text{no}}$*, the smallest eigenvalue of* H *is at least* b*.*

Note that given the normalization factors, we can see that each term ($X_i X_j$ or $Z_i Z_j$) is associated with the probability J_{ij}. We denote the 2-local ZX-Hamiltonian problem with parameters $a, b \in \mathbb{R}$ by $\text{ZX}_{a,b}$.

When working with Hamiltonian terms S, we overload the notation for convenience. First, we write S_j to denote the Pauli operator assigned by S to qubit j, so that $S = \bigotimes_j S_j$. Second, we write $i \in S$ to indicate that i is a qubit index for which S does not act as the

identity, i.e., $S_i \neq \mathbb{1}$.

Morimae and Fitzsimons present a protocol (the "MF protocol") with a quantum prover and a limited verifier who only needs the ability to perform single-qubit X and Z basis measurements [47]. The prover $\mathcal{P}$ prepares the ground state of the Hamiltonian and sends it to $\mathcal{V}$, who then samples a term S with probability p_S and performs the corresponding measurement. Notice that to estimate the energy of term S, only Z or X basis measurements are necessary. In the original protocol of [47], the qubits are sent individually; in the variant below, we simply have the prover send the entire state all at once.

Setup. $\mathcal{P}$ and $\mathcal{V}$ receive an instance of Problem 1, namely a Hamiltonian

$$H_x = \sum_S p_S \frac{\mathbb{1} + m_S S}{2}, \tag{2.9}$$

where each S is a tensor product of $X, Z, \mathbb{1}$ and $m_S \in \{\pm 1\}$.

Round 1. $\mathcal{P}$ prepares a quantum state ρ and sends it to $\mathcal{V}$.

Verdict. $\mathcal{V}$ samples a term S with probability p_S and performs the measurement $\{M_1 = \frac{\mathbb{1}+S}{2}, M_{-1} = \frac{\mathbb{1}-S}{2}\}$ on ρ, getting an outcome e. $\mathcal{V}$ accepts if $e = -m_S$.

Protocol 2.1: A variant of the MF protocol.

Since $M_{m_S} = \frac{\mathbb{1}+m_S S}{2} = \mathbb{1} - M_{-m_S}$ and $H_x = \sum_S p_S M_{m_S}$, the success probability of the protocol with input state ρ is

$$\sum_S p_S \operatorname{tr}(M_{-m_S}\rho) = 1 - \sum_s p_S \operatorname{tr}(M_{m_s}\rho) = 1 - \operatorname{tr}(H_x\rho). \tag{2.10}$$

Since $b - a > \text{poly}(|x|)^{-1}$, the error in the MF protocol can be made negligible by parallel repetition: $\mathcal{V}$ receives T copies of the ground state of H and performs an independent test on

each copy. By accepting if at least $(2-a-b)T/4$ copies accept, both the completeness and soundness errors are suppressed to negligible with polynomial $T(|x|)$ (cf. [6, Theorem 8.4]). For a detailed proof of QMA gap amplification, see [77, Section 3].

In the following discussion, the term S is encoded by an n-bit string $h(S)$: for each qubit $i \in S$, set $h_i = 0$ for a Z basis measurement and $h_i = 1$ for an X basis measurement. For other qubits, the choice is irrelevant but we set $h_i = 0$ for concreteness. We let $\alpha_{h,\rho} := \mathrm{tr}(M_{-m_S}\rho)$ denote the success probability with ρ when $h = h(S)$ is sampled in Protocol 2.1.

2.3.3 The Mahadev protocol for BQP verification

Required primitives. The protocol relies crucially on two special classes of functions: Noisy Trapdoor Claw-free Functions (NTCFs) $\mathbb{F}$ and Noisy Trapdoor Injective Functions (NTIFs) $\mathcal{G}$. Both classes of functions are constructed based on the presumed hardness of the Learning with Errors (LWE) problem [6, 9]. We now sketch the properties of these function families. For complete details, and for the LWE construction, see [9]. Let λ be a security parameter and let $q \geq 2$ be prime. Choose parameters $\ell = \mathrm{poly}(\lambda)$, $n = \Omega(\ell \log q)$, and $m = \Omega(n \log q)$.

The NTCF family $\mathcal{F} = \{f_{pk}\}_{pk \in \mathcal{K}_\mathcal{F}}$ is a family of keyed functions

$$f_{pk} \colon \{0,1\} \times \mathcal{X} \to D_\mathcal{Y} \tag{2.11}$$

which, on input a public key $pk \in \mathcal{K}_\mathcal{F} := \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$, a bit b, and $x \in \mathcal{X} := \mathbb{Z}_q^n$, outputs a distribution $f_{pk}(b,x)$ over $\mathcal{Y} := \mathbb{Z}_q^m$. Each function $f_{pk} \in \mathcal{F}$ satisfies the *injective pair* property: there exists a perfect matching $\mathcal{R}_{pk} \subset \mathcal{X} \times \mathcal{X}$ such that $f_{pk}(0,x_0) = f_{pk}(1,x_1)$ if and only if $(x_0,x_1) \in \mathcal{R}_{pk}$.

The NTCF family is equipped with the following polynomial-time algorithms:

1. $\mathsf{Gen}_{\mathcal{F}}$, on input 1^λ, outputs a secret-public key pair (pk, sk).

2. $\mathsf{Chk}_{\mathcal{F}}$ is a deterministic algorithm for checking if (b, x) and y form a preimage-image pair of f_{pk}. On input b, x, y, $\mathsf{Chk}_{\mathcal{F}}$ outputs 1 iff $y \in \mathrm{supp}(f_{pk}(b, x))$.

3. $\mathsf{Inv}_{\mathcal{F}}$ is a deterministic algorithm for inverting the function f_{pk}. On input secret key sk, bit b, and image y, $\mathsf{Inv}_{\mathcal{F}}$ returns the preimage $x_{b,y}$ such that $y \in \mathrm{supp}(f_{pk}(b, x_{b,y}))$, or outputs reject if no such preimage exists.

4. $\mathsf{Samp}_{\mathcal{F}}$ is an efficient quantum process which, on input pk and $b \in \{0, 1\}$, returns a quantum state negligibly close to

$$\frac{1}{|\mathcal{X}|^{1/2}} \sum_{x \in \mathcal{X}} |b\rangle|x\rangle|\psi_{f_{pk}(b,x)}\rangle, \tag{2.12}$$

where $|\psi_p\rangle := \sum_{y \in \mathcal{Y}} \sqrt{p(y)}|y\rangle$ for distribution p. By the injective pair property, we have $\langle \psi_{f_{pk}(b,x)} | \psi_{f_{pk}(b',x')} \rangle = 1$ if $(b, x) = (b', x')$, or there exists $(x_0, x_1) \in \mathcal{R}_{pk}$ such that $(b, x, b', x') = (0, x_0, 1, x_1)$ or $(1, x_1, 0, x_0)$. This implies that the states in $\{|\psi_{f_{pk}(b,x)}\rangle\}$ can be perfectly distinguished by performing a standard basis measurement. Thus, intuitively, we may consider an *ideal* version of these functions, i.e., the distribution p is concentrated at a single point. The state in (2.12) describes the encoding of a single qubit. For encoding a quantum state $|b_1, \ldots, b_n\rangle$ using n public keys $pk = (pk_1, \ldots, pk_n)$, we run $\mathsf{Samp}_{\mathcal{F}}$ in parallel. For convenience, we define the distribution $f_{pk}(b, x)$ of density $f_{pk}(b, x)(y) := \prod_{i=1}^n f_{pk_i}(b_i, x_i)(y_i)$ and the state $|\psi_{f_{pk}(b,x)}\rangle := \bigotimes_{i=1}^n |\psi_{f_{pk_i}(b_i,x_i)}\rangle$ for each b, x.

Similarly, the NTIF family $\mathcal{G} = \{g_{pk}\}_{pk \in \mathcal{K}_\mathcal{G}}$ is a family of keyed functions

$$g_{pk} \colon \{0,1\} \times \mathcal{X} \to D_\mathcal{Y} \tag{2.13}$$

which, on input a public key $pk \in \mathcal{K}_\mathcal{G}$, a bit b, and $x \in \mathcal{X}$, outputs a distribution $g_{pk}(b,x)$ over $\mathcal{Y}$. Instead of the injective pair property of NTCFs, NTIFs satisfy an *injectivity* property: for all $(x,b) \neq (x',b')$, $\mathrm{supp} g_{pk}(b,x) \cap \mathrm{supp} g_{pk}(b',x') = \emptyset$. An NTIF family is also equipped with a tuple of four polynomial-time algorithms $(\mathsf{Gen}_\mathcal{G}, \mathsf{Chk}_\mathcal{G}, \mathsf{Inv}_\mathcal{G}, \mathsf{Samp}_\mathcal{G})$, defined exactly as in the NTCF case (but with g in place of f, and $\mathcal{G}$ instead of $\mathcal{F}$.) For convenience, when describing the encoding of n-qubit state using public keys $pk = (pk_1, \ldots, pk_n)$, we define the distribution $g_{pk}(b,x)$ for of density $g_{pk}(b,x)(y) := \prod_{i=1}^n g_{pk_i}(b_i,x_i)(y_i)$ and the state $|\psi_{g_{pk}(b,x)}\rangle := \bigotimes_{i=1}^n |\psi_{g_{pk_i}(b_i,x_i)}\rangle$ for each b, x.

We remark that the states (2.12) prepared by $\mathsf{Samp}_\mathcal{F}$ and $\mathsf{Samp}_\mathcal{G}$ should be contrasted with the "idealized" state described in (2.1) in our sketch of the protocol.

The protocol. The Mahadev protocol [6] for BQP verification allows $\mathcal{V}$ to request an X or Z basis measurement outcome without revealing the basis to $\mathcal{P}$. The aim of the protocol is to verify that the prover's response, when appropriately decoded, is close to the measurement outcomes of some n-qubit quantum state ρ. Crucially, this guarantee holds simultaneously for all basis choices $h \in \{0,1\}^n$, where 0 denotes a Z basis measurement and 1 denotes an X basis measurement. With this guarantee, the verifier can then apply the verification procedure of the MF protocol to the decoded responses of the prover, knowing that this will correctly decide whether the instance should be accepted or rejected.

In the following protocol, for each qubit, if $\mathcal{V}$ requests a Z basis measurement, then an NTIF key is sent; if $\mathcal{V}$ requests an X basis measurement, then an NTCF key is sent. Since $\mathsf{Chk}_{\mathcal{F}}$ and $\mathsf{Chk}_{\mathcal{G}}$ are identical, we denote them by Chk. Similarly, $\mathsf{Samp}_{\mathcal{F}}$ and $\mathsf{Samp}_{\mathcal{G}}$ are identical, so we denote them by Samp.[8] We let $\mathsf{Gen}(1^\lambda, h)$ for $h \in \{0,1\}^*$ denote the following key generation algorithm: for every bit i of h, run $(pk_i, sk_i) \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^\lambda)$ if $h_i = 0$ and $(pk_i, sk_i) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda)$ if $h_i = 1$. Set $pk = (pk_i)_i$ and $sk = (sk_i)_i$ and output the key pairs (pk, sk).

We now the describe the protocol when $\mathcal{V}$ and $\mathcal{P}$ are honest.

Theorem 2.3.1 (Theorems 1.1 and 8.6 in [6])**.** *Under the* LWE *assumption, Protocol 2.2 is a four-message quantum-prover interactive argument for the class* BQP *with completeness error* $\mathrm{negl}(n)$ *and soundness error* $3/4 + \mathrm{negl}(n)$.

2.4 Instance-independent key generation

In this section, we modify the MF protocol such that the sampling of the Hamiltonian term is independent of the performed measurements. This change allows the keys in the Mahadev protocol to be generated before the parties receive the input Hamiltonian, in an offline setup phase.

In our variant, for some $r = \mathrm{poly}(n)$, the verifier $\mathcal{V}$ samples r n-bit strings $h_1, \ldots, h_r$ uniformly and r independent 2-local terms $S_1, \ldots, S_r$ according to distribution π (in which S is sampled with the probability p_S from Protocol 2.1). We say the bases h_i and the terms S_i are *consistent* if, when the observable for the jth qubit in S_i is Z (resp., X) then the jth

[8] These algorithms are identical from the instantiation of NTCF/NTIF functions based on LWE, as described in [6].

Setup. Choose a security parameter $\lambda \geq n$. Both $\mathcal{P}$ and $\mathcal{V}$ receive an instance of Problem 1, namely $H = \sum_S p_S \frac{\mathbb{1}+m_S S}{2}$.

Round $\mathcal{V}_1$. $\mathcal{V}$ samples r terms $S = (S_1, \ldots, S_r)$ and computes $h = h(S)$, the concatenation of $h(S_1), \ldots, h(S_r)$. $\mathcal{V}$ generates the key pair $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda, h)$ and sends pk to $\mathcal{P}$.

Round $\mathcal{P}_1$. $\mathcal{P}$ prepares r copies of the n-qubit ground state $|\phi\rangle^{\otimes r} = \sum_{b\in\{0,1\}^{nr}} \phi_b |b\rangle_W$ of H in register W. For $j \in [r], \ell \in [n]$ and each qubit $W_{j\ell}$ in W, $\mathcal{P}$ performs Samp on input the key $pk_{j\ell}$ coherently and yields a quantum state negligibly close to

$$\frac{1}{|\mathcal{X}|^{n/2}} \sum_{x\in\mathcal{X}^n} \sum_{b\in\{0,1\}^{nr}} \phi_b |b\rangle_W |x\rangle_X |\psi_{f_{pk}(b,x)}\rangle_Y \,, \tag{2.14}$$

where $|\psi_{f_{pk}(b,x)}\rangle := \bigotimes_{j\in[r],\ell\in[n]} |\psi_{f_{pk_{j\ell}}(b_{j\ell},x_{j\ell})}\rangle$. Next, $\mathcal{P}$ measures Y and sends the outcome y to $\mathcal{V}$.

Round $\mathcal{V}_2$. $\mathcal{V}$ responds with a uniformly random "challenge" bit $c \in \{0,1\}$. We call $c = 0$ "test round" and $c = 1$ "Hadamard round," and set labels $\mathfrak{t} = 0$ and $\mathfrak{h} = 1$.

Round $\mathcal{P}_2$. If $c = \mathfrak{t}$, $\mathcal{P}$ measures WX in the computational basis. If $c = \mathfrak{h}$, $\mathcal{P}$ measures WX in the Hadamard basis. In either case, $\mathcal{P}$ sends the measurement outcome (w, t) to $\mathcal{V}$;

Verdict. In a test round, $\mathcal{V}$ accepts if $\bigwedge_{j\in[r],\ell\in[n]} \mathsf{Chk}(pk_{j\ell}, w_{j\ell}, t_{j\ell}, y_{j\ell}) = 1$.

In a Hadamard round, $\mathcal{V}$ performs the following: for each copy j and qubit $\ell \in [n]$,

1. if $h_{j\ell} = 0$, run $(e_{j\ell}, x_{b_{j\ell},y_{j\ell}}) \leftarrow \mathsf{Inv}_{\mathcal{G}}(sk_{j\ell}, y_{j\ell})$. If $h_{j\ell} = 1$, run $x_{0,y_{j\ell}} \leftarrow \mathsf{Inv}_{\mathcal{F}}(sk_{j\ell}, 0, y_{j\ell})$ and $x_{1,y_{j\ell}} \leftarrow \mathsf{Inv}_{\mathcal{F}}(sk_{j\ell}, 1, y_{j\ell})$ to get both preimages. Then compute $e_{j\ell} = t_{j\ell} \cdot (x_{0,y_{j\ell}} \oplus x_{1,y_{j\ell}}) \oplus w_{j\ell}$.
   If any of the above algorithms returns reject for any j, ℓ, or $t_{j\ell}$ is trivial (e.g., 0^n, see [9]), reject. Otherwise store $e = (e_{j\ell})_{j\in[r],\ell\in[n]}$ as the witness to the next check.
2. $\mathcal{V}$ sets $v_j = 1$ if the witness $(e_{j\ell})_{\ell\in S_j}$ satisfies $M_{-m_{S_j}}$ (defined in Protocol 2.1).[a]

Finally, $\mathcal{V}$ accepts if $\sum_{j\in[r]} v_j \geq (2 - a - b)r/4$.

[a] A string e satisfies a projector M if $\langle e|M|e\rangle = 1$, i.e., $|e\rangle$ lies in the support of M, by convention.

Protocol 2.2: The Mahadev protocol.

bit of h_i is 0 (resp., 1). Since h_i is uniformly sampled and S_i is 2-local, we have

$$\Pr_{S_i \leftarrow \pi, h_i \leftarrow \{0,1\}^n}[S_i \text{ and } h_i \text{ are consistent}] \geq \frac{1}{4}. \tag{2.15}$$

In an r-copy protocol, we let $A := \{i \in [r] : h_i \text{ and } S_i \text{ are consistent}\}$ and denote $t = |A|$. For each $i \in A$, $\mathcal{V}_i$ decides as in the MF protocol: if $i \notin A$, then $\mathcal{V}_i$ accepts. Thus we consider the following protocol. For sufficiently large r, with high probability, there are around

Setup.	$\mathcal{V}$ samples the bases $h_1, \ldots, h_r \leftarrow \{0,1\}^n$ uniformly.
Round 1.	$\mathcal{P}$ sends the witness state ρ (r copies of the ground state).
Round 2.	$\mathcal{V}$ measures the quantum state ρ in the bases $h_1, \ldots, h_r$. For each copy $i \in [r]$, $\mathcal{V}$ samples terms $S_1, \ldots, S_r \leftarrow \pi$. $\mathcal{V}$ records the subset $A \subseteq [r]$ of consistent copies. For each copy $i \in A$, $\mathcal{V}$ sets $v_i = 1$ if the outcome satisfies M_{-m_S} and 0 otherwise. $\mathcal{V}$ accepts if $\sum_{i \in A} v_i \geq (2 - a - b)|A|/4$.

Protocol 2.3: A modified parallel-repeated MF protocol for $\text{ZX}_{a,b}$.

$r/4$ consistent copies. Thus to achieve the same completeness and soundness, it suffices to increase the number of copies by a constant factor. We thus have the following fact.

Lemma 2.4.1. *The completeness error and soundness error of Protocol 2.3 are negligible, provided $r = \omega\big(\frac{\log n}{(b-a)^2}\big)$ copies are used.*

Proof. First we observe that for each copy, with probability 1/4, $\mathcal{V}$ measures the quantum state with a term sampled from the distribution π; otherwise $\mathcal{V}$ accepts. Thus for an instance H, the effective Hamiltonian to verify is $\widetilde{H}^{\otimes r}$ where $\widetilde{H} = \frac{3\mathbb{1}+H}{4}$. Following the standard parallel repetition theorem for QMA, we know that $\mathcal{P}$'s optimal strategy is to present the the ground state of $\widetilde{H}$, which is also the ground state of H.

With probability $\binom{r}{t}(\frac{1}{4})^t(\frac{3}{4})^{r-t}$, there are t consistent copies. Now for $i \in A$, we let X_i be a binary random variable corresponding to the decision of $\mathcal{V}_i$. For soundness, by Hoeffding's inequality[9] the success probability for A such that $|A| = t$ is

$$\begin{aligned}\Pr[\text{accept}|A] &= \Pr\left[\frac{1}{t}\sum_{i\in A} X_i \geq \frac{c+s}{2}\right] \\ &\leq \Pr\left[\frac{1}{t}\sum_{i\in A} X_i - s \geq \frac{c-s}{2}\right] \leq 2e^{-\frac{tg^2}{2}},\end{aligned} \tag{2.16}$$

where $g = c - s$ is the promise gap. Then the overall success probability is

$$\begin{aligned}\Pr[\text{accept}] &= 2\cdot 4^{-r}\sum_{t=0}^{r}\binom{r}{t}3^{r-t}e^{-tg^2/2} \\ &= 2\left(\frac{e^{-g^2/2}+3}{4}\right)^r \leq 2(1-g^2/16)^r \leq 2e^{-rg^2/16}\end{aligned} \tag{2.17}$$

since $1 - x/2 \geq e^{-x}$ for $x \in [0, 1]$ and $1 - x \leq e^{-x}$ for $x \geq 0$. Thus $r = \omega(g^{-2}\log n)$ suffices to suppress the soundness error to $n^{-\omega(1)}$. Since $g^{-1} = \text{poly}(n)$, polynomially many copies suffice to achieve negligible soundness error.

For completeness, again by Hoeffding's inequality,

$$\begin{aligned}\Pr[\text{reject}|A] &= \Pr\left[\frac{1}{t}\sum_{i\in A} X_i < \frac{c+s}{2}\right] \\ &\leq \Pr\left[c - \frac{1}{t}\sum_{i\in A} X_i > \frac{c-s}{2}\right] \leq 2e^{-\frac{tg^2}{2}}.\end{aligned} \tag{2.18}$$

By the same calculation as in (2.17), the completeness error is negligible if we set $r = \omega(g^{-2}\log n)$. □

[9] $\Pr[\frac{1}{n}\sum_i X_i - \mu \geq \delta] \leq e^{-2t\delta^2}$ for i.i.d. $X_1, \ldots, X_n \in [0, 1]$.

Remark 2.4.1. *We stress that the terms S_i are sampled* independently *of the interaction in the protocol. We let* $\mathsf{term}(H, s)$ *denote the deterministic algorithm that outputs a term from H according to distribution π when provided the randomness $s \in \{0,1\}^p$ for sufficiently large polynomial p. For bases $h \in \{0,1\}^{nr}$ and $s \in \{0,1\}^p$, we let $\alpha_{h,s,\rho}$ denote the success probability when $\mathcal{P}$ presents quantum state ρ.*

We embed Protocol 2.3 into our classical verification protocol by modifying the Mahadev protocol (Protocol 2.2) as follows. First, note that the measurement in random bases h can be achieved by sending random keys, so the key generation can be done before the parties receive the instance. In the Verdict stage, if the protocol enters a test round (i.e., $c = 0$), then $\mathcal{V}$ checks as in Protocol 2.2. If the protocol enters a Hadamard round (i.e., $c = 1$), $\mathcal{V}$ uses the secret key to compute the measurement outcome, as in check 1 of the Verdict stage of Protocol 2.2. Once the outcomes are successfully computed, $\mathcal{V}$ samples terms $S_1, \ldots, S_r \leftarrow \pi$ and checks the consistent copies. Since the outcome must be computationally indistinguishable from measuring a quantum state in bases h, Lemma 2.4.1 applies, and Theorem 2.3.1 holds for our variant protocol. We refer to this variant as "the three-round Mahadev protocol" and denote it by $\mathfrak{M}$.

2.5 A parallel repetition theorem for the Mahadev protocol

In a k-fold parallel repetition of $\mathfrak{M}$, the honest prover runs the honest single-fold prover independently for each copy of the protocol. Meanwhile, the honest verifier runs the single-fold verifier independently for each copy, accepting if and only if all k verifiers accept. The completeness error clearly remains negligible. We now analyze the soundness error and

establish a parallel repetition theorem.

In preparation, we fix the following notation related to the Verdict stage of $\mathfrak{M}$. We will refer frequently to the notation established in our description of Protocol 2.2 above, which applies to $\mathfrak{M}$ as well. First, the check $\bigwedge_{j\in[r],\ell\in[n]} \mathsf{Chk}(pk_{j\ell}, w_{j\ell}, t_{j\ell}, y_{j\ell}) = 1$ in a test round is represented by a projection $\Pi_{sk,\mathfrak{t}}$ acting on registers WXY. Specifically, this is the projector whose image is spanned by all inputs (w, t, y) that are accepted by the verifier in the Verdict stage. Note that running Chk does not require the trapdoor sk, but the relation implicitly depends on it. For notational convenience, we will also denote $\Pi_{sk,\mathfrak{t}}$ as $\Pi_{s,sk,\mathfrak{t}}$, though the projector does not depend on s.

Second, the two Hadamard round checks 1 and 2 of the Verdict stage are represented by projectors $\Lambda_{sk,\mathfrak{h},1}$ and $\Lambda_{s,sk,\mathfrak{h},2}$, respectively. These two projectors commute since they are both diagonal in the standard basis. We define the overall Hadamard round projector $\Pi_{s,sk,\mathfrak{h}} := \Lambda_{sk,\mathfrak{h},1}\Lambda_{s,sk,\mathfrak{h},2}$.

2.5.1 A lemma for the single-copy protocol

We begin by showing an important fact about the single-copy protocol: the verifier's accepting paths associated to the two challenges (denoted $\mathfrak{t}$ and $\mathfrak{h}$ for "test" and "Hadamard," respectively) correspond to nearly orthogonal[10] projectors. Moreover, in a certain sense this property holds even for input states that are adaptively manipulated by a dishonest prover after they have learned which challenge will take place. This fact is essential in our analysis of the parallel repetition of many copies in the following sections.

[10]Strictly speaking, the projectors are only nearly orthogonal when applied to states prepared by efficient provers.

The setup. As discussed in [6], any prover $\mathcal{P}$ can be characterized as follows. First, pick a state family $|\Psi_{pk}\rangle$; this state is prepared on registers $WXYE$ after receiving pk. Here Y is the register that will be measured in Round $\mathcal{P}_1$, W and X are the registers that will be measured in Round $\mathcal{P}_2$, and E is the private workspace of $\mathcal{P}$. Then, choose two unitaries $U_{\mathfrak{t}}$ and $U_{\mathfrak{h}}$ to describe the Round $\mathcal{P}_2$ actions of $\mathcal{P}$ before any measurements, in the test round and Hadamard round, respectively. Both $U_{\mathfrak{t}}$ and $U_{\mathfrak{h}}$ act on $WXYE$, but can only be classically controlled on Y, as they must be implemented after $\mathcal{P}$ has measured Y and sent the result to the verifier. (Of course, a cheating prover is not constrained to follow the honest protocol, but we can nevertheless designate a fixed subsystem Y that carries their message.) We will write $\mathcal{P} = (|\Psi_{pk}\rangle, U_{\mathfrak{t}}, U_{\mathfrak{h}})$, where it is implicit that $|\Psi_{pk}\rangle$ is a family of states parameterized by pk.

At the end of the protocol, the registers WXY are measured and given to the verifier. Recall that we can view the final actions of the verifier as applying one of two measurements: a test-round measurement or a Hadamard-round measurement. Let $\Pi_{s,sk,\mathfrak{t}}$ and $\Pi_{s,sk,\mathfrak{h}}$ denote the "accept" projectors for those measurements, respectively. For a given prover $\mathcal{P}$, we additionally define

$$\begin{aligned}\Pi^{U_{\mathfrak{t}}}_{s,sk,\mathfrak{t}} &:= U_{\mathfrak{t}}^{\dagger}(\Pi_{s,sk,\mathfrak{t}} \otimes \mathbb{1}_E)U_{\mathfrak{t}}\,,\\ \Pi^{U_{\mathfrak{h}}}_{s,sk,\mathfrak{h}} &:= U_{\mathfrak{h}}^{\dagger}(H_{WX}\Pi_{s,sk,\mathfrak{h}}H_{WX} \otimes \mathbb{1}_E)U_{\mathfrak{h}}\,,\end{aligned} \tag{2.19}$$

where H_{WX} denotes the Hadamard transform on registers WX, i.e., the Hadamard gate applied to every qubit in those registers. These projectors have a natural interpretation: they describe the action of the two accepting projectors of the verifier on the initial state

$|\Psi_{pk}\rangle$ of the prover, taking into account the (adaptive) attacks the prover makes in Round $\mathcal{P}_2$.

A key lemma. We now prove a fact about the single-copy protocol. The proof is largely a matter of making some observations about the results from [6], and then combining them in the right way.

Recall that, after the setup phase, for any instance H of the ZX-Hamiltonian problem (Problem 1), $\mathfrak{M}$ begins with the verifier $\mathcal{V}$ making a measurement basis choice $h \in \{0,1\}^{nr}$ for all the qubits. After interacting with a prover $\mathcal{P}$, the verifier either rejects or produces a candidate measurement outcome, which is then tested as in Protocol 2.3. We let $D_{\mathcal{P},h}$ denote the distribution of this candidate measurement outcome for a prover $\mathcal{P}$ and basis choice h, averaged over all measurements and randomness of $\mathcal{P}$ and $\mathcal{V}$. It is useful to compare $D_{\mathcal{P},h}$ with an "ideal" distribution $D_{\rho,h}$ obtained by simply measuring some (nr)-qubit quantum state ρ (i.e., a candidate ground state) according to the basis choices specified by h, with no protocol involved.

Lemma 2.5.1. *Let $\mathcal{P} = (|\Psi_{pk}\rangle, U_{\mathfrak{t}}, U_{\mathfrak{h}})$ be a prover in $\mathfrak{M}$ such that, for every $h \in \{0,1\}^{nr}$ and $s \in \{0,1\}^{p}$,*

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow \mathsf{Gen}(1^\lambda,h)} [\langle \Psi_{pk}| \Pi^{U_{\mathfrak{t}}}_{s,sk,\mathfrak{t}} |\Psi_{pk}\rangle] \geq 1 - \mathrm{negl}(n)\,. \tag{2.20}$$

Then there exists an (nr)-qubit quantum state ρ such that, for every h, s,

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow \mathsf{Gen}(1^\lambda,h)} [\langle \Psi_{pk}| \Pi^{U_{\mathfrak{h}}}_{s,sk,\mathfrak{h}} |\Psi_{pk}\rangle] \leq \alpha_{h,s,\rho} + \mathrm{negl}(n)\,, \tag{2.21}$$

where $\alpha_{h,s,\rho}$ (see Remark 2.4.1) is the success probability in the MF protocol with basis choice

h and quantum state ρ.

Proof. Up to negligible terms, (2.20) means that $\mathcal{P}$ is what Mahadev calls a *perfect prover*. She establishes two results ([6, Claim 7.3] and [6, Claim 5.7]) which, when taken together, directly imply the following fact about perfect provers. For every perfect prover $\mathcal{P}$, there exists an efficiently preparable quantum state ρ such that $D_{\mathcal{P},h}$ is computationally indistinguishable from $D_{\rho,h}$ for all basis choices $h \in \{0,1\}^{nr}$. In particular, the proof is obtained in two steps. First, for every perfect prover, there exists a nearby "trivial prover" whose attack in a Hadamard round commutes with standard basis measurement on the committed state [6, Claim 5.7]. Second, for every trivial prover, the distribution is computationally indistinguishable from measuring a consistent quantum state ρ in any basis h [6, Claim 7.3]. Mahadev shows this for exactly perfect provers, but the proofs can be easily adapted to our "negligibly-far-from-perfect" case.

Now consider two ways of producing a final accept/reject output of the verifier. In the first case, an output is sampled from the distribution $D_{\mathcal{P},h}$ and the verifier applies the final checks in $\mathfrak{M}$. In this case, the final outcome is obtained by performing the measurement $\{\Pi^{U_\mathfrak{h}}_{s,sk,\mathfrak{h}}, \mathbb{1} - \Pi^{U_\mathfrak{h}}_{s,sk,\mathfrak{h}}\}$ on the state $|\Psi_{pk}\rangle$, and accepting if the first outcome is observed. In the second case, an output is sampled from the distribution $D_{\rho,h}$ and the verifier applies the final checks in the MF protocol. In this case, the acceptance probability is $\alpha_{h,s,\rho}$ simply by definition. The result then follows directly. □

Notice that for the soundness case, there is no state that succeeds non-negligibly in the MF protocol. In this case, Lemma 2.5.1 implies that for perfect provers the averaged projection $\mathbb{E}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h),h,s}[\langle\Psi_{pk}|\Pi^{U_\mathfrak{h}}_{s,sk,\mathfrak{h}}|\Psi_{pk}\rangle]$ is negligible. In other words, provers who

succeed almost perfectly in the test round must almost certainly fail in the Hadamard round. We emphasize that this is the case even though the prover can adaptively change their state (by applying $U_{\mathfrak{t}}$ or $U_{\mathfrak{h}}$) after learning which round will take place. This formalizes the intuitive claim we made at the beginning of the section about "adaptive orthogonality" of the two acceptance projectors corresponding to the two round types.

2.5.2 The parallel repetition theorem

Characterization of a prover in the k-fold protocol. We now discuss the behavior of a general prover in a k-fold protocol. We redefine some notation, and let $\mathcal{V}$ be the verifier and $\mathcal{P}$ an arbitrary prover in the k-fold protocol.

In the Setup phase, the key pairs $(pk_1, sk_1), \ldots, (pk_k, sk_k)$ are sampled according to the correct NTCF/NTIF distribution.[11] The secret keys $sk = (sk_1, \ldots, sk_k)$ and the corresponding bases h are given to $\mathcal{V}$, whereas $pk = (pk_1, \ldots, pk_k)$ is given to $\mathcal{P}$ (in the register $PK = (PK_1, \ldots, PK_k)$).

In Round $\mathcal{P}_1$, without loss of generality, the action of $\mathcal{P}$ prior to measurement is to apply a unitary $U_0 = \sum_{pk} |pk\rangle\langle pk|_{PK} \otimes (U_{0,pk})_{WXYE}$ to the input state $|pk\rangle_{PK}|0\rangle_{WXYE}$. Each of W, X, Y is now a k-tuple of registers, and E is the prover's workspace. To generate the "commitment" message to $\mathcal{V}$, $\mathcal{P}$ performs standard basis measurement on Y. We write $|\Psi_{pk}\rangle_{WXYE} = \sum_y \beta_y |\Psi_{pk,y}\rangle_{WXE}|y\rangle_Y$. When the measurement outcome is y, the side state $\mathcal{P}$ holds is then $|\Psi_{pk,y}\rangle_{WXE}$. In the following analysis of the success probability of $\mathcal{P}$, we consider the superposition $|\Psi_{pk}\rangle_{WXYE}$ instead of a classical mixture of the states $|\Psi_{pk,y}\rangle_{WXE}$ using the principle of deferred measurement.

[11] Recall that the keys are sampled by choosing uniform bases h followed by running $\mathsf{Gen}(1^\lambda, h)$.

In Round $\mathcal{V}_2$, after receiving the commitment $y = (y_1, \ldots, y_k)$ from the prover, $\mathcal{V}$ sends challenge coins $c := (c_1, \ldots, c_k) \in \{0,1\}^k$. For the remainder of the protocol, we take the following point of view. We assume that $\mathcal{P}$ and $\mathcal{V}$ share access to a register Y whose state is fixed forever to be the standard basis state y. This is clearly equivalent to the real situation, as there $\mathcal{P}$ measured Y and committed to the outcome y by sending it to $\mathcal{V}$.

In Round $\mathcal{P}_2$, without loss of generality, the action of $\mathcal{P}$ consists of a general operation (that can depend on c), followed by the honest action. The general operation is some efficient unitary U_c on $WXYE$. The honest action is measurement in the right basis, i.e., for each i, W_iX_i is measured in the standard basis (if $c_i = 0$) or the Hadamard basis (if $c_i = 1$). Equivalently, the honest action is *(i.)* apply $\mathfrak{H}^c_{WX} := \bigotimes_{i=1}^k (H^{c_i})_{W_iX_i}$, i.e., for each $\{i : c_i = 1\}$ apply a Hadamard to every qubit of W_iX_i, and then *(ii.)* apply standard basis measurement.

In the Verdict stage, $\mathcal{V}$ first applies for each i the two-outcome measurement corresponding to the Π_{s_i,sk_i,c_i} from the single-copy protocol. The overall decision is then to accept if the measurements accept for all i. We let

$$(\Pi_{s,sk,c})_{WXY} := \bigotimes_{i=1}^{k} (\Pi_{s_i,sk_i,c_i})_{W_iX_iY_i} \tag{2.22}$$

denote the corresponding acceptance projector for the entire k-copy protocol. The effective measurement on $|\Psi_{pk}\rangle_{WXYE}$ is then described by the projection

$$\left(\Pi^{U_c}_{s,sk,c}\right)_{WXYE} := (U_c^\dagger)_{WXYE}(\mathfrak{H}^c \Pi_{s,sk,c,y} \mathfrak{H}^c \otimes \mathbb{1}_E)(U_c)_{WXYE}\,. \tag{2.23}$$

The success probability of $\mathcal{P}$, which is characterized by the state $|\Psi_{pk}\rangle$ and family of unitaries $\{U_c\}_{c\in\{0,1\}^n}$, is thus

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h),h,s,c}\left[\langle\Psi_{pk}|\Pi^{U_c}_{s,sk,c}|\Psi_{pk}\rangle\right]. \tag{2.24}$$

The proof of parallel repetition. Recall that Lemma 2.5.1 states that the projectors corresponding to the two challenges in $\mathfrak{M}$ are nearly orthogonal, even when one takes into account the prover's adaptively applied unitaries. We show that this property persists in the k-copy protocol. Specifically, we show that all 2^k challenges are nearly orthogonal (in the same sense as in Lemma 2.5.1) with respect to any state $|\Psi_{pk}\rangle$ and any post-commitment unitaries U_c of the prover.

This can be explained informally as follows. For any two distinct challenges $c \neq c'$, there exists a coordinate i such that $c_i \neq c'_i$, meaning that one enters a test round in that coordinate while the other enters a Hadamard round. In coordinate i, by the single-copy result (Lemma 2.5.1), the prover who succeeds with one challenge should fail with the other. A complication is that, since we are dealing with an interactive argument, we must show that a violation of this claim leads to an *efficient* single-copy prover that violates the single-copy result. Once we have shown this, we can then apply it to any distinct challenge pairs $c \neq c'$. It then follows that we may (approximately) decompose $|\Psi_{pk}\rangle$ into components accepted in each challenge, each of which occurs with probability 2^{-k}. We can then use this decomposition to express the overall success probability of $\mathcal{P}$ in terms of this decomposition. As $|\Psi_{pk}\rangle$ is of course a normalized state, it will follow that the overall soundness error is negligibly close to 2^{-k}.

The "adaptive orthogonality" discussed above is formalized in the following lemma. Recall that any prover in the k-fold parallel repetition of $\mathfrak{M}$ can be characterized by a state family $\{|\Psi_{pk}\rangle\}_{pk}$ that is prepared in Round $\mathcal{P}_1$ and a family of unitaries $\{U_c\}_{c\in\{0,1\}^k}$ that are applied in Round $\mathcal{P}_2$.

Lemma 2.5.2. *Let $\mathcal{P}$ be a prover in the k-fold parallel repetition of $\mathfrak{M}$ that prepares $|\Psi_{pk}\rangle$ in Round $\mathcal{P}_1$ and performs U_c in Round $\mathcal{P}_2$. Let $a, b \in \{0,1\}^k$ such that $a \neq b$ and choose i such that $a_i \neq b_i$. Then there is an (nr)-qubit quantum state ρ such that for every basis choice h and randomness s,*

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}\left[\langle\Psi_{pk}|\Pi^{U_b}_{s,sk,b}\Pi^{U_a}_{s,sk,a}+\Pi^{U_a}_{s,sk,a}\Pi^{U_b}_{s,sk,b}|\Psi_{pk}\rangle\right]\leq 2\alpha^{1/2}_{h_i,s_i,\rho}+\mathrm{negl}(n)\,, \tag{2.25}$$

where $\alpha_{h_i,s_i,\rho}$ (see Remark 2.4.1) is the success probability with ρ conditioned on the event that h_i is sampled.

Proof. Since we are proving an upper bound for a quantity that is symmetric under the interchange of b and a, we can assume that $b_i = 0$ and $a_i = 1$ without loss of generality.

We first claim that there exists a quantum state ρ such that

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}\left[\langle\Psi_{pk}|\Pi^{U_b}_{s,sk,b}\Pi^{U_a}_{s,sk,a}\Pi^{U_b}_{s,sk,b}|\Psi_{pk}\rangle\right]\leq \alpha_{h_i,s_i,\rho}+\mathrm{negl}(n) \tag{2.26}$$

for all basis choices h and randomness s. For a contradiction, suppose that is not the case. Then there exists a basis choice h^* and s^* and a polynomial η such that for every state ρ,

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h^*)}\left[\langle\Psi_{pk}|\Pi^{U_b}_{s^*,sk,b}\Pi^{U_a}_{s^*,sk,a}\Pi^{U_b}_{s^*,sk,b}|\Psi_{pk}\rangle\right]> \alpha_{h^*_i,s^*_i,\rho}+1/\eta(n)\,. \tag{2.27}$$

We show that this implies the existence of an efficient prover $\mathcal{P}^*$ for the single-copy three-round Mahadev protocol $\mathfrak{M}$ who violates Lemma 2.5.1. Define the following projector on $WXYE$:

$$\Sigma_a := U_a^\dagger(H^a \otimes \mathbb{1}_E)((\mathbb{1} \otimes \cdots \otimes \mathbb{1} \otimes \Pi \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1}) \otimes \mathbb{1}_E)(H^a \otimes \mathbb{1}_E)U_a\,. \tag{2.28}$$

Here Π denotes the single-copy protocol acceptance projector for the Hadamard round, with key sk_i and basis choice h_i^*, s_i^*. In the above, Π acts on the ith set of registers, i.e., $W_iX_iY_i$. The projector Σ_a corresponds to performing the appropriate Hadamard test in the ith protocol copy, and simply accepting all other copies unconditionally. It follows that $\Pi^{U_a}_{s,sk,a} \preceq \Sigma_a$, and we thus have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h^*)}\left[\langle\Psi_{pk}|\Pi^{U_b}_{s^*,sk,b}\Sigma_a\Pi^{U_b}_{s^*,sk,b}|\Psi_{pk}\rangle\right] \\
&\geq \mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h^*)}\left[\langle\Psi_{pk}|\Pi^{U_b}_{s^*,sk,b}\Pi^{U_a}_{s^*,sk,a}\Pi^{U_b}_{s^*,sk,b}|\Psi_{pk}\rangle\right] \\
&> \alpha_{h_i^*,s_i^*,\rho} + 1/\eta.
\end{aligned} \tag{2.29}$$

The single-copy prover $\mathcal{P}^*$ interacts with the single-copy verifier $\mathcal{V}^*$ as follows.

- In the Setup phase, after receiving the public key pk^*, initialize $k-1$ internally simulated verifiers, and set pk to be the list of their keys, with pk^* inserted in the ith position. Let $h = (h_1, \ldots, h_k)$ be the basis choices, and note that all but h_i are known to $\mathcal{P}^*$.

- Using the algorithms of $\mathcal{P}$, perform the following repeat-until-success (RUS) procedure

for at most $q = \eta^4$ steps.

1. Prepare the state $|\Psi_{pk}\rangle$ on registers $WXYE$, and then apply the unitary U_b.
2. Apply the measurement determined by $\Pi_{s,sk,b}$ (defined in (2.22)); for index i we can use pk^* because $b_i = 0$; for the rest we know the secret keys.
3. If the measurement rejects, go to step (1.), and otherwise apply $U_b^\dagger$ and output the state.

If the RUS procedure does not terminate within q steps, then $\mathcal{P}^*$ prepares a state[12] $|\Phi^*_{pk}\rangle$ by performing Samp coherently on $|0^n\rangle_W$ (see Round 2 of Protocol 2.2).

Note that if $\mathcal{P}^*$ terminates within q steps, the resulting state is

$$|\Phi_{pk}\rangle := \frac{\Pi^{U_b}_{s^*,sk,b}|\Psi_{pk}\rangle}{\|\Pi^{U_b}_{s^*,sk,b}|\Psi_{pk}\rangle\|}\,; \tag{2.30}$$

otherwise $|\Phi^*_{pk}\rangle$ is prepared.

- For the Round $\mathcal{P}_1$ message, measure the Y_i register of $|\Phi_{pk}\rangle$ and send the result to $\mathcal{V}^*$.
- When $\mathcal{V}^*$ returns the challenge bit w in Round 3, if $w = b_i = 0$, apply U_b (resp. $\mathbb{1}$) to $|\Phi_{pk}\rangle$ (resp. $|\Phi^*_{pk}\rangle$), and otherwise apply U_a. Then behave honestly, i.e., measure W_iX_i in computational or Hadamard bases as determined by w, and send the outcomes.

By the RUS construction and the fact that $b_i = 0$, the state $|\Phi_{pk}\rangle$ or $|\Phi^*_{pk}\rangle$ is in the image of the test-round acceptance projector in the ith coordinate. This means that, when $\mathcal{V}^*$ enters

[12] To pass the test round, any efficiently preparable state suffices.

a test round, i.e., $w = 0 = b_i$, $\mathcal{P}^*$ is accepted perfectly. In other words, $\mathcal{P}^*$ is a perfect prover[13] and thus satisfies the hypotheses of Lemma 2.5.1.

Now consider the case when $\mathcal{V}^*$ enters a Hadamard round, i.e., $w = 1$. Let

$$\Omega := \{(pk, sk) : \langle \Psi_{pk} | \Pi^{U_b}_{s^*,sk,b} | \Psi_{pk} \rangle > q^{-1/2}\} \tag{2.31}$$

denote the set of "good" keys. For $(pk, sk) \in \Omega$, the probability of not terminating within $q = \mathrm{poly}(n)$ steps is at most $(1 - q^{-1/2})^q \leq e^{-\sqrt{q}}$. Therefore, the success probability of RUS for the good keys is $1 - \mathrm{negl}(n)$. Thus we have

$$\mathop{\mathbb{E}}_{sk|\Omega} [\langle \Phi_{pk} | \Sigma_a | \Phi_{pk} \rangle] \Pr[\Omega] \leq \alpha_{h_i^*, s_i^*, \rho} + \mathrm{negl}(n) \tag{2.32}$$

where we let $\mathbb{E}_{X|E}[f(X)] := \frac{1}{\Pr[E]} \sum_{x \in E} p(x) f(x)$ denote the expectation value of $f(X)$ conditioned on event E for random variable X over finite set $\mathcal{X}$ with distribution p and function $f \colon \mathcal{X} \to [0, 1]$. Now we divide (2.29) into two terms and find

$$\begin{aligned}
\alpha_{h_i^*, s_i^*, \rho} + \eta^{-1} &< \mathop{\mathbb{E}}_{(pk,sk)} \left[\langle \Psi_{pk} | \Pi^{U_b}_{s^*,sk,b} \Sigma_a \Pi^{U_b}_{s^*,sk,b} | \Psi_{pk} \rangle \right] \\
&= \Pr[\Omega] \mathop{\mathbb{E}}_{(pk,sk)|\Omega} \left[\langle \Psi_{pk} | \Pi^{U_b}_{s^*,sk,b} \Sigma_a \Pi^{U_b}_{s^*,sk,b} | \Psi_{pk} \rangle \right] \\
&\quad + \Pr[\overline{\Omega}] \mathop{\mathbb{E}}_{(pk,sk)|\overline{\Omega}} \left[\langle \Psi_{pk} | \Pi^{U_b}_{s^*,sk,b} \Sigma_a \Pi^{U_b}_{s^*,sk,b} | \Psi_{pk} \rangle \right] \\
&\leq \Pr[\Omega] \mathop{\mathbb{E}}_{(pk,sk)|\Omega} \left[\langle \Psi_{pk} | \Pi^{U_b}_{s^*,sk,b} \Sigma_a \Pi^{U_b}_{s^*,sk,b} | \Psi_{pk} \rangle \right] + q^{-1/2} \\
&\leq \alpha_{h_i^*, \rho} + \mathrm{negl}(n) + q^{-1/2}.
\end{aligned} \tag{2.33}$$

[13] While we used $\Pi_{h^*,sk,b}$ in the RUS procedure, and h_i^* is (almost always) not equal to the h_i selected by $\mathcal{V}^*$, the result is still a perfect prover state. This is because, as described in Protocol 2.2, the acceptance test in the test round is independent of the basis choice.

Since $q = \eta^4$, this is a contradiction. Therefore (2.26) holds for every h, s, i.e.,

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}[\langle\Psi_{pk}|\Pi^{U_b}_{s,sk,b}\Pi^{U_a}_{s,sk,a}\Pi^{U_b}_{s,sk,b}|\Psi_{pk}\rangle] \leq \alpha_{h_i,s_i,\rho} + \mathrm{negl}(n). \tag{2.34}$$

It then follows that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}\left[\langle\Psi_{pk}|\Pi^{U_b}_{h,sk,b}\Pi^{U_a}_{h,sk,a} + \Pi^{U_a}_{h,sk,a}\Pi^{U_b}_{h,sk,b}|\Psi_{pk}\rangle\right] \\
&\quad= 2\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}\left[\mathrm{Re}(\langle\Psi_{pk}|\Pi^{U_b}_{h,sk,b}\Pi^{U_a}_{h,sk,a}|\Psi_{pk}\rangle)\right] \\
&\quad\leq 2\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}\left[|\langle\Psi_{pk}|\Pi^{U_b}_{h,sk,b}\Pi^{U_a}_{h,sk,a}|\Psi_{pk}\rangle|\right] \\
&\quad\leq 2\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}\left[\langle\Psi_{pk}|\Pi^{U_b}_{h,sk,b}\Pi^{U_a}_{h,sk,a}\Pi^{U_b}_{h,sk,b}|\Psi_{pk}\rangle^{1/2}\right] \\
&\quad\leq 2\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}\left[\langle\Psi_{pk}|\Pi^{U_b}_{h,sk,b}\Pi^{U_a}_{h,sk,a}\Pi^{U_b}_{h,sk,b}|\Psi_{pk}\rangle\right]^{1/2} \leq 2\alpha^{1/2}_{h_i,s_i,\rho} + \mathrm{negl}(n)
\end{aligned} \tag{2.35}$$

as claimed. □

We remark that this adaptive orthogonality is guaranteed under a computational assumption. Assuming that no efficient quantum adversary can break the underlying security properties based on plain LWE, the projections are pairwise orthogonal in the sense of averaging over the key pairs (pk, sk) and with respect to any quantum state $|\Psi_{pk}\rangle$ prepared by an efficient quantum circuit.

We also emphasize that, in Lemma 2.5.2, for each pair $a \neq b$ the left-hand side is upper-bounded by the acceptance probability of measuring some state ρ in the basis h_i, and the quantum state ρ may be different among distinct choices of (a, b) and i. This implies

that if $\mathcal{P}$ succeeds with one particular challenge perfectly[14] when we average over h and s, Lemma 2.5.2 and standard amplification techniques (see Section 2.4) imply that it succeeds on challenge $b \neq a$ with probability at most $\mathbb{E}_{(pk,sk)\leftarrow \mathsf{Gen}(1^\lambda)} \langle \Psi_{pk} | \Pi_{s,sk,b} | \Psi_{pk} \rangle \leq \mathrm{negl}(n)$. We note that this strategy leads to acceptance probability at most $2^{-k} + \mathrm{negl}(n)$.

Unfortunately, the above observation does not rule out that $\mathcal{P}$ can succeed on several challenges with appreciable probability, achieving a better overall success probability. For instance, one could try to implement this by coherently simulating distinct provers who perfectly win different challenges.

We rule out this possibility by showing that if the projectors are pairwise nearly orthogonal with respect to a quantum state ρ, then the above strategy is optimal with respect to ρ. Since pairwise orthogonality holds with respect to *any* efficiently preparable quantum state by Lemma 2.5.2, our parallel repetition theorem follows.

First, we state a key technical lemma.

Lemma 2.5.3. *Let $A_1, \ldots, A_m$ be projectors and $|\psi\rangle$ be a quantum state. Suppose there are real numbers $\delta_{ij} \in [0,2]$ such that $\langle \psi | A_i A_j + A_j A_i | \psi \rangle \leq \delta_{ij}$ for all $i \neq j$. Then $\langle \psi | A_1 + \cdots + A_m | \psi \rangle \leq 1 + \left(\sum_{i<j} \delta_{ij}\right)^{1/2}$.*

Proof. Let $\alpha := \langle \psi | A_1 + \ldots + A_m | \psi \rangle$. We have

$$\begin{aligned} \alpha^2 &\leq \langle \psi | (A_1 + \cdots + A_m)^2 | \psi \rangle \\ &= \alpha + \sum_{i<j} \langle \psi | A_i A_j + A_j A_i | \psi \rangle \qquad (2.36) \\ &\leq \alpha + \sum_{i<j} \delta_{ij} \qquad (2.37) \end{aligned}$$

[14] More concretely, if for some fixed a, $\Pi_{s,sk,a} | \Psi_{pk} \rangle = | \Psi_{pk} \rangle$.

The first inequality holds since $|\psi\rangle\langle\psi| \preceq \mathbb{1}$, and thus

$$\langle\psi|(A_1 + \cdots + A_m)|\psi\rangle\langle\psi|(A_1 + \cdots + A_m)|\psi\rangle \leq \langle\psi|(A_1 + \cdots + A_m)^2|\psi\rangle. \tag{2.38}$$

The equality (2.36) holds since each A_i is idempotent, and thus

$$\begin{aligned}\langle\psi|(A_1 + \cdots + A_m)^2|\psi\rangle &= \langle\psi|A_1^2 + \cdots + A_m^2|\psi\rangle + \sum_{i<j}\langle\psi|A_iA_j + A_jA_i|\psi\rangle \\ &= \langle\psi|A_1 + \cdots + A_m|\psi\rangle + \sum_{i<j}\langle\psi|A_iA_j + A_jA_i|\psi\rangle. \end{aligned} \tag{2.39}$$

Now observe that for $\beta > 0$, $x^2 \leq x + \beta$ implies $x \leq \frac{1}{2}(1 + \sqrt{1+4\beta}) \leq \frac{1}{2}(1 + (1 + 2\sqrt{\beta})) = 1 + \sqrt{\beta}$. Thus $\alpha \leq 1 + \sqrt{\sum_{i<j}\delta_{ij}}$ as claimed. □

Observe that when the projectors are mutually orthogonal, we have $A_1 + \cdots + A_m \preceq \mathbb{1}$ and the bound clearly holds. Lemma 2.5.3 describes a relaxed version of this fact. In our application, the projectors and the state are parameterized by the key pair, and we use this bound to show that the average of pairwise overlaps is small. We are now ready to establish our parallel repetition theorem.

Lemma 2.5.4. *Let k be a positive integer and let $\{U_c\}_{c\in\{0,1\}^k}$ be any set of unitaries that may be implemented by $\mathcal{P}$ after the challenge coins are sent. Let $|\Psi_{pk}\rangle$ be any state $\mathcal{P}$ holds in the commitment round, and suppose $\mathcal{P}$ applies U_c followed by honest measurements when the coins are c. Then there exists a negligible function ϵ such that $\mathcal{V}_1, \ldots, \mathcal{V}_k$ accept $\mathcal{P}$ with probability at most $2^{-k} + \epsilon(n)$.*

Proof. The success probability of any prover in the k-fold protocol is

$$\Pr[\text{success}] = 2^{-k} \mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h),h,s}[\langle\Psi_{pk}|\sum_c \Pi^{U_c}_{s,sk,c}|\Psi_{pk}\rangle] \tag{2.40}$$

where h, s are drawn from uniform distributions.

Define a total ordering on $\{0,1\}^k$ such that $a < b$ if $a_i < b_i$ for the smallest index i such that $a_i \neq b_i$. Then by Lemma 2.5.3, we have

$$\Pr[\text{success}] \leq 2^{-k} + 2^{-k} \mathop{\mathbb{E}}_{h,s}\left[\sum_{a<b} \mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}[\langle\Psi_{pk}|\Pi^{U_a}_{s,sk,a}\Pi^{U_b}_{s,sk,b} + \Pi^{U_b}_{s,sk,b}\Pi^{U_a}_{s,sk,a}|\Psi_{pk}\rangle]\right]^{1/2}. \tag{2.41}$$

By Lemma 2.5.2, there exists a negligible function δ such that

$$\mathop{\mathbb{E}}_{(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda,h)}[\langle\Psi_{pk}|\Pi^{U_a}_{s,sk,a}\Pi^{U_b}_{s,sk,b} + \Pi^{U_b}_{s,sk,b}\Pi^{U_a}_{s,sk,a}|\Psi_{pk}\rangle] \leq 2\alpha^{1/2}_{h_{i(a,b)},\rho_{ab}} + \delta \tag{2.42}$$

for every pair (a, b). Here $i(a, b)$ is the smallest index i such that $a_i \neq b_i$ and ρ_{ab} is the reduced quantum state associated with a, b, as guaranteed by Lemma 2.5.2. Let μ be the

soundness error of the MF protocol. We have

$$\begin{aligned}
\Pr[\text{success}] &\leq 2^{-k} + 2^{-k} \mathop{\mathbb{E}}_{h,s} \left[\sum_{a<b} \left(2\alpha^{1/2}_{h_{i(a,b)}, s_{i(a,b)}, \rho_{ab}} + \delta \right) \right]^{1/2} \\
&\leq 2^{-k} + 2^{-k} \mathop{\mathbb{E}}_{h,s} \left[\sum_{a<b} 2\alpha^{1/2}_{h_{i(a,b)}, s_{i(a,b)}, \rho_{ab}} \right]^{1/2} + 2^{-k} \sqrt{\binom{2^k}{2}} \delta^{1/2} \\
&\leq 2^{-k} + 2^{-k} \left[\sum_{a<b} 2 \left(\mathop{\mathbb{E}}_{h,s} [\alpha_{h_{i(a,b)}, s_{i(a,b)}, \rho_{ab}}] \right)^{1/2} \right]^{1/2} + \delta^{1/2} \\
&\leq 2^{-k} + 2^{-k} \left[\sum_{a<b} 2\mu^{1/2} \right]^{1/2} + \delta^{1/2} \\
&\leq 2^{-k} + 2^{-k} \left[2^k (2^k - 1) \mu^{1/2} \right]^{1/2} + \delta^{1/2} \\
&\leq 2^{-k} + \mu^{1/4} + \delta^{1/2} \qquad (2.43)
\end{aligned}$$

where the second and third inequalities hold by Jensen's inequality. Amplifying the soundness of the MF protocol, μ is negligible using polynomially many copies by Lemma 2.4.1. Thus the soundness error is negligibly close to 2^{-k}. □

We note that Mahadev shows the soundness error for a single-copy protocol is negligibly close to 3/4 [6], whereas Lemma 2.5.4 implies the error can be upper bounded by $1/2 + \text{negl}(n)$. Mahadev obtains soundness error $3/4 + \text{negl}(n)$ by considering a general prover $\mathcal{P}$ who, for each basis h, succeeds in the test round (characterized by $\Pi_{h,sk,\mathfrak{t}}$) with probability $1 - p_{h,\mathfrak{t}}$, in the first stage of the Hadamard round (characterized by $\Lambda_{h,sk,\mathfrak{h},1}$) with probability $1 - p_{h,\mathfrak{h}}$, and in the second stage of the Hadamard round (characterized by $\Lambda_{h,sk,\mathfrak{h},2}$) with probability at most $\sqrt{p_{h,\mathfrak{t}}} + p_{h,\mathfrak{h}} + \alpha_{h,\rho} + \text{negl}(n)$ for some state ρ [6, Claim 7.1]. These contributions are combined by applying the triangle inequality for trace distance. This analysis is loose since $\Lambda_{h,sk,\mathfrak{h},1}$ and $\Lambda_{h,sk,\mathfrak{h},2}$ commute, and $\mathcal{P}$ must pass both stages to win

the Hadamard round.

Finally, Lemma 2.5.4 immediately implies the following theorem.

Theorem 2.5.1. *Let $\mathfrak{M}^k$ be the k-fold parallel repetition of the three-round Mahadev protocol $\mathfrak{M}$. Under the* LWE *assumption, the soundness error of $\mathfrak{M}^k$ is at most $2^{-k} + \mathrm{negl}(n)$.*

We conclude with the following three-round protocol.

Theorem 2.5.2. *For $r = \omega(\frac{\log n}{(b-a)^2})$ and $k = \omega(\log n)$, Protocol 2.4 is a quantum prover interactive argument for $\mathrm{ZX}_{a,b}$ with negligible completeness error and soundness error.*

2.6 A classical zero-knowledge argument for QMA

To turn Protocol 2.4 into a zero-knowledge protocol, we first consider an intermediate protocol in which $\mathcal{P}$ first encrypts the witness state $|\psi\rangle^{\otimes rk}$ with a quantum one-time pad. Then in Round $\mathcal{P}_2$, $\mathcal{P}$ sends the one-time pad key β, γ along with the response u. In the verdict stage, $\mathcal{V}$ uses the keys to decrypt the response. We denote the verdict function as

$$\mathsf{verdict}'(H, s, sk, y, c, \beta, \gamma, u) := \mathsf{verdict}(H_{\beta,\gamma}, s, sk, y, c, u) \tag{2.44}$$

where $H_{\beta,\gamma} := X^\beta Z^\gamma H Z^\gamma X^\beta$ is the instance conjugated by the one-time pad.

Obviously, this is not zero-knowledge yet, as the verifier can easily recover the original measurement outcomes on the witness state. To address this issue, we take the approach of [63, 66] and invoke a NIZK protocol for NP languages. The language $\mathcal{L}$ that we consider is

Setup. $\mathcal{V}$ samples random bases $h \in \{0,1\}^{nrk}$ and runs the key generation algorithm $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda, h)$. $\mathcal{V}$ samples a string $s \in \{0,1\}^{prk}$ uniformly. $\mathcal{V}$ sends the public keys pk to $\mathcal{P}$.

Round $\mathcal{P}_1$. $\mathcal{P}$ queries Samp coherently on the witness state $|\psi\rangle^{\otimes rk}$, followed by a standard basis measurement on register Y. The outcome is sent to $\mathcal{V}$.

Round $\mathcal{V}_2$. $\mathcal{V}$ samples $c_1, \ldots, c_k \leftarrow \{0,1\}$ and sends $c = (c_1, \ldots, c_k)$ to $\mathcal{P}$.

Round $\mathcal{P}_2$. For each $i \in [k]$, $j \in [r]$, $\ell \in [n]$,

1. if $c_i = 0$, $\mathcal{P}$ performs a standard basis measurement and gets $u_{ij\ell} = (w_{ij\ell}, t_{ij\ell})$;
2. if $c_i = 1$, $\mathcal{P}$ performs a Hadamard basis measurement and gets $u_{ij\ell} = (w_{ij\ell}, t_{ij\ell})$.

$\mathcal{P}$ sends u to $\mathcal{V}$.

Verdict. For each $i \in [k]$,

1. If $c_i = 0$, $\mathcal{V}$ accepts iff $\bigwedge_{j,\ell} \mathsf{Chk}(pk_{j\ell}, w_{j\ell}, t_{j\ell}, y_{j\ell}) = 1$.
2. If $c_i = 1$, $\mathcal{V}$ records the set $A_i \subseteq [r]$ of consistent copies. For each $j \in A_i$ and $\ell \in [n]$:
   (a) If $h_{ij\ell} = 0$, run $(b_{ij}, x_{b_{ij}, y_{ij}}) \leftarrow \mathsf{Inv}_{\mathcal{G}}(sk_{ij}, y_{ij})$. Set $e_{ij\ell} = b_{ij\ell}$; if $h_{ij} = 1$, compute $x_{0,y_{ij\ell}}, x_{1,y_{ij\ell}}$ and $e_{ij\ell} = t_{ij\ell} \cdot (x_{0,y_{ij\ell}} \oplus x_{1,y_{ij\ell}}) \oplus w_{ij}$. If any of the algorithms rejects or any of $t_{ij\ell}$ is trivial (e.g., $t_{ij\ell} = 0$, see [6]), $\mathcal{V}$ sets $v_{ij} = 0$; otherwise enters the next step.
   (b) $\mathcal{V}$ computes the terms $S_{ij} = \mathsf{term}(H, s_{ij})$ for each $i \in [k], j \in [r]$. Set $v_{ij} = 1$ if $(e_{ij\ell})_{\ell \in S_{ij}}$ satisfies $M_{-m_{S_{ij}}}$ and $v_{ij} = 0$ otherwise.

   Then $\mathcal{V}$ sets $v_i = 1$ if $\sum_{j \in A_i} v_{ij} \geq (2 - a - b)|A_i|/4$ and 0 otherwise.

$\mathcal{V}$ accepts iff $v_i = 1$ for every $i \in [k]$.
The verdict function is $\mathsf{verdict}(H, s, sk, y, c, u) := \bigwedge_{i=1}^{k} v_i$.

Protocol 2.4: Verification with instance-independent setup.

defined as follows:

$$\mathcal{L} := \{(H, s, sk, \xi, y, c, \chi) : \exists\, \tau = (\beta, \gamma, u, r_1, r_2),$$

$$\xi = \mathsf{commit}(u; r_1) \wedge \chi = \mathsf{commit}(\beta, \gamma; r_2)$$

$$\wedge\, \mathsf{verdict}'(H, s, sk, y, c, \beta, \gamma, u) = 1\}, \tag{2.45}$$

where r_1, r_2 are the randomness for a computationally secure bit commitment scheme (see Section 2.2.1). However, this alone is insufficient since, to agree on an instance without introducing more message exchanges, $\mathcal{V}$ must reveal sk, s before $\mathcal{P}$ sends a NIZK proof. Revealing sk, s enables a simple attack on soundness: $\mathcal{P}$ can ensure the verifier accepts all instances by using the secret key to forge a valid response u, committing to it, and computing the NIZK proof.

The solution is to invoke a quantum-secure classical FHE scheme and to let $\mathcal{P}$ homomorphically compute a NIZK proof. This requires $\mathcal{P}$ to only use an encrypted instance. In the setup phase, $\mathcal{P}$ is given the ciphertexts $csk = \mathsf{FHE.Enc}_{hpk}(sk)$ and $cs = \mathsf{FHE.Enc}_{hpk}(s)$. Next, in Round $\mathcal{P}_2$, $\mathcal{P}$ computes $cx = \mathsf{FHE.Enc}_{hpk}(x)$ where $x := (H, s, sk, \xi, y, c, \chi)$ since the partial transcript (y, c, ξ, χ) has already been fixed. $\mathcal{P}$ then computes

$$ce = \mathsf{FHE.Eval}_{hpk}(\mathsf{NIZK.P}, cc, cx, c\tau) = \mathsf{FHE.Enc}_{hpk}(\mathsf{NIZK.P}(\mathsf{crs}, x, \tau)), \tag{2.46}$$

where $c\tau = \mathsf{FHE.Enc}_{hpk}(\tau)$, and sends ce to $\mathcal{V}$. Finally, $\mathcal{V}$ decrypts the encrypted NIZK proof ce and outputs $\mathsf{NIZK.V}(\mathsf{crs}, x, e)$. The above transformation yields a three-message zero-knowledge protocol for quantum computation with trusted setup from a third party, described as follows.

2.6.1 Completeness and soundness

To prove soundness, we consider the following hybrid protocols:

H_0: Protocol 2.4.

The algorithm setup takes two integers N, M as input, and outputs two strings $\mathsf{st}_{\mathcal{V}}, \mathsf{st}_{\mathcal{P}}$ with the following steps.

1. Run $\mathsf{crs} \leftarrow \mathsf{NIZK.Setup}(1^{\lambda})$.
2. Sample uniform bases $h \leftarrow \{0,1\}^N$ and run $(pk, sk) \leftarrow \mathsf{Gen}(1^{\lambda}, h)$.
3. Run the FHE key generation algorithm $(hpk, hsk) \leftarrow \mathsf{FHE.Gen}(1^{\lambda})$.
4. Run encryption on the secret key $csk \leftarrow \mathsf{FHE.Enc}_{hpk}(sk)$.
5. Choose keys (β, γ) and randomness r_1 uniformly and compute $\xi = \mathsf{commit}(\beta, \gamma; r_1)$.
6. Sample a random string $s_1, \ldots, s_M \in \{0,1\}^p$ (see Remark 2.4.1) uniformly and compute its encryption $cs = \mathsf{FHE.Enc}_{hpk}(s)$.

Output $\mathsf{st}_{\mathcal{V}} = (\mathsf{crs}, sk, hsk, hpk, \xi)$ and $\mathsf{st}_{\mathcal{P}} = (\mathsf{crs}, pk, hpk, csk, cs, \beta, \gamma, r_1)$.

Protocol 2.5: The setup phase $\mathsf{setup}(\lambda, N, M)$.

Setup. Run $\mathsf{st}_{\mathcal{V}}, \mathsf{st}_{\mathcal{P}} \leftarrow \mathsf{setup}(\lambda, nrk, rk)$ (defined in Protocol 2.5). Send $\mathsf{st}_{\mathcal{V}} = (\mathsf{crs}, sk, hsk, hpk, \xi)$ to $\mathcal{V}$ and $\mathsf{st}_{\mathcal{P}} = (\mathsf{crs}, pk, hpk, csk, cs, \beta, \gamma, r_1)$ to $\mathcal{P}$.

Round $\mathcal{P}_1$. $\mathcal{P}$ aborts if pk is invalid. $\mathcal{P}$ queries Samp coherently on the witness state $X^{\beta}Z^{\gamma}|\psi\rangle^{\otimes rk}$.

Round $\mathcal{V}_2$. $\mathcal{V}$ samples $c_1, \ldots, c_k \leftarrow \{0,1\}$ and sends $c = (c_1, \ldots, c_k)$ to $\mathcal{P}$.

Round $\mathcal{P}_2$. For each $i \in [k]$, $j \in [r]$, $\ell \in [n]$,

1. if $c_i = 0$, $\mathcal{P}$ performs a standard basis measurement and gets $u_{ij\ell} = (w_{ij\ell}, t_{ij\ell})$.
2. if $c_i = 1$, $\mathcal{P}$ performs a Hadamard basis measurement and gets $u_{ij\ell} = (w_{ij\ell}, t_{ij\ell})$.

$\mathcal{P}$ sends $\chi := \mathsf{commit}(u; r_2)$ and

$$ce := \mathsf{FHE.Eval}_{hpk}(\mathsf{NIZK.P}, cc, cx, c\tau), \tag{2.47}$$

where cc, cx and $c\tau$ are the encryptions of crs, x and τ respectively.

Verdict. $\mathcal{V}$ accepts if $\mathsf{NIZK.V}(\mathsf{crs}, x, \mathsf{FHE.Dec}_{hsk}(ce)) = 1$.

Protocol 2.6: An interactive zero-knowledge protocol for QMA.

H_1: Same as H_0 except that in the setup phase, $\mathcal{P}$ receives uniform one-time pad keys β, γ and randomness r_1 for the commitment and $\mathcal{V}$ receives the commitment $\xi = \mathsf{commit}(\beta, \gamma, r_1)$. Moreover, $\mathcal{P}$ reveals β, γ and r_1 in Round $\mathcal{P}_2$. $\mathcal{V}$ accepts if $\mathsf{verdict}'$ returns 1 and $\xi = \mathsf{commit}(\beta, \gamma, r_1)$.

H_2: Same as H_1 except that $\mathcal{P}$ also sends the commitment to u with message χ, and the randomness r_2. $\mathcal{V}$ accepts if $\mathsf{verdict}'$ returns 1, $\xi = \mathsf{commit}(\beta, \gamma, r_1)$, and $\chi = \mathsf{commit}(u, r_2)$.

H_3: Same as H_2 except that in the setup phase, both $\mathcal{V}$ and $\mathcal{P}$ also receive $csk = \mathsf{FHE.Enc}_{hpk}(sk)$ and $cs = \mathsf{FHE.Enc}_{hpk}(s)$.

H_4: Same as H_3 except that in the setup phase both $\mathcal{V}$ and $\mathcal{P}$ get crs and in Round $\mathcal{P}_3$, $\mathcal{P}$ sends ce and in Verdict, $\mathcal{V}$ accepts if ce is a valid ciphertext and $\mathsf{NIZK.V}(\mathsf{crs}, x, \mathsf{Dec}_{hsk}(ce)) = 1$. Notice that this is Protocol 2.6.

Let A_i be the maximum acceptance probability of the prover in the hybrid H_i among all no-instances. We show that $A_i \leq \mathrm{negl}(n)$ for all $i \in \{0, \ldots, 4\}$. Notice that $A_0 \leq \mathrm{negl}(n)$ by Theorem 2.5.2.

Lemma 2.6.1. $A_1 \leq \mathrm{negl}(n)$.

Proof. Let the soundness error of H_0 be $\epsilon = \mathrm{negl}(n)$. First we introduce a hybrid protocol $H_{0.1}$, which is the same as H_1 except that in the setup phase, both $\mathcal{P}$ and $\mathcal{V}$ receive a uniform key pair (β, γ) (and the instance H), and $\mathcal{V}$ verifies the instance $H_{\beta,\gamma}$, which has the same ground state energy as H.

Conditioned on any key pair (β, γ), let the success probability of any prover $\mathcal{P}$ be $\epsilon_{\beta,\gamma}$ in $H_{0.1}$. Furthermore, since we have fixed a pair (β, γ), $H_{0.1}$ is exactly the same as H_0 except that H is replaced with $H_{\beta,\gamma}$. Thus we conclude that $\epsilon_{\beta,\gamma} \leq \epsilon$. The success probability of $\mathcal{P}$ is $\mathbb{E}_{\beta,\gamma}[\epsilon_{\beta,\gamma}] \leq \epsilon$, and thus $H_{0.1}$ has soundness error ϵ.

Next, we observe that H_1 is the same as $H_{0.1}$ except that $\mathcal{V}$ receives a commitment ξ to a uniform key pair (β, γ) in the setup phase, and $\mathcal{P}$ reveals the key pair in Round $\mathcal{P}_2$. Since $\mathcal{V}$'s only message (Round $\mathcal{V}_2$) does not depend on the key pair in either $H_{0.1}$ or H_1 (public coins) and the commitment is perfectly binding (and therefore the prover cannot change the values of β and γ depending on the verifier's challenge), in the reduction, $\mathcal{V}$ does not reject with higher probability if $\mathcal{V}$ knows the keys in an earlier step. □

Lemma 2.6.2. $A_2 \leq \text{negl}(n)$.

Proof. The difference between H_1 and H_2 is whether $\mathcal{P}$ sends a commitment to u in Round $\mathcal{P}_2$, and $\mathcal{V}$ checks if χ is a commitment to u. Since the probability can only become smaller under this change, the success probability in H_2 is negligible. □

Lemma 2.6.3. $A_3 \leq \text{negl}(n)$.

Proof. Let us suppose that there is a prover who wins H_3 with non-negligible probability ϵ. We can construct an adversary in H_2 that simulates such a prover by providing encryptions of $(0, 0)$ instead of (s, sk). By the security of the FHE scheme, such an adversary cannot distinguish both ciphertexts, and therefore H_3 would still make the verifier accept with probability at least $\epsilon - \text{negl}(n)$. In this case the adversary in H_2 would succeed with the same probability, which is a contradiction. □

Lemma 2.6.4. $A_4 \leq \text{negl}(n)$.

Proof. In H_3, by the perfect binding property of the commitment scheme, for an accepting view $x = (H, s, sk, \xi, y, c, \chi)$, there is a unique witness $\tau = (u, \beta, \gamma, r_1, r_2)$ such that $\mathsf{verdict}'(x, \tau) = 1$; otherwise x is unsatisfiable. Since no efficient quantum prover can generate (ξ, y, χ) such that $x \in \mathcal{L}$ except with negligible probability (otherwise it would be possible for a prover in H_3 to succeed with non-negligible probability), for any prover $\mathcal{P}$ and $H \in \mathrm{ZX}_{no}$,

$$\Pr_{s,sk,c}[(\xi, y, \chi) \leftarrow \mathcal{P} : \ (H, s, sk, \xi, y, c, \chi) \notin \mathcal{L}] > 1 - \delta \tag{2.48}$$

for some negligible function δ.

By the soundness property of the NIZK scheme, no bounded provers could then provide a proof that makes the verifier accept such a no instance except with negligible probability. The fact that the NIZK is encrypted does not change this. □

Theorem 2.6.1. *Protocol 2.6 has negligible completeness and soundness errors.*

Proof. The completeness follows from the correctness of FHE, the completeness of NIZK, and the completeness of Protocol 2.4. For soundness, we note that H_4 is indeed Protocol 2.6. By Lemma 2.6.4, the soundness error is negligible. □

2.6.2 The zero-knowledge property

To show Protocol 2.6 is zero-knowledge, we consider the following simulator.

Finally, $\mathcal{S}$ outputs $(\mathsf{st}_{\mathcal{V}}, \xi, y, c, \chi, ce)$.

To prove zero-knowledge, we consider an intermediate (quantum) simulator $\mathcal{S}_Q$ that

Setup. $\mathcal{S}$ samples pair $\mathsf{st}_{\mathcal{V}}, \mathsf{st}_{\mathcal{P}}$ according to the correct distributions.

$\mathcal{P}_1$. For each bit, $\mathcal{S}$ samples random $b_{ij\ell}, x_{ij\ell}$ and queries the function $f_{pk_{ij\ell}}$ to get a sample $y_{ij\ell}$. $\mathcal{S}$ computes $\xi = \mathsf{commit}(\beta, \gamma; r_1)$ for random β and γ and feeds $\mathcal{V}^*$ with y.

$\mathcal{V}_2$. $\mathcal{S}$ queries $\mathcal{V}_2^*$ to get coins $c = (c_1, \ldots, c_k)$.

$\mathcal{P}_2$. $\mathcal{S}$ uses $\mathsf{st}_{\mathcal{V}}$ to compute a valid response u.[a] $\mathcal{S}$ computes the commitment $\chi = \mathsf{commit}(u; r_2)$ and $ce = \mathsf{FHE.Enc}_{hpk}(\mathsf{NIZK.S}(x))$.

[a]The response u is computed as follows: for each i such that $c_i = 0$, $\mathcal{S}$ sets $u_{ij\ell} = (b_{ij\ell}, x_{ij\ell})$. For each i such that $c_i = 1$, $\mathcal{S}$ computes the terms $S_{ij} = \mathsf{term}(H, s_{ij})$. For each copy i, j such that S_{ij} is consistent with h_{ij}, do the following: $\mathcal{S}$ samples nontrivial $t_{ij\ell}$ uniformly and uses the secret key $sk_{ij\ell}$ to compute the hardcore bit $o_{ij\ell} = t_{ij\ell} \cdot (x_{0y_{ij\ell}} \oplus x_{1y_{ij\ell}})$. Then for each copy i, j and qubits $\ell_1, \ell_2 \in S_{ij}$ such that $S_{ij\ell_1} = X$ (and $S_{ij\ell_2} = Z$), set $b'_{ij\ell_2}$ randomly and $b'_{ij\ell_1} = \beta_{ij\ell} \oplus b_{ij\ell_2} \oplus \frac{1-m_{S_{ij}}}{2}$; for $\ell \notin S_{ij}$, samples $b'_{ij\ell}$ uniformly. The response $u_{ij\ell} = (b'_{ij\ell} \oplus o_{ij\ell} \oplus \gamma_{ij\ell}, t_{ij\ell})$. For inconsistent copies, $\mathcal{S}$ samples a random response.

Protocol 2.7: The simulator $\mathcal{S}(H)^{\mathcal{V}_2^*}$.

simulates the interaction between $\mathcal{P}$ and $\mathcal{V}^*$, but that sets the last message as $ce = \mathsf{FHE.Enc}_{hpk}(\mathsf{NIZK.S}(x))$ instead of the NIZK proof.

Lemma 2.6.5. *The output distributions of the original protocol and $\mathcal{S}_Q$ are computationally indistinguishable.*

Proof. We prove this by showing an intermediate quantum simulator as follows:

1. $\tilde{\mathcal{S}}_0$: A quantum simulator that fully simulates $\mathcal{P}$ and $\mathcal{V}$ in the original protocol.

2. $\tilde{\mathcal{S}}_1$: Same as $\tilde{\mathcal{S}}_0$, except that $\tilde{\mathcal{S}}_1$ sets $ce = \mathsf{FHE.Enc}_{hpk}(\mathsf{NIZK.P}(\mathsf{crs}, x, \tau))$, instead of computing it homomorphically (this is possible since the simulator has access to the private keys).

3. $\tilde{\mathcal{S}}_2$: Same as $\tilde{\mathcal{S}}_1$, except that $\tilde{\mathcal{S}}_2$ sets $ce = \mathsf{FHE.Enc}_{hpk}(\mathsf{NIZK.S}(x))$. Notice that this is $\mathcal{S}_Q$.

The output distribution of $\tilde{\mathcal{S}}_0$ is indistinguishable from the output of $\tilde{\mathcal{S}}_1$ by the circuit privacy property of the FHE scheme and that the encryption of sk and s are provided by a trusted party.

The output distribution of $\tilde{\mathcal{S}}_1$ is indistinguishable from the output of $\tilde{\mathcal{S}}_2$ by the computational zero-knowledge property of NIZK. □

Lemma 2.6.6. *The output distributions of $\mathcal{S}_Q$ and $\mathcal{S}$ are computationally indistinguishable.*

Proof. The main difference between the protocols is that the instance x for the NP language $\mathcal{L}$ is generated by the quantum simulator $\mathcal{S}_Q$, who honestly measures the quantum witness. Here again we introduce an intermediate step $\tilde{\mathcal{S}}_Q$ which is the same as $\mathcal{S}_Q$ but the commitment $\xi' = \mathsf{commit}(\beta', \gamma'; r_1)$ is provided instead of the commitment of the one-time pad key β, γ, for uniformly independently random β' and γ'.

Notice that the output distribution of $\mathcal{S}_Q$ is computationally indistinguishable from the output of $\tilde{\mathcal{S}}_Q$ from the hiding property of the commitment scheme, and we just need to argue now that the output of $\tilde{\mathcal{S}}_Q$ is computationally indistinguishable from the output of $\mathcal{S}$.

Since the last message is obtained by homomorphic evaluation of $\mathsf{NIZK.S}$ on the encryption of crs and x, it suffices to show the instances from the simulators are computationally indistinguishable.

Notice that in $\tilde{\mathcal{S}}_Q$, ξ' is independent of the other parts of the message, and in particular there is no information leaked from β and γ that were used to one-time pad the state. In this case, the distribution of the output Samp on the maximally mixed state, i.e., for each qubit $(i, j, \ell) \in [k] \times [r] \times [n]$, $y_{ij\ell}$ is obtained by measuring $\mathsf{Samp}(pk, \mathbb{1}/2)$, is negligibly close

to

$$\frac{1}{2|\mathcal{X}|}\sum_{b\in\{0,1\}}\sum_{x,x'\in\mathcal{X}}|b\rangle\langle b|_{W_{ij\ell}}\otimes|x\rangle\langle x'|_{X_{ij\ell}}\otimes|\psi_{f_{pk}(b,x)}\rangle\langle\psi_{f_{pk}(b,x')}|_{Y_{ij\ell}}. \tag{2.49}$$

Measuring register $Y_{ij\ell}$, by the injective pair property, there exists a unique x such that y lies in the support of $f_{pk}(b,x)$, and thus upon measurement the state collapses to

$$\frac{1}{2|\mathcal{X}|}\sum_{b,x}\sum_{y}f_{pk}(b,x)(y)|b\rangle\langle b|_{W_{ij\ell}}\otimes|x\rangle\langle x|_{X_{ij\ell}}\otimes|y\rangle\langle y|_{Y_{ij\ell}}, \tag{2.50}$$

which is identical to the state obtained by querying f_{pk} on uniform b,x. This implies that the marginal distributions of y on the two experiments are statistically close. By the hiding property of the commitment scheme, the distributions over ξ are computationally indistinguishable.[15]

In Round $\mathcal{V}_2$, since the distribution over c depends on $\mathsf{st}_{\mathcal{V}},\xi,y$, the distributions over $(\mathsf{st}_{\mathcal{V}},\xi,y,c)$ are computationally indistinguishable.

In Round $\mathcal{P}_2$, by the hiding property, conditioned on $(\mathsf{st}_{\mathcal{V}},\xi,y,c)$, the distributions over χ are computationally indistinguishable.[16] □

From Lemma 2.6.5 and Lemma 2.6.6, we conclude the following theorem.

Theorem 2.6.2. *Assuming the existence of a non-interactive bit commitment scheme with perfect binding and computational hiding, Protocol 2.6 is zero-knowledge.*

[15] Note that y of $\mathcal{S}$ is independent of β,γ, whereas in Protocol 2.6, $y,(\beta,\gamma)$ may not be independent; for example, consider the special case where the witness is $|\psi\rangle=|0^n\rangle$.

[16] Namely, the commitment to a response sampled from $\mathcal{P}$'s witness state is computationally indistinguishable from a deterministically computed response by $\mathcal{S}$.

2.7 Round reduction by Fiat-Shamir transformation

In this section we show that the Fiat-Shamir transformation can be used make the k-fold parallel three-round Mahadev protocol $\mathfrak{M}$ non-interactive with a setup phase, while keeping both the completeness and the soundness errors negligible. This will also be the case for the zero-knowledge variant of the same, i.e., Protocol 2.6.

2.7.1 Fiat-Shamir for Σ-protocols in the QROM

The Fiat-Shamir (FS) transformation turns any public-coin three-message interactive argument, also called a Σ-protocol, into a single-message protocol in the random oracle model (ROM). A Σ-protocol consists of the following type of interaction between $\mathcal{V}$ and $\mathcal{P}$:

$\mathcal{V}$ and $\mathcal{P}$ receive an input x.

Round 1. $\mathcal{P}$ sends a message y (called the *commitment*).
Round 2. $\mathcal{V}$ samples a random c (called the *challenge*) uniformly from a finite set $\mathcal{C}$.
Round 3. $\mathcal{P}$ responds with a message m (called the *response*).
Verdict. $\mathcal{V}$ outputs $V(x, y, c, m)$, where V is some Boolean function.

Protocol 2.8: A Σ-protocol for a language $\mathcal{L}$.

In the Fiat-Shamir transformation, the prover generates their own challenge by making a query to hash function $\mathcal{H}$, and then computing their response m as usual. The transformed protocol thus does not require $\mathcal{V}$ to send any messages.

In the standard approach, one proves that the Fiat-Shamir transformation preserves soundness in the ROM. In this idealized cryptographic model, all parties receive oracle

$\mathcal{V}_{FS}$ and $\mathcal{P}_{FS}$ receive an input x and are given access to a random oracle $\mathcal{H}$.

Round 1. $\mathcal{P}_{FS}$ sends a message (y, m).

Verdict. $\mathcal{V}_{FS}$ outputs $V(x, y, \mathcal{H}(x, y), m)$.

Protocol 2.9: The FS-transformed protocol for $\mathcal{L}$.

access to a uniformly random function $\mathcal{H}$. Against quantum adversaries, there is a well-known complication: a quantum computer can easily evaluate any actual instantiation of $\mathcal{H}$ (with a concrete public classical function) in superposition via

$$U_{\mathcal{H}} \colon |x, y\rangle|z\rangle \mapsto |x, y\rangle|z \oplus \mathcal{H}(x, y)\rangle \,. \tag{2.51}$$

We thus work in the Quantum Random Oracle Model (QROM), in which all parties receive quantum oracle access to $U_{\mathcal{H}}$.

While this is nontrivial to prove, the Fiat-Shamir transformation remains secure in the QROM, up to some mild conditions on the Σ-protocol [64]. The proof proceeds by a reduction that uses a successfully cheating prover $\mathcal{A}^{\mathcal{H}}$ in the FS-transformed protocol to build a successfully cheating prover $\mathcal{S}$ in the Σ-protocol. This can actually be done in a black-box way, so that $\mathcal{S}$ only needs to query $\mathcal{A}$.

We let Y, C, M, E denote the registers storing the commitment, challenge, response, and prover workspace, respectively. A q-query prover $\mathcal{A}^{\mathcal{H}}$ is characterized by a sequence of unitaries $A_0, A_1, \ldots, A_q$ on registers Y, C, M, E. After i queries, it is in the state

$$|\psi_i^{\mathcal{H}}\rangle := A_i U_{\mathcal{H}} A_{i-1} U_{\mathcal{H}} \ldots A_1 U_{\mathcal{H}} A_0 |0\rangle_{YCME} \,. \tag{2.52}$$

In a normal execution, $\mathcal{A}^{\mathcal{H}}$ first prepares $|\psi_q^{\mathcal{H}}\rangle$, then measures Y and M in the standard basis and sends the outcome to $\mathcal{V}_{FS}$ (see Protocol 2.9). The success probability is

$$\Pr_{\mathcal{H}}[V(x, y, \mathcal{H}(x, y), m) = 1,\ (y, m) \leftarrow \mathcal{A}^{\mathcal{H}}(x)]. \tag{2.53}$$

The prover $\mathcal{S}^{\mathcal{A}}$ begins by internally instantiating a quantum-secure pseudorandom function[17] $\mathcal{F}$ [78]. It then chooses a random index $i \in \{0, \ldots, q\}$ and prepares the state $|\psi_i^{\mathcal{F}}\rangle$. Note that the queries of $\mathcal{A}$ are answered using $\mathcal{F}$. Next, the simulator performs a standard basis measurement on register Y and sends the outcome y to $\mathcal{V}$. After $\mathcal{V}$ returns a random challenge Θ, $\mathcal{S}^{\mathcal{A}}$ constructs a reprogrammed oracle, denoted by $\mathcal{F} * \Theta y$, with

$$\mathcal{F} * \Theta y(x, y') = \begin{cases} \Theta & \text{if } y' = y \\ \mathcal{F}(x, y') & \text{otherwise} \end{cases} \tag{2.54}$$

which will be used for the remaining simulation of $\mathcal{A}$. $\mathcal{S}^{\mathcal{A}}$ then tosses a random coin b and performs the following:

1. If $b = 0$, $\mathcal{S}^{\mathcal{A}}$ runs $\mathcal{A}$ with the reprogrammed oracle $\mathcal{F} * \Theta y$ for query $i + 1, \ldots, q$.

2. If $b = 1$, $\mathcal{S}^{\mathcal{A}}$ runs $\mathcal{A}$ with the original oracle $\mathcal{F}$ for query $i + 1$ and with $\mathcal{F} * \Theta y$ for query $i + 2, \ldots, q$.

To generate the message to $\mathcal{V}$, $\mathcal{S}^{\mathcal{A}}$ measures registers Y, M and obtains the outcomes y'', m. If $y'' \neq y$ then $\mathcal{S}^{\mathcal{A}}$ aborts; otherwise $\mathcal{S}^{\mathcal{A}}$ outputs the measurement outcome (y, m). The

[17] If an upper bound on q is known, a $2q$-wise independent function would suffice.

success probability of $\mathcal{S}^{\mathcal{A}}$ is

$$\Pr_{\Theta}[V(x,y,\Theta,m)=1:(y,m)\leftarrow\langle\mathcal{S}^{\mathcal{A}},\Theta\rangle], \tag{2.55}$$

where $\langle\mathcal{S}^{\mathcal{A}},\Theta\rangle$ denotes the interaction between $\mathcal{V}$, who sends a challenge Θ, and $\mathcal{S}^{\mathcal{A}}$ in the Σ-protocol. Theorem 2.7.1 by [64] establishes that the success probabilities of $\mathcal{S}^{\mathcal{A}}$ and $\mathcal{A}^{\mathcal{H}}$ are polynomially related.

Theorem 2.7.1 (Quantum security of Fiat-Shamir [64, Theorem 2])**.** *For every QPT prover $\mathcal{A}^{\mathcal{H}}$ in Protocol 2.9, there exists a QPT prover $\mathcal{S}$ for the underlying Σ-protocol such that*

$$\begin{aligned}&\Pr_{\Theta}[V(x,y,\Theta,m)=1:(y,m)\leftarrow\langle\mathcal{S}^{\mathcal{A}},\Theta\rangle]\\ &\geq\frac{1}{2(2q+1)(2q+3)}\Pr_{\mathcal{H}}[V(x,y,\mathcal{H}(x,y),m)=1,\ (y,m)\leftarrow\mathcal{A}^{\mathcal{H}}(x)]-\frac{1}{(2q+1)|\mathcal{Y}|}.\end{aligned} \tag{2.56}$$

In the above, $(y,m)\leftarrow\langle\mathcal{S}^{\mathcal{A}},\Theta\rangle$ indicates that y and m are the first-round and third-round (respectively) messages of $\mathcal{S}^{\mathcal{A}}$, when it is given the random challenge Θ in the second round.

2.7.2 Extension to generalized Σ-protocols

In this section, we show that Fiat-Shamir also preserves soundness for a more general family of protocols, which we call "generalized Σ-protocols." In such a protocol, $\mathcal{V}$ can begin the protocol by sending an initial message to $\mathcal{P}$.

Notice that the original Mahadev protocol [6] is a generalized Σ-protocol: the distribution D describes the distribution for the secret key, and f computes the public key. Similarly,

Select a public function $f\colon \mathcal{R} \times \mathcal{L} \to \mathcal{W}$, a finite set $\mathcal{C}$, and a distribution D over $\mathcal{R}$. The protocol begins with $\mathcal{P}$ and $\mathcal{V}$ receiving an input x.

Round 1. $\mathcal{V}$ samples randomness $r \in \mathcal{R}$ from distribution D and computes message $w = f(r, x)$, which is sent to $\mathcal{P}$.

Round 2. $\mathcal{P}$ sends a message y to $\mathcal{V}$.

Round 3. $\mathcal{V}$ responds with a uniformly random classical challenge $c \in \mathcal{C}$.

Round 4. $\mathcal{P}$ sends a response m to $\mathcal{V}$.

Verdict. $\mathcal{V}$ outputs a bit computed by a Boolean function $V(r, x, y, c, m)$.

Protocol 2.10: A generalized Σ-protocol.

the k-fold parallel repetition of our instance-independent protocol is also a generalized Σ-protocol since our trusted setup phase can be seen as a message from the verifier.

Fiat-Shamir for generalized Σ protocols. The FS transformation for generalized Σ-protocols is similar to standard ones: in the Verdict stage, $\mathcal{V}$ computes $c = \mathcal{H}(x, w, y)$ and accepts if and only if $V(r, x, y, c, m) = 1$.

Select a public function $f : \mathcal{R} \times \mathcal{L} \to \mathcal{W}$, a finite set $\mathcal{C}$, and a distribution D over $\mathcal{R}$. $\mathcal{P}$ and $\mathcal{V}$ receive an input x and are given access to a random oracle $\mathcal{H}$.

Round 1. $\mathcal{V}$ samples randomness $r \in \mathcal{R}$ from distribution D, and computes message $w = f(r, x)$, which is sent to $\mathcal{P}$.

Round 2. $\mathcal{P}$ sends a message (y, m) to $\mathcal{V}$.

Verdict. $\mathcal{V}$ computes $c = \mathcal{H}(x, w, y)$ and then outputs a bit computed by a Boolean function $V(r, x, y, c, m)$.

Protocol 2.11: The FS-transformed generalized Σ-protocol.

To show that generalized Σ-protocols remain secure under FS transformation, similarly to the idea for Σ-protocols in Section 2.7.1, we give a reduction. Conditioned on any random-

ness r, the prover $\mathcal{A}_r^{\mathcal{H}}(x) := \mathcal{A}^{\mathcal{H}}(x, f(r,x))$ is characterized by unitaries $A_{0,f(r,x)}, \ldots, A_{q,f(r,x)}$. The prover $\mathcal{B}$ in the Σ-protocol runs $\mathcal{S}^{\mathcal{A}_r}$ and outputs its decision. Given the success probability of $\mathcal{A}$, we establish a lower bound on that of $\mathcal{B}$, formally stated as follows.

Lemma 2.7.1 (Fiat-Shamir transformation for generalized Σ-protocols)**.** *Suppose that*

$$\Pr_{r,\mathcal{H}}[V(r,x,y,\mathcal{H}(x,f(r,x),y),m)=1:\ (y,m)\leftarrow\mathcal{A}^{\mathcal{H}}(x,f(r,x))]=\epsilon. \tag{2.57}$$

Then

$$\Pr_{r,\Theta}[V(r,x,y,\Theta,m)=1:\ (y,m)\leftarrow\langle\mathcal{B},\Theta\rangle]\geq\frac{\epsilon}{2(2q+1)(2q+3)}-\frac{1}{(2q+1)|\mathcal{Y}|}. \tag{2.58}$$

Proof. Conditioned on $r=r^*$, the success probability of $\mathcal{A}$ is

$$\begin{aligned}&\Pr_{\mathcal{H}}[V(r,x,y,\mathcal{H}(x,f(r,x),y),m)=1:\ (y,m)\leftarrow\mathcal{A}^{\mathcal{H}}(x,f(x,r))|r=r^*]\\&\quad=\Pr_{\mathcal{H}_{r^*}}[V(r^*,x,y,\mathcal{H}_{r^*}(x,y),m)=1:\ (y,m)\leftarrow\mathcal{A}_{r^*}^{\mathcal{H}}(x)]=:\epsilon(r^*),\end{aligned} \tag{2.59}$$

where $\mathcal{H}_{r^*}(x,y) := \mathcal{H}(x,f(r,x),y)$ and by definition $\mathbb{E}_r[\epsilon(r)] = \epsilon$. Since $\mathcal{H}$ is a random function over $\mathcal{L}\times\mathcal{W}\times\mathcal{Y}\to\mathcal{C}$, for any r^*, $\mathcal{H}_{r^*}$ is a random function over $\mathcal{L}\times\mathcal{Y}\to\mathcal{C}$. Then by Theorem 2.7.1, the success probability of $\mathcal{S}^{\mathcal{A}_{r^*}}$ is

$$\Pr_{\Theta}[V(r^*,x,y,\Theta,m)=1:\ (y,m)\leftarrow\langle\mathcal{S}^{\mathcal{A}_{r^*}},\Theta\rangle]\geq\frac{\epsilon(r^*)}{2(2q+1)(2q+3)}-\frac{1}{(2q+1)|\mathcal{Y}|}. \tag{2.60}$$

Taking the average over r^*, we have

$$\begin{aligned}
&\Pr_{r,\Theta}[V(r,x,y,\Theta,m)=1:\ (y,m)\leftarrow\langle\mathcal{B},\Theta\rangle] \\
&\quad=\mathop{\mathbb{E}}_{r^*}\Big[\Pr_{\Theta}[V(r^*,x,y,\Theta,m)=1:\ (y,m)\leftarrow\langle\mathcal{S}^{\mathcal{A}_{r^*}},\Theta\rangle|r=r^*]\Big] \\
&\quad=\mathop{\mathbb{E}}_{r^*}\left[\frac{\epsilon(r^*)}{2(2q+1)(2q+3)}-\frac{1}{(2q+1)|\mathcal{Y}|}\right] \\
&\quad=\frac{\epsilon}{2(2q+1)(2q+3)}-\frac{1}{(2q+1)|\mathcal{Y}|}
\end{aligned} \tag{2.61}$$

as claimed. □

Lemma 2.7.1 immediately gives the following theorem.

Theorem 2.7.2. *If a language $\mathcal{L}$ admits a generalized Σ-protocol with soundness error s, then after the Fiat-Shamir transformation, the soundness error against provers who make up to q queries to a random oracle is $O(sq^2+q|\mathcal{Y}|^{-1})$.*

Proof. Suppose that there is a prover who succeeds in the transformed protocol with success probability ϵ. Then by Lemma 2.7.1, we may construct a prover who succeeds with probability at least $\frac{\epsilon}{O(q^2)}-O\left(\frac{1}{q|\mathcal{Y}|}\right)$. By the soundness guarantee, we have $\frac{\epsilon}{O(q^2)}-O\left(\frac{1}{q|\mathcal{Y}|}\right)\leq s$ and thus $\epsilon\leq O(q^2s+q|\mathcal{Y}|^{-1})$. □

By Theorem 2.7.2, if both s and $|\mathcal{Y}|^{-1}$ are negligible in security parameter λ, the soundness error of the transformed protocols remains negligible against an efficient prover who makes $q=\mathrm{poly}(\lambda)$ queries. Theorem 2.1.3 follows directly from Theorem 2.7.2.

2.7.3 Non-interactive zero-knowledge for QMA

We now show that, using the Fiat-Shamir transformation, our three-round protocol proposed in Protocol 2.6 can be converted into a non-interactive zero-knowledge argument (with trusted setup) for QMA in the Quantum Random Oracle model. The resulting protocol is defined exactly as Protocol 2.6, with two modifications: (i.) instead of Round $\mathcal{V}_2$, the prover $\mathcal{P}$ computes the coins c by evaluating the random oracle $\mathcal{H}$ on the protocol transcript thus far, and (ii.) the NIZK instance x is appropriately redefined using these coins.

We remark that since the setup in this protocol is trusted, it follows from Theorem 2.7.2 that the compressed protocol is complete and sound, and therefore we just need to argue about the zero-knowledge property.

Theorem 2.7.3. *The Fiat-Shamir transformation of Protocol 2.6 is zero-knowledge.*

Proof. The simulator $\mathcal{S}^{\mathcal{V}_2^*}$ can sample the trapdoor keys for NTCF/NTIF functions and private keys for the FHE scheme, enabling simulation of the transcript for every challenge sent by the verifier. In particular, one can run the same proof of Section 2.6.2 with the variant $\mathcal{S}^{\mathcal{H}}$ that queries the random oracle $\mathcal{H}$ for the challenges instead of receiving it from a malicious verifier $\mathcal{V}^*$. □

Chapter 3: Polynomial interpolation

3.1 Introduction

Let $f(X) = c_d X^d + \cdots + c_1 X + c_0 \in \mathbb{F}_q[X]$ be an unknown polynomial of degree d, specified by its coefficient vector $c \in \mathbb{F}_q^{d+1}$. Suppose q and d are known[1] and we are given a black box that evaluates f on any desired $x \in \mathbb{F}_q$. In the *polynomial interpolation problem*, our goal is to learn f—that is, to determine the vector c—by querying this black box. We would like to determine how many queries are required to solve this problem.

The classical query complexity of polynomial interpolation is well known: $d+1$ queries to f are clearly sufficient and are also necessary to determine the polynomial, even with bounded error. Shamir [26] used this fact to construct a cryptographic protocol that divides a secret into $d+1$ parts such that knowledge of all the parts can be used to infer the secret, but any d parts give no information about the secret. The security of this protocol relies on the fact that if f is chosen uniformly at random, and if we only know d function values $f(x_1), \ldots, f(x_d)$, then we cannot guess the value $f(x_{d+1})$ for a point $x_{d+1} \notin \{x_1, \ldots, x_d\}$ with probability greater than $1/q$ (that is, there is no advantage over random guessing). This example motivates understanding the query complexity of polynomial interpolation precisely, since a single query can dramatically increase the amount of information that can

[1] We assume $q > d$ so that different coefficients correspond to distinct functions $f \colon \mathbb{F}_q \to \mathbb{F}_q$.

be extracted.

The quantum query complexity of polynomial interpolation has also been studied previously. Kane and Kutin [27] and Meyer and Pommersheim [28] independently showed that $d/2 + 1/2$ quantum queries are needed to solve the problem with bounded error. Furthermore, Kane and Kutin conjectured that $d + 1$ quantum queries might be necessary. This was refuted by Boneh and Zhandry, who showed that d quantum queries suffice to solve the problem with probability[2] $1 - O(1/q)$ [79]. To show this, they described a 1-query quantum algorithm that determines a linear polynomial with probability $1 - O(1/q)$. The result for general d follows because $d - 1$ classical queries can be used to reduce the case of a degree-d polynomial to that of a linear polynomial. However, this work left a substantial gap between the lower and upper bounds.

Here we present an improved quantum algorithm for polynomial interpolation. We show that the aforementioned lower bounds are tight: with d fixed, $k = d/2 + 1/2$ queries suffice to solve the problem with constant success probability. While the success probability at this value of k has a q-independent lower bound, it decreases rapidly with k, scaling like $1/k!$. This raises the question of how the success probability increases as we make more queries. We show that there is a sharp transition as k is increased: in particular, with $k = d/2 + 1$ queries, the algorithm succeeds with a probability that approaches 1 for large q.

Our algorithm is motivated by the pretty good measurement (PGM) approach to the hidden subgroup problem (HSP) [80]. In this approach, one queries the black box on uniform superpositions to create *coset states* and then makes entangled measurements on several

[2]While the notation $O(\cdot)$ only indicates an asymptotic upper bound on the absolute value, we sometimes write $1 - O(\cdot)$ to indicate a bound on a quantity that is at most 1.

coset states to infer the hidden subgroup. As in the PGM approach (and in other approaches to the HSP using the so-called standard method), our algorithm makes nonadaptive queries to the black box and performs collective postprocessing. Also, similarly to previous analysis of the PGM approach, we can express our success probability in terms of the number of solutions of a system of polynomial equations.

However, our approach to polynomial interpolation also has significant differences from the PGM approach to the HSP. In particular, we introduce a different way to query the black box that simplifies both the algorithm and its analysis. In the PGM approach, we query the black box on a uniform superposition and then uncompute uniform superpositions over certain sets. For polynomial interpolation, we instead query a carefully-chosen non-uniform superposition of inputs so that the subsequent uncomputation is classical. Furthermore, the success probability of our method is higher, and its analysis is more straightforward, than if we used a direct analog of the PGM approach. We hope that these techniques will prove useful for other quantum algorithms, perhaps for the hidden subgroup problem or for other applications of the PGM approach [81, 82].

We also show that our strategy is precisely optimal: for any number of queries k, we describe a k-query algorithm with the highest possible success probability. We give a simple algebraic characterization of this success probability, as follows.

Theorem 3.1.1. *The maximum success probability of any k-query quantum algorithm for interpolating a polynomial of degree d over $\mathbb{F}_q$ is $|R_k|/q^{d+1}$, where $R_k := Z(\mathbb{F}_q^k \times \mathbb{F}_q^k)$ is the range of the function $Z\colon \mathbb{F}_q^k \times \mathbb{F}_q^k \to \mathbb{F}_q^{d+1}$ defined by $Z(x,y)_j := \sum_{i=1}^k y_i x_i^j$ for $j \in \{0,1,\ldots,d\}$.*

We present an explicit quantum algorithm that achieves this success probability, and

we show that no algorithm can do better. We establish optimality with an argument based on the dimension of the space spanned by the possible output states, which appears to be distinct from arguments using the two main approaches to proving limitations on quantum algorithms, the polynomial and adversary methods. Instead, our approach is closely related to a linear-algebraic lower bound technique of Radhakrishnan, Sen, and Venkatesh [83] and to the "rank method" of Boneh and Zhandry [79].

We characterize the query complexity by proving bounds on $|R_k|$, as follows.

Theorem 3.1.2. *For any fixed positive integer d, the success probability of Theorem 3.1.1 is*

(i) $|R_k|/q^{d+1} = \frac{1}{k!}(1 - O(1/q))$ *if d is odd and $k = \frac{d}{2} + \frac{1}{2}$, and*

(ii) $|R_k|/q^{d+1} = 1 - O(1/q)$ *if d is even and $k = \frac{d}{2} + 1$.*

To show the former bound, we explicitly characterize the possible $(x, y) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$ such that $Z(x, y)$ takes a particular value. We prove the latter bound in a completely different way, using a second moment argument.

Theorem 3.1.2 shows that the success probability has a sharp transition as a function of k, from subconstant for $k < d/2 + 1/2$ (by known lower bounds [27, 28]), to a (d-dependent) constant for $k = d/2 + 1/2$, to $1 - o(1)$ for $k = d/2 + 1$. Note that since k must be an integer, the success probability varies differently with k depending on whether d is odd or even. For fixed even d, $k = d/2 + 1$ queries give success probability $1 - o(1)$, whereas $k = d/2$ queries give success probability $o(1)$. For fixed odd d, the success probability is $o(1)$ for $k = d/2 - 1/2$ and constant for $k = d/2 + 1/2$. To achieve higher success probability, we can make $k = d/2 + 3/2$ queries and treat f as a polynomial of degree $d + 1$ with $c_{d+1} = 0$,

giving success probability $1 - o(1)$.

In light of these results, polynomial interpolation is reminiscent of the task of computing the parity of n bits, where the classical query complexity is n (even for bounded error) and the quantum query complexity is $n/2$ [20, 84]. More generally, a similar factor-of-two improvement is possible for the oracle interrogation problem, where the goal is to learn the entire n-bit string encoded by a black box [85]. However, polynomial interpolation is qualitatively different in that the oracle returns values over $\mathbb{F}_q$ rather than $\mathbb{F}_2$. Note that for the oracle interrogation problem over $\mathbb{F}_q$, one can only achieve speedup by a factor of about $1 - 1/q$ [79, Section 4], which is negligible for large q.

Our algorithm improves results of Boneh and Zhandry giving quantum attacks on certain cryptographic protocols [79]. For a version of the Shamir secret sharing scheme [26] where the shares can be quantum superpositions, their d-query interpolation algorithm shows that a subset of only d parties can recover the secret. Our algorithm considerably strengthens this, showing that a subset of $d/2 + 1/2$ parties can recover the secret with constant probability, and $d/2 + 1$ can recover it with probability $1 - O(1/q)$. Boneh and Zhandry also formulate a model of quantum message-authentication codes (MACs), where the goal is to tag messages to authenticate the sender. Informally, a MAC is called d-time if, given the ability to create d valid message-tag pairs, an attacker cannot forge another valid message-tag pair. Boneh and Zhandry show that there are $(d + 1)$-wise independent functions that are not d-time quantum MACs. Our result improves this to show that there are $(d + 1)$-wise independent functions that are not $(d/2 + 1/2)$-time quantum MACs.

We consider the gate complexity of polynomial interpolation. We call an algorithm *gate-efficient* if it can be implemented with a number of 2-qubit gates that is only larger

than its query complexity by a factor of $\mathrm{poly}(\log q)$. We construct a gate-efficient variant of our algorithm that achieves almost the same success probability.[3]

Theorem 3.1.3. *For any fixed positive integer d, there is a gate-efficient quantum algorithm for interpolating a polynomial of degree d over $\mathbb{F}_q$ using*

(i) $k = \frac{d}{2} + \frac{1}{2}$ *queries, succeeding with probability* $\frac{1}{k!}(1 - O(1/q))$*, if d is odd; and*

(ii) $k = \frac{d}{2} + 1$ *queries, succeeding with probability* $1 - o(1)$*, if d is even.*

The main step in implementing the algorithm is to invert the function Z described in the statement of Theorem 3.1.1, i.e., to find some $x, y \in \mathbb{F}_q^k$ so that $Z(x, y)$ takes a given value. We achieve this by characterizing the solutions in terms of a polynomial equation and a system of linear equations.

For multivariate polynomials, in [30], we conjectured that a straightforward analog of the univariate algorithm solves the interpolation problem with probability $1 - o(1)$ using $\lfloor \frac{1}{n+1}\binom{n+d}{d} \rfloor + 1$ queries. However, while that conjecture is natural, the analysis of the algorithm appeared to require solving a difficult problem in algebraic geometry and was left open. (In addition, Montanaro considered the quantum query complexity of interpolating a multilinear polynomial [86], but this is quite different from the general multivariate case.)

To the best of our knowledge, all previous work on quantum algorithms for polynomial interpolation has focused on finite fields. Cryptographic applications of interpolation typically use finite fields, and the multivariate case could lead to new applications in that domain. However, polynomial interpolation over infinite fields is also a natural problem,

[3] Note that while our algorithm for $k = d/2 + 1/2$ has gate complexity polynomial in both $\log q$ and d, the algorithm for $k = d/2 + 1$ has gate complexity $k! \, \mathrm{poly}(\log q)$. Improving the dependence on d is a natural open question.

especially considering the ubiquity of real- and complex-valued polynomials in numerical analysis.

We propose an approach to the quantum query complexity of polynomial interpolation in the continuum limit. To obtain a well-defined initial state, the algorithm prepares a superposition over a bounded working region. The bounded region limits the precision that can be achieved due to the uncertainty principle, but the algorithm can be made arbitrarily precise by taking an arbitrarily large region. Using this strategy, we present a quantum algorithm for multivariate polynomial interpolation over the real and complex numbers. To simplify the analysis, we allow the algorithm to work with arbitrarily precise inputs and outputs over $\mathbb{R}$ or $\mathbb{C}$; in practice, sufficiently fine discretization of the space could achieve similar performance. We also consider multivariate polynomial interpolation over finite fields, where our algorithm can be viewed as a generalization of the univariate polynomial interpolation algorithm proposed in [30].

To analyze the success probability of our approach, we relate it to the tensor rank problem. The rank of a given tensor, which is the smallest integer k such that the tensor can be decomposed as a linear combination of k simple tensors (i.e., those that can be written as tensor products), was first introduced nearly a century ago. A half century later, with the advent of principal component analysis on multidimensional arrays, the study of tensor rank attracted further attention. However, it has recently been shown that most tensor problems, including tensor rank, are NP-hard [87, 88, 89], and restricting these problems to symmetric tensors does not seem to alleviate their NP-hardness [88, 89]. More specifically, tensor rank is NP-hard over any field extension of $\mathbb{Q}$ and NP-complete over a finite field $\mathbb{F}_q$.

Fortunately, analyzing the success probability of multivariate polynomial interpolation

does not require exactly computing the rank of a symmetric tensor. The number of queries needed to achieve success probability 1 can be translated to the smallest integer k such that *almost every* symmetric tensor can be decomposed as a linear combination of no more than k simple tensors. In turn, this quantity can be related to properties of certain secant varieties, which lets us take advantage of recent progress in algebraic geometry [90, 91].

The success probability of our algorithm behaves differently as a function of the number of queries for the three fields we consider. Specifically, by introducing

$$k_{\mathbb{C}}(n,d) := \begin{cases} n+1 & d=2, n\geq 2; \\ \lceil \frac{1}{n+1}\binom{n+d}{d} \rceil + 1 & (n,d) = (4,3),(2,4),(3,4),(4,4); \\ \lceil \frac{1}{n+1}\binom{n+d}{d} \rceil & \text{otherwise,} \end{cases} \tag{3.1}$$

we have the following upper bounds on the query complexity:

Theorem 3.1.4. *For positive integers d and n, there exists a quantum algorithm for interpolating an n-variate polynomial of degree d over the field $\mathbb{K}$ using at most*

1. $\frac{d}{n+d}\binom{n+d}{d}$ *queries for $\mathbb{K} = \mathbb{F}_q$, succeeding with probability $1 - O(1/q)$;*
2. $2k_{\mathbb{C}}$ *queries for $\mathbb{K} = \mathbb{R}$, succeeding with probability 1;*
3. $k_{\mathbb{C}}$ *queries for $\mathbb{K} = \mathbb{C}$, succeeding with probability 1.*

The remainder of the paper is organized as follows. After introducing some definitions in Section 3.3.1, we describe our k-query algorithm in Section 3.3.2. We analyze the success probability of this algorithm for $k = d/2 + 1/2$ in Section 3.3.3, and for $k = d/2 + 1$ in Section 3.3.4. We also show in Section 3.3.5 that essentially the same performance can

be achieved using k independent queries to the oracle, each on a uniform superposition of inputs (which might make some cryptographic attacks easier, depending on the model). In Section 3.3.6, we describe the gate-efficient version of our algorithm.

For multivariate polynomial interpolation, we describe the query model in Section 3.4.1 and present our algorithms in Section 3.4.2. We then analyze the algorithm to establish our query complexity upper bounds in Section 3.4.3.

Finally, we establish optimality of our algorithm over finite fields in Section 3.5.

3.2 Algebraic geometry concepts

A subset V of $\mathbb{K}^n$ is an *algebraic set* if it is the set of common zeros of a finite collection of polynomials $g_1, g_2, \ldots, g_r$ with $g_i \in \mathbb{K}[x_1, x_2, \ldots, x_n]$ for $1 \leq i \leq r$.

A finite union of algebraic sets is an algebraic set, and an arbitrary intersection of algebraic sets is again an algebraic set. Thus by taking the open subsets to be the complements of algebraic sets, we can define a topology, called the *Zariski topology*, on $\mathbb{K}^n$.

A nonempty subset V of a topological space X is called *irreducible* if it cannot be expressed as the union of two proper (Zariski) closed subsets. The empty set is not considered to be irreducible. An *affine algebraic variety* is an irreducible closed subset of some $\mathbb{K}^n$.

We define *projective n-space*, denoted by $\mathbb{P}^n$, to be the set of equivalence classes of $(n+1)$-tuples $(a_0, \ldots, a_n)$ of complex numbers, not all zero, under the equivalence relation given by $(a_0, \ldots, a_n) \sim (\lambda a_0, \ldots, \lambda a_n)$ for all $\lambda \in \mathbb{K}$, $\lambda \neq 0$.

A notion of algebraic variety may also be introduced in projective spaces, giving the notion of a projective algebraic variety: a subset $V \subseteq \mathbb{P}^n$ is an *algebraic set* if it is the

set of common zeros of a finite collection of homogeneous polynomials $g_1, g_2, \ldots, g_r$ with $g_i \in \mathbb{K}[x_0, x_1, \ldots, x_n]$ for $1 \leq i \leq r$. We call open subsets of irreducible projective varieties quasi-projective varieties.

For any integers n and d, we define the Veronese map of degree d as

$$V_d \colon [x_0 : x_1 : \cdots : x_n] \mapsto [x_0^d : x_0^{d-1} x_1 : \cdots : x_n^d] \tag{3.2}$$

where the notation with square brackets and colons denotes homogeneous coordinates and the expression in the output of V_d ranges over all monomials of degree d in $x_0, x_1, \ldots, x_n$. The image of V_d is an algebraic variety called a *Veronese variety*.

Finally, for an irreducible algebraic variety V, its k^{th} *secant variety* $\sigma_k(V)$ is the Zariski closure of the union of subspaces spanned by k distinct points chosen from V.

For more information about Veronese and secant varieties, refer to Example 2.4 and Example 11.30 in [92].

3.3 Univariate polynomial interpolation

3.3.1 Preliminaries

Let $f(X) = c_d X^d + \cdots + c_1 X + c_0 \in \mathbb{F}_q[X]$ be an unknown polynomial of degree d that is specified by the vector of coefficients $c \in \mathbb{F}_q^{d+1}$, where $q = p^r$ a power of a prime p. Access to f is provided by a black box acting as $|x, y\rangle \mapsto |x, y + f(x)\rangle$ for all $x, y \in \mathbb{F}_q$.

Let $e \colon \mathbb{F}_q \to \mathbb{C}$ be the exponential function $e(z) = \mathrm{e}^{2\pi \mathrm{i} \operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(z)/p}$, where the trace function $\operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p} \colon \mathbb{F}_q \to \mathbb{F}_p$ is defined by $\operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(z) = z + z^p + z^{p^2} + \cdots + z^{p^{r-1}}$. The Fourier

transform over $\mathbb{F}_q$ is the unitary transformation acting as $|x\rangle \mapsto \frac{1}{\sqrt{q}} \sum_{y\in\mathbb{F}_q} e(xy)|y\rangle$ for all $x \in \mathbb{F}_q$.

We can compute the value of f into the phase by Fourier transforming the second query register. If we apply the inverse Fourier transform, perform a query, and then apply the Fourier transform, we have the transformation

$$\begin{aligned} |x, y\rangle &\mapsto \frac{1}{\sqrt{q}} \sum_{z\in\mathbb{F}_q} e(-yz)|x, z\rangle \\ &\mapsto \frac{1}{\sqrt{q}} \sum_{z\in\mathbb{F}_q} e(-yz)|x, z + f(x)\rangle \\ &\mapsto \frac{1}{q} \sum_{z,w\in\mathbb{F}_q} e(-yz + (z + f(x))w)|x, w\rangle \\ &= e(yf(x))|x, y\rangle \end{aligned} \tag{3.3}$$

for any $x, y \in \mathbb{F}_q$, where we used the fact that $\sum_{z\in\mathbb{F}_q} e(zv) = q\delta_{z,v}$. We call the transformation $|x, y\rangle \mapsto e(yf(x))|x, y\rangle$ a *phase query*. Since a phase query can be implemented with a single standard query and vice versa, the query complexity of a problem does not depend on which type of query we use.

For vectors $x, y \in \mathbb{F}_q^k$, we denote the inner product over $\mathbb{F}_q$ by $x \cdot y := \sum_{i=1}^k x_i y_i$. The k-fold Fourier transform (i.e., the Fourier transform acting independently on each register) acts as $|x\rangle \mapsto \frac{1}{\sqrt{q^k}} \sum_{y\in\mathbb{F}_q^k} e(x \cdot y)|y\rangle$ for any $x \in \mathbb{F}_q^k$.

3.3.2 The algorithm

We now describe our algorithm for polynomial interpolation. An ideal algorithm would produce the Fourier transform of the coefficient vector $c \in \mathbb{F}_q^{d+1}$, that is, the state

$$|\hat{c}\rangle = \frac{1}{\sqrt{q^{d+1}}} \sum_{z \in \mathbb{F}_q^{d+1}} e(c \cdot z)|z\rangle. \tag{3.4}$$

Instead we use k quantum queries to create the approximate state

$$|\hat{c}_{R_k}\rangle := \frac{1}{\sqrt{|R_k|}} \sum_{z \in R_k} e(c \cdot z)|z\rangle \tag{3.5}$$

for some set $R_k \subseteq \mathbb{F}_q^{d+1}$. A measurement of this state in the Fourier basis gives c with probability $|\langle \hat{c}_{R_k} | \hat{c} \rangle|^2 = |R_k|/q^{d+1}$.

Our algorithm performs k phase queries in parallel, each acting on a separate register. On input $|x, y\rangle$ for $x, y \in \mathbb{F}_q^k$, these k queries introduce the phase $e(\sum_{i=1}^k y_i f(x_i))$. To define the set R_k, recall the function $Z \colon \mathbb{F}_q^k \times \mathbb{F}_q^k \to \mathbb{F}_q^{d+1}$ defined by

$$Z(x, y)_j := \sum_{i=1}^k y_i x_i^j \text{ for } j \in \{0, 1, \ldots, d\}. \tag{3.6}$$

Then we have $\sum_{i=1}^k y_i f(x_i) = \sum_{i=1}^k \sum_{j=0}^d y_i c_j x_i^j = c \cdot Z(x, y)$ for all $x, y \in \mathbb{F}_q^k$. The range $R_k := Z(\mathbb{F}_q^k \times \mathbb{F}_q^k)$ of the function Z is the set

$$R_k = \{Z(x, y) : (x, y) \in \mathbb{F}_q^k \times \mathbb{F}_q^k\} \subseteq \mathbb{F}_q^{d+1}. \tag{3.7}$$

For each $z \in R_k$ we choose a unique $(x, y) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$ such that $Z(x, y) = z$. Let $T_k \subseteq \mathbb{F}_q^k \times \mathbb{F}_q^k$ be the set of these representatives. Clearly, $Z \colon T_k \to R_k$ is a bijection.

To create the state $|\hat{c}_{R_k}\rangle$, we prepare a uniform superposition over T_k, perform k phase queries, and compute Z in place (i.e., perform the unitary transformation $|x, y\rangle \mapsto |Z(x, y)\rangle$), giving

$$\begin{aligned} \frac{1}{\sqrt{|T_k|}} \sum_{(x,y)\in T_k} |x, y\rangle &\mapsto \frac{1}{\sqrt{|T_k|}} \sum_{(x,y)\in T_k} e(c \cdot Z(x, y))|x, y\rangle \\ &\mapsto \frac{1}{\sqrt{|R_k|}} \sum_{z\in R_k} e(c \cdot z)|z\rangle. \end{aligned} \tag{3.8}$$

The above procedure is a k-query algorithm for polynomial interpolation that succeeds with probability $|R_k|/q^{d+1}$, establishing the lower bound on the success probability stated in Theorem 3.1.1. To analyze the algorithm, it remains to lower bound $|R_k|$ as a function of k.

3.3.3 Performance using $d/2 + 1/2$ queries

We now consider the performance of the above algorithm using $k = d/2 + 1/2$ queries. Let

$$Z^{-1}(z) = \{(x, y) \in \mathbb{F}_q^k \times \mathbb{F}_q^k : Z(x, y) = z\} \tag{3.9}$$

be the set of those $(x, y) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$ corresponding to a particular $z \in \mathbb{F}_q^{d+1}$. Clearly $|R_k|$ is the number of values of z such that $Z^{-1}(z)$ is nonempty. To analyze this, we focus on "good" values of (x, y). Define $X_k^{\text{good}} := \{x \in \mathbb{F}_q^k : x_i \neq x_j \; \forall\, i \neq j\}$ and $Y_k^{\text{good}} := (\mathbb{F}_q^\times)^k$ and let $Z^{-1}(z)^{\text{good}} := Z^{-1}(z) \cap (X_k^{\text{good}} \times Y_k^{\text{good}})$. We claim the following:

Lemma 3.3.1. *If $k = d/2 + 1/2$, then for all $z \in \mathbb{F}_q^{d+1}$, either $|Z^{-1}(z)^{\text{good}}| = 0$ or $|Z^{-1}(z)^{\text{good}}| = k!$.*

Proof. We can write the condition $Z(x, y) = z$ in the form $\sum_i y_i \boldsymbol{x}_i = z$, where $\boldsymbol{x}_i := (1, x_i, x_i^2, \ldots, x_i^d)$. We claim that for a given $z \in \mathbb{F}_q^{d+1}$, the values $(x, y) \in X^{\text{good}} \times Y^{\text{good}}$ that satisfy this equation are unique up to a permutation of the indices. To see this, suppose that $Z(x, y) = Z(u, v)$ for some good values $(x, y) \neq (u, v)$. By permuting the indices, we can ensure that $x_i = u_i$ for $i \in \{1, \ldots, m\}$ and $x_i \neq u_i$ for $i \in \{m + 1, \ldots, k\}$, where m is the number of positions at which x and u agree. Then we have

$$\sum_{i=1}^{m} (y_i - v_i)\boldsymbol{x}_i + \sum_{i=m+1}^{k} y_i \boldsymbol{x}_i + \sum_{i=m+1}^{k} v_i \boldsymbol{u}_i = 0. \tag{3.10}$$

It is well known that the Vandermonde matrix

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_{d+1} \\ x_1^2 & x_2^2 & \cdots & x_{d+1}^2 \\ \vdots & \vdots & & \vdots \\ x_1^d & x_2^d & \cdots & x_{d+1}^d \end{pmatrix} \tag{3.11}$$

is invertible provided the values $x_1, x_2, \ldots, x_{d+1}$ are distinct. Because the values x_i for $i \in \{1, \ldots, k\}$ and u_i for $i \in \{m+1, \ldots, k\}$ are all distinct, and because the number of terms in (3.10) is at most $2k \leq d+1$, the vectors $\boldsymbol{x}_i$ for $i \in \{1, \ldots, k\}$ and $\boldsymbol{u}_i$ for $i \in \{m+1, \ldots, k\}$ are linearly independent. Thus we have $y_i = v_i$ for all $i \in \{1, \ldots, m\}$ and $y_i = v_i = 0$ for all $i \in \{m + 1, \ldots, k\}$. Since $y \in Y^{\text{good}}$, we cannot have $y_i = 0$ for any i, so we must have

$m = k$. Therefore $x = u$ and $y = v$. It follows that the only way to obtain a distinct (x, y) is to permute the indices, and therefore we either have $|Z^{-1}(z)^{\text{good}}| = 0$ (if there is no $(x, y) \in X^{\text{good}} \times Y^{\text{good}}$ such that $Z(x, y) = z$) or $|Z^{-1}(z)^{\text{good}}| = k!$. □

Using Lemma 3.3.1, we can show that $k = d/2 + 1/2$ queries suffice to perform polynomial interpolation with probability that is independent of q, but that decreases with d.

Proof of Theorem 3.1.2(i): $k = d/2 + 1/2$. We have $|X_k^{\text{good}}| = q!/(q-k)!$ and $|Y_k^{\text{good}}| = (q-1)^k$, so

$$\sum_{z \in \mathbb{F}_q^{d+1}} |Z^{-1}(z)^{\text{good}}| = |X_k^{\text{good}}| \cdot |Y_k^{\text{good}}| = \frac{q!}{(q-k)!}(q-1)^k = q^{2k}(1 - O(1/q)). \tag{3.12}$$

Thus, invoking Lemma 3.3.1, the number of values of z for which $|Z^{-1}(z)^{\text{good}}| = k!$ is at least $\frac{q^{2k}}{k!}(1 - O(1/q))$. Since $k = d/2 + 1/2$, it follows that $|R_k|/q^{d+1}$ is at least $\frac{1}{k!}(1 - O(1/q))$, as claimed. □

3.3.4 Performance using $d/2 + 1$ queries

Next we show that with more than $d/2+1/2$ queries, the success probability approaches 1 for large q.

Proof of Theorem 3.1.2(ii): $k = d/2 + 1$. Under the uniform distribution on $z \in \mathbb{F}_q^{d+1}$, we have

$$|R_k|/q^{d+1} = 1 - \Pr[|Z^{-1}(z)| = 0]. \tag{3.13}$$

We use a second moment argument to upper bound the number of $z \in \mathbb{F}_q^{d+1}$ for which $|Z^{-1}(z)| = 0$. The mean of $|Z^{-1}(z)|$ is

$$\mu := \frac{1}{q^{d+1}} \sum_{z \in \mathbb{F}_q^{d+1}} |Z^{-1}(z)| = q^{2k-(d+1)}. \tag{3.14}$$

Let $\delta[\mathcal{P}]$ be 1 if $\mathcal{P}$ is true and 0 if $\mathcal{P}$ is false. For the second moment, we compute

$$\begin{aligned}
\sum_{z \in \mathbb{F}_q^{d+1}} |Z^{-1}(z)|^2 &= \sum_{u,v,x,y \in \mathbb{F}_q^k} \delta[Z(u,v) = Z(x,y)] \\
&= \sum_{u,v,x,y \in \mathbb{F}_q^k} \frac{1}{q^{d+1}} \sum_{\lambda \in \mathbb{F}_q^{d+1}} e(\lambda \cdot (Z(u,v) - Z(x,y))) \\
&= \frac{q^{4k}}{q^{d+1}} + \frac{1}{q^{d+1}} \sum_{\lambda \in \mathbb{F}_q^{d+1} \setminus (0,\ldots,0)} \left(\sum_{x,y \in \mathbb{F}_q} e\left(y \sum_{j=0}^{d} \lambda_j x^j \right) \right)^{2k} \\
&= q^{4k-(d+1)} + \frac{1}{q^{d+1}} \sum_{\lambda \in \mathbb{F}_q^{d+1} \setminus (0,\ldots,0)} \left(q \sum_{x \in \mathbb{F}_q} \delta\left[\sum_{j=0}^{d} \lambda_j x^j = 0 \right] \right)^{2k} \\
&\leq q^{4k-(d+1)} + (qd)^{2k}.
\end{aligned} \tag{3.15}$$

Thus for the variance, we have

$$\sigma^2 := \frac{1}{q^{d+1}} \sum_{z \in \mathbb{F}_q^{d+1}} |Z^{-1}(z)|^2 - \mu^2 \leq \frac{(qd)^{2k}}{q^{d+1}}. \tag{3.16}$$

(note that $\sigma^2 \geq 0$ by the Cauchy inequality). Applying the Chebyshev inequality, we find

$$\Pr[Z^{-1}(z) = 0] \leq \frac{\sigma^2}{\mu^2} \leq \frac{(qd)^{2k}/q^{d+1}}{q^{4k-2(d+1)}} = d^{2k} q^{d+1-2k}. \tag{3.17}$$

Therefore $|R_k|/q^{d+1} = 1 - \Pr[Z^{-1}(z) = 0] \geq 1 - d^{2k}q^{d+1-2k}$. With $k = d/2 + 1$, we have

$$|R_k|/q^{d+1} \geq 1 - d^{2k}/q = 1 - O(1/q) \tag{3.18}$$

as claimed. □

Note that one can improve the dependence on d in (3.16) using results on the distribution of zeros in random polynomials [93].

3.3.5 An alternative algorithm

The algorithm described above queries the oracle nonadaptively, that is, all k queries can be performed in parallel. However, the input state to these queries is correlated across all k copies. In this section, we describe an alternative algorithm that queries the black box on a state that is independent and identical for each of the k queries, namely, a uniform superposition over all inputs. This algorithm is suboptimal, but its performance is not significantly worse than that of the optimal algorithm described in Section 3.3.2.

Analogous to the so-called standard method for the hidden subgroup problem, querying f on a uniform superposition gives the state $\frac{1}{\sqrt{q}} \sum_{x \in \mathbb{F}_q^k} |x, f(x)\rangle$. If we use k queries to prepare k copies of this state and then perform the Fourier transform on the second register (or equivalently, perform k independent phase queries), we obtain the state

$$\frac{1}{q^k} \sum_{x,y \in \mathbb{F}_q^k} e(c \cdot Z(x,y))|x,y\rangle = \frac{1}{q^k} \sum_{z \in \mathbb{F}_q^{d+1}} e(c \cdot z) \sqrt{|Z^{-1}(z)|}\, |Z^{-1}(z)\rangle \tag{3.19}$$

where $|Z^{-1}(z)\rangle := \sum_{(x,y) \in Z^{-1}(z)} |x,y\rangle / |Z^{-1}(z)|^{1/2}$. Motivated by the PGM approach to the

hidden subgroup problem [80], suppose we perform the transformation $|Z^{-1}(z)\rangle \mapsto |z\rangle$, giving the state

$$|\phi_k^c\rangle := \frac{1}{q^k} \sum_{z \in \mathbb{F}_q^{d+1}} e(c \cdot z) \sqrt{|Z^{-1}(z)|}\, |z\rangle. \tag{3.20}$$

Measuring this state in the Fourier basis gives the outcome c with probability

$$|\langle \phi_k^c | \hat{c} \rangle|^2 = \frac{1}{q^{2k+d+1}} \left(\sum_{z \in \mathbb{F}_q^{d+1}} \sqrt{|Z^{-1}(z)|} \right)^2. \tag{3.21}$$

If $k = d/2 + 1/2$, we claim that this algorithm succeeds with constant probability. From the proof of Theorem 3.1.2 for $k = d/2 + 1/2$, we have that $|Z^{-1}(z)| \geq k!$ for at least $\frac{q^{2k}}{k!}(1 - O(1/q))$ values of z. Therefore the success probability is at least $\frac{1}{k!}(1 - O(1/q))$.

If $k = d/2 + 1$, then this algorithm succeeds with probability that approaches 1 for large q. To see this, recall from the proof of Theorem 3.1.2 for $k = d/2 + 1$ that, under a uniform distribution over $z \in \mathbb{F}_q^{d+1}$, the quantity $Z^{-1}(z)$ has mean $\mu = q$ and standard deviation $\sigma = \sqrt{q} d^k$. Thus, by the Chebyshev inequality, we have

$$\Pr\left[|Z^{-1}(z)| \leq q - \alpha \sqrt{q} d^k \right] \leq \frac{1}{\alpha^2}. \tag{3.22}$$

It follows that

$$|\langle \phi_k^c | \hat{c} \rangle|^2 \geq \left(1 - \frac{\alpha d^k}{\sqrt{q}} \right) \left(1 - \frac{1}{\alpha^2} \right)^2. \tag{3.23}$$

Choosing $\alpha = \Theta(q^{1/6})$, this gives a success probability of $|\langle \phi_k^c | \hat{c} \rangle|^2 = 1 - O(q^{-1/3})$, which

approaches 1 for large q.

3.3.6 Gate complexity

In Section 3.3, we analyzed the query complexity of our polynomial interpolation algorithm. Here we describe a $(d/2+1/2)$-query algorithm whose gate complexity is $\mathrm{poly}(\log q)$, and whose success probability is close to that of the best algorithm using this number of queries (in particular, for fixed d it still succeeds with constant probability). We also give an algorithm for the case $k = d/2+1$ whose gate complexity is larger by a factor of $\mathrm{poly}(\log q)$, but with an additional factor of $k!$.

3.3.6.1 Algorithm for $k = d/2 + 1/2$ queries

To simplify the computation of unique representatives of values $z \in R_k$, we restrict attention to the "good" case considered in Section 3.3.3. Let

$$R_k^{\mathrm{good}} := \{Z(x,y) : x \in X_k^{\mathrm{good}},\, y \in Y_k^{\mathrm{good}}\}. \tag{3.24}$$

For any $z \in R_k^{\mathrm{good}}$, we show how to efficiently compute representative values $x \in X_k^{\mathrm{good}}$ and $y \in Y_k^{\mathrm{good}}$ with $Z(x,y) = z$, defining a set of representatives T_k^{good}. Then we consider an algorithm as described in Section 3.3.2, but with R_k replaced by R_k^{good} and T_k replaced by T_k^{good}. Clearly the success probability of this algorithm is $|R_k^{\mathrm{good}}|/q^{d+1}$. Our lower bound on $|R_k|$ in Section 3.3.3 was actually a bound on $|R_k^{\mathrm{good}}|$, so this algorithm still succeeds with probability $\frac{1}{k!}(1 + O(1/q))$.

To give a gate-efficient algorithm, it suffices to show how to efficiently compute the

function $Z^{-1}\colon R_k^{\text{good}} \to T_k^{\text{good}}$ (that is, to compute this function using $\text{poly}(\log q)$ gates).

Lemma 3.3.2. *Suppose there is an efficient algorithm to compute $Z^{-1}\colon R_k^{\text{good}} \to T_k^{\text{good}}$. Then the algorithm of Section 3.3.2 can be made gate-efficient (with R_k replaced by R_k^{good} and T_k by T_k^{good}).*

Proof. It is trivial to compute $Z\colon T_k^{\text{good}} \to R_k^{\text{good}}$ efficiently. Given an efficient procedure for computing $Z^{-1}\colon R_k^{\text{good}} \to T_k^{\text{good}}$, this gives us the ability to efficiently compute Z in place (that is, to perform the transformation $|x, y\rangle \mapsto |z\rangle$ as required by the algorithm). To do this, we first compute z in an ancilla register by evaluating Z (which only requires arithmetic over $\mathbb{F}_q$) and then uncompute (x, y) by applying the circuit for Z^{-1} in reverse.

It remains to prepare the initial uniform superposition over T_k^{good}. This can also be done using the ability to compute Z^{-1}. Suppose we create a uniform superposition over all of $z \in \mathbb{F}_q^{d+1}$ and then attempt to compute Z^{-1}. If $z \notin R_k^{\text{good}}$, this is detected, and we can set a flag qubit indicating failure. Thus we can prepare a state of the form

$$\frac{1}{\sqrt{q^{d+1}}}\left(\sum_{(x,y)\in T_k^{\text{good}}} |Z(x,y),0,x,y\rangle + \sum_{z\in\mathbb{F}_q^{d+1}\setminus R_k^{\text{good}}} |z,1,0,0\rangle\right). \tag{3.25}$$

A measurement of the flag qubit gives the outcome 0 with probability $|R_k^{\text{good}}|/q^{d+1}$. Since this is our lower bound on the success probability of the overall algorithm, we do not have to repeat this process too many times before we successfully prepare the initial state (and by sufficiently many repetitions, we can make the error probability arbitrarily small). When the measurement succeeds, we can uncompute the first register to obtain the state $\sum_{(x,y)\in T_k^{\text{good}}} |x, y\rangle / |T_k^{\text{good}}|^{1/2}$ as desired. □

In the remainder of this section, we describe how to efficiently compute $Z^{-1}(z)$ for $z \in R_k^{\text{good}}$. Our approach appeals to "Prony's method" [94] (a precursor to Fourier analysis) and the theory of linear recurrences. We start with the following technical result, where e_j denotes the jth elementary symmetric polynomial in k variables, i.e.,

$$e_j(x_1, \ldots, x_k) = \sum_{1 \le i_1 < i_2 < \cdots < i_j \le k} x_{i_1} x_{i_2} \cdots x_{i_j}. \tag{3.26}$$

Lemma 3.3.3. *We have*

$$x_i^k = -\sum_{j=1}^{k} x_i^{k-j} (-1)^j e_j(x_1, \ldots, x_k) \tag{3.27}$$

for all $i \in \{1, \ldots, k\}$.

Proof. Observe that it suffices to prove the lemma for $i = 1$, since if we interchange the roles of x_1 and x_ℓ in (3.27) with $i = 1$, we obtain (3.27) with $i = \ell$.

We apply induction on k. If $k = 1$ then the claim is trivial. Now suppose the claim holds for a given value of k. We have

$$\begin{aligned} e_j(x_1, \ldots, x_{k+1}) &= e_j(x_1, \ldots, x_k) + x_{k+1} e_{j-1}(x_1, \ldots, x_k) \\ &= e_j(x_1, \ldots, x_k) + x_{k+1} \tfrac{\partial}{\partial x_{k+1}} e_j(x_1, \ldots, x_{k+1}). \end{aligned} \tag{3.28}$$

Therefore

$$\begin{aligned}-\sum_{j=1}^{k+1} x_1^{k+1-j}(-1)^j e_j(x_1,\ldots,x_{k+1}) &= -\sum_{j=1}^{k+1} x_1^{k+1-j}(-1)^j \left[e_j(x_1,\ldots,x_k)\right. \\ &\qquad \left. + x_{k+1}\tfrac{\partial}{\partial x_{k+1}} e_j(x_1,\ldots,x_{k+1})\right] \\ &= x_1^{k+1} - x_{k+1}\tfrac{\partial}{\partial x_{k+1}} x_i^{k+1} \\ &= x_1^{k+1} \end{aligned} \tag{3.29}$$

(where the second equality uses the induction hypothesis). □

Using this fact, we can show that each component of $Z(x,y)$ satisfies a kth-order linear recurrence.

Lemma 3.3.4. *If $z_j = \sum_{i=1}^k y_i x_i^j$ for all nonnegative integers j, then we have (for all nonnegative integers n)*

$$z_{n+k} = -\sum_{j=0}^{k-1} (-1)^{k-j} e_{k-j}(x_1,\ldots,x_k) z_{n+j}. \tag{3.30}$$

Proof. The right-hand side of (3.30) is

$$\begin{aligned}-\sum_{j=0}^{k-1}(-1)^{k-j}e_{k-j}(x_1,\ldots,x_k)\sum_{i=1}^{k} y_i x_i^{n+j} &= -\sum_{i=1}^{k} y_i \sum_{j=0}^{k-1}(-1)^{k-j}e_{k-j}(x_1,\ldots,x_k)x_i^{n+j} \\ &= -\sum_{i=1}^{k} y_i \sum_{j=1}^{k}(-1)^{j}e_{j}(x_1,\ldots,x_k)x_i^{n+k-j} \\ &= \sum_{i=1}^{k} y_i x_i^{n+k} = z_{n+k} \end{aligned} \tag{3.31}$$

as claimed, where the third equality uses Lemma 3.3.3. □

We are now ready to describe the gate-efficient algorithm for polynomial interpolation.

Proof of Theorem 3.1.3(i): $k = d/2 + 1/2$. By Lemma 3.3.2, it suffices to give an efficient algorithm for computing a representative $(x, y) \in Z^{-1}(z)^{\text{good}}$ for any given $z \in R_k^{\text{good}}$.

By Lemma 3.3.4, the coefficients $a_j = -(-1)^{k-j} e_{k-j}(x_1, \ldots, x_k)$ of the linear recurrence (3.30) satisfy

$$H_k \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} = \begin{pmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{2k-1} \end{pmatrix}, \quad \text{where} \quad H_k := \begin{pmatrix} z_0 & z_1 & \cdots & z_{k-1} \\ z_1 & z_2 & \cdots & z_k \\ \vdots & \vdots & \ddots & \vdots \\ z_{k-1} & z_k & \cdots & z_{2(k-1)} \end{pmatrix} \tag{3.32}$$

is a Hankel matrix. Observe that

$$H_k = V_k^\top \begin{pmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & y_k \end{pmatrix} V_k, \quad \text{where} \quad V_k := \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{k-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{k-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^{k-1} \end{pmatrix}. \tag{3.33}$$

For $x \in X_k^{\text{good}}$, the Vandermonde matrix V_k (and its transpose) are invertible, and for

$y \in Y_k^{\text{good}}$, the diagonal matrix is invertible. Then H_k is invertible, and we have

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} = H_k^{-1} \begin{pmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{2k-1} \end{pmatrix}. \tag{3.34}$$

We claim that for any $(x, y) \in Z^{-1}(z)^{\text{good}}$, the values $x_1, \ldots, x_k$ must be roots of the characteristic polynomial

$$\chi(x) := x^k - \sum_{j=0}^{k-1} a_j x^j. \tag{3.35}$$

To see this, observe that

$$\begin{pmatrix} z_k \\ z_{k+1} \\ \vdots \\ z_{2k-1} \end{pmatrix} - H_k \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} = V_k^\top \begin{pmatrix} \chi(x_1) \\ \chi(x_2) \\ \vdots \\ \chi(x_k) \end{pmatrix}. \tag{3.36}$$

This must be the zero vector, and since the Vandermonde matrix is invertible, we see that $\chi(x_i) = 0$ for all $i \in \{1, \ldots, k\}$.[4]

[4]Since a polynomial of degree k can have at most k roots, this shows that the values $x_1, \ldots, x_k$ are unique up to permutation, giving $|Z^{-1}(z)^{\text{good}}| = k!$ as shown in Lemma 3.3.1.

Finally, observe that the values $y_1, \ldots, y_k$ satisfy

$$V_k^\top \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_{k-1} \end{pmatrix}. \tag{3.37}$$

Since the Vandermonde matrix is invertible, we see that y is uniquely determined by z and x.

To compute a unique representative of a given $z \in R_k^{\text{good}}$, we use equation (3.34) to efficiently compute the coefficients $a_0, \ldots, a_{k-1}$ of the characteristic polynomial $\chi(x)$. We can then determine $x \in X_k^{\text{good}}$ by finding the roots of this polynomial, which can be done in time $\text{poly}(k, \log q)$ using a randomized algorithm [95, Chapter 14]. Finally, we can determine $y \in Y_k^{\text{good}}$ by solving a linear system of equations, namely (3.37).

This procedure does not uniquely specify (x, y) because any permutation of the indices (acting identically on x and y) gives an equivalent solution. To choose a unique $(x, y) \in T_k^{\text{good}}$, we simply require that the entries of x occur in lexicographic order with respect to some fixed representation of $\mathbb{F}_q$. □

3.3.6.2 Algorithm for $k = d/2 + 1$ queries

We now present a similar algorithm for the case $k = d/2 + 1$ that also has gate complexity $\text{poly}(\log q)$, although it has more overhead as a function of d.

To apply the approach of Section 3.3.6.1, we again focus on solutions of $Z(x, y) = z$

with $(x, y) \in X^{\text{good}} \times Y^{\text{good}}$. However, recall that our lower bound on the success probability for $k = d/2+1$ in Section 3.3.4 used all solutions $(x, y) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$. Thus we begin by showing that the success probability of the algorithm remains close to 1 even when restricted to good solutions.

Lemma 3.3.5. *If $k = d/2 + 1$, then $|R_k^{\text{good}}|/q^{d+1} = 1 - O(1/q)$.*

Proof. We repeat the second moment argument of Section 3.3.4, but now restricted to good solutions. Under the uniform distribution on $z \in \mathbb{F}_q^{d+1}$, we have

$$\mu^{\text{good}} := \frac{1}{q^{d+1}} \sum_{z \in \mathbb{F}_q^{d+1}} |Z^{-1}(z)^{\text{good}}| = q^{2k-(d+1)}(1 - O(1/q)) \tag{3.38}$$

by (3.12). Similarly to the previous second moment calculation, we have

$$\begin{aligned} \sum_{z \in \mathbb{F}_q^{d+1}} |Z^{-1}(z)^{\text{good}}|^2 &= \sum_{u,x \in X_k^{\text{good}}} \sum_{v,y \in Y_k^{\text{good}}} \delta[Z(u, v) = Z(x, y)] \\ &= q^{d+1}(\mu^{\text{good}})^2 + \frac{1}{q^{d+1}} \sum_{u,x \in X_k^{\text{good}}} \sum_{v,y \in Y_k^{\text{good}}} \sum_{\lambda \in \mathbb{F}_q^{d+1}} e(\lambda \cdot (Z(u, v) - Z(x, y))). \end{aligned} \tag{3.39}$$

Thus we have

$$
\begin{aligned}
(\sigma^{\text{good}})^2 &:= \frac{1}{q^{d+1}} \sum_{z \in \mathbb{F}_q^{d+1}} |Z^{-1}(z)^{\text{good}}|^2 - (\mu^{\text{good}})^2 \\
&= \frac{1}{q^{2(d+1)}} \sum_{\lambda \in \mathbb{F}_q^{d+1}} \sum_{u,x \in X_k^{\text{good}}} \sum_{v,y \in Y_k^{\text{good}}} \prod_{i=1}^{k} e(v_i \textstyle\sum_{j=0}^{d} \lambda_j u_i^j)\, e(-y_i \sum_{j=0}^{d} \lambda_j x_i^j) \\
&= \frac{1}{q^{2(d+1)}} \sum_{\lambda \in \mathbb{F}_q^{d+1}} \sum_{u,x \in X_k^{\text{good}}} \prod_{i=1}^{k} (q\,\delta[\textstyle\sum_{j=0}^{d} \lambda_j u_i^j = 0] - 1)(q\,\delta[\sum_{j=0}^{d} \lambda_j x_i^j = 0] - 1) \\
&\le \frac{1}{q^{2(d+1)}} \sum_{\lambda \in \mathbb{F}_q^{d+1}} \sum_{u,x \in \mathbb{F}_q^{k}} \prod_{i=1}^{k} (q\,\delta[\textstyle\sum_{j=0}^{d} \lambda_j u_i^j = 0] + 1)(q\,\delta[\sum_{j=0}^{d} \lambda_j x_i^j = 0] + 1) \\
&\le \frac{(q(d+1))^{2k}}{q^{d+1}}
\end{aligned}
\tag{3.40}
$$

(which is identical to the previous bound for σ^2 except that d is replaced by $d+1$). Therefore, by the Chebyshev inequality, we have

$$
\Pr[Z^{-1}(z)^{\text{good}}] \le \frac{(\sigma^{\text{good}})^2}{(\mu^{\text{good}})^2} \le (d+1)^{2k} q^{d+1-2k}. \tag{3.41}
$$

With $k = d/2 + 1$, we find

$$
\frac{|R_k^{\text{good}}|}{q^{d+1}} = 1 - \Pr[Z^{-1}(z)^{\text{good}} = 0] \ge 1 - \frac{(d+1)^{2k}}{q} = 1 - O(1/q) \tag{3.42}
$$

as claimed. □

Now consider the problem of computing a value $(x, y) \in X^{\text{good}} \times Y^{\text{good}}$ such that $Z(x, y) = z$ for some given $z \in R_k^{\text{good}}$. We can approach this task using the strategy outlined in Section 3.3.6.1. With $k = d/2 + 1$, we have $2k - 1 = d + 1$, so the last entry in the vector

on the right-hand side of (3.34) is not specified. Nevertheless, for any fixed $(x, y) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$, the value $z_{d+1} = Z(x, y)_{d+1}$ is well-defined by extending (3.6) to $j = d + 1$, so we can find $(x, y) \in X^{\text{good}} \times Y^{\text{good}}$ by searching for some value of $z_{d+1} \in \mathbb{F}_q$ for which the algorithm of Section 3.3.6.1 succeeds at finding k distinct roots $x_1, \ldots, x_k \in \mathbb{F}_q$ of the characteristic polynomial (3.35).

We claim that choosing a random $z_{d+1} \in \mathbb{F}_q$ gives a solution with probability nearly $1/k!$.

Lemma 3.3.6. *Suppose $z = (z_0, \ldots, z_d)$ is chosen uniformly at random from $\mathbb{F}_q^{d+1}$. Then with probability $1 - o(1)$ (over the choice of z), choosing z_{d+1} uniformly at random from $\mathbb{F}_q$ and solving for $(x, y) \in Z^{-1}(z)^{\text{good}}$ as in the proof of Theorem 3.1.3(ii) gives a solution with probability $(1 - o(1))/k!$ (over the choice of z_{d+1}).*

Proof. For any $z \in R_k^{\text{good}}$, each value of $z_{d+1} \in \mathbb{F}_q$ gives a unique set of roots of the characteristic polynomial (3.35), and hence corresponds to either 0 or $k!$ solutions $(x, y) \in Z^{-1}(z)^{\text{good}}$. By a similar second moment argument as in (3.22), but using the mean (3.38) and variance (3.40) of $Z^{-1}(z)^{\text{good}}$, we have $|Z^{-1}(z)^{\text{good}}| = q(1 - o(1))$ with probability $1 - o(1)$ over the uniform choice of $z \in \mathbb{F}_q^{d+1}$. Thus the number of values of z_{d+1} that lead to a valid solution $(x, y) \in Z^{-1}(z)^{\text{good}}$ is at least $q(1 - o(1))/k!$ with probability $1 - o(1)$ over the choice of z. Since there are q possible values of z_{d+1}, choosing z_{d+1} at random leads to a valid representative $(x, y) \in Z^{-1}(z)^{\text{good}}$ with probability $(1 - o(1))/k!$, again with probability $1 - o(1)$ over the uniform choice of z. □

Lemma 3.3.6 gives a method for computing a representative $(x, y) \in X^{\text{good}} \times Y^{\text{good}}$ such that $Z(x, y) = z$: simply choose $z_{d+1} \in \mathbb{F}_q$ at random until we find a solution. Repeating

this process $O(k!)$ times suffices to find a solution with constant probability (for almost all z). However, since this approach constructs a random $(x, y) \in Z^{-1}(z)^{\text{good}}$ rather than a unique representative, it does not define a set T_k^{good}, and it cannot be directly applied to our quantum algorithm as described so far. Instead, we construct an equivalent algorithm that represents the sets $Z^{-1}(z)^{\text{good}}$ using quantum superpositions.

Lemma 3.3.7. *Suppose there is an efficient algorithm to generate the quantum state*

$$|Z^{-1}(z)^{\text{good}}\rangle := \frac{1}{\sqrt{|Z^{-1}(z)^{\text{good}}|}} \sum_{(x,y)\in Z^{-1}(z)^{\text{good}}} |x, y\rangle \tag{3.43}$$

for any given $z \in R_k^{\text{good}}$. Then there is a gate-efficient k-query quantum algorithm for the polynomial interpolation problem, succeeding with probability $|R_k^{\text{good}}|/q^{d+1}$.

Proof. We essentially replace $(x, y) \in T_k^{\text{good}}$ by $|Z^{-1}(Z(x, y))^{\text{good}}\rangle$ throughout the algorithm. More concretely, we proceed as follows.

Observe that the ability to perform the given state generation map $|z\rangle \mapsto |z\rangle|Z^{-1}(z)^{\text{good}}\rangle$ implies the ability to perform the in-place transformation

$$|z\rangle \mapsto |Z^{-1}(z)^{\text{good}}\rangle. \tag{3.44}$$

After applying the state generation map, we simply uncompute the map Z to erase the register $|z\rangle$.

The algorithm begins by creating a uniform superposition over all of $z \in \mathbb{F}_q^{d+1}$ and applying the map (3.44). As in the proof of Lemma 3.3.2, we can detect whether $z \notin R_k^{\text{good}}$, and we can postselect on the outcomes for which $z \in R_k^{\text{good}}$ with reasonable overhead, giving

the state $\sum_{z \in R_k^{\text{good}}} |Z^{-1}(z)^{\text{good}}\rangle / |R_k^{\text{good}}|^{1/2}$. Then perform k phase queries and apply the inverse of the transformation (3.44), giving the state

$$\frac{1}{\sqrt{|R_k^{\text{good}}|}} \sum_{z \in R_k^{\text{good}}} e(c \cdot z) |Z^{-1}(z)^{\text{good}}\rangle \mapsto \frac{1}{\sqrt{|R_k^{\text{good}}|}} \sum_{z \in R_k^{\text{good}}} e(c \cdot z) |z\rangle. \tag{3.45}$$

As discussed in Section 3.3.2, measuring this state gives c with probability $|R_k^{\text{good}}|/q^{d+1}$. □

Finally, we show how to prepare $|Z^{-1}(z)^{\text{good}}\rangle$ and thereby give a gate-efficient quantum algorithm for polynomial interpolation with $k = d/2 + 1$ queries.

Proof of Theorem 3.1.3(ii): $k = d/2 + 1$. We use $|Z^{-1}(z)^{\text{good}}\rangle$ as a quantum representative of the set of solutions $Z^{-1}(z)^{\text{good}}$ as described in Lemma 3.3.7. We claim that we can efficiently perform the transformation $|z\rangle \mapsto |Z^{-1}(z)^{\text{good}}\rangle$ for a fraction $1 - o(1)$ of those $z \in R_k^{\text{good}}$, which in turn are a fraction $1 - o(1)$ of all $z \in \mathbb{F}_q^{d+1}$ (by Lemma 3.3.5), giving the claimed success probability.

To prepare $|Z^{-1}(z)^{\text{good}}\rangle$, we first prepare a uniform superposition over $z_{d+1} \in \mathbb{F}_q$ and use the procedure of Section 3.3.6.1 to compute the corresponding (x, y), if it exists. Lemma 3.3.6 shows that a fraction $(1 - o(1))/k!$ of the values of z_{d+1} correspond to a valid (x, y), so this process can be boosted to prepare a state close to $|Z^{-1}(z)^{\text{good}}\rangle$ with overhead $O(k!)$ (or with amplitude amplification, $O(\sqrt{k!})$), which in particular is independent of q. We can easily uncompute z_{d+1} given (x, y), giving the desired transformation. □

3.4 Multivariate polynomial interpolation

3.4.1 The query model

Using the standard concept of phase kickback, we encode the results of queries in the phase by performing standard queries in the Fourier basis. We briefly explain these queries for the three types of fields we consider.

3.4.1.1 Finite field $\mathbb{F}_q$

As in the univariate case, our algorithm is nonadaptive, making all queries in parallel for a carefully chosen superposition of inputs. With k parallel queries, we generate a phase $\sum_{i=1}^{k} y_i f(x_i) = \sum_{i=1}^{k} \sum_{j \in \mathbb{J}} y_i x_i^j c_j$ for the input $(x, y) \in \mathbb{F}_q^k \times \mathbb{F}_q^k$. For convenience, we define $Z \colon \mathbb{F}_q^{nk} \times \mathbb{F}_q^k \to \mathbb{F}_q^J$ by $Z(x, y)_j = \sum_{i=1}^{k} y_i {x_i}^j$ for $j \in \mathbb{J}$, so that $\sum_{i=1}^{k} y_i f(x_i) = Z(x, y) \cdot c$.

3.4.1.2 Real numbers $\mathbb{R}$

As in the finite field case, we construct a phase query by making a standard query in the Fourier basis, giving

$$\int_{\mathbb{R}^2} \mathrm{d}x \, \mathrm{d}y \, \psi(x, y)|x, y\rangle \mapsto \int_{\mathbb{R}^2} \mathrm{d}x \, \mathrm{d}y \, e(yf(x))\psi(x, y)|x, y\rangle. \tag{3.46}$$

An algorithm making k parallel queries generates a phase $Z(x, y) \cdot c$, where we similarly define $Z \colon \mathbb{R}^{nk} \times \mathbb{R}^k \to \mathbb{R}^J$ by $Z(x, y)_j = \sum_{i=1}^{k} y_i {x_i}^j$ for $j \in \mathbb{J}$.

3.4.1.3 Complex numbers $\mathbb{C}$

An algorithm making k parallel queries generates a phase $\sum_{i=1}^{k} \overline{y_i} f(x_i) = \sum_{i=1}^{k} \sum_{j \in \mathbb{J}} \overline{y_i} x_i^j c_j$. We define $Z \colon \mathbb{C}^{nk} \times \mathbb{C}^k \to \mathbb{C}^J$ satisfying $Z(x,y)_j = \sum_{i=1}^{k} y_i \overline{x_i}^j$ for $j \in \mathbb{J}$, so that $\sum_{i=1}^{k} \overline{y_i} f(x_i) = Z(x,y) \cdot c$.

3.4.2 The algorithm

Our algorithm follows the same idea as in [30]: we perform k phase queries in parallel for a carefully-chosen superposition of inputs, such that the output states corresponding to distinct polynomials are as distinguishable as possible. For a k-query quantum algorithm, we consider the mapping $Z \colon \mathbb{K}^{nk} \times \mathbb{K}^k \to \mathbb{K}^J$ defined in Section 3.4.1 for $\mathbb{K} = \mathbb{F}_q$, $\mathbb{R}$, and $\mathbb{C}$. Reference [30] gave an optimal algorithm for $n = 1$ using a uniform superposition over a unique set of preimages of the range $R_k := Z(\mathbb{K}^{nk}, \mathbb{K}^k)$ of Z, so we apply the same strategy here. For each $z \in R_k$, we choose a unique $(x,y) \in \mathbb{K}^{nk} \times \mathbb{K}^k$ such that $Z(x,y) = z$. Let T_k be some set of unique representatives, so that $Z \colon T_k \to R_k$ is a bijection.

3.4.2.1 $\mathbb{K} = \mathbb{F}_q$

The algorithm generates a uniform superposition over T_k, performs k phase queries, and computes Z in place, giving

$$\frac{1}{\sqrt{|T_k|}} \sum_{(x,y) \in T_k} |x,y\rangle \mapsto \frac{1}{\sqrt{|T_k|}} \sum_{(x,y) \in T_k} e(Z(x,y) \cdot c) |x,y\rangle \mapsto \frac{1}{\sqrt{|R_k|}} \sum_{z \in R_k} e(z \cdot c) |z\rangle. \tag{3.47}$$

We then measure in the basis of Fourier states $|\widetilde{c}\rangle := \frac{1}{\sqrt{q^J}} \sum_{z \in \mathbb{F}_q^J} e(z \cdot c) |z\rangle$. A simple

calculation shows that the result of this measurement is the correct vector of coefficients with probability $|R_k|/q^J$.

3.4.2.2 $\mathbb{K} = \mathbb{R}$

We consider a bounded subset $S \subseteq \mathbb{R}^J$ and a set T'_k of unique preimages of each element in $R_k \cap S$ such that $Z(T'_k) = R_k \cap S$ and $Z\colon T'_k \to R_k \cap S$ is bijective. The algorithm on input $|\psi\rangle$ with support $\mathrm{supp}(\psi) \subseteq R_k \cap S$ gives

$$\begin{aligned} |\psi\rangle = \int_{R_k \cap S} \mathrm{d}^J z\, \psi(z)|z\rangle &\mapsto \int_{R_k \cap S} \mathrm{d}^J z\, \psi(z)|z\rangle|Z^{-1}(z)\rangle \\ &\mapsto \int_{R_k \cap S} \mathrm{d}^J z\, \psi(z) e(z \cdot c)|z\rangle|Z^{-1}(z)\rangle \\ &\mapsto \int_{R_k \cap S} \mathrm{d}^J z\, \psi(z) e(z \cdot c)|z\rangle =: |\psi_c\rangle. \end{aligned} \tag{3.48}$$

The choice of S constrains the set of inputs that can be perfectly distinguished by this procedure, as captured by the following lemma.

Lemma 3.4.1 (Orthogonality)**.** *For positive integer n, let $m(A) := \int_A \mathrm{d}^n z$ be the measure of the set $A \subseteq \mathbb{R}^n$. Let S be a bounded subset of $\mathbb{R}^n$ with nonzero measure. Let $|\widetilde{c}\rangle = \frac{1}{\sqrt{m(S)}} \int_S \mathrm{d}^n z\, e(c \cdot z)|z\rangle$ and let U be the maximal subset of $\mathbb{R}^n$ such that for any $c, c' \in U$ with $c \neq c'$,*

$$\langle \widetilde{c'}|\widetilde{c}\rangle = \frac{1}{m(S)} \int_S \mathrm{d}^n z\, e((c - c') \cdot z) = 0. \tag{3.49}$$

Then there is a lattice Λ such that $U \in \mathbb{R}^n/\Lambda$.

Proof. By definition, $c - c'$ must be a zero of the Fourier transform $\mathbb{F}(\mathbb{I}_S)$ of the indicator function $\mathbb{I}_S(z)$. We denote $\Lambda := \{c : \mathbb{F}(\mathbb{I}_S)(c) = 0\} \cup \{0\}$ and let $c_0 \in U$. Clearly $U \subseteq c_0 + \Lambda$ as Λ contains all zeros. Since $\langle \widetilde{c + c_0} | \widetilde{c_0} \rangle = 0$ for all $c \in \Lambda \backslash \{0\}$, we have $c_0 + \Lambda \subseteq U$ and $U = c_0 + \Lambda$. If $c \in \Lambda \backslash \{0\}$, then $\langle \widetilde{c_0 + c} | \widetilde{c_0} \rangle = \langle \widetilde{c_0} | \widetilde{c_0 - c} \rangle = 0$ implies that $-c \in \Lambda$. If $c, c' \in \Lambda \backslash \{0\}$, then $\langle \widetilde{c + c_0} | \widetilde{-c' + c_0} \rangle = \langle \widetilde{c + c' + c_0} | \widetilde{c_0} \rangle = 0$ implies $c + c' \in \Lambda \backslash \{0\}$. Therefore Λ is an additive subgroup of $\mathbb{R}^n$.

Now we prove that Λ is a lattice. For $\epsilon > 0$, $\delta \in B(\epsilon)$, and $c \in \Lambda$,

$$|\langle \widetilde{c + \delta} | \widetilde{c} \rangle|^2 = \left| \int_S \mathrm{d}^n z \, e(\delta \cdot z) \right|^2 \geq \left| \int_S \mathrm{d}^n z \, \cos(2\pi \delta \cdot z) \right|^2 > 0 \tag{3.50}$$

if $S \subseteq B(r)$ for $r < \frac{1}{4\epsilon}$. Thus $B(\epsilon)$ contains exactly one element in Λ and hence Λ is discrete. □

Roughly speaking, Lemma 3.4.1 is a consequence of the uncertainty principle: restricting the support to a finite window limits the precision with which we can determine the Fourier transform. In the proof, note that a larger window offers better resolution of the coefficients.

We have shown that the set Λ of perfectly distinguishable coefficients forms a lattice. We also require the set $\{|\widetilde{c}\rangle : c \in \Lambda\}$ to be a complete basis. Since $\langle z | \widetilde{c} \rangle = \frac{1}{\sqrt{m(S)}} e(z \cdot c)$, completeness implies that $|z\rangle$ is of the form $\sum_{c \in \Lambda} e(-z \cdot c) |\widetilde{c}\rangle$ up to a normalization constant. More formally, we have the following lemma.

Lemma 3.4.2 (Completeness)**.** *For positive integer n, let $m(A) := \int_A \mathrm{d}^n z$ be the measure of the set $A \subseteq \mathbb{R}^n$. Let Λ be a discrete additive subgroup of $\mathbb{R}^n$. Let S be a bounded set*

with nonzero measure and $|\widetilde{c}\rangle = \frac{1}{\sqrt{m(S)}} \int_S \mathrm{d}^n z\, e(z\cdot c)|z\rangle$. *Then* $\{|\widetilde{c}\rangle : c \in \Lambda\}$ *forms a complete basis over support* S *if and only if* S *is a fundamental domain of the dual lattice of* Λ.

Proof. Let $\widetilde{\Lambda}$ be the dual lattice of Λ. We observe that (ignoring the normalization constant)

$$\begin{aligned}\sum_{c\in\Lambda} e(-z\cdot c)|\widetilde{c}\rangle &= \int_S \mathrm{d}^J z' \sum_{c\in\Lambda} e((z'-z)\cdot c)|z'\rangle = \int_S \mathrm{d}^J z' \sum_{z_0\in\widetilde{\Lambda}} \delta(z'-z-z_0)|z'\rangle \\ &= \sum_{z_0\in\widetilde{\Lambda}} \mathbb{I}_S(z+z_0)|z+z_0\rangle = |(z+\widetilde{\Lambda})\cap S\rangle. \end{aligned} \tag{3.51}$$

In (3.51), $\sum_{c\in\Lambda} e(z\cdot c) = \sum_{z_0\in\widetilde{\Lambda}} \delta(z-z_0)$ up to a constant factor [96, Section 7.2]. The set $(z+\widetilde{\Lambda})\cap S$ cannot be empty, so a fundamental domain of $\widetilde{\Lambda}$ is a subset of S. For $z, z' \in \mathbb{R}^n$, $\langle (z+\widetilde{\Lambda})\cap S | (z'+\widetilde{\Lambda})\cap S\rangle = 0$ if $z' \notin z+\widetilde{\Lambda}$, which implies that S is a subset of a fundamental domain of $\widetilde{\Lambda}$. □

Lemma 3.4.2 further restricts the bounded set S has to be a fundamental region of $\widetilde{\Lambda}$. Without loss of generality, one may choose S to be a fundamental domain of a lattice centered at zero. In the last step, the algorithm applies the unitary operator

$$\frac{1}{\sqrt{m(S)}} \sum_{c'\in\Lambda} \int_S \mathrm{d}^J z\, e(-z\cdot c')|c'\rangle\langle z| \tag{3.52}$$

to the state $|\psi_c\rangle$ in (3.48). The algorithm outputs $c' \in \Lambda$ with probability

$$\frac{1}{m(R_k\cap S)m(S)} \left| \int_{R_k\cap S} \mathrm{d}^J z\, \psi(z) e(z\cdot(c-c')) \right|^2 \le \frac{m(R_k\cap S)}{m(S)}, \tag{3.53}$$

where the upper bound follows from the Cauchy-Schwarz inequality. The maximum is reached if $\psi(z) = \frac{1}{\sqrt{m(R_k\cap S)}} \mathbb{I}_{R_k\cap S}(z)$ and c happens to be a lattice point. If $c \notin \Lambda$, the

algorithm returns the closest lattice point with high probability.

To achieve arbitrarily high precision, one may want to take $S \to \mathbb{R}^J$. In this limit, the basis of coefficients is normalized to the Dirac delta function, i.e., $\langle \widetilde{c}'|c\rangle = \delta^{(J)}(c-c')$. In this case, $\Lambda \to \mathbb{R}^J$ and the unitary operator in (3.52) becomes the J-dimensional QFT over the real numbers. However, for the interpolation problem, the success probability $\frac{m(R_k \cap S)}{m(S)}$ is not well-defined in the limit $S \to \mathbb{R}^J$ since different shapes for S can give different probabilities. Thus it is necessary to choose a bounded region, and we leave the optimal choice as an open question.

Though the size of the fundamental domain S affects the resolution of the coefficients, it does not affect the maximal success probability $\frac{m(R_k \cap S)}{m(S)}$. This can be seen by scale invariance: for every $z \in R_k$, there is a preimage (x, y) such that $Z(x, y) = z$. Then $\lambda z \in R_k$ since $Z(x, \lambda y) = \lambda z$ for any $\lambda \in \mathbb{R}$. In terms of the bijection $\ell \colon z \mapsto \lambda z$ for $\lambda \in \mathbb{R}^\times$, we have $\ell(R_k) = R_k$ and $\ell(R_k \cap S) = R_k \cap \ell(S)$. Then $m(R_k \cap \ell(S)) = m(\ell(R_k \cap S)) = \lambda^J m(R_k \cap S)$ and hence $\frac{m(R_k \cap \ell(S))}{m(\ell(S))} = \frac{m(R_k \cap S)}{m(S)}$. Thus we can make the precision arbitrarily high by taking S arbitrarily large, and we call $\frac{m(R_k \cap S)}{m(S)}$ the success probability of the algorithm.

3.4.2.3 $\mathbb{K} = \mathbb{C}$

We consider a bounded set $S \subseteq \mathbb{C}^J$ and a set T'_k of unique preimages of each element in $R_k \cap S$ such that $Z(T'_k) = R_k$ and $Z \colon T'_k \to R_k \cap S$ is bijective. The algorithm on input

$|\psi\rangle$ with support $\mathrm{supp}(\psi) \subseteq R_k \cap S$ gives

$$\begin{aligned}|\psi\rangle = \int_{\phi(R_k\cap S)} \mathrm{d}^{2J}\mathbf{z}\,\psi(\mathbf{z})|\mathbf{z}\rangle &\mapsto \int_{\phi(R_k\cap S)} \mathrm{d}^{2J}\mathbf{z}\,\psi(\mathbf{z})|\mathbf{z}\rangle|\phi(Z^{-1}(z))\rangle \\ &\mapsto \int_{\phi(R_k\cap S)} \mathrm{d}^{2J}\mathbf{z}\,\psi(\mathbf{z})e(z\cdot c)|\mathbf{z}\rangle|\phi(Z^{-1}(z))\rangle \\ &\mapsto \int_{\phi(R_k\cap S)} \mathrm{d}^{2J}\mathbf{z}\,\psi(\mathbf{z})e(z\cdot c)|\mathbf{z}\rangle =: |\psi_{\mathbf{c}}\rangle. \end{aligned} \tag{3.54}$$

By Lemma 3.4.1 and Lemma 3.4.2, the set S must be a fundamental domain in $\mathbb{C}^J$. Let $\{|\widetilde{\mathbf{c}}\rangle : c \in \Lambda\}$ be the measurement basis. In the last step of the algorithm, we apply the unitary operator

$$\frac{1}{\sqrt{m(S)}} \sum_{\mathbf{c}'\in\phi(\Lambda)} \int_{\phi(S)} \mathrm{d}^{2J}\mathbf{z}\,e(-z\cdot c')|\mathbf{c}'\rangle\langle\mathbf{z}| \tag{3.55}$$

to the state $|\psi_{\mathbf{c}}\rangle$ in (3.54). The algorithm outputs $c' \in \Lambda$ with probability

$$\frac{1}{m(R_k\cap S)m(S)}\left|\int_{\phi(R_k\cap S)} \mathrm{d}^{2J}\mathbf{z}\,\psi(\mathbf{z})e(z\cdot(c-c'))\right|^2. \tag{3.56}$$

Again, since $|\psi\rangle$ is normalized, (3.56) cannot be arbitrarily large. By the Cauchy-Schwarz inequality, (3.56) is upper bounded by $\frac{m(R_k\cap S)}{m(S)}$; this maximal success probability is obtained if $\psi(\mathbf{z}) = \frac{1}{\sqrt{m(R_k\cap S)}}\mathbb{I}_{\phi(R_k\cap S)}(\mathbf{z})$ and c happens to be a lattice point. If $c \notin \Lambda$, the algorithm returns the closest lattice point with high probability.

By the same argument as in Section 3.4.2.2, we can show scale invariance holds for complex numbers: for $\ell\colon z \mapsto \lambda z$ where $z \in \mathbb{C}^J$ and $\lambda \in \mathbb{R}^\times$, $\frac{m(R_k\cap S)}{m(S)} = \frac{m(R_k\cap \ell(S))}{m(\ell(S))}$. Thus we can make the precision of the algorithm arbitrarily high by taking S arbitrarily large without

affecting the maximal success probability.

3.4.3 Performance

We have shown in Section 3.4.2.1 that the optimal success probability is at most $|R_k|/q^J$ for $\mathbb{K} = \mathbb{F}_q$. For real and complex numbers, we consider a bounded support S in which the algorithm is performed. The success probability of the algorithm with this choice is at most $\frac{m(R_k \cap S)}{m(S)}$, as shown in (3.53) and (3.56). To establish the query complexity, first we show that if $\dim R_k = J$, the algorithm outputs the coefficients with bounded error.

Lemma 3.4.3. *For positive integers n, k, d, let $J := \binom{n+d}{d}$ and let $m(A) := \int_A \mathrm{d}^J z$ be the volume of $A \subseteq R^J$. Let $Z \colon \mathbb{K}^{nk} \times \mathbb{K}^k \to \mathbb{K}^J$, $Z(x, y) = \sum_{i=1}^k y_i x_i^j$ for an infinite field $\mathbb{K}$. Let $R_k = Z(\mathbb{K}^{nk}, \mathbb{K}^k)$ be the range of Z. If $\dim R_k = J$, then $\frac{m(R_k \cap S)}{m(S)} > 0$ if S is a fundamental domain centered at 0.*

Proof. R_k is a constructible set for $\mathbb{K} = \mathbb{C}$ and it is a semialgebraic set for $\mathbb{K} = \mathbb{R}$. By [90] and [91], R_k has non-empty interior if $\dim(R_k) = J$ for both cases.

S is a fundamental domain centered at 0 with finite measure, so we only need to prove that $m(R_k \cap S)$ is of positive measure, or equivalently, that the interior of R_k and the interior of S have non-empty intersection.

If this is not the case, then any interior point of S cannot be in the interior of R_k. By scale invariance of R_k, any point in $\mathbb{K}^n$ except 0 cannot be in the interior of R_k, which contradicts the fact that R_k has non-empty interior given $\dim(R_k) = J$. □

Lemma 3.4.3 shows that for infinite fields, although we perform the algorithm over a bounded support, the query complexity can be understood by considering the dimension of

the entire set R_k. Moreover, by invoking recent work on typical ranks, we can establish the minimum number of queries to determine the coefficients almost surely.

Now let $v_d(x_1, x_2, \ldots, x_n)$ be the $\binom{n+d}{d}$-dimensional vector that contains all monomials with variables $x_1, \ldots, x_n$ of degree no more than d as its entries. Let

$$X_{n,d} := \{v_d(x_1, x_2, \ldots, x_n) : x_1, x_2, \ldots, x_n \in \mathbb{K}\} \tag{3.57}$$

where $\mathbb{K}$ is a given ground field. For example, we have

$$X_{3,2} = \{(x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3, x_1, x_2, x_3, 1)^\top : x_1, x_2, x_3 \in \mathbb{K}\}. \tag{3.58}$$

Our question is to determine the smallest number k such that a generic vector in $\mathbb{K}^{\binom{n+d}{d}}$ can be written as a linear combination of no more than k elements from $X_{n,d}$. More precisely, we have $R_k = \{\sum_{i=1}^k c_i v_i : c_i \in \mathbb{K}, v_i \in X_{n,d}\}$, and we ask what is the smallest number k such that R_k has full measure in $\mathbb{K}^{\binom{n+d}{d}}$.

Our approach requires basic knowledge of algebraic geometry—specifically, the concepts of Zariski topology, Veronese variety, and secant variety. Formal definitions can be found in Section 3.2. For the reader's convenience, we also explain these concepts briefly when we first use them.

Now we make two simple observations.

1. In general, $v_d(x_1, x_2, \ldots, x_n)$ can be treated as an $\binom{n+d}{d}$-dimensional vector that contains all monomials with variables $x_1, \ldots, x_n, x_{n+1}$ of degree d as its entries, by simply taking the map $(x_1, x_2, \ldots, x_n) \mapsto (\frac{x_1}{x_{n+1}}, \ldots, \frac{x_n}{x_{n+1}})$ and multiplying by x_{n+1}^d. For ex-

ample, applying this mapping to $X_{3,2}$ gives

$$X'_{3,2} = \{(x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3, x_1x_4, x_2x_4, x_3x_4, x_4^2)^\top : x_1, x_2, x_3, x_4 \in \mathbb{K}\}.$$

The new set $X'_{n,d}$ is slightly bigger than $X_{n,d}$ since it also contains those points corresponding to $x_{n+1} = 0$, but this will not affect our calculation since the difference is just a measure zero set in $X'_{n,d}$.

2. The set $X'_{n,d}$ is the Veronese variety. One may also notice that this set is isomorphic to $\left((x_1, x_2, \ldots, x_{n+1})^\top\right)^{\otimes d}$ in the symmetric subspace.

These observations imply that instead of studying R_k, we can study the new set

$$R'_k = \left\{ \sum_{i=1}^{k} c_i v'_i : c_i \in \mathbb{K}, v'_i \in X'_{n,d} \right\}. \tag{3.59}$$

In general, we have a sequence of inclusions:

$$X'_{n,d} = R'_1 \subseteq R'_2 \subseteq \cdots \subseteq R'_k \subseteq \cdots \subseteq \mathbb{K}^{\binom{n+d}{d}}. \tag{3.60}$$

By taking the Zariski closure, we also have

$$\overline{X'_{n,d}} = \overline{R'_1} \subseteq \overline{R'_2} \subseteq \cdots \subseteq \overline{R'_k} \subseteq \cdots \subseteq \mathbb{K}^{\binom{n+d}{d}} \tag{3.61}$$

where $\overline{R'_k}$ is the kth secant variety of the Veronese variety $X'_{n,d}$.

Palatini showed the following [97, 98]:

Lemma 3.4.4. *If* $\dim \overline{R'_{k+1}} \leq \dim \overline{R'_k} + 1$, *then* $\overline{R'_{k+1}}$ *is linear.*

In particular, this shows that if $\dim \overline{R'_k} = \binom{n+d}{d}$, then $\overline{R'_k} = \mathbb{K}^{\binom{n+d}{d}}$.

For an infinite field $\mathbb{K}$, define $k_{\mathbb{K}}$ to be the smallest integer such that $\frac{m(R_{k_{\mathbb{K}}} \cap S)}{m(S)} = 1$. Thus $k_{\mathbb{K}}$ represents the minimal number of queries such that our algorithm succeeds with probability 1. For the finite field case $\mathbb{K} = \mathbb{F}_q$, we only require that $\frac{m(R_{k_{\mathbb{F}_q}} \cap S)}{m(S)}$ goes to 1 when q tends to infinity.

3.4.3.1 $\mathbb{K} = \mathbb{C}$

A theorem due to Alexander and Hirschowitz [99] implies an upper bound on the query complexity of polynomial interpolation over $\mathbb{C}$.

Theorem 3.4.1 (Alexander-Hirschowitz Theorem, [99])**.** *The dimension of* $\overline{R'_k}$ *satisfies*

$$\dim \overline{R'_k} = \begin{cases} k(n+1) - \frac{k(k-1)}{2} & d = 2, 2 \leq k \leq n; \\ \binom{n+d}{d} - 1 & (d,n,k) = (3,4,7), (4,2,5), (4,3,9), (4,4,14); \\ \min\{k(n+1), \binom{n+d}{d}\} & \textit{otherwise.} \end{cases} \tag{3.62}$$

Thus, the minimum k to make $\overline{R'_k} = \mathbb{C}^{\binom{n+d}{d}}$ is

$$k_{\mathbb{C}}(n,d) := \begin{cases} n+1 & d = 2, n \geq 2; \\ \lceil \frac{1}{n+1}\binom{n+d}{d} \rceil + 1 & (n,d) = (4,3), (2,4), (3,4), (4,4); \\ \lceil \frac{1}{n+1}\binom{n+d}{d} \rceil & \text{otherwise.} \end{cases} \tag{3.63}$$

By parameter counting, we see that R_k is of full measure in R'_k. It remains to show that R'_k

is of full measure in its Zariski closure $\overline{R'_k}$:

Theorem 3.4.2. *R'_k is of full measure in $\overline{R'_k}$.*

Proof. R'_k is just the image of the map $(Q_1, Q_2, \ldots, Q_k) \mapsto (Q_1 + Q_2 + \cdots + Q_k)$. By Exercise 3.19 in Chapter II of [100], R'_k is a constructible set, so it contains an open subset of each connected component of $\overline{R'_k}$. Therefore its complement is of measure 0. □

This immediately implies the following:

Corollary 3.4.1. *R_k has measure 0 in $\mathbb{C}^{\binom{n+d}{d}}$ for $k < k_{\mathbb{C}}(n,d)$ and measure 1 in $\mathbb{C}^{\binom{n+d}{d}}$ for $k \geq k_{\mathbb{C}}(n,d)$.*

Thus, as the integer k increases, $\frac{m(R'_k \cap S)}{m(S)}$ suddenly jumps from 0 to 1 at the point $k_{\mathbb{C}}(n,d)$, and so does $\frac{m(R_k \cap S)}{m(S)}$. This implies part (3) of Theorem 3.1.4.

3.4.3.2 $\mathbb{K} = \mathbb{R}$

Now consider the case $\mathbb{K} = \mathbb{R}$. For $d = 2$, $(n+1)$-variate symmetric tensors are simply $(n+1) \times (n+1)$ symmetric matrices, so a random $(n+1)$-variate symmetric tensor will be of rank $n+1$ with probability 1. However, if the order of the symmetric tensors is larger than 2, the situation is much more complicated. For example, a random bivariate symmetric tensor of order 3 will be of two different ranks, 2 and 3, both with positive probabilities.

From the perspective of algebraic geometry, it still holds that $\overline{R'_k} = \mathbb{R}^{\binom{n+d}{d}}$ for $k \geq k_{\mathbb{C}}(n,d)$, and for $k < k_{\mathbb{C}}(n,d)$, $\overline{R'_k}$ is of measure zero in $\mathbb{R}^{\binom{n+d}{d}}$. It also holds that R_k is of full measure in R'_k. However, the claim that R'_k has full measure in $\overline{R'_k}$ no longer holds over $\mathbb{R}$. As we explained in the proof of Theorem 3.4.2, R'_k is the image of the map

$(Q_1, Q_2, \ldots, Q_k) \mapsto (Q_1 + Q_2 + \cdots + Q_k)$. For an algebraically closed field $\mathbb{K}$, it is known that the image of any map is always a constructible set in its Zariski closure. Thus R'_k is of full measure in $\overline{R'_k}$. Over $\mathbb{R}$, it is easy to verify that the image may not be of full measure in its Zariski closure (a simple counterexample is $x \mapsto x^2$). Consequently, over $\mathbb{C}$, R'_k has non-empty interior for a unique value of k, and this value of k is called the *generic rank*. Over $\mathbb{R}$, R'_k is just a semialgebraic set and it has non-empty interior for several values of k, which are called the *typical ranks*.

For the univariate case, we have the following theorem:

Theorem 3.4.3 ([101, 102]). *For $n = 1$, all integers from $k_{\mathbb{C}} = \lceil \frac{d+1}{2} \rceil$ to $k_{\mathbb{R}} = d$ are typical ranks.*

For the multivariate case $n \geq 2$, it still holds that $k_{\mathbb{C}}(n, d)$ defined in Section 3.4.3.1 is the smallest typical rank [91]. According to [90], every rank between $k_{\mathbb{C}}(n, d)$ and the top typical rank $k_{\mathbb{R}}(n, d)$ is also typical. Thus we only need to study the top typical rank $k_{\mathbb{R}}(n, d)$. Unfortunately, the top typical rank in general is not known. In the literature, considerable effort has been devoted to understanding the maximum possible rank $k_{\max}(n, d)$, which, by definition, is also an upper bound for $k_{\mathbb{R}}(n, d)$. In particular, we have $k_{\max}(n, 2) \leq n + 1$ for $n \geq 2$, $k_{\max}(2, 4) \leq 11$, $k_{\max}(3, 4) \leq 19$, $k_{\max}(4, 4) \leq 29$, $k_{\max}(4, 3) \leq 15$, and $k_{\max}(n, d) \leq 2\lceil \frac{1}{n+1} \binom{n+d}{d} \rceil$ otherwise [91].

The above result implies $k_{\mathbb{R}}(n, d) \leq k_{\max}(n, d) \leq 2k_{\mathbb{C}}(n, d)$. We also mention a few other upper bounds on $k_{\max}(n, d)$. Trivially we have $k_{\max}(n, d) \leq \binom{n+d}{d}$. In [103, 104], this was improved to $k_{\max}(n, d) \leq \binom{n+d}{d} - n$. Later work showed that $k_{\max}(n, d) \leq \binom{n+d-1}{n}$ [105]. Jelisiejew then proved that $k_{\max}(n, d) \leq \binom{n+d-1}{n} - \binom{n+d-5}{n-2}$ [106], and Ballico and De Paris

then improved this to $k_{\max}(n,d) \le \binom{n+d-1}{n} - \binom{n+d-5}{n-2} - \binom{n+d-6}{n-2}$ [107]. For small cases, these bounds may be stronger than the bound $k_{\max}(n,d) \le 2k_{\mathbb{C}}(n,d)$ mentioned above.

To summarize, we have the following, which implies part (2) of Theorem 3.1.4:

Theorem 3.4.4. *As the integer k increases from $k_{\mathbb{C}}(n,d)-1$ to $k_{\mathbb{R}}(n,d) \le 2k_{\mathbb{C}}(n,d)$, $\frac{m(R'_k \cap S)}{m(S)}$ forms a strictly increasing sequence from 0 to 1, and so does $\frac{m(R_k \cap S)}{m(S)}$.*

3.4.3.3 $\mathbb{K} = \mathbb{F}_q$

We link the finite field case with the complex case using the Lang-Weil theorem:

Theorem 3.4.5 (Lang-Weil Theorem, [108])**.** *There exists a constant $A(n,d,r)$ depending only on n,d,r such that for any variety $V \subseteq \mathbb{P}^n$ with dimension r and degree d, if we define V over a finite field $\mathbb{F}_q$, the number of points in V must satisfy*

$$|N - q^r| \le (d-1)(d-2)q^{r-\frac{1}{2}} + A(n,d,r)q^{r-1}. \tag{3.64}$$

The Lang-Weil theorem shows that when q is large enough, the number of points in a variety over $\mathbb{F}_q$ is very close to $q^{\dim V}$. So it actually tells us that $\frac{m(R'_k \cap S)}{m(S)} = 0$ if $k < k_{\mathbb{C}}(n,d)$. It remains unclear whether $\frac{m(R'_k \cap S)}{m(S)} > 0$ for $k = k_{\mathbb{C}}(n,d)$. Once again, for the finite field case, when we talk about the measure, we always assume q is sufficiently large. As in the real field case, the main challenge now is to study the measure of R'_k in $\overline{R'_k}$.

For the upper bound, recall our notation that $v_d(x_1, x_2, \ldots, x_n)$ is the $\binom{n+d}{d}$-dimensional vector that contains all monomials with degree no more than d as its entries.

Here we make a slight change to the definition in which we require all those x_is in v_d to be nonzero. We can similarly define $X''_{n,d}$ and R''_k. We prove the following:

Lemma 3.4.5. *Let $r_{n,d}$ be the minimum number such that $|R''_{r_{n,d}}| = q^{\binom{n+d}{d}} - O(q^{\binom{n+d}{d}-1})$. Then $r_{n,d} \leq r_{n-1,d} + r_{n,d-1}$.*

Proof. The proof is by induction on $n + d$.

For $n + d = 2$, it is easy to verify $r_{2,2} = 3 \leq r_{1,2} + r_{2,1} = 2 + 1$. Assume Lemma 3.4.5 holds for $n + d \leq m - 1$ and consider the pair (n, d) with $n + d = m$. For the sake of readability, we first explain how the induction works for the specific example $(n, d) = (3, 2)$, and then generalize our idea to any (n, d).

The vector

$$v_2(x_1, x_2, x_3) = (x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3, x_1, x_2, x_3, 1)^\top \in X''_{3,2} \tag{3.65}$$

can be rearranged as $(x_3, x_3x_1, x_3x_2, x_3^2, x_1^2, x_2^2, x_1x_2, x_1, x_2, 1)^\top$. The first 4 entries can be rewritten as $x_3^2(\frac{1}{x_3}, \frac{x_1}{x_3}, \frac{x_2}{x_3}, 1)^\top = x_3^2 v_1(\frac{1}{x_3}, \frac{x_1}{x_3}, \frac{x_2}{x_3})$, and the last 6 entries form $v_2(x_1, x_2)$.

When (x_1, x_2, x_3) ranges over all 3-tuples in $\mathbb{F}_q \setminus \{0\}$, $(\frac{1}{x_3}, \frac{x_1}{x_3}, \frac{x_2}{x_3})$ also ranges over all possible 3-tuples in $\mathbb{F}_q \setminus \{0\}$. By assumption, if we take linear combinations of $r_{3,1}$ vectors chosen from $X''_{3,2}$, the first 4 entries will range over no fewer than $q^{\binom{3+1}{1}} - O(q^{\binom{3+1}{1}-1})$ different vectors in $\mathbb{F}_q^{\binom{3+1}{1}}$.

For any such linear combination, we can add $r_{2,2}$ extra vectors from $X_{3,2}$ with the restriction that $x_3 = 0$, which will guarantee these extra vectors do not affect the first 4 entries. By assumption, the last 6 entries will range over no fewer than $q^{\binom{2+2}{2}} - O(q^{\binom{2+2}{2}-1})$ different vectors in $\mathbb{F}_q^{\binom{2+2}{2}}$.

Thus, in total, we have $\left(q^{\binom{3+1}{1}} - O(q^{\binom{3+1}{1}-1})\right)\left(q^{\binom{2+2}{2}} - O(q^{\binom{2+2}{2}-1})\right)$ different vectors in $\mathbb{F}_q^{\binom{3+2}{2}}$ if we take linear combinations of $r_{3,1} + r_{2,2}$ vectors from $X''_{3,2}$, which implies $r_{3,2} \leq$

$r_{3,1} + r_{2,2}$.

For general (n, d), the analogous partition of $v_d(x_1, x_2, \ldots, x_n)$ is still valid. Those $\binom{n+d}{d} - \binom{n-1+d}{d} = \binom{n+d-1}{d-1}$ entries involving x_n will form $x_n^{d-1} v_{d-1}(\frac{1}{x_n}, \frac{x_1}{x_n}, \ldots, \frac{x_{n-1}}{x_n})$ and the rest will form $v_d(x_1, x_2, \ldots, x_{n-1})$. All arguments follow straightforwardly, so we have $r_{n,d} \leq r_{n-1,d} + r_{n,d-1}$ for $n + d = m$ and for any (n, d) by induction. □

Corollary 3.4.2. $r_{n,d} \leq \binom{n+d-1}{d-1}$.

Proof. We use induction on $n+d$. For $n+d = 2$, it is easy to verify. If it is true for $n+d = m$, then for $n + d = m + 1$, we have $r_{n,d} \leq r_{n-1,d} + r_{n,d-1} \leq \binom{n+d-2}{d-1} + \binom{n+d-2}{d-2} = \binom{n+d-1}{d-1}$. □

R''_k is obviously a subset of R'_k, so $k_{\mathbb{F}_q}(n, d) \leq r_{n,d}$. By combining Theorem 3.4.5 and Corollary 3.4.2, we have the following, which implies part (1) of Theorem 3.1.4:

Corollary 3.4.3. $k_{\mathbb{C}}(n, d) \leq k_{\mathbb{F}_q}(n, d) \leq r_{n,d} \leq \binom{n+d-1}{d-1} = \frac{d}{n+d}\binom{n+d}{d}$.

Remark 3.4.1. *By combining Corollary 3.4.2 with the fact $r_{n,2} \geq k_{\mathbb{C}}(n, 2)$, we have $r_{n,2} = n + 1$.*

Remark 3.4.2. *It was previously known that $r_{n,1} = 1$ [109, 110] and $r_{1,d} = \lceil \frac{d+1}{2} \rceil$ [30]. We can further refine the upper bound using these boundary conditions:*

$$\begin{aligned} r_{n,d} &\leq \sum_{i=0}^{d-2} \binom{n-2+i}{i} r_{1,d-i} + \binom{d+n-3}{d-1} \leq \sum_{i=0}^{d-2} \binom{n-2+i}{i} \frac{d-i+3}{2} + \binom{d+n-3}{d-1} \\ &= \frac{n+d+2}{2}\binom{n+d-3}{n-1} - \frac{n-1}{2}\binom{n+d-2}{n} + \binom{d+n-3}{d-1}. \end{aligned} \tag{3.66}$$

3.5 Optimality

In this section, we show that the query complexity of our algorithms over a finite field is precisely optimal: no k-query algorithm can succeed with a probability larger than $|R_k|/q^{d+1}$. We begin with a basic result showing that m states spanning an n-dimensional subspace can be distinguished with probability at most n/m.

Lemma 3.5.1. *Suppose we are given a state $|\psi_c\rangle$ with $c \in C$ chosen uniformly at random. Then the probability of correctly determining c with some orthogonal measurement is at most* $\dim \operatorname{span}\{|\psi_c\rangle : c \in C\}/|C|$.

Proof. Consider a measurement with orthogonal projectors E_c, and let Π denote the projection onto $\operatorname{span}\{|\psi_c\rangle : c \in C\}$. Then we have

$$\Pr[\text{success}] = \frac{1}{|C|}\sum_{c\in C}\langle\psi_c|E_c|\psi_c\rangle \leq \frac{1}{|C|}\sum_{c\in C}\operatorname{tr}(E_c\Pi) = \frac{\operatorname{tr}(\Pi)}{|C|} = \frac{\dim \operatorname{span}\{|\psi_c\rangle : c \in C\}}{|C|} \tag{3.67}$$

as claimed. □

We apply this lemma where $|\psi_c\rangle$ is the final state of a given quantum query algorithm when the black box contains $c \in \mathbb{F}_q^J$. There is no loss of generality in considering an orthogonal measurement at the end of the algorithm since we allow the use of an arbitrary-sized ancilla.

Lemma 3.5.2. *Let $J = \binom{n+d}{d}$ and $|\psi_c\rangle$ be the state of any quantum polynomial interpolation algorithm after k queries, where the black box contains $c \in \mathbb{F}_q^J$. Then* $\dim \operatorname{span}\{|\psi_c\rangle : c \in$

$\mathbb{F}_q^J\} \leq |R_k|$.

Proof. We claim that

$$|\psi_c\rangle = \sum_{x,y\in\mathbb{F}_q^k} e(Z(x,y)\cdot c)|\phi_{x,y}\rangle \tag{3.68}$$

for some set of (unnormalized) states $\{|\phi_{x,y}\rangle : x \in \mathbb{F}_q^{nk}, y \in \mathbb{F}_q^k\}$ that do not depend on c. Then the result follows, since

$$|\psi_c\rangle = \sum_{z\in\mathbb{F}_q^{d+1}} e(z\cdot c) \sum_{x,y\in Z^{-1}(z)} |\phi_{x,y}\rangle \in \operatorname{span}\left\{ \sum_{x,y\in Z^{-1}(z)} |\phi_{x,y}\rangle : z \in \mathbb{F}_q^J \right\}, \tag{3.69}$$

which has dimension at most $|R_k| = |\{Z(x,y) : (x,y) \in \mathbb{F}_q^{nk} \times \mathbb{F}_q^k\}|$.

To see the claim, consider a general k-query algorithm $U_k Q_c U_{k-1} \dots Q_c U_1 Q_c U_0$ acting on states of the form $|x,y,w\rangle$ for an arbitrary-sized workspace register $|w\rangle$, starting in the state $|x_0,y_0,w_0\rangle = |0,0,0\rangle$. Here $Q_c\colon |x,y,w\rangle \mapsto e(yf(x))|x,y,w\rangle$ is a phase query. The final state $|\psi_c\rangle$ equals

$$\sum_{\substack{x\in\mathbb{F}_q^{nk} y\in\mathbb{F}_q^k \\ x_{k+1}\in\mathbb{F}_q^n, y_{k+1}\in\mathbb{F}_q \\ w\in I^{k+1}}} e\left(\sum_{j=1}^k y_j f(x_j)\right)\left(\prod_{j=0}^k \langle x_{j+1},y_{j+1},w_{j+1}|U_j|x_j,y_j,w_j\rangle\right)|x_{k+1},y_{k+1},w_{k+1}\rangle, \tag{3.70}$$

with $x_0 = y_0 = w_0 = 0$, $x = (x_1,\dots,x_k)$, $y = (y_1,\dots,y_k)$, $w = (w_1,\dots,w_{k+1})$, and I some appropriate index set. This expression has the claimed form when we define

$$|\phi_{x,y}\rangle = \sum_{\substack{x_{k+1}\in\mathbb{F}_q^n, y_{k+1}\in\mathbb{F}_q \\ w\in I^{k+1}}} \left(\prod_{j=0}^k \langle x_{j+1},y_{j+1},w_{j+1}|U_j|x_j,y_j,w_j\rangle\right)|x_{k+1},y_{k+1},w_{k+1}\rangle. \qquad \square$$

We can now prove our upper bound on the success probability of quantum algorithms for polynomial interpolation.

Proof of Theorem 3.1.1 (upper bound on success probability). By combining Lemma 3.5.1 with Lemma 3.5.2, we see that if the coefficients $c \in \mathbb{F}_q^J$ are chosen uniformly at random, no algorithm can succeed with probability greater than $|R_k|/q^J$. Since the minimum cannot be larger than the average, this implies a lower bound on the success probability in the worst case of $|R_k|/q^J$. □

This result also shows that the exact quantum query complexity of univariate polynomial interpolation is maximal.

Corollary 3.5.1. *The exact quantum query complexity of interpolating a degree-d univariate polynomial is $d+1$.*

Proof. This follows from Theorem 3.1.1 and the fact that if $k < d+1$, we have $|R_k| < q^{d+1}$. To see this, observe that if $k < d+1$, then vectors of the form $(0, \ldots, 0, z_d)$ for $z_d \neq 0$ are not in the range of Z. We can assume there is an $(x, y) \in Z^{-1}(z)$ with $x_1, \ldots, x_k$ all distinct, since if $x_i = x_j$ for some $i \neq j$, then we could delete index j and replace y_i by $y_i + y_j$. Then in equation (3.37), the Vandermonde matrix on the left-hand side is invertible, so $y_1 = \cdots = y_k = 0$. However, this implies that $\sum_i y_i x_i^d = 0 \neq z_d$. □

Chapter 4: Matrix-vector products

4.1 Introduction

Algorithms for linear algebra problems—for example, solving linear systems and determining basic properties of matrices such as rank, trace, determinant, eigenvalues, and eigenvectors—constitute a fundamental research area in applied mathematics and theoretical computer science. Such tasks have widespread applications in scientific computation, statistics, operations research, and many other related areas. Algorithmic linear algebra also provides a fundamental toolbox that can inspire the design of algorithms in general.

There are several possible models of access to a matrix, and linear-algebraic algorithms can depend significantly on how the input is represented (as discussed further below). One natural model is the *matrix-vector product* (Mv) oracle. For a matrix $M \in \mathbb{F}^{n \times m}$ in a given field $\mathbb{F}$, the Mv oracle takes $x \in \mathbb{F}^m$ as input and outputs $Mx \in \mathbb{F}^n$. Matrix-vector products arise, for example, as the elementary step of the power method (and the related Lanczos method) for computing the largest eigenvector of a matrix. Matrix-vector products also commonly appear in streaming algorithms, especially in the technique of sketching (see the survey [111] for more information).

Recent work has studied the classical complexity of various basic problems in the Mv model. Specifically, Sun, Woodruff, Yang, and Zhang [32] studied the complexities of

various linear algebra, statistics, and graph problems using matrix-vector products, and Braverman, Hazan, Simchowitz, and Woodworth [33] proved tight bounds on maximum eigenvalue computation and linear regression in this model. Rashtchian, Woodruff, and Zhu [112] considered a generalization to the vector-matrix-vector product (vMv) oracle, which returns $x^\top My$ for given input vectors $x \in \mathbb{F}^n, y \in \mathbb{F}^m$, and studied the complexity of various linear algebra, statistics, and graph problems in this setting. Table 4.1 includes a partial summary of these results.

Quantum computers can solve certain problems much faster than classical computers, so it is natural to study quantum query complexity with matrix-vector products. Lee, Santha, and Zhang recently studied the quantum query complexity of graph problems with cut queries [113], which are closely related to matrix-vector queries. For a weighted graph $G = (V, w)$ where $|V| = n$ and w assigns a nonnegative integer weight to each edge, the input of a cut query is a subset $S \subseteq V$ and the output is $|w(S, V \setminus S)|$, the total weight of the edges between S and $V \setminus S$. This can be viewed as a version of the vMv model over $\mathbb{Z}$, with the extra assumptions that $x \in \{0,1\}^n, y \in \{0,1\}^m$ are both boolean and M is a symmetric matrix with nonnegative integer entries. Reference [113] gives quantum algorithms for determining all connected components of G with $O(\log^6 n)$ quantum cut queries, and for outputting a spanning forest of G with $O(\log^8 n)$ quantum cut queries. Both problems require $\Omega(n/\log n)$ classical cut queries, so the quantum algorithms provide exponential speedups.

In other recent work on structured queries for graph problems, Montanaro and Shao studied the problem of learning an unknown graph with "parity queries" [114]: for an unknown graph with adjacency matrix A, the parity oracle takes as input a string x that encodes a subset of the vertices, and returns $x^\top Ax \bmod 2$. This query model is the vMv

model over $\mathbb{F}_2$ with the extra restriction that the left and right vectors are identical.

Van Apeldoorn and Gribling studied Simon's problem for linear functions over a prime field $\mathbb{F}_p$ [115]. In this problem, the oracle encodes a linear function $f\colon \mathbb{F}_p \to \mathbb{F}_p$, and the task is to determine if the function is one-to-one, or if there is a one-dimensional subspace $H \subset \mathbb{F}_p$ such that for every $x, x' \in \mathbb{F}_p^n$, $f(x) = f(x')$ if and only if $x - x' \in H$. Such a function can be represented by a square matrix over $\mathbb{F}_p$, and the problem is equivalent to determining whether that matrix is full rank or has nullity 1 using matrix-vector product queries.

Other past work has developed linear algebraic quantum algorithms using different input models. Quantum algorithms for high-dimensional linear algebra have been studied extensively since Harrow, Hassidim, and Lloyd introduced a method for generating a quantum state proportional to the solution of a large, sparse system of linear equations [116]. This algorithm assumes a quantum oracle that determines the locations and values of the nonzero entries of a matrix in any given row or column, and the ability to generate a quantum state that encodes the right-hand side of the linear system. Subsequent work has led to improved and generalized algorithms under similar assumptions. However, it is challenging to find practical applications that achieve speedup over classical computation [117, 118]. Recent work by Apers and de Wolf [119] gives polynomial quantum speedup for producing an explicit classical description of the solution of a Laplacian linear system, assuming adjacency-list access to the underlying graph of the Laplacian. Note also that for various problems including determinant estimation, rank testing, linear regression, etc., there is a large separation between the classical query complexities under Mv and entrywise queries ($\tilde{\Theta}(n)$ [32] and $\Theta(n^2)$, respectively). These results show how the model of access to a ma-

trix can significantly impact the complexity of solving linear-algebraic problems. A better understanding of the quantum matrix-vector oracle could therefore provide a useful tool for the design of future quantum algorithms.

Contributions. We conduct a systematic study of quantum query complexity with a matrix-vector oracle for a matrix $M \in \mathbb{F}_q^{m\times n}$, where $\mathbb{F}_q$ is a given finite field. Using this model, we provide results on the quantum query complexities of linear algebra and statistics problems.

First, we prove that various linear algebra problems, including computing the trace $\mathrm{tr}(M)$ of $M \in \mathbb{F}_q^{n\times n}$; computing the determinant $\det(M)$ of $M \in \mathbb{F}_q^{n\times n}$; solving the linear system $Ax = b$ for $A \in \mathbb{F}_q^{n\times n}$; and testing whether $\mathrm{rank}(M) = n$ or $\mathrm{rank}(M) \leq n/2$ for a matrix $M \in \mathbb{F}_q^{m\times n}$; require $\Omega(n)$ quantum queries to the Mv oracle. Since $O(n)$ queries suffice to determine the entire matrix, even classically, these results show that no quantum speedup is possible. (As a side effect, we improve the $\Omega(n/\log n)$ classical lower bound for trace computation [32] to $\Omega(n)$.)

Our quantum lower bound for trace computation applies results of Copeland and Pommersheim [29] by viewing the problem as a special case of *coset identification*. Our lower bounds for other linear algebra problems are all proved by the polynomial method [20, 120]. We show how to symmetrize the success probability to a univariate polynomial, and then give a lower bound on the polynomial degree using an observation of Koiran, Nesme, and Portier [121].

On the other hand, we determine the matrix-vector quantum query complexity of several statistics problems, including computing the row and column parities, deciding if there exist two identical columns, and deciding if there exist two identical rows for $M \in \mathbb{F}_2^{m\times n}$.

Specifically, we prove that their quantum query complexities with an Mv oracle are $O(1)$, $O(\log n)$, and $O(\log m)$, respectively. Compared to the classical bounds using either the Mv oracle [32] or the vMv oracle [112], our quantum algorithms achieve *exponential* quantum speedups.

Technically, these results build upon our observation that the quantum query complexities in the Mv model under left or right multiplication are *identical* (Theorem 4.3.1). In particular, one right Mv query can be simulated using one left Mv query, and vice versa. In contrast, classically there is a significant difference between matrix-vector (Mv) and vector-matrix (vM) queries—for example, computing the parity of rows over $\mathbb{F}_2$ only takes $O(1)$ Mv queries, but computing the parity of columns over $\mathbb{F}_2$ requires $\Theta(n)$ Mv queries. In contrast, for both problems a quantum computer can achieve the smaller query complexity by switching to the easier side.

Our results are summarized in Table 4.1, including some implications of our results for classical query complexity and a few additional results over $\mathbb{R}$. Note that there can be large gaps between the classical query complexities with Mv and vMv queries, but they are the same in the quantum setting due to an equivalence between quantum Mv and vMv queries (Theorem 4.3.3), which follows along similar lines to the equivalence between Mv and vM queries. The Mv–vMv equivalence is closely related to a similar equivalence shown in the work of Lee, Santha, and Zhang [113], as we discuss further in Section 4.3.2.

Open questions. Our paper leaves several natural open questions for future investigation: First of all, for linear algebra problems such as those we studied, can we also prove quantum query lower bounds for matrices over the real field $\mathbb{R}$? Our proofs rely on the polynomial

method, and it is unclear how to adapt them to a setting with continuous input.

Can we prove a quantum lower bound for the task of minimizing a quadratic form $f(x) = \frac{1}{2}x^\top Ax + b^\top x$, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$? Note that f is minimized at $x = -A^{-1}b$, and we can determine the vector b and implement Mv queries to the matrix A using fast quantum gradient computation [122], so this is closely related to the previous open question. Quadratic form minimization is a special case of optimizing a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ by quantum evaluation queries, where previous works [123, 124, 125] left a quadratic gap between the best known quantum upper and lower bounds of $\tilde{O}(n)$ and $\Omega(\sqrt{n})$, respectively.

For the finite field case, can we identify other problems with quantum speedup over the classical matrix-vector oracle, or find advantage compared to other quantum oracles such as entrywise queries?

Organization. We review necessary background in Section 4.2. We prove the equivalence of quantum matrix-vector and vector-matrix-vector product oracles in Section 4.3. In Section 4.4, we prove tight quantum query complexity lower bounds on various linear algebra problems, including trace, determinant, linear systems, and rank.

4.2 Preliminaries

4.2.1 The quantum query model

Given a set X and an abelian group G, let $f\colon X \to G$ be a function. Access to f is provided by a black-box unitary operation $U_f\colon |x, y\rangle \mapsto |x, y + f(x)\rangle$ for all $x \in X$ and $y \in G$. We call an application of U_f a (standard) query.

For a finite abelian group G, the Fourier transform over G is

$$F_G \coloneqq \frac{1}{|G|^{1/2}} \sum_{x \in G} \sum_{y \in \hat{G}} \chi_y(x) |y\rangle\langle x|, \tag{4.1}$$

where $\hat{G}$ is a complete set of characters of G, and $\chi_y \colon G \to \mathbb{C}$ denotes the y^{th} character of G. Since $\hat{G} \cong G$, we label elements of $\hat{G}$ using elements of G. Note that χ_y is a group homomorphism, i.e., $\chi_y(x+z) = \chi_y(x)\chi_y(z)$. In addition, the characters satisfy the orthogonality condition

$$\frac{1}{|G|} \sum_{z \in G} \chi_y(z)^* \chi_w(z) = \delta_{yw}. \tag{4.2}$$

A phase query is defined as a standard query conjugated by the Fourier transform acting on the output register. In other words, for $x \in X$ and $y \in G$, a phase query acts as

$$\begin{aligned} |x, y\rangle &\xmapsto{\mathbb{1} \otimes F_G^\dagger} \frac{1}{|G|^{1/2}} \sum_{z \in G} \chi_y(z)^* |x, z\rangle \\ &\xmapsto{U_f} \frac{1}{|G|^{1/2}} \sum_{z \in G} \chi_y(z)^* |x, z + f(x)\rangle \\ &\xmapsto{\mathbb{1} \otimes F_G} \frac{1}{|G|} \sum_{z \in G} \chi_y(z)^* \chi_w(z + f(x)) |x, w\rangle = \chi_y(f(x)) |x, y\rangle. \end{aligned} \tag{4.3}$$

The equality in (4.3) follows from the orthogonality condition in (4.2). Since one can simulate a phase query using a single standard query and vice versa, the query complexities of any problem are equal with these two models.

Over a finite field $\mathbb{F}_q$ for prime power $q = p^r$, the Fourier transform over $\mathbb{F}_q$ is the unitary transformation $|x\rangle \mapsto q^{-1/2} \sum_{y \in \mathbb{F}_q} e(xy)|y\rangle$, where the exponential function $e \colon \mathbb{F}_q \to \mathbb{C}$ is

defined as $e(z) \coloneqq e^{2\pi i \operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(z)/p}$ and the trace function $\operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p} \colon \mathbb{F}_q \to \mathbb{F}_p$ is defined as $\operatorname{Tr}_{\mathbb{F}_q/\mathbb{F}_p}(z) \coloneqq z + z^p + z^{p^2} + \cdots + z^{p^{r-1}}$.

Over the field of real numbers, the quantum Fourier transform is

$$F_{\mathbb{R}} \coloneqq \int_{\mathbb{R}} \mathrm{d}y \int_{\mathbb{R}} \mathrm{d}x \, e^{2\pi i y x} |y\rangle\langle x|. \tag{4.4}$$

The basis states $\{|x\rangle : x \in \mathbb{R}\}$ are normalized to the Dirac delta function, i.e., for $x, x' \in \mathbb{R}$, $\langle x'|x\rangle = \delta(x - x')$. Here the Dirac delta function δ satisfies $\int_{\mathbb{R}} \mathrm{d}x' \, \delta(x - x') f(x') = f(x)$ for any function f. Furthermore, we have $\int_{\mathbb{R}} \mathrm{d}y \, e^{2\pi i y(x-x')} = \delta(x - x')$. By direct calculation using these facts, $F_{\mathbb{R}}^\dagger F_{\mathbb{R}} = \int_{\mathbb{R}} \mathrm{d}x \, |x\rangle\langle x| = \mathbb{1}$.

While we can formally consider a model of query complexity over $\mathbb{R}$ with arbitrary precision, its practical instantiation requires discrete approximation. We can achieve precision ϵ by approximating real numbers with $s = O(\log(1/\epsilon))$ bits, and can then replace the continuous Fourier transform with the discrete Fourier transform over $\mathbb{Z}_{2^s}$. It is straightforward to show that a discretized phase query over $\mathbb{Z}_{2^s}$ can be implemented by Fourier transforming a standard query that maps discretized inputs to discretized function values.

4.2.2 The coset identification problem

Copeland and Pommersheim studied a kind of quantum query problem that they call the *coset identification problem* [29]. They define this problem in a generalized query model where the black box does not necessarily perform a standard or phase query, although their definition includes those cases. In the coset identification problem, we fix a finite group G and a subgroup $H \leq G$. The algorithm is given access to a unitary transformation $\pi(g)$,

where π is a representation of G on vector space V. When π is given, the vector space V is called the representation space (or simply, the representation) of G [126, Chapter 1]. The goal is to determine which coset of H the unknown element $g \in G$ belongs to.

Definition 4.2.1 (Coset identification problem [29]). *A coset identification problem for a finite group G and subgroup $H \leq G$ is a 3-tuple (π, V, F) such that*

- *π is a unitary representation of G in the complex vector space V, and*
- *F is a function constant on left cosets of $H \leq G$ and distinct on distinct cosets, i.e., $F(g) = F(g')$ if and only if $g' = gh$ for some $h \in H$.*

Given a black box that performs the unitary transformation $\pi(g)$, the goal is to compute $F(g)$.

Copeland and Pommersheim show that the optimal success probability of a t-query algorithm for a coset identification problem can be calculated by taking, over all irreps Y of H, the maximum of the fraction of the induced representation $Y^{\uparrow}$ of G shared with $V^{\otimes t}$. Furthermore, the optimal algorithm can be non-adaptive. For a representation V, let $I(V)$ denote the set of irreducible characters of G appearing in V.

Theorem 4.2.1 (Optimal success probability of coset identification [29, Corollary 5.7]). *The optimal success probability of any t-query quantum algorithm $\mathcal{A}$ for the coset identification problem (π, V, F) for finite group G and subgroup $H \leq G$, under uniformly random inputs in G, is*

$$\Pr[\mathcal{A}^{\pi(g)} = F(g)] = \max_{Y} \frac{\dim Y^{\uparrow}_{V^{\otimes t}}}{\dim Y^{\uparrow}}, \tag{4.5}$$

where the probability is maximized over all irreducible representations Y of H, $Y^{\uparrow}$ is the in-

duced representation of G, and A_B is the maximal subrepresentation of A such that $I(A_B) \subseteq I(B)$ for representations A, B.

The *oracle discrimination problem* is the special case of the coset identification problem where H is the trivial group, i.e., the function F is injective. In this case, $Y^{\uparrow} = \mathrm{span}\{|g\rangle : g \in G\}$.

Corollary 4.2.1 (Optimal success probability of oracle discrimination [29, Theorem 4.2])**.** *The optimal success probability of the oracle discrimination problem is $\frac{1}{|G|}\sum_{i \in I(V^{\otimes t})} d_i^2$, where $I(V^{\otimes t})$ is the irrep content of $(\pi^{\otimes t}, V^{\otimes t})$ and d_i is the dimension of irrep $i \in I(V^{\otimes t})$.*

We consider the complexity of standard queries in the matrix-vector model. In this model, oracle access to a matrix $M \in \mathcal{F}^{m \times n}$ for field $\mathcal{F}$ and positive integers m, n is the unitary operation $U(M)\colon |x, y\rangle \mapsto |x, y + Mx\rangle$. The map U is a representation of the additive group of matrices since it is a group homomorphism satisfying $U(M)U(N) = U(M + N)$ for all matrices M, N of the same dimensions. A phase query is also a unitary representation since it is a standard query conjugated by a fixed unitary matrix (the quantum Fourier transform).

4.2.3 The polynomial method

We will use the polynomial method to obtain quantum lower bounds. Here we state a version for non-boolean functions as used in [120].

Lemma 4.2.1. *Let $\mathcal{A}$ be a t-query quantum algorithm given access to the input $x \in [m]^n$ for $m, n \in \mathbb{Z}$ through oracle $U_x\colon |i, j\rangle \mapsto |i, j + x_i\rangle$ for $i \in [n]$ and $j \in [m]$. The acceptance probability of $\mathcal{A}$ on input x is a degree-$(2t)$ polynomial in $x_1, \ldots, x_n$.*

4.3 Equivalence of matrix-vector and vector-matrix-vector products

In this section, we show that the matrix-vector and vector-matrix-vector models are equivalent, i.e., for any problem, the quantum query complexities in these models differ by at most a constant factor. Furthermore, we show that in the matrix-vector model, left matrix-vector products and right matrix-vector products are equivalent. This is in stark contrast to the classical case where these query complexities can differ significantly, as mentioned in Section 4.1 and discussed further below.

4.3.1 Left and right matrix-vector queries

We first show that left matrix-vector products and right matrix-vector products are equivalent.

Theorem 4.3.1. *Quantum query complexities in the left and right matrix-vector models over a finite field are identical. In particular, one right* Mv *query can be simulated using one left* Mv *query, and vice versa.*

Proof. For input matrix $M \in \mathbb{F}_q^{n \times m}$, a matrix-vector ($\mathsf{Mv}$) query applies the unitary transformation

$$U^{\mathsf{Mv}}(M)\colon |x, y\rangle \mapsto |x, y + Mx\rangle \tag{4.6}$$

for every $x \in \mathbb{F}_q^m$ and $y \in \mathbb{F}_q^n$. Conjugating by a quantum Fourier transform on the output

register yields a phase query

$$
\begin{aligned}
|x, y\rangle &\xmapsto{\mathbb{1}\otimes F_{\mathbb{F}_q^n}^\dagger} q^{-1/2} \sum_z e(-y^\top z)|x, z\rangle \\
&\xmapsto{U^{\mathsf{Mv}}(M)} q^{-1/2} \sum_z e(-y^\top z)|x, z + Mx\rangle \\
&\xmapsto{\mathbb{1}\otimes F_{\mathbb{F}_q^n}} q^{-1} \sum_{z,w} e(-y^\top z + w^\top (z + Mx))|x, w\rangle \\
&= \sum_w \delta[y = w] e(-y^\top z + w^\top (z + Mx))|x, w\rangle \\
&= e(y^\top Mx)|x, y\rangle. \qquad (4.7)
\end{aligned}
$$

We denote this unitary transformation by $U^{\widetilde{\mathsf{Mv}}}(M)$.

Conjugating a phase query by a swap gate, we have

$$
\begin{aligned}
|x, y\rangle &\xmapsto{\mathsf{SWAP}} |y, x\rangle \\
&\xmapsto{U^{\widetilde{\mathsf{Mv}}}(M)} e(x^\top My)|y, x\rangle \\
&\xmapsto{\mathsf{SWAP}} e(x^\top My)|x, y\rangle \\
&= e(y^\top M^\top x)|x, y\rangle. \qquad (4.8)
\end{aligned}
$$

This yields $U^{\widetilde{\mathsf{Mv}}}(M^\top)$, which in turn gives $U^{\mathsf{Mv}}(M^\top)$ upon conjugation by an inverse quantum Fourier transform on the output register. Thus one can simulate the oracle $U^{\mathsf{Mv}}(M^\top)$ using one query to $U^{\mathsf{Mv}}(M)$, showing equivalence of the two models. □

In contrast to Theorem 4.3.1, Sun, Woodruff, Yang, and Zhang show that for the task of computing the row parities of an $m \times n$ matrix M over $\mathbb{F}_2$, the left query complexity is

$\Omega(m)$, whereas the right query complexity is 1 [32]. Thus we have shown that computing column parities over $\mathbb{F}_2$ in the Mv model has quantum query complexity 1, significantly less than the classical query complexity of $\Omega(n)$.

Corollary 4.3.1. *The query complexity of computing the row parities and the column parities of an $m \times n$ matrix over $\mathbb{F}_2$ is 1.*

Note that it is easy to understand the randomized query complexities of these problems in the vMv model.

Lemma 4.3.1. *The randomized query complexities of computing the row parities and the column parities of an $m \times n$ matrix over $\mathbb{F}_2$ are $\Theta(m)$ and $\Theta(n)$, respectively.*

Proof. Each query reveals one bit of information, while the row parities convey m bits, giving a lower bound of $\Omega(m)$. An algorithm querying $(e_1, 1^n), \ldots, (e_m, 1^n)$ learns the row parities with probability 1, giving an upper bound of m. The query complexity of column parities follows immediately from the symmetry of the vMv oracle. □

The randomized query complexities of determining if there exist identical columns or identical rows are $\Theta(n/m)$ and $\Theta(\log m)$, respectively [32]. Theorem 4.3.1 implies that for identical columns, there is an exponential quantum speedup.

Corollary 4.3.2. *The query complexities of deciding if there exist two identical columns and rows in a $m \times n$ matrix over $\mathbb{F}_2$ are $O(\log n)$ and $O(\log m)$, respectively.*

Proof. By Theorem 4.3.1, it suffices to give an algorithm for determining if there are two identical rows. To make the proof self-contained, we describe the algorithm of Sun, Woodruff, Yang, and Zhang [32, Section 4.2]. The algorithm makes q random queries $v_1, \ldots, v_q$, the

entries of which are sampled uniformly. The algorithm outputs 1 if and only if there exist two entries i, j such that $(Mv_k)_i = (Mv_k)_j$ for $k \in [q]$.

To analyze the performance, for any two identical rows $m_i^\top, m_j^\top$, $\Pr_v[m_i^\top v = m_j^\top v] = 1$. For $m_i \neq m_j$, $\Pr_v[m_i^\top v = m_j^\top v] \leq 1/2$. Therefore for a matrix that has two identical rows, the algorithm outputs 1 with probability 1. On the other hand, for a matrix that has no identical rows, the algorithm outputs 1 with probability

$$\begin{aligned}\Pr_{v_1,\ldots,v_q}[\exists i,j \in [m],\ \forall \ell \in [q], m_i^\top v_\ell = m_j^\top v_\ell] &\leq \sum_{i,j\in[m],i\neq j} \Pr_{v_1,\ldots,v_q}[\forall \ell \in [q], m_i^\top v_\ell = m_j^\top v_\ell] \\ &\leq \binom{m}{2} 2^{-q}. \end{aligned} \tag{4.9}$$

Taking $q = 2\log m$, the probability is no more than $\frac{1}{2} - \frac{1}{2m}$. □

The equivalence of left and right queries also holds over the reals.

Theorem 4.3.2. *Quantum query complexities in the left and the right matrix-vector models over $\mathbb{R}$ are identical. In particular, one right* Mv *query can be simulated using one left* Mv *query, and vice versa.*

Proof. The same idea as in the proof of Theorem 4.3.1 applies. First, a phase query can be simulated by conjugating a standard query by the quantum Fourier transform. This yields $U^{\widetilde{\mathsf{Mv}}}(M)$. Conjugating a phase query by a swap gate gives $U^{\widetilde{\mathsf{Mv}}}(M^\top)$ with the same calculation as in (4.8). This in turn yields $U^{\mathsf{Mv}}(M^\top)$ upon conjugating $U^{\widetilde{\mathsf{Mv}}}(M^\top)$ by an inverse quantum Fourier transform. □

Note that with finite precision, a phase query can be simulated using the quantum Fourier transform over an integer modulus (see Section 4.2.1 for details).

As an example, we determine the query complexity of the majority of rows or columns: given a binary matrix $M \in \{0,1\}^{m\times n}$, compute the majority of each row or column over $\mathbb{R}$.

Corollary 4.3.3. *The query complexities of computing the majorities of rows and columns of an $m \times n$ matrix over $\mathbb{R}$ are 1.*

Proof. By Theorem 4.3.2, it suffices to show the query complexity of the majority of rows is 1. With a single query $(1, 1, \ldots, 1)^\top$, the majority of each row is determined. □

This result is not significantly affected by considering computation with finite precision. The number of 1s in each row and each column is an integer in $[0, k]$ for $k = \max\{m, n\}$. Thus a truncation with $O(\log k)$ bits suffices to perform the computation with no error.

4.3.2 The vector-matrix-vector model

We now relate the power of the matrix-vector and vector-matrix-vector query models. In the vector-matrix-vector model, the algorithm is given access to M via $U^{\mathsf{vMv}}\colon |x, y, a\rangle \mapsto |x, y, a + y^\top Mx\rangle$. We can simulate one vMv query using two Mv queries and an ancilla space storing a matrix-vector product:

$$\begin{aligned} |x, y, a\rangle &\xmapsto{U^{\mathsf{Mv}}(M)} |x, y, a\rangle|Mx\rangle \\ &\mapsto |x, y, a + y^\top Mx\rangle|Mx\rangle \\ &\xmapsto{U^{\mathsf{Mv}}(M)^\dagger} |x, y, a + y^\top Mx\rangle|0\rangle. \end{aligned} \tag{4.10}$$

On the other hand, an Mv phase query (defined previously in (4.7)) can be simulated

using a vMv phase query by setting $a = 1$:

$$|x, y, 1\rangle \mapsto e(y^\top M x)|x, y, 1\rangle. \tag{4.11}$$

Such a vMv phase query can be constructed using one application of U^{vMv}:

$$\begin{aligned}
|x, y, a\rangle &\xmapsto{\mathbb{1}\otimes\mathbb{1}\otimes F_{\mathbb{F}_q}^\dagger} \sum_b e(-ab)|x, y, b\rangle \\
&\xmapsto{U^{\mathsf{vMv}}(M)} \sum_b e(-ab)|x, y, b + y^\top M x\rangle \\
&\xmapsto{\mathbb{1}\otimes\mathbb{1}\otimes F_{\mathbb{F}_q}} \sum_{bc} e(-ab + c(b + y^\top M x))|x, y, c\rangle \\
&= e(ay^\top M x)|x, y, a\rangle.
\end{aligned} \tag{4.12}$$

Thus we have shown the following.

Theorem 4.3.3. *Quantum query complexities in the matrix-vector and vector-matrix-vector models differ by at most a constant factor. In particular, one* vMv *query can be simulated using two* Mv *queries, and one* Mv *query can be simulated using one* vMv *query.*

This is again in stark contrast to the classical case, where the Mv model can be much more powerful than the vMv model. For example, for distinguishing a full-rank matrix from a rank-$(n-1)$ matrix, the randomized query complexity in the vMv model is $\Omega(n^2)$ [112], while the randomized query complexity in the Mv model is $O(n)$ [32].

Note that Lee, Santha, and Zhang [113] previously studied the equivalence between quantum Mv and vMv oracles. They focus on the special case where the matrix M is the adjacency matrix of a graph with nonnegative integer weights and the inputs $x \in \{0,1\}^n, y \in$

$\{0,1\}^m$ are boolean. In that setting, they prove equivalence between the vMv oracle and the additive oracle $a\colon 2^{[n]} \to \mathbb{Z}$ that returns $a(S) = \sum_{(u,v)\in S^{(2)}} w(u,v)$ for $S \subseteq [n]$, where $S^{(2)}$ denotes the set of cardinality-2 subsets of S. They also study relationships with other oracles that encode specific information about graphs (cuts, disjoint cuts, etc.; see Section 4 of [113]). In contrast, our Theorem 4.3.1, Theorem 4.3.2, and Theorem 4.3.3 work for inputs and matrices in fields, and do not apply to other graph oracles. While these results are, strictly speaking, incomparable, they are closely related, both following from a generalization of the Bernstein-Vazirani algorithm [109].

4.4 Linear algebra over finite fields

We now consider the quantum query complexity of particular linear algebra problems in the matrix-vector query model. Specifically, we consider learning the trace (Section 4.4.1), computing the null space and determinant (Section 4.4.2), solving linear systems (Section 4.4.3), and estimating the rank (Section 4.4.4).

4.4.1 Trace

In this section, we show that the quantum query complexity of computing the trace of an $n \times n$ matrix over $\mathbb{F}_q$ is $\Theta(n)$. Since there is a trivial algorithm that computes the trace by learning the entire matrix using n queries, we focus on the lower bound.

Learning the trace can be regarded as a coset identification problem (defined in Section 4.2.2) in the group $G = \mathbb{F}_q^{n\times n}$ with subgroup $H = \{M \in \mathbb{F}_q^{n\times n} : \operatorname{tr} M = 0\} \cong \mathbb{F}_q^{n^2-1}$. The irreducible characters χ_Z of H are indexed by $Z \in \mathbb{Z}_m^{n\times n}$ with $Z_{nn} = 0$, and satisfy

$\chi_Z(M) = e(\langle Z, M\rangle)$ where $\langle Z, M\rangle \coloneqq \sum_{i,j=1}^{n} Z_{ij}M_{ij}$.

4.4.1.1 Learning the trace over $\mathbb{F}_2$

First we consider the case $q = 2$. Then the irreducible characters χ_Z of H for $Z \in \mathbb{Z}_m^{n\times n}$ (with $Z_{nn} = 0$) satisfy

$$\chi_Z(M) = (-1)^{\langle Z,M\rangle}. \tag{4.13}$$

For irredicible character Z, the induced representation can be decomposed into two irreducible characters of G:

$$\chi_{Z,0}(M) = (-1)^{\langle Z,M\rangle}; \qquad \chi_{Z,1}(M) = (-1)^{\langle Z,M\rangle + \operatorname{tr} M}. \tag{4.14}$$

It is easy to check that for $M \in G$, $\chi_{Z,0}(M+E_{nn}) = \chi_{Z,0}(M)$ and $\chi_{Z,1}(M+E_{nn}) = -\chi_{Z,1}(M)$, where E_{ij} is an $n \times n$ matrix whose entries are zero except that $(E_{ij})_{ij} = 1$. We emphasize that in (4.14), $M \in G$ (rather than in H since we are now looking at the representations of the entire group), and $Z_{nn} = 0$.

On the other hand, recall that the phase query oracle is $U(M)\colon |x, y\rangle \mapsto (-1)^{y^\top Mx}|x, y\rangle$, which is a unitary representation of M with character $\xi(M) \coloneqq \operatorname{tr}(U(M)) = \sum_{x,y\in\mathbb{F}_2^n}(-1)^{y^\top Mx}$. To determine the optimal success probability, we calculate the irrep content of $U^{\otimes t}$. The

character of $U^{\otimes t}$ is ξ^t, satisfying

$$\begin{aligned}\mathrm{tr}(U^{\otimes t}(M)) &= \mathrm{tr}(U(M))^t = (\xi(M))^t \\ &= \left(\sum_{x,y\in\mathbb{F}_2^n} (-1)^{y^\top M x}\right)^t = \sum_{x_1,\ldots,x_t,y_1,\ldots,y_t\in\mathbb{F}_2^n} (-1)^{\sum_i y_i M x_i}. \end{aligned} \tag{4.15}$$

Thus it has non-zero Fourier coefficient at W if and only if $W \in R_t$, where R_t is the set of matrices of rank no more than t.

We now check containment of the irreps (4.14) in $U^{\otimes t}$. We find

$$m_{Z,0}^{(t)} = \langle \xi^t, \chi_{Z,0}\rangle > 0 \iff Z \in R_t, \quad m_{Z,1}^{(t)} = \langle \xi^t, \chi_{Z,0}\rangle > 0 \iff Z + \mathbb{1}_n \in R_t. \tag{4.16}$$

By Theorem 4.2.1, to succeed with probability better than $1/2$, we must choose a Z such that both $m_{Z,0}^{(t)} > 0$ and $m_{Z,1}^{(t)} > 0$. However, now we show this is impossible with $t < n/2$.

Lemma 4.4.1. *The set* $\{Z : m_{Z,0}^{(t)} > 0 \wedge m_{Z,1}^{(t)} > 0\}$ *is empty for* $t < n/2$.

Proof. We show that the set is non-empty only if $t \geq n/2$. Suppose there exists Z such that $m_{Z,0} > 0$ and $m_{Z,1} > 0$. By (4.16), $Z \in R_t$ and $Z + \mathbb{1}_n \in R_t$. Since the ranks of Z and $Z + \mathbb{1}_n$ are no more than t, we conclude that the rank of $\mathbb{1}_n = Z + Z + \mathbb{1}_n$ is no more than $2t$. Therefore $t \geq n/2$. □

This implies an $n/2$ lower bound, formally stated as follows.

Lemma 4.4.2. *For* $t < n/2$, *any* t*-query quantum algorithm computing the trace of an* $n \times n$ *matrix over* $\mathbb{F}_2$ *succeeds with probability at most* $1/2$.

Proof. By Theorem 4.2.1 and Lemma 4.4.1, the optimal success probability for a uniformly

random matrix in $\mathbb{F}_2^{n\times n}$ is

$$\frac{1}{2}\max_Z \sum_{b=0}^{1} \delta[m_{Z,b} > 0] \leq \frac{1}{2} \tag{4.17}$$

for $t < n/2$. □

On the upper bound side, we present an $\lceil n/2 \rceil$-query quantum algorithm, showing that the above lower bound is achievable.

Lemma 4.4.3. *In the matrix-vector query model, there exists an $\lceil n/2 \rceil$-query quantum algorithm that computes the trace of an $n \times n$ matrix over $\mathbb{F}_2$ with probability 1.*

Proof. First we pad the matrix with one extra zero row and one extra zero column if n is odd, and denote the padded matrix by M'. Let $\ell = \lceil n/2 \rceil$. It is clear that one query to $M' \in \mathbb{F}_2^{2\ell\times 2\ell}$ can be simulated using one query to M. By Theorem 4.2.1, it suffices to find an irreducible character such that both $m_{Z,0} > 0$ and $m_{Z,1} > 0$. Now consider

$$Z = \begin{bmatrix} \mathbb{1}_\ell & 0 \\ 0 & 0 \end{bmatrix} = \sum_{i=1}^{\ell} e_i e_i^\top, \qquad Z + \mathbb{1}_{2\ell} = \begin{bmatrix} 0 & 0 \\ 0 & \mathbb{1}_\ell \end{bmatrix} = \sum_{i=\ell+1}^{2\ell} e_i e_i^\top. \tag{4.18}$$

The algorithm first prepares the state

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}|e_1, \ldots, e_\ell\rangle|e_1, \ldots, e_\ell\rangle + \frac{1}{\sqrt{2}}|e_{\ell+1}, \ldots, e_{2\ell}\rangle|e_{\ell+1}, \ldots, e_{2\ell}\rangle. \tag{4.19}$$

Making ℓ phase queries in parallel, we have

$$\begin{aligned}|\psi_M\rangle &= U^{\widetilde{\mathsf{Mv}}}(M')|\psi_0\rangle \\ &= \frac{1}{\sqrt{2}}(-1)^{\sum_{i=1}^{\ell} M'_{ii}}|e_1,\ldots,e_\ell\rangle|e_1,\ldots,e_\ell\rangle \\ &\quad + \frac{1}{\sqrt{2}}(-1)^{\sum_{i=\ell+1}^{2\ell} M'_{ii}}|e_{\ell+1},\ldots,e_{2\ell}\rangle|e_{\ell+1},\ldots,e_{2\ell}\rangle. \end{aligned} \tag{4.20}$$

Measuring in the basis $\{|\psi_0\rangle\langle\psi_0|, |\psi_1\rangle\langle\psi_1|\}$, where

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}|e_1,\ldots,e_\ell\rangle|e_1,\ldots,e_\ell\rangle - \frac{1}{\sqrt{2}}|e_{\ell+1},\ldots,e_{2\ell}\rangle|e_{\ell+1},\ldots,e_{2\ell}\rangle, \tag{4.21}$$

the algorithm outputs the trace with probability 1. □

The results of this section are summarized in the following theorem.

Theorem 4.4.1. *In the matrix-vector query model, no quantum algorithm can compute the trace of an $n \times n$ matrix over $\mathbb{F}_2$ with probability better than 1/2 using fewer than $n/2$ queries, and there exists a quantum algorithm that succeeds with probability 1 using $\lceil n/2 \rceil$ queries.*

4.4.1.2 Learning the trace over $\mathbb{F}_q$

Now we prove a linear lower bound for the task of learning the trace over $\mathbb{F}_q$. The proof idea is the same as in the case $q = 2$, generalized to any finite field.

Theorem 4.4.2. *In the matrix-vector query model over $\mathbb{F}_q$, computing the trace of an $n \times n$ matrix with probability more than $1/q$ requires at least $n/2$ queries.*

Proof. The induced representation of Z (defined in the second paragraph of Section 4.4.1)

can be decomposed into q 1-dimensional irreps whose characters are

$$\chi_{Z,s}(M) = e(\langle Z, M\rangle + s \cdot \operatorname{tr} M) = e(\langle Z + s\mathbb{1}_n, M\rangle) \tag{4.22}$$

for $s \in \mathbb{F}_q$. Again, recall that a phase query oracle $U(M)\colon |x, y\rangle \mapsto e(y^\top Mx)|x, y\rangle$ is a unitary representation of M. The character of U is the trace $\xi(M) \coloneqq \operatorname{tr}(U(M)) = \sum_{x,y\in\mathbb{F}_q^n} e(y^\top Mx)$. The optimal success probability is determined by the irrep content of $U^{\otimes t}$, and the character of $U^{\otimes t}$ is ξ^t, satisfying

$$\operatorname{tr}(U^{\otimes t}(M)) = \xi^t(M) = \sum_{x_1,\ldots,x_t,y_1,\ldots,y_t\in\mathbb{F}_q^n} e\left(\sum_{i=1}^{t} y_i^\top Mx_i\right). \tag{4.23}$$

Thus for every $s \in \mathbb{Z}_m$,

$$m_{Z,s}^{(t)} = \langle \xi^t, \chi_{Z,s}\rangle > 0 \iff Z + s \cdot \mathbb{1}_n \in R_t, \tag{4.24}$$

where R_t is the set of matrices of rank no more than t. Since $\mathbb{1}_n \notin R_{n-1}$, we conclude for $t < n/2$ the success probability is at most $1/q$, as claimed. □

4.4.2 Null space

In this section, we show a linear lower bound on the matrix-vector quantum query complexity of computing the rank of a matrix $M \in \mathbb{F}_q^{m\times n}$ for $m \geq n$. This is without loss of generality since for $m < n$, by Theorem 4.3.1, we can simulate oracle access to $M^\top$ using one query to M.

The rank problem is an instance of the hidden subgroup problem (HSP) over $\mathbb{F}_q^m$ since two vectors map to the same value if and only if their difference is in the null space. However, the lower bound for the abelian HSP [121] does not directly apply to this problem since the instance is more structured—specifically, the subgroup hiding function is a linear transformation.

We recall some standard facts from linear algebra over finite fields. For $\ell \geq m$, let $\binom{\ell}{m}_q := \frac{\prod_{i=0}^{m-1}(q^\ell - q^i)}{\prod_{i=0}^{m-1}(q^m - q^i)}$ denote a Gaussian binomial coefficient.

Lemma 4.4.4. *The number of m-dimensional subspaces of an ℓ-dimensional space over $\mathbb{F}_q$ is $\binom{\ell}{m}_q$.*

Lemma 4.4.5. *For integers $k \leq m \leq \ell$ and any k-dimensional space V over $\mathbb{F}_q$, the number of m-dimensional subspaces of an ℓ-dimensional space containing V is $\binom{\ell-k}{m-k}_q$.*

For proofs of these facts, see for example [127, Lemma 9.3.2].

Computing the rank. Now we consider the problem of computing the rank of a matrix $M \in \mathbb{F}_q^{m \times n}$ for $m \geq n$. A matrix M has rank r if and only if its null space is $(n-r)$-dimensional.

By Lemma 4.2.1, the success probability of a t-query algorithm is a degree-$2t$ polynomial in δ_{xy}. This polynomial P can be written as

$$P(\delta) = \sum_{S \subseteq \mathbb{F}_q^n \times \mathbb{F}_q^m} c_S \prod_{(x,y) \in S} \delta_{xy}, \tag{4.25}$$

with $c_S = 0$ for $|S| > \deg(P)$. For an input M, the assignments to these variables are $\delta_{xy} = \delta[Mx = y]$; we will sometimes write $\delta_{xy} = \delta_{xy}(M)$ to emphasize that δ is a function of

M.

Now symmetrize by averaging over all matrices with nullity d, giving

$$\begin{aligned} Q(d) &\coloneqq \mathop{\mathbb{E}}_{M\sim Y_d}[P(\delta(M))] \\ &= \sum_{S\subseteq \mathbb{F}_q^n\times\mathbb{F}_q^m} c_S \mathop{\mathbb{E}}_{M\sim Y_d}\left[\prod_{(x,y)\in S}\delta_{xy}(M)\right] \\ &= \sum_{S\subseteq \mathbb{F}_q^n\times\mathbb{F}_q^m} c_S \Pr_{M\sim Y_d}[Mx = y\ \forall (x,y)\in S], \end{aligned} \tag{4.26}$$

where Y_d is the set of matrices of nullity d. Here M is drawn uniformly from Y_d. Since $0 \leq P(\delta(M)) \leq 1$, we have $0 \leq Q(d) \leq 1$. The following lemma states that we can approximate $Q(d)$ with a low-degree polynomial. Van Apeldoorn and Gribling previously showed the same statement in their proof of a lower bound for Simon's problem for linear functions [115, Lemma 3]. That problem can be viewed as a special case of our problem with $m = n$. We observe that essentially the same proof establishes this lemma for $m \geq n$.

Lemma 4.4.6. *There exists a polynomial R of degree at most $2t$ such that for each $d \in [n]$, $R(q^d) = Q(d)$.*

We emphasize that we do not bound the degree of $Q(d)$ because we do not know how to represent it as a polynomial in d. Instead, the lower bound is established by showing (i) a lower bound on the degree of the polynomial R and (ii) that the degree of R is no more than $2t$.

Next, recall a lemma by Koiran, Nesme, and Portier [121, Lemma 5].

Lemma 4.4.7. *Let $c > 0$ and $\xi > 1$ be constants and let f be a real polynomial with the following properties:*

1. *for any integer* $0 \leq i \leq n$, $|f(\xi^i)| \leq 1$;

2. *for some real number* $1 \leq x_0 \leq \xi$, $|f'(x_0)| \geq c$.

Then $\deg f = \Omega(n)$.

Lemma 4.4.6 and Lemma 4.4.7 imply an $\Omega(\min\{m, n\})$ lower bound for distinguishing a matrix is full-rank or has nullity 1. The case $m = n$ was previously shown by van Apeldoorn and Gribling [115, Theorem 1]. We briefly explain the main ideas for completeness. By Lemma 4.4.6, for $d \in \{0, 1, \ldots, n-1\}$, $R(q^d) = Q(d)$ and $\deg(R) \leq 2t$. For distinguishing a full-rank matrix (i.e., $d = 0$) from a rank $n-1$ matrix (i.e, $d = 1$), we set $R(1) \geq 1 - \epsilon$ and $R(q) \leq \epsilon$. There exists $x_0 \in [1, q]$ such that $R'(x_0) \geq \frac{|R(q)-R(1)|}{q-1} \geq \frac{1-2\epsilon}{q-1}$. By Lemma 4.4.7, $t = \Omega(n)$ for $m \geq n$. For $m < n$, an $\Omega(m)$ lower bound follows from Theorem 4.3.1. Overall, this gives the following.

Theorem 4.4.3. *The bounded-error matrix-vector quantum query complexity of deciding if an* $m \times n$ *matrix over* $\mathbb{F}_q$ *is full-rank is* $\Omega(\min\{m, n\})$. *In particular,* $\Omega(\min\{m, n\})$ *queries are needed to decide whether the matrix is full-rank or has nullity* 1.

There is a trivial algorithm that learns an entire $m \times n$ matrix using $\min\{m, n\}$ queries. Thus the query complexity of computing the rank is $\Theta(\min\{m, n\})$.

Corollary 4.4.1. *The bounded-error query matrix-vector quantum complexity of computing the rank of an* $m \times n$ *matrix over* $\mathbb{F}_q$ *is* $\Theta(\min\{m, n\})$.

With the same argument, the quantum query complexity of computing the determinant of an $n \times n$ matrix over $\mathbb{F}_q$ is $\Theta(n)$. Moreover, the classical query complexity is $\Theta(n^2)$, implied by the $\Omega(n^2)$ lower bound for rank testing by Rashtchian, Woodruff, and Zhu [112, Theorem 3.3].

Corollary 4.4.2 (Determinant)**.** *The bounded-error classical and quantum query complexities of computing the determinant of an $n \times n$ matrix over $\mathbb{F}_q$ through matrix-vector products are $\Theta(n^2)$ and $\Theta(n)$, respectively.*

4.4.3 Solving linear systems

In this section, we consider the quantum query complexity of solving the linear system $Ax = b$ for $A \in \mathbb{F}_q^{n \times n}$ is $\Theta(n)$. Since there is an n-query algorithm learning the entire matrix using n matrix-vector queries, we focus on the lower bound.

Our proof is based on a randomized reduction from deciding whether a submatrix is full rank. For a square matrix A, let A^{ij} be the submatrix obtained by deleting the i^{th} row and the j^{th} column, and let A_{ij} denote the (i, j) element of A. The elements of A^{-1} can be computed as

$$(A^{-1})_{ij} = \frac{\det A^{ij}}{\det A}. \tag{4.27}$$

Given an invertible A, one can use a linear system solver to decide whether $(A^{-1})_{11}$ is non-zero, and thus decide if the minor A^{11} is full-rank.

In our reduction, to decide whether $M \in \mathbb{F}_q^{n \times n}$ is full-rank given access to matrix-vector products, we pad M with one extra random row and one extra random column, giving a matrix $A \in \mathbb{F}_q^{(n+1)\times(n+1)}$. We show that with sufficiently high probability, the padded matrix is full-rank. Thus, invoking a linear system solver with $b = e_1$, we learn whether $\det M = 0$. Thus the linear regression lower bound follows from Theorem 4.4.3.

Theorem 4.4.4. *The bounded-error matrix-vector quantum query complexity of solving an*

$n \times n$ linear system is $\Omega(n)$.

Proof. Assume toward contradiction that $\mathcal{A}$ is a t-query quantum algorithm for determining whether $(A^{-1})_{11}$ is non-zero for any invertible $A \in \mathbb{F}_q^{(n+1)\times(n+1)}$, succeeding with probability $p \geq 1/3$ with $t = o(n)$. We present a t-query algorithm for determining whether an $n \times n$ matrix is full-rank with probability $p(1 - 1/q)^2 \geq 1/12$.

Given access to $M \in \mathbb{F}_q^{n\times n}$, the algorithm first samples two random vectors $u, v \in \mathbb{F}_q^n$ and a random element $a \in \mathbb{F}_q$ to give the padded matrix

$$A = \begin{bmatrix} a & u^\top \\ v & M \end{bmatrix}. \tag{4.28}$$

The matrix-vector product $A(x_0, x^\top)^\top$ for $x_0 \in \mathbb{F}_q, x \in \mathbb{F}_q^n$ can be computed using one Mv query to Mx since

$$A \begin{bmatrix} x_0 \\ x \end{bmatrix} = \begin{bmatrix} a_0 + u^\top x \\ x_0 v + Mx \end{bmatrix}. \tag{4.29}$$

We show that with probability at least $(1 - 1/q)^2$, the matrix A is invertible (i.e., $\det A \neq 0$) given that $\operatorname{rank}(M) \geq n - 1$. If M is invertible, the submatrix $B = (v, M)$ is full-rank. If $\operatorname{rank}(M) = n - 1$, then without loss of generality, we consider the case that the first $n - 1$ rows of M are linearly independent, and the last row is a linear combination of the first $n - 1$ rows, since other cases can be handled accordingly by rearranging the rows.

We let

$$M = \begin{bmatrix} M' \\ w^\top \end{bmatrix}. \tag{4.30}$$

for an $(n-1) \times n$ matrix M' and an $n \times 1$ vector w. Since $w^\top$ is a linear combination of the first $n-1$ rows, we write $w^\top = c^\top M'$ for an $(n-1) \times 1$ vector c. Since M' is full-rank, the vector c satisfying $w^\top = c^\top M'$ is unique. Now write the vector

$$v = \begin{bmatrix} z \\ b \end{bmatrix} \tag{4.31}$$

for an $(n-1) \times 1$ matrix z and $b \in \mathbb{F}_q$. The matrix B is not full rank if and only if the last row is a linear combination of the first $n-1$ rows, i.e., $c^\top z = b$, since the first $n-1$ rows of B are linearly independent. Since v is a random vector with each element chosen independently, we have

$$\Pr[B \text{ is not full-rank}] = \Pr_{z,b}[c^\top z = b] = 1/q. \tag{4.32}$$

Thus with probability at least $1 - 1/q$ the matrix B is full-rank.

Conditioned on B being full-rank, the matrix A is not full-rank if and only if the vector $(a, u^\top)$ is in the vector space spanned by the rows of B. The number of vectors in the vector space is $q^{(n-1)}$. Thus

$$\Pr_{a,u,v}[A \text{ is not full-rank} \mid B \text{ is full-rank}] = 1/q. \tag{4.33}$$

Therefore with probability at least $1 - 1/q$, A is invertible. Conditioned on successfully simulating Mv queries of an invertible A, the algorithm $\mathcal{A}$ determines whether $(A^{-1})_{11}$ is nonzero with probability p. Thus the algorithm succeeds with probability at least $p(1 - 1/q)^2 \geq 1/12$ using $t = o(n)$ queries to M. By Theorem 4.4.3 we have a contradiction. □

The same proof idea shows that a lower bound for rank testing implies a lower bound for linear regression in the vMv model. Rashtchian, Woodruff, and Zhu show that the query complexity of distinguishing rank-n matrices from rank-$(n-1)$ matrices over $\mathbb{F}_q$ is $\Omega(n^2)$ [112, Theorem 3.3].

Corollary 4.4.3. *The bounded-error classical* vMv *query complexity of solving an* $n \times n$ *linear system over* $\mathbb{F}_q$ *is* $\Omega(n^2)$.

Proof. By the same idea as in the proof of Theorem 4.4.4, it suffices to show that one vMv query to the $(n+1) \times (n+1)$ matrix A in (4.28) can be simulated with one vMv query to the $n \times n$ matrix M. For any query x, y, we let $x = (x_0, x_1^\top)^\top$ and $y = (y_0, y_1^\top)^\top$ for $n \times 1$ matrices x_1, y_1. The product $y^\top A x$ can be computed using one vMv query to M since $y^\top A x = a y_0 x_0 + y_0 u^\top x_1 + y_1^\top v x_0 + y_1^\top M x_1$. Since no $o(n^2)$-query classical algorithm can distinguish rank-n matrices from rank-$(n-1)$ matrices [112, Theorem 3.3], the bounded-error query complexity of solving linear systems is $\Omega(n^2)$. □

4.4.4 Rank testing

In this section, we show a linear lower bound on distinguishing whether an $m \times n$ matrix M has $\mathrm{rank}(M) = n$ or $\mathrm{rank}(M) \leq n/2$, where $m \geq n$. First we show the following lemma using ideas from [121].

Lemma 4.4.8. *Let $\xi \geq 2$ and let n be an even integer. Then any polynomial f satisfying*

1. *$0 \leq f(\xi^i) \leq 1$ for $i \in \{0, 1, \ldots, n-1\}$ and*
2. *$f(1) \leq 1/3$ and $f(\xi^i) \geq 2/3$ for $i \in \{n/2, n/2+1, \ldots, n-1\}$*

has $\deg(f) = \Omega(n)$.

Proof. Let $d = \deg(f)$. Toward contradiction, we assume $d = o(n)$. For intervals $S_i := [\xi^i, \xi^{i+1})$, since $\deg(f'), \deg(f'') = o(n)$, there exists an index $a \in \{9n/10, \ldots, n-3, n-2\}$ such that none of the roots of f' and f'' has its real part in S_a. This implies that f' is monotonically increasing or decreasing in S_a, i.e., f is concave or convex. In each case, $f(\frac{\xi^a+\xi^{a+1}}{2}) \in [0, 1]$. If f is convex in S_a,

$$\left| f'\Big(\frac{\xi^a + \xi^{a+1}}{2}\Big) \right| \leq \frac{1}{\xi^{a+1} - \frac{\xi^{a+1}+\xi^a}{2}} = \frac{2}{\xi^{a+1} - \xi^a} \leq \frac{2}{\xi^a} \leq 2\xi^{-9n/10}. \tag{4.34}$$

If f is concave in S_a, reflecting about the x-axis gives the same bound.

By the second constraint, there exists $x_0 \in [1, \xi^{n/2}]$ such that

$$|f'(x_0)| \geq \frac{|f(\xi^{n/2}) - f(1)|}{\xi^{n/2} - 1} \geq \xi^{-n/2}/3. \tag{4.35}$$

Therefore

$$\left| \frac{f'(\frac{\xi^a+\xi^{a+1}}{2})}{f'(x_0)} \right| \leq 6\xi^{-2n/5} \leq \xi^{3-2n/5}. \tag{4.36}$$

On the other hand, since $\deg(f') = d-1$, denoting the roots $a_1, \ldots, a_{d-1} \in \mathbb{C}$, we write

$$f'(x) = \lambda \prod_{i=1}^{d-1} (x - a_i). \tag{4.37}$$

Thus

$$\left| \frac{f'(\frac{\xi^a + \xi^{a+1}}{2})}{f'(x_0)} \right| = \prod_{i=1}^{d-1} \left| \frac{\frac{\xi^a + \xi^{a+1}}{2} - a_i}{x_0 - a_i} \right| = \prod_{i=1}^{d-1} |g(a_i)|, \tag{4.38}$$

where

$$g(x) = \frac{x - \frac{\xi^a + \xi^{a+1}}{2}}{x - x_0}. \tag{4.39}$$

Our goal is to show that for each i, $|g(a_i)| \geq \frac{1}{2\xi}$. Recall that for each i, $\mathrm{Re}(a_i) \notin S_a$. Also for real $x \notin S_a$, $x \geq x_0$, we have $|g(x)| \geq \frac{\xi - 1}{2\xi} \geq \frac{1}{2\xi}$. For real roots, $|g(a_i)| \geq \frac{1}{2\xi}$. Now we consider the case where $a_i = \alpha + \beta i$ for $\beta \neq 0$, giving

$$|g(\alpha + \beta i)|^2 = \frac{(\alpha - \frac{\xi^a + \xi^{a+1}}{2})^2 + \beta^2}{(\alpha - x_0)^2 + \beta^2}. \tag{4.40}$$

If $(\alpha - \frac{\xi^a + \xi^{a+1}}{2})^2 \geq (\alpha - x_0)^2$, then $|g(\alpha + \beta i)| \geq 1$. Otherwise,

$$|g(\alpha + \beta i)| \geq \left| \frac{\alpha - \frac{\xi^a + \xi^{a+1}}{2}}{\alpha - x_0} \right| \geq \frac{1}{2\xi}. \tag{4.41}$$

We have shown that $|g(a_i)| \geq \frac{1}{2\xi}$ for every root a_i. Now we have

$$\left| \frac{f'(\frac{\xi^a + \xi^{a+1}}{2})}{f'(x_0)} \right| = \prod_{i=1}^{d-1} |g(a_i)| \geq (2\xi)^{-d+1} \geq \xi^{2-2d}. \tag{4.42}$$

Thus by (4.36), we have $\xi^{3-2n/5} \geq \xi^{2-2d}$ and conclude $d \geq n/5 - 1/2 = \Omega(n)$—a contradiction.

□

Lemma 4.4.6 and Lemma 4.4.8 imply the following theorem.

Theorem 4.4.5. *The bounded-error matrix-vector quantum query complexity of determining whether a matrix* $M \in \mathbb{F}_q^{m \times n}$ *has* $\operatorname{rank}(M) = n$ *or* $\operatorname{rank}(M) \leq n/2$ *is* $\Omega(n)$.

Problem	Classical Mv	Classical vMv	Quantum (this paper)
Trace	$O(n), \Omega(n/\log n)$ for matrix with entries in $\{0,1,\ldots,n^3\}$ & queries with entries in $\{0,1,\ldots,n^C\}$, $C \in \mathbb{N}$ [32]; $\Theta(n)$ over $\mathbb{F}_q$ (Theorem 4.4.2)	$O(n), \Omega(n/\log n)$ for matrix with entries in $\{0,1,\ldots,n^3\}$ & queries with entries in $\{0,1,\ldots,n^C\}$, $C \in \mathbb{N}$ [112]; $\Theta(n)$ over $\mathbb{F}_q$ (Theorem 4.4.2)	$\Theta(n)$ over $\mathbb{F}_q$ (Theorem 4.4.2)
Linear regression	$\Theta(n)$ over $\mathbb{R}$ [33]; $\Theta(n)$ over $\mathbb{F}_q$ (Theorem 4.4.4)	$\Theta(n^2)$ over $\mathbb{F}_q$ (Corollary 4.4.3)	$\Theta(n)$ over $\mathbb{F}_q$ (Theorem 4.4.4)
Rank testing	$k+1$ to distinguish rank $\leq k$ from $k' > k$ over $\mathbb{R}$ [32]; $\Theta(n)$ over $\mathbb{F}_q$ (Theorem 4.4.5)	$\Omega(k^2)$ to distinguish rank k from $k+1$ over $\mathbb{F}_q$ [112]; $\Omega(n^{2-O(\epsilon)})$ for non-adaptive $(1 \pm \epsilon)$-approximation over $\mathbb{R}$ [112]	$\Theta(\min\{m,n\})$ to distinguish rank $\min\{m,n\}$ from $\leq \frac{1}{2}\min\{m,n\}$ over $\mathbb{F}_q$ (Theorem 4.4.5)
Two identical columns	$O(n/m)$, $m = \Omega(\log(n/\epsilon))$ over $\mathbb{F}_2$ [32]	$O(n\log n), \Omega(n)$ over $\mathbb{F}_2$ [112]	$O(\log n)$ over $\mathbb{F}_2$ (Corollary 4.3.2)
Two identical rows	$O(\log m)$ over $\mathbb{F}_2$ [32]	$O(n\log n), \Omega(n)$ over $\mathbb{F}_2$ [112]	$O(\log m)$ over $\mathbb{F}_2$ (Corollary 4.3.2)
Majority of columns	$\Omega(n/\log n)$ for binary matrices over $\mathbb{R}$ [32]	$\Theta(n^2)$ over $\mathbb{F}_2$ [112]	$O(1)$ for binary matrices over $\mathbb{R}$ (Corollary 4.3.3)
Majority of rows	$O(1)$ for binary matrices over $\mathbb{R}$ [32]	$\Theta(n^2)$ over $\mathbb{F}_2$ [112]	$O(1)$ for binary matrices over $\mathbb{R}$ (Corollary 4.3.3)
Parity of columns	$\Theta(n)$ over $\mathbb{F}_2$ [32]	$\Theta(n)$ over $\mathbb{F}_2$ (Lemma 4.3.1)	$O(1)$ over $\mathbb{F}_2$ (Corollary 4.3.1)
Parity of rows	$O(1)$ over $\mathbb{F}_2$ [32]	$\Theta(m)$ over $\mathbb{F}_2$ (Lemma 4.3.1)	$O(1)$ over $\mathbb{F}_2$ (Corollary 4.3.1)

Table 4.1: Comparison of classical and quantum query complexities with matrix-vector (Mv) and vector-matrix-vector (vMv) product oracles for an $m \times n$ matrix. For trace and linear regression, $m = n$. Known query complexities over $\mathbb{R}$ and $\mathbb{F}_q$ are included for completeness; results over different fields are incomparable in general.

Chapter 5: Quantitative robustness analysis

5.1 Introduction

In this chapter, we apply formal methods to reason about the quality of error-affected programs, in the following steps. First, we extend the syntax and semantics to include noisy operations. In particular, we modify the quantum gate applications to express noisy operations Φ (a superoperator) occurring with probability p. This approach permits modeling any local noise occurring during the execution of a quantum program, which is the standard noise model considered in the study of quantum error correction and fault-tolerant quantum computation [128].

Secondly, we define the notion of quantum robustness. In particular, we say a noisy program is ϵ-robust under (Q, λ) if it computes a quantum state at most ϵ distance away from that of its ideal equivalent when starting both from states satisfying quantum predicate Q to degree λ (see Section 5.2.3 for definition).

Thirdly, we define a logic for reasoning about quantum robustness, with judgment in this form $(Q, \lambda) \vdash \tilde{P} \leq \epsilon$. The judgment states that for any input state satisfying the predicate Q to degree λ, the distance between the denotation of $\tilde{P}$ (a superoperator) and that of its ideal equivalent P is at most ϵ. Imposing the constraint on the input state satisfying Q to degree λ allows us to leverage prior knowledge about input states to obtain more accurate

bounds.

We prove our logic is sound. A particular challenge is the rule for loops, owing to the termination problem first studied by Li and Ying [129]. In particular, if the loop body generates some error and the loop does not terminate, it is hard to prove any non-trivial bound on the final accumulated error. To avoid this difficulty in the setting of approximate computing, Carbin, Misailovic and Rinard [35, 130] simply assume that the loop will terminate within a bounded number of iterations or a trivial upper bound will be applied. To capture more complicated cases, we introduce a concept called the (a, n)-boundedness of the loop. Intuitively, a loop is (a, n)-bounded if, for every input state, after n iterations it is guaranteed that with probability at least $1 - a$ it has exited the loop. The probability is due to the quantum measurement in the loop guard. It is easy to see that (a, n)-boundedness implies the termination of the loop. Pleasantly, (a, n)-bounding the ideal loop is sufficient to reason about a noisy loop with any error model Φ in the loop body.

5.2 Quantum programs

5.2.1 Syntax

Define *Var* as the set of quantum variables. We use the symbol q as a metavariable ranging over quantum variables and define a *quantum register* $\overline{q}$ to be a finite set of distinct variables. For each $q \in Var$, its state space is denoted by $\mathcal{H}_q$. The quantum register $\overline{q}$ is associated with the Hilbert space $\mathcal{H}_{\overline{q}} = \bigotimes_{q \in \overline{q}} \mathcal{H}_q$. If $type(q) = \mathbf{Bool}$ then $\mathcal{H}_q$ is the two-dimensional Hilbert space with basis $\{|0\rangle, |1\rangle\}$. If $type(q) = \mathbf{Int}$ then $\mathcal{H}_q$ is the Hilbert space

with basis $\{|n\rangle : n \in \mathbb{Z}\}$. The syntax of a *quantum* ***while*** *program* P is defined as follows.

$$\begin{aligned} P \;::=\; & \textbf{skip} \mid q := |0\rangle \mid \overline{q} := U[\overline{q}] \mid P_1; P_2 \mid \\ & \textbf{case } M[\overline{q}] = \overline{m \to P_m} \textbf{ end} \mid \textbf{while } M[\overline{q}] = 1 \textbf{ do } P_1 \textbf{ done} \end{aligned} \tag{5.1}$$

The language constructs above are similar to their classical counterparts:

1. **skip** does nothing.

2. $q := |0\rangle$ sets quantum variable q to the basis state $|0\rangle$.

3. $\overline{q} := U[\overline{q}]$ applies the unitary U to the qubits in $\overline{q}$.

4. Sequencing $P_1; P_2$ has the same behavior as its classical counterpart.

5. **case** $M[\overline{q}] = \overline{m \to P_m}$ **end** performs the measurement $M = \{M_m\}$ on the qubits in $\overline{q}$, and executes program P_m if the outcome of the measurement is m. The bar over $\overline{m \to P_m}$ indicates that there may be one or more repetitions of this expression.[1]

6. **while** $M[\overline{q}] = 1$ **do** P_1 **done** performs the measurement $M = \{M_0, M_1\}$ on the qubits in $\overline{q}$, and executes P_1 if measurement produces the outcome corresponding to M_1 or terminates if measurement produces the outcome corresponding to M_0.

We highlight two differences between quantum and classical while languages:

- Qubits may only be initialized to the basis state $|0\rangle$. There is no quantum analogue for initialization to any expression (i.e. $x := e$) because of the no-cloning theorem of

[1] Our syntax for conditional/case statements differs from that presented by [131] to make it more clear that there are multiple programs P_m.

quantum states. Any state $|\psi\rangle \in \mathcal{H}_q$, however, can be constructed by applying some unitary U to $|0\rangle$.[2]

- Evaluating the guard of a case statement or loop, which performs a measurement, potentially disturbs the state of the system.

We now present an example program written in the quantum **while**-language syntax. The *quantum walk* [132] is a widely considered example in quantum programming, quantum algorithms, and quantum simulation literature [133, 134]. Here we consider a quantum walk on a circle with n points. We let the initial position of the walker be 0, and say that the program halts if and only if the walker arrives at position 1.

Example 5.2.1 (Quantum Walk)**.** *Define the coin (or "direction") space $\mathcal{H}_c$ to be the 2-dimensional Hilbert space with orthonormal basis states $|L\rangle$ and $|R\rangle$, for* Left *and* Right *respectively. Define the position space $\mathcal{H}_p$ to be the n-dimensional Hilbert space with orthonormal basis states $|0\rangle, |1\rangle, ..., |n-1\rangle$, where vector $|i\rangle$ represents position i for $0 \leq i < n$. Now the state space of the walk is $\mathcal{H} = \mathcal{H}_c \otimes \mathcal{H}_p$ and the initial state is $|L\rangle|0\rangle$. In each step of the walk:*

1. *Measure the position of the system to determine whether the walker has reached position 1. If the walker has reached position 1, the walk terminates. Otherwise, it continues. We use the measurement $M = \{|1\rangle\langle 1|, \sum_{i\neq 1}|i\rangle\langle i|\}$.*

2. *Apply the "coin-tossing" operator H to the coin space $\mathcal{H}_c$.*

[2] In our examples, we may write $q := |\psi\rangle$ for some fixed basis state $|\psi\rangle$. What we mean in this case is $q := |0\rangle; q := U[q]$ where U is the unitary operation that transforms $|0\rangle$ into $|\psi\rangle$.

3. *Perform the shift operator* S *defined by* $S|L,i\rangle = |L, i-1(mod\ n)\rangle, \quad S|R,i\rangle = |R, i+1(mod\ n)\rangle$ *for* $i = 0, 1, ..., n-1$ *to the space* $\mathcal{H}$. *The* S *operator can be written as*

$$S = \sum_{i=0}^{n-1} |L\rangle\langle L| \otimes |i-1(mod\ n)\rangle\langle i| + \sum_{i=0}^{n-1} |R\rangle\langle R| \otimes |i+1(mod\ n)\rangle\langle i|.$$

In this algorithm, the walker takes one step left or one step right corresponding to the coin flip result $|L\rangle$ *or* $|R\rangle$. *However, unlike the classical case, the result of the coin flip may be a superposition of* $|L\rangle$ *and* $|R\rangle$, *allowing the walker to take a step to the left and right* simultaneously. *This quantum walk can be described by the following program*

$$QW_n \equiv p := |0\rangle; c := |L\rangle; \textbf{while } M[p] = 1 \textbf{ do } c := H[c]; c, p := S[c, p] \textbf{ done}. \tag{5.2}$$

5.2.2 Denotational semantics

The *denotational* semantics of a quantum **while** program is given in Figure 5.1. It defines $[\![P]\!]$ as a superoperator that acts on $\rho \in \mathcal{H}_{Var}$ [131]. The semantics of each term

$$\begin{aligned}
[\![\textbf{skip}]\!]\rho &= \rho \\
[\![q := |0\rangle]\!]\rho &= \begin{cases} \mathcal{E}^{\text{bool}}_{q\to 0}(\rho) & \text{if } type(q) = \textbf{Bool} \\ \mathcal{E}^{\text{int}}_{q\to 0}(\rho) & \text{if } type(q) = \textbf{Int} \end{cases} \\
[\![\overline{q} := U[\overline{q}]]\!]\rho &= U\rho U^\dagger \\
[\![P_1; P_2]\!]\rho &= [\![P_2]\!]([\![P_1]\!]\rho) \\
[\![\textbf{case } M[\overline{q}] = \overline{m \to P_m} \textbf{ end}]\!]\rho &= \textstyle\sum_m [\![P_m]\!](M_m \rho M_m^\dagger) \\
[\![\textbf{while } M[\overline{q}] = 1 \textbf{ do } P_1 \textbf{ done}]\!]\rho &= \bigsqcup_{k=0}^{\infty} [\![\textbf{while}^{(k)}]\!]\rho
\end{aligned}$$

Figure 5.1: The denotational semantics of quantum **while** programs.

is given compositionally. We write $\mathbf{while}^{(k)}$ for the k^{th} syntactic approximation (i.e., unrolling) of **while** and $\bigsqcup$ for the *least upper bound* operator in the complete partial order generated by Löwner comparison. The superoperators $\mathcal{E}^{\text{bool}}_{q\to 0}$ and $\mathcal{E}^{\text{int}}_{q\to 0}$, which initialize the variable q in ρ to the zero state, is defined as $\mathcal{E}^{\text{bool}}_{q\to 0}(\rho) = \sum_{b=0}^{1} |0\rangle_q\langle b|\rho|b\rangle_q\langle 0|$ and $\mathcal{E}^{\text{int}}_{q\to 0}(\rho) = \sum_{n=-\infty}^{\infty} |0\rangle_q\langle n|\rho|n\rangle_q\langle 0|$. For more detail on the semantics of loops, we refer the reader to [131, 135].

The semantics presented so far assume that no noise will occur during computation. In Section 5.3, we extend the semantics to include possible errors that may occur during unitary application.

5.2.3 Quantum predicates and Hoare logic

A *quantum predicate* is a Hermitian operator M such that $0 \sqsubseteq M \sqsubseteq I$ [136]. For a predicate M and state ρ, $\text{tr}(M\rho)$ is the expectation of the truth value of predicate M in state ρ. Restricting M to be between 0 and I ensures that $0 \leq \text{tr}(M\rho) \leq 1$ for any $\rho \in \mathcal{D}(\mathcal{H})$. A quantum state ρ satisfying predicate M to degree λ if $\text{tr}(M\rho) = \lambda$.

The identity matrix corresponds to the **true** predicate because for any density operator ρ, $\text{tr}(I\rho) = 1$. The zero matrix corresponds to the **false** predicate because for any density operator ρ, $\text{tr}(0\rho) = 0$. For example, $|0\rangle\langle 0|$ is the predicate that says that a state is in the subspace spanned by $|0\rangle$. The density operator ρ_0 corresponding to the state $|0\rangle$ is such that $\text{tr}(|0\rangle\langle 0|\rho_0) = 1$, and the density operator ρ_1 corresponding to the state $\sqrt{1/3}|0\rangle + \sqrt{2/3}|1\rangle$ is such that $\text{tr}(|0\rangle\langle 0|\rho_1) = \frac{1}{3}$.

Ying [131, 135] uses quantum predicates as the basis for defining quantum preconditions

and postconditions in his quantum Hoare logic. Let M and N be quantum predicates and let P be a quantum **while** program. Then M is a precondition of N with respect to P, written $\{M\}P\{N\}$, if

$$\forall \rho.\ \mathrm{tr}(M\rho) \leq \mathrm{tr}(N[\![P]\!]\rho). \tag{5.3}$$

This inequality can be seen as the quantum analogue of the following statement: if state ρ satisfies predicate M, then after applying the program P the resulting state will satisfy predicate N. If we include an auxiliary space $\mathcal{A}$, then the equivalent statement is

$$\forall \rho.\ \mathrm{tr}((M \otimes I_{\mathcal{A}})\rho) \leq \mathrm{tr}((N \otimes I_{\mathcal{A}})([\![P]\!] \otimes I_{\mathcal{A}})(\rho)). \tag{5.4}$$

5.3 Noisy quantum programs

In this section, we present the syntax and semantics for the quantum **while**-language with noise, as an extension of the quantum **while**-language. Our syntax allows one to explicitly encode any error model that describes local noise during the execution of a quantum program.

5.3.1 Noise in quantum computation

Here we briefly discuss how noise is modeled in the study of quantum error correction and fault-tolerant quantum computation [128], which in turn comes from the noise model in quantum physical experiments. It is a convention to only consider *local noise* rather than correlated noise, because benign white noise is more likely than *adversarial noise* in actual quantum devices. A few types of natural local noise arise in realistic quantum systems, which

generalize classical bit-flip errors, including:

- The *bit flip* noise flips the state with probability p, and can be represented by

$$\Phi_{p,\text{bit}} = (1-p)I \circ I + pX \circ X. \tag{5.5}$$

- The *phase flip* noise flips the phase with probability p, and can be represented by

$$\Phi_{p,\text{phase}} = (1-p)I \circ I + pZ \circ Z. \tag{5.6}$$

Other types of noise include depolarization, amplitude damping, and phase damping [137].

This model of noise is used by experimental physicists for building and benchmarking quantum devices in both academia and industry [138]. Noisy information of specific quantum devices can also be publicly available (e.g., the IBM Q-experience[3]).

5.3.2 Syntax

The syntax of a noisy quantum program $\widetilde{P}$ is defined as follows

$$\begin{aligned} \widetilde{P} \quad ::= \quad & \textbf{skip} \mid q := |0\rangle \mid \overline{q} :\cong_{p,\Phi} U[\overline{q}] \mid \widetilde{P_1}; \widetilde{P_2} \mid \\ & \textbf{case } M[\overline{q}] = \overline{m \to \widetilde{P_m}} \textbf{ end} \mid \textbf{while } M[\overline{q}] = 1 \textbf{ do } \widetilde{P_1} \textbf{ done} \end{aligned} \tag{5.7}$$

This syntax is identical to that of the standard quantum while language described in Section 5.2.1, except that we have annotated the unitary application construct with an error

[3] `https://www.research.ibm.com/ibm-q/technology/devices/`.

probability p and an error model Φ, which is the superoperator of the noisy operation. The statement $\overline{q} :\cong_{p,\Phi} U[\overline{q}]$ will apply the correct operation U on $\overline{q}$ with probability $1-p$ and will apply the noisy (or erroneous) operation Φ on $\overline{q}$ with probability p. The nature of Φ will depend on the underlying hardware, and is a parameter to our language.

For any noisy program $\widetilde{P}$, its corresponding *ideal* program can be obtained by simply replacing any noisy unitary operations by their ideal versions (i.e., we ignore p and Φ). We write $\texttt{ideal}(\widetilde{P})$ for this program, or simply P when there is no ambiguity.

We remark that noisy unitary operations are already expressive enough to capture many types of noise. First, any noise can depend on the quantum state of the system by the nature of modeling it as a quantum operation Φ. Second, noisy initialization can be modeled as initialization followed by application of a noisy identity operation, and noisy measurement can be modeled as application of a noisy identity operation followed by measurement. Third, errors that occur between applications of subsequent unitaries can also be modeled by noisy identity operations.

5.3.3 Semantics

$$[\![\overline{q} :\cong_{p,\Phi} U[\overline{q}]]\!]\rho = (1-p)U\rho U^\dagger + p\,\Phi(\rho)$$

Figure 5.2: Denotation of noisy unitary operation.

The denotational semantics of noisy quantum **while** programs are also identical to those of the standard quantum **while** programs, except that the rules related to unitary

application now include an error term, as shown in Figure 5.2. Note that we do not require p and Φ to be the same in every instance of a unitary application. They may depend on the type of unitary being applied, or on other features of the program such as the number of operations performed so far. We use *explicit* characterizations of the noise given by Φ to enable us to argue about the effect of error-correcting gadgets explicitly written in the program. If we only considered error probability (like [35]) then we could only reason about error as accumulating throughout the program, and we would not be able to show that error-correcting gadgets reduce noise by cancelling previous errors.

5.4 Quantum robustness

This section defines a notion of *quantum robustness*, which bounds the distance between the output of a noisy execution of a program and the ideal (noise-free) execution of the same program. We first introduce distance measures in quantum information, and then present the semantic definition of robustness and define a logic for reasoning about it, which we prove sound.

5.4.1 Definition

To capture how noise (error) impacts the execution of quantum program $\widetilde{P}$, we want to compare $[\![\widetilde{P}]\!]$ and $[\![P]\!]$, which are superoperators representing the execution of quantum program P with and without noise respectively. A natural candidate is to use the aforementioned diamond norm to measure the distance between $[\![\widetilde{P}]\!]$ and $[\![P]\!]$. To account for prior knowledge of the input state, we extend the definition of the diamond norm to consider only

input states that satisfy predicate Q to degree at least λ. More explicitly, we have

Definition 5.4.1 ((Q,λ)-diamond norm)**.** *Given superoperators $\mathcal{E}$, $\mathcal{E}'$, quantum predicate Q over $\mathcal{H}$, and $0 \leq \lambda \leq 1$, the (Q,λ)-diamond norm between $\mathcal{E}$ and $\mathcal{E}'$, denoted $\|\mathcal{E}-\mathcal{E}'\|_{Q,\lambda}$, is defined by*

$$\|\mathcal{E}-\mathcal{E}'\|_{Q,\lambda} \equiv \max_{\rho\in\mathcal{D}(\mathcal{H}\otimes\mathcal{A}) \,:\, \mathrm{tr}(\rho)=1,\ \mathrm{tr}(Q\rho)\geq\lambda} \mathrm{T}(\mathcal{E}\otimes I_{\mathcal{A}}(\rho),\ \mathcal{E}'\otimes I_{\mathcal{A}}(\rho)), \tag{5.8}$$

where $\mathcal{A}$ is any auxiliary space.

We remark that $\mathcal{A}$ can be assumed to be $\mathcal{H}$ without loss of generality due to a similar reason for the original diamond norm (see, for example, [139, Chapter 3]).

We argue that the (Q,λ)-diamond norm is a seminorm. Intuitively, this is conceivable since we only restrict the input state from all density operators to a convex subset satisfying $\mathrm{tr}(Q\rho) \geq \lambda$. Note that, by definition, $\|\cdot\|_\diamond \equiv \|\cdot\|_{I,\lambda}$ for any $0 \leq \lambda \leq 1$.

Theorem 5.4.1. *The (Q,λ)-diamond norm is a seminorm.*

Proof. Recall that $0 \leq \mathrm{T}(\mathcal{E}\otimes I_{\mathcal{A}}(\rho),\ \mathcal{E}'\otimes I_{\mathcal{A}}(\rho)) \leq 1$ for any quantum state ρ, so $\|\mathcal{E}-\mathcal{E}'\|_{Q,\lambda} \geq 0$.

1. *Positive scalability:* This property is inherited directly from the diamond norm. One may also observe that for any $\alpha \in \mathbb{C}$,

$$\begin{aligned}\mathrm{T}(\mathcal{E}\otimes I_{\mathcal{A}}(\alpha\rho),\ \mathcal{E}'\otimes I_{\mathcal{A}}(\alpha\rho)) &= \mathrm{T}(\alpha(\mathcal{E}\otimes I_{\mathcal{A}}(\rho)),\ \alpha(\mathcal{E}'\otimes I_{\mathcal{A}}(\rho))) \\ &= \alpha\mathrm{T}(\mathcal{E}\otimes I_{\mathcal{A}}(\rho),\ \mathcal{E}'\otimes I_{\mathcal{A}}(\rho)).\end{aligned} \tag{5.9}$$

2. *Triangle Inequality:* For quantum superoperators $\mathbb{E}, \mathbb{E}', \mathcal{F}, \mathcal{F}'$, we have

$$\|(\mathcal{E} - \mathcal{E}') + (\mathcal{F} - \mathcal{F}')\|_{Q,\lambda} = \max_{\substack{\rho \in \mathcal{D}(\mathcal{H}\otimes\mathcal{A}) \,:\\ \mathrm{tr}(\rho)=1,\ \mathrm{tr}(Q\rho)\geq\lambda}} \mathrm{T}((\mathcal{E}+\mathcal{F})\otimes I_{\mathcal{A}}(\rho),\ (\mathcal{E}'+\mathcal{F}')\otimes I_{\mathcal{A}}(\rho)). \tag{5.10}$$

Suppose the maximal value on the right hand side is attained at a state $\rho^* \in \mathcal{D}(\mathcal{H}\otimes\mathcal{A})$. Note that, by definition, this requires ρ^* to satisfy $\mathrm{tr}(\rho^*) = 1, \mathrm{tr}(Q\rho^*) \geq \lambda$. Then,

$$\begin{aligned}
&\|(\mathcal{E} - \mathcal{E}') + (\mathcal{F} - \mathcal{F}')\|_{Q,\lambda} \\
&\quad = \max_{0\sqsubseteq P\sqsubseteq I} \mathrm{tr}(P((\mathcal{E}-\mathcal{E}')\otimes I_{\mathcal{A}}(\rho^*) + (\mathcal{F}-\mathcal{F}')\otimes I_{\mathcal{A}}(\rho^*))) \\
&\quad \leq \max_{0\sqsubseteq P_1\sqsubseteq I} \mathrm{tr}(P_1((\mathcal{E}-\mathcal{E}')\otimes I_{\mathcal{A}}(\rho^*))) + \max_{0\sqsubseteq P_2\sqsubseteq I} \mathrm{tr}(P_2(\mathcal{F}-\mathcal{F}')\otimes I_{\mathcal{A}}(\rho^*))) \\
&\quad \leq \max_{\substack{\rho_1 \in \mathcal{D}(\mathcal{H}\otimes\mathcal{A}) \,:\\ \mathrm{tr}(\rho_1)=1,\ \mathrm{tr}(Q\rho_1)\geq\lambda}} \Big(\max_{0\sqsubseteq P_1\sqsubseteq I} \mathrm{tr}(P_1((\mathcal{E}-\mathcal{E}')\otimes I_{\mathcal{A}}(\rho_1))) \Big) \\
&\qquad + \max_{\substack{\rho_2 \in \mathcal{D}(\mathcal{H}\otimes\mathcal{A}) \,:\\ \mathrm{tr}(\rho_2)=1,\ \mathrm{tr}(Q\rho_2)\geq\lambda}} \Big(\max_{0\sqsubseteq P_2\sqsubseteq I} \mathrm{tr}(P_2((\mathcal{F}-\mathcal{F}')\otimes I_{\mathcal{A}}(\rho_2))) \Big) \\
&\quad = \max_{\substack{\rho_1 \in \mathcal{D}(\mathcal{H}\otimes\mathcal{A}) \,:\\ \mathrm{tr}(\rho_1)=1,\ \mathrm{tr}(Q\rho_1)\geq\lambda}} \mathrm{T}(\mathcal{E}\otimes I_{\mathcal{A}}(\rho_1),\ \mathcal{E}'\otimes I_{\mathcal{A}}(\rho_1)) \\
&\qquad + \max_{\substack{\rho_2 \in \mathcal{D}(\mathcal{H}\otimes I_{\mathcal{A}}) \,:\\ \mathrm{tr}(\rho_2)=1,\ \mathrm{tr}(Q\rho_2)\geq\lambda}} \mathrm{T}(\mathcal{F}\otimes I_{\mathcal{A}}(\rho_2),\ \mathcal{F}'\otimes I_{\mathcal{A}}(\rho_2)) \\
&\quad = \|\mathcal{E}-\mathcal{E}'\|_{Q,\lambda} + \|\mathcal{F}-\mathcal{F}'\|_{Q,\lambda}.
\end{aligned} \tag{5.11}$$

□

Let $\mathcal{E}$ and $\mathcal{E}'$ be superoperators over $\mathcal{H}$. By extending [140], we show that $\|\mathcal{E} - \mathcal{E}'\|_{Q,\lambda}$

can be efficiently computed by the following semidefinite program (SDP):

$$\begin{aligned} \max \quad & \operatorname{tr}(J(\Phi)W) \\ \text{s.t.} \quad & W \leq I_{\mathcal{H}} \otimes \rho, \ \ \operatorname{tr}(Q\rho) \geq \lambda, \\ & \rho \in \mathcal{D}(\mathcal{H}), \ W \text{ is a positive semidefinite operator over } \mathcal{H} \otimes \mathcal{H}, \end{aligned} \tag{5.12}$$

where $\Phi = \mathcal{E} - \mathcal{E}'$ and $J(\Phi)$ is the *Choi-Jamiolkowski* representation of Φ. Note that the above SDP is identical to the SDP used in [140, Section 4] to compute $\|\mathcal{E} - \mathcal{E}'\|_\diamond$, except that we have added the additional constraint $\operatorname{tr}(Q\rho) \geq \lambda$ to capture the requirement on input states. The correctness of the above SDP then basically follows from the analysis of [140] and the definition of (Q, λ)-diamond norm.

The standard diamond norm and (Q, λ)-diamond norm can be significantly different for the same pair of superoperators $\mathcal{E}, \mathcal{E}'$. For example, consider $\mathcal{E} = H \circ H$ and $\mathcal{E}' = HZ \circ ZH$. We can show[4] that $\|\mathcal{E} - \mathcal{E}'\|_\diamond = 1$, whereas $\|\mathcal{E} - \mathcal{E}'\|_{|0\rangle\langle 0|, \frac{3}{4}} = \sqrt{3}/2$. Thus, the (Q, λ)-diamond norm can help us leverage prior knowledge about input states to obtain more accurate bounds.

Using the (Q, λ)-diamond norm, we define a notion of quantum robustness as follows.

Definition 5.4.2 (Quantum Robustness)**.** *The noisy program $\widetilde{P}$ (over $\mathcal{H}$, having ideal pro-*

[4]For the normal diamond norm, consider the input state to be $\rho = |+\rangle\langle +| \otimes \sigma$ for some ancilla state σ. It is easy to see that $\mathcal{E}(\rho) = |0\rangle$ and $\mathcal{E}'(\rho) = |1\rangle$, which are perfectly distinguishable. For the $(|0\rangle\langle 0|, \frac{3}{4})$-diamond norm, without loss of generality, consider any input state $|\psi\rangle = \cos\theta|0\rangle|\psi_0\rangle + \sin\theta|1\rangle|\psi_1\rangle$ where $\theta \in [0, \frac{\pi}{2}]$. Requiring $|\psi\rangle$ to satisfy $|0\rangle\langle 0|$ to degree at least $\frac{3}{4}$, we have $\cos^2\theta \geq \frac{3}{4}$ and therefore $\theta \in [0, \frac{\pi}{6}]$. Simple calculation gives $H|\psi\rangle = \cos\theta|+\rangle|\psi_0\rangle + \sin\theta|-\rangle|\psi_1\rangle$ and $HZ|\psi\rangle = \cos\theta|+\rangle|\psi_0\rangle - \sin\theta|-\rangle|\psi_1\rangle$. The projector that maximally distinguishes these states is $|0\rangle\langle 0| \otimes I$, so we have

$$\|\mathcal{E} - \mathcal{E}'\|_{|0\rangle\langle 0|, 3/4} = \frac{1}{2}\big((\cos\theta + \sin\theta)^2 - (\cos\theta - \sin\theta)^2\big) = \sin 2\theta \leq \frac{\sqrt{3}}{2},$$

the equality of which holds when $\theta = \pi/6$.

gram $P = \texttt{ideal}(\widetilde{P})$*) is* ϵ-robust under (Q, λ) *if and only if*

$$\left\| [\![\widetilde{P}]\!] - [\![P]\!] \right\|_{Q,\lambda} \leq \epsilon. \tag{5.13}$$

Here, Q *is a quantum predicate over* $\mathcal{H}$ *and* $0 \leq \lambda, \epsilon \leq 1$*.*

By the definition of the (Q, λ)*-diamond norm and the trace distance, (5.13) can be equivalently stated as the following.*

$$\forall \rho \in \mathcal{D}(\mathcal{H} \otimes \mathcal{H}),\ \ \mathrm{tr}(Q\rho) \geq \lambda \,\mathrm{tr}(\rho) \Rightarrow \frac{1}{2} \left\| [\![\widetilde{P}]\!] \otimes I_{\mathcal{H}}(\rho) - [\![P]\!] \otimes I_{\mathcal{H}}(\rho) \right\|_1 \leq \epsilon \,\mathrm{tr}(\rho). \tag{5.14}$$

Since ϵ measures the distance between $[\![\widetilde{P}]\!]$ and $[\![P]\!]$, the *smaller* ϵ is, the *closer* the noisy program $\widetilde{P}$ is to the ideal program P. One could think of ϵ as measuring both the probability that noise can happen and intensity of that noise. When the noise is strong, a noisy program $\widetilde{P}$ being ϵ-robust implies that the probability of noise is at most ϵ. When the noise is weak, it could occur with greater probability, but its effect will be much smaller.

Intuitively, the use of precondition Q can help us obtain more accurate bounds. For example, one can use Q to characterize prior information about the input state due to the nature of underlying physical systems. Even without any prior knowledge about the input state, preconditions can still be leveraged for different branches of the program in case statements and loops.

5.4.2 Logic

A program's robustness can be proved by working out the (denotational) semantics of programs P and $\widetilde{P}$ and applying Definition 5.4.2 directly. However, the computation is difficult in general, since the complexity of computing the semantics of a quantum program scales exponentially in the size of the program. As an alternative, we present a logic for proving judgments of the form $(Q, \lambda) \vdash \widetilde{P} \leq \epsilon$, meaning that program $\widetilde{P}$ is ϵ-robust under (Q, λ). In Section 5.4.3 we prove the logic is sound.

$$\frac{}{(Q,\lambda) \vdash \mathbf{skip} \leq 0}\ (\text{Skip}) \qquad \frac{}{(Q,\lambda) \vdash (q := |0\rangle) \leq 0}\ (\text{Init})$$

$$\frac{\left\|U \circ U^\dagger - \Phi\right\|_{Q,\lambda} \leq \epsilon}{(Q,\lambda) \vdash (\overline{q} :\cong_{p,\Phi} U[\overline{q}]) \leq p\epsilon}\ (\text{Unitary})$$

$$\frac{(Q',\lambda') \vdash \widetilde{P} \leq \epsilon' \qquad \epsilon' \leq \epsilon \qquad Q \sqsubseteq Q' \qquad \lambda' \leq \lambda}{(Q,\lambda) \vdash \widetilde{P} \leq \epsilon}\ (\text{Weaken})$$

$$\frac{(Q/\delta,\lambda/\delta) \vdash \widetilde{P} \leq \epsilon \qquad 0 \sqsubseteq Q, Q/\delta \sqsubseteq I \qquad 0 \leq \lambda, \lambda/\delta \leq 1}{(Q,\lambda) \vdash \widetilde{P} \leq \epsilon}\ (\text{Rescale})$$

$$\frac{(Q_1,\lambda) \vdash \widetilde{P_1} \leq \epsilon_1 \qquad (Q_2,\lambda) \vdash \widetilde{P_2} \leq \epsilon_2 \qquad \{Q_1\}P_1\{Q_2\}}{(Q_1,\lambda) \vdash (\widetilde{P_1}; \widetilde{P_2}) \leq \epsilon_1 + \epsilon_2}\ (\text{Sequence})$$

$$\frac{\forall m, (Q_m, 1-\delta) \vdash \widetilde{P_m} \leq \epsilon \qquad t, \delta \in [0,1]}{(\textstyle\sum_m M_m^\dagger Q_m M_m, 1 - t\delta) \vdash (\mathbf{case}\ M[\overline{q}] = \overline{m \to \widetilde{P_m}}\ \mathbf{end}) \leq (1-t)\epsilon + t}\ (\text{Case})$$

$$\frac{(Q,\lambda) \vdash \widetilde{P_1} \leq \epsilon \qquad \{Q\}P_1\{\lambda M_0^\dagger M_0 + M_1^\dagger Q M_1\} \qquad \widetilde{P} \equiv \mathbf{while}\ M[\bar{q}] = 1\ \mathbf{do}\ \widetilde{P_1}\ \mathbf{done} \qquad P \text{ is } (a,n)\text{-bounded}}{(\lambda M_0^\dagger M_0 + M_1^\dagger Q M_1, \lambda) \vdash \widetilde{P} \leq n\epsilon/(1-a)}\ (\text{While-Bounded})$$

$$\frac{}{(Q,\lambda) \vdash (\mathbf{while}\ M[\bar{q}] = 1\ \mathbf{do}\ \widetilde{P_1}\ \mathbf{done}) \leq 1}\ (\text{While-Unbounded})$$

Figure 5.3: Rules for logic of quantum robustness.

The rules for our logic are given in Figure 5.3.

5.4.2.1 Simple rules

The *Skip* and *Init* rules say that the skip and initialization operations are always error-free. These operations will not increase the distance between $[\![P]\!]$ and $[\![\widetilde{P}]\!]$. The *Unitary* rule says that if we can bound the (Q, λ)-diamond norm between the intended operation U and the noise operation Φ by ϵ, then we can bound the total distance by $p\epsilon$. The *Weaken* rule says that we can always safely make the precondition more restrictive, increase the degree to which an input state must satisfy the predicate, or increase the upper bound on the distance between the noisy and ideal programs. The *Rescale* rule says that equivalent forms of our judgment can be obtained by rescaling Q and λ. Note that the *Rescale* rule does not weaken the judgment, but rather provides some flexibility in choosing Q, λ compatible with other rules; one can scale by δ so long as Q/δ and λ/δ are still well defined.

The *Sequence* rule allows us to compose two judgments by summing their computed upper bounds. Note that in the *Sequence* and *While-Bounded* rules we define Hoare triples as in Section 5.2.3. While providing an accurate precondition (Q, λ) allows a better estimation on the error bound, it remains a practical challenge to compute the postcondition Q_2. To tackle the issue, Tao, Shi, Yao, Hui, Chong, and Gu [141] propose a method for automatedly computing the postcondition using matrix product state tensor network.

5.4.2.2 The case rule

The *Case* rule says that, given appropriate bounds for every branch of a case statement, we can bound the error of the entire case statement. Note that $\sum_m M_m^\dagger Q_m M_m$ is the weakest precondition of the case construct in quantum Hoare logic [131]. In a logic for

classical programs, one might expect each branch of a case statement to satisfy the precondition perfectly. However, in a quantum logic, as we will discuss in the soundess proof (see Section 5.4.3), this is not necessarily true. To see this, note that in our rule we start with the precondition $\sum_m M_m^\dagger Q_m M_m$ and $\lambda = 1 - t\delta$ on the input state to the case statement, but we can only guarantee that a weighted fraction of $1 - t$ of the branches satisfy a weaker precondition Q_m and $\lambda' = 1 - \delta$ for some choice of $t \in [0, 1]$. (Note that $1 - \delta \leq 1 - t\delta$ for $t, \delta \in [0, 1]$.)

When applying this rule, one can make $1 - \delta$ and ϵ the same for every $\widetilde{P_m}$ by applying the *Weaken* and/or *Rescale* rules. The choice of t will represent a tradeoff between a lower error bound and a more restrictive requirement on the satisfaction of the predicate.

5.4.2.3 The loop rule

Finally, the *While-Bounded* and *While-Unbounded* rules allow us to compute an upper bound for the distance between the noisy and ideal versions of a while loop. The *While-Unbounded* rule is a trivial bound on the distance. The *While-Bounded* rule, however, demonstrates a non-trivial upper bound with an assumption called (a, n)-boundedness. Such an assumption is necessary for us to get around the potential issue of termination and to be able to reason about interesting programs in our case study.

Intuitively, (a, n)-boundedness is a condition on how fast the *ideal* loop will converge, which is inherent to the control flow of the program and does not depend on any specific error model. We view this as an advantage as we do not need a new analysis for every possible noise model.

Definition 5.4.3 ((a,n)-boundedness)**.** *A while loop* $P \equiv$ **while** $M[q]=1$ **do** P_1 **done** *is said to be* (a,n)*-bounded for* $0 \leq a < 1$ *and integer* $n \geq 1$ *if*

$$(\mathcal{E}^*)^n(M_1^\dagger M_1) \sqsubseteq a M_1^\dagger M_1 \tag{5.15}$$

where the linear map $\mathcal{E}(\rho)$ *is defined as* $[\![P_1]\!](M_1 \rho M_1^\dagger)$ *and* $\mathcal{E}^*$ *is the dual map of* $\mathcal{E}$*.*[5]

Intuitively speaking, a loop is (a,n)-bounded if, for every state ρ, after n iterations it is guaranteed that at least a $(1-a)$-fraction of the state has exited the loop. A while loop with this nice property is guaranteed to terminate with probability 1 on all input states, which helps avoid the termination issue. As we will show in the examples that follow and in Section 5.4.4, specific (a,n) can be derived analytically or numerically for concrete programs.[6] We also remark that by assuming (a,n)-boundedness, we avoid weakening λ like in the *Case* rule.

In [35], loops are assumed to have a bounded number of iterations or a trivial upper bound will be used (i.e, our rule *While-Unbounded*). Because of the use of (a,n)-boundedness, we can handle more complicated loops. One also has the freedom to choose appropriate values of Q for different purposes. A simple choice is $Q = \lambda I$. A less trivial choice of Q is shown in the following example.

Example 5.4.2 (Slow state preparation)**.** *In this example, we consider the following program*

[5] If $[\![P_1]\!]$ can be written as $\sum_k F_k \circ F_k^\dagger$ for some set of Kraus operators $\{F_k\}_k$, the Kraus form of $\mathcal{E}^*$ is $\sum_k M_1^\dagger F_k^\dagger \circ F_k M_1$.

[6] The rough idea is to guess (or enumerate) n and prove a either analytically or numerically (with a simple SDP).

which prepares the standard basis state $|1\rangle$*:*

$$SSP \equiv q := |0\rangle; \textbf{while } M[q] = 0 \textbf{ do } q := H[q]; q := I[q] \textbf{ done}, \tag{5.16}$$

where M *is the standard basis measurement* $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$*. We consider the case where there is a bit flip error with probability 0.01 when applying the* I *gate, i.e., the ideal and the noisy loop bodies are* $P_1 \equiv q := H[q]; q := I[q]$*; and* $\widetilde{P_1} \equiv q := H[q]; q :\cong_{0.01,X} I[q]$*. Then* $[\![P_1]\!] = H \circ H$ *and* $[\![\widetilde{P_1}]\!] = 0.99H \circ H + 0.01XH \circ HX$*. Consider* $Q = |0\rangle\langle 0|$ *and* $\lambda = 1$*. Since* $[\![P_1]\!](|0\rangle\langle 0|) = [\![\widetilde{P_1}]\!](|0\rangle\langle 0|) = |+\rangle\langle +|$*, we have that* $\left\| [\![P_1]\!] - [\![\widetilde{P_1}]\!] \right\|_{|0\rangle\langle 0|,1} = 0$*, and by the* Unitary *rule,* $(|0\rangle\langle 0|, 1) \vdash \widetilde{P_1} \leq 0$*.*

We can use this choice of Q *and* λ *when applying the* While-Bounded *rule. First, note that the statement* $\{Q\}P_1\{\lambda M_0^\dagger M_0 + M_1 Q M_1^\dagger\}$ *holds because* $\lambda M_0^\dagger M_0 + M_1 Q M_1^\dagger = I$ *for our choice of* M_0, M_1, Q, λ*. Next, we need to show* (a, n)*-boundedness of the loop. This requires us to consider the behavior of* $\mathcal{E}^*$ *where* $\mathcal{E}^*$ *is the dual of* $\mathcal{E} = HM_1 \circ M_1 H$*. We claim that the while loop is* $(1/2, 1)$*-bounded because*

$$\mathcal{E}^*(M_1^\dagger M_1) = |0\rangle\langle 0|H|0\rangle\langle 0|H|0\rangle\langle 0| = \frac{1}{2}|0\rangle\langle 0| = \frac{1}{2}M_1^\dagger M_1. \tag{5.17}$$

Now by the While-Bounded *rule, we have that* $(I, 1) \vdash \widetilde{SSP} \leq 0$*, i.e., the program is perfectly robust.*

In Example 5.4.2, if we use the precondition I in our judgment of the robustness of the loop body, the best upper bound we can argue is $\epsilon = 0.01$, i.e., $(I, 1) \vdash \widetilde{P_1} \leq 0.01$. Then, applying the *While-Bounded* rule yields $(I, 1) \vdash \widetilde{SSP} \leq 0.02$ since $\frac{n\epsilon}{1-a} = 2\epsilon =$

0.02. Therefore, restricting the state space with the predicate $Q = |0\rangle\langle 0|$, as we did in Example 5.4.2, was a better choice, as it yielded a better bound. Moreover, this choice of Q is natural since the post-measurement state entering the loop satisfies Q perfectly. We note that the program SSP might look contrived for preparing $|1\rangle$ when compared to the more straightforward program $SP \equiv q := |0\rangle; q := X[q]$. We argue that SSP might be preferred over SP in the presence of noise. Consider the same noise model as above. Namely, let $\widetilde{SP} \equiv q := |0\rangle; q := X[q]; q :\cong_{0.01,X} I[q]$. We have shown that $(I, 1) \vdash \widetilde{SSP} \leq 0$ given this noise model, which says that $\widetilde{SSP}$ is perfectly robust. By directly applying Definition 5.4.2, we can show that $\widetilde{SP}$ is 0.01-robust given the same noise model,[7] which suggests that $\widetilde{SP}$ is less desirable in this situation.

5.4.3 Soundness

In this section, we show that logic given in Figure 5.3 is sound.

Theorem 5.4.3 (Soundness)**.** *If* $(Q, \lambda) \vdash \widetilde{P} \leq \epsilon$ *then* $\widetilde{P}$ *is* ϵ*-robust under* (Q, λ)*.*

Proof. The proof proceeds by induction on the derivation $(Q, \lambda) \vdash \widetilde{P} \leq \epsilon$. We will work primarily from definition of robustness given in (5.14). We note that superoperators $[\![\widetilde{P}]\!]$ and $[\![P]\!]$ apply on the same space. By the definition of the (Q, λ)-diamond norm, we need to consider $[\![\widetilde{P}]\!] \otimes I$ and $[\![P]\!] \otimes I$. However, to simplify the presentation, we will omit "$\otimes I$" in the following proof whenever there is no ambiguity.

Now we consider each possible rule used in the final step of the derivation.

[7] Note that $[\![\widetilde{SP}]\!] = 0.99X \circ X + 0.01I \circ I$, and hence we have $\left\| [\![SP]\!] - [\![\widetilde{SP}]\!] \right\|_{I,1} = 0.01 \|X \circ X - I \circ I\|_{I,1} = 0.01$.

1. *Skip:* This rule holds by observing $[\![\widetilde{\mathbf{skip}}]\!] = [\![\mathbf{skip}]\!]$, i.e., they refer to the same super-operator. Thus, any (Q, λ)-diamond norm between them is 0. We choose $Q = I$ and $\lambda = 0$.

2. *Init:* for the same reason as the proof for *Skip*.

3. *Unitary:* For every state ρ satisfying $\mathrm{tr}(Q\rho) \geq \lambda$,

$$\begin{aligned}\frac{1}{2}\|[\![\bar{q} :\cong_{p,\Phi} U[\bar{q}]]\!]\rho - [\![\bar{q} := U[\bar{q}]]\!]\rho\|_1 &= \frac{1}{2}\big\|\big((1-p)U\rho U^\dagger + p\Phi(\rho)\big) - U\rho U^\dagger\big\|_1 \\ &= \frac{p}{2}\big\|U\rho U^\dagger - \Phi(\rho)\big\|_1 \leq p\big\|U\cdot U^\dagger - \Phi\big\|_{Q,\lambda} \\ &\leq p\epsilon. \end{aligned} \tag{5.18}$$

   The second from last inequality holds by the definition of the (Q, λ)-diamond norm and the last inequality follows from the premise.

4. *Weaken:* By induction, the premise $(Q', \lambda') \vdash \widetilde{P} \leq \epsilon'$ implies

$$\forall \rho, \mathrm{tr}(Q'\rho) \geq \lambda' \,\mathrm{tr}(\rho) \implies \frac{1}{2}\left\|[\![\widetilde{P}]\!]\rho - [\![P]\!]\rho\right\|_1 \leq \epsilon' \,\mathrm{tr}(\rho). \tag{5.19}$$

   For any density matrix ρ, constants $0 \leq \lambda' \leq \lambda \leq 1$, and predicates $Q \sqsubseteq Q'$, $\mathrm{tr}(Q\rho) \geq \lambda \,\mathrm{tr}(\rho)$ implies that $\mathrm{tr}(Q'\rho) \geq \lambda' \,\mathrm{tr}(\rho)$. And for $0 \leq \epsilon' \leq \epsilon \leq 1$, $\frac{1}{2}\left\|[\![\widetilde{P}]\!]\rho - [\![P]\!]\rho\right\|_1 \leq \epsilon' \,\mathrm{tr}(\rho)$ implies that $\frac{1}{2}\left\|[\![\widetilde{P}]\!]\rho - [\![P]\!]\rho\right\|_1 \leq \epsilon \,\mathrm{tr}(\rho)$. Therefore, we have that

$$\forall \rho, \mathrm{tr}(Q\rho) \geq \lambda \,\mathrm{tr}(\rho) \implies \frac{1}{2}\left\|[\![\widetilde{P}]\!]\rho - [\![P]\!]\rho\right\|_1 \leq \epsilon \,\mathrm{tr}(\rho). \tag{5.20}$$

So $\widetilde{P}$ is ϵ-robust under (Q, λ).

5. *Rescale:* This rule follows by observing that the condition $\operatorname{tr}(\rho Q) \geq \lambda$ is equivalent to the condition $\operatorname{tr}(\rho Q/\delta) \geq \lambda/\delta$ for $\delta > 0$. We only require that $Q, Q/\delta$ and $\lambda, \lambda/\delta$ are well defined, namely, $0 \sqsubseteq Q, Q/\delta \sqsubseteq I$ and $0 \leq \lambda, \lambda/\delta \leq 1$.

6. *Sequence:* For every state ρ,

$$\begin{aligned}
\left\| [\![\widetilde{P_1}; \widetilde{P_2}]\!]\rho - [\![P_1; P_2]\!]\rho \right\|_1 &= \left\| [\![\widetilde{P_2}]\!][\![\widetilde{P_1}]\!]\rho - [\![P_2]\!][\![P_1]\!]\rho \right\|_1 \\
&\leq \left\| [\![\widetilde{P_2}]\!][\![\widetilde{P_1}]\!]\rho - [\![\widetilde{P_2}]\!][\![P_1]\!]\rho \right\|_1 + \left\| [\![\widetilde{P_2}]\!][\![P_1]\!]\rho - [\![P_2]\!][\![P_1]\!]\rho \right\|_1 \\
&\leq \left\| [\![\widetilde{P_1}]\!]\rho - [\![P_1]\!]\rho \right\|_1 + \left\| [\![\widetilde{P_2}]\!][\![P_1]\!]\rho - [\![P_2]\!][\![P_1]\!]\rho \right\|_1 . \qquad (5.21)
\end{aligned}$$

The inequality $\left\| [\![\widetilde{P_2}]\!][\![\widetilde{P_1}]\!]\rho - [\![\widetilde{P_2}]\!][\![P_1]\!]\rho \right\|_1 \leq \left\| [\![\widetilde{P_1}]\!]\rho - [\![P_1]\!]\rho \right\|_1$ follows because quantum superoperators are contractive.

Now assume that $\operatorname{tr}(Q_1\rho) \geq \lambda \operatorname{tr}(\rho)$. By induction, the premise $(Q_1, \lambda) \vdash \widetilde{P_1} \leq \epsilon_1$ implies that $\frac{1}{2}\left\| [\![\widetilde{P_1}]\!]\rho - [\![P_1]\!]\rho \right\|_1 \leq \epsilon_1 \operatorname{tr}(\rho)$. Also, by the premise $\{Q_1\}P_1\{Q_2\}$, we have that $\operatorname{tr}(Q_2[\![P_1]\!]\rho) \geq \operatorname{tr}(Q_1\rho) \geq \lambda \operatorname{tr}(\rho) \geq \lambda \operatorname{tr}([\![P_1]\!]\rho)$. Now we can use our induction hypothesis and the premise $(Q_2, \lambda) \vdash \widetilde{P_2} \leq \epsilon_2$ to conclude $\frac{1}{2}\left\| [\![\widetilde{P_2}]\!][\![P_1]\!]\rho - [\![P_2]\!][\![P_1]\!]\rho \right\|_1 \leq \epsilon_2 \operatorname{tr}([\![P_1]\!]\rho)$. So, finally, we have that

$$\frac{1}{2}\left\| [\![\widetilde{P_1}; \widetilde{P_2}]\!]\rho - [\![P_1; P_2]\!]\rho \right\|_1 \leq \epsilon_1 \operatorname{tr}(\rho) + \epsilon_2 \operatorname{tr}([\![P_1]\!]\rho) \leq (\epsilon_1 + \epsilon_2) \operatorname{tr}(\rho). \qquad (5.22)$$

7. *Case:* Let $\widetilde{P} = \textbf{case } M[\overline{q}] = \overline{m \to \widetilde{P_m}} \textbf{ end}$. Assume the input state ρ to the case statement satisfies $\sum_m M_m^\dagger Q_m M_m$ to degree λ', i.e., $\sum_m \operatorname{tr}(M_m^\dagger Q_m M_m \rho) \geq \lambda' \operatorname{tr}(\rho)$. To

leverage the premise $(Q_m, \lambda) \vdash \widetilde{P_m} \leq \epsilon$ and the induction hypothesis to conclude that $\frac{1}{2}\left\| [\![\widetilde{P_m}]\!]\rho - [\![P_m]\!]\rho \right\|_1 \leq \epsilon \operatorname{tr}(\rho)$, one must show the precondition (Q_m, λ) holds for state ρ entering branch m. A naive approach is to show that $\operatorname{tr}(M_m^\dagger Q_m M_m \rho) \geq \lambda \operatorname{tr}(M_m^\dagger M_m \rho)$ holds for *every* branch m, which implies that $\operatorname{tr}(Q_m M_m \rho M_m^\dagger) \geq \lambda \operatorname{tr}(M_m \rho M_m^\dagger)$ where $M_m \rho M_m^\dagger$ is the (sub-normalized) post-measurement state entering branch m. We argue that this is in general impossible when $\lambda' = \lambda$. For instance, consider a collection of projective measurement operators $\{M_m\}_m$. If there exists a branch i and a state ρ supported on M_i such that $\operatorname{tr}(M_i^\dagger Q_i M_i \rho) \geq \lambda \operatorname{tr}(\rho)$ for some $\lambda > 0$, obviously $\sum_m M_m^\dagger Q_m M_m \rho \geq \lambda \operatorname{tr}(\rho)$, but none of the preconditions is satisfied except for the one for branch i since $\operatorname{tr}(M_j^\dagger Q_j M_j \rho) = 0$ for each $j \neq i$.

Instead, we show that for a majority of the clauses $\operatorname{tr}(M_m^\dagger Q_m M_m \rho) \geq \lambda \operatorname{tr}(M_m^\dagger M_m \rho)$ holds for some λ strictly less than λ'. To that end, let $p_m = \operatorname{tr}(M_m^\dagger M_m \rho)$ and $q_m = \operatorname{tr}(M_m^\dagger Q_m M_m \rho)$. Define δ_m to be such that $q_m = (1 - \delta_m) p_m$. Note that $0 \leq \delta_m \leq 1$ because $0 \leq q_m \leq p_m$ for every m. Without loss of generality we assume $\operatorname{tr}(\rho) = 1$. Let $S(\rho)$ denote the collection of branches such that the precondition (Q_m, λ) holds, i.e., $S(\rho) = \{m : q_m \geq \lambda p_m, \text{ i.e., } \delta_m \leq 1 - \lambda\}$. We will determine a lower bound for $\sum_{m \in S(\rho)} p_m$ for each state ρ using a probabilistic argument.

First, note that $\{p_m\}$ is a probability distribution since $\sum_m p_m = 1$ and $p_m \geq 0$ for each m. Also, since (by our assumption) $\sum_m q_m \geq \lambda' \sum_m p_m = \lambda'$, we have that $\sum_m \delta_m p_m \leq (1 - \lambda')$. Now we can define a random variable Δ to be such that $\Pr[\Delta = \delta_m] = p_m$ for each m. The expected value of Δ is $\mathbb{E}[\Delta] \leq (1 - \lambda')$, and Markov's

inequality yields

$$\sum_{m \notin S(\rho)} p_m = \Pr[\Delta \geq 1 - \lambda] \leq \frac{\mathbb{E}[\Delta]}{1-\lambda} \leq \frac{1-\lambda'}{1-\lambda} =: t. \tag{5.23}$$

This says that a weighted fraction t of the branches will not satisfy the precondition (Q_m, λ). Note that $t \in [0,1]$ since $\lambda \leq \lambda'$. For $m \notin S(\rho)$, only the trivial upper bound $\epsilon_m = 1$ is guaranteed. Therefore, for each state ρ,

$$\begin{aligned}
\frac{1}{2}\left\| [\![\widetilde{P}]\!]\rho - [\![P]\!]\rho \right\|_1 &= \frac{1}{2}\left\| \sum_m \left([\![\widetilde{P_m}]\!](M_m \rho M_m^\dagger) - [\![P_m]\!](M_m \rho M_m^\dagger) \right) \right\|_1 \\
&\leq \frac{1}{2} \sum_m \left\| [\![\widetilde{P_m}]\!](M_m \rho M_m^\dagger) - [\![P_m]\!](M_m \rho M_m^\dagger) \right\|_1 \\
&\leq \sum_{m \in S(\rho)} \operatorname{tr}(M_m \rho M_m^\dagger)\epsilon + \sum_{m \notin S(\rho)} \operatorname{tr}(M_m \rho M_m^\dagger) \\
&\leq (1-t)\epsilon + t = ((1-t)\epsilon + t)\operatorname{tr}(\rho).
\end{aligned} \tag{5.24}$$

Rewriting $\lambda = 1 - \delta$ for ease of notation, we have $\lambda' = 1 - t\delta$. Finally, we note that the case $t = 0$ implies that the precondition of each branch is satisfied, and the error of the case statement can be bounded by ϵ.

8. *While-Bounded:* Let $P \equiv$ **while** $M[q] = 1$ **do** P_1 **done**. Let S_k be the bounded while loop of k iterations. Define the linear maps $\mathcal{E}(\rho) := [\![P_1]\!](M_1 \rho M_1^\dagger)$ and $\widetilde{\mathcal{E}}(\rho) := [\![\widetilde{P_1}]\!](M_1 \rho M_1^\dagger)$ and let $[\![S_k]\!]\rho = M_0 \rho M_0^\dagger + [\![S_{k-1}]\!](\mathcal{E}(\rho))$ for $k \geq 1$ (with $[\![S_0]\!]\rho = \rho$). In order to bound the distance between $[\![P]\!]$ and $[\![\widetilde{P}]\!]$, we first upper bound the distance

between $[\![S_k]\!]$ and $[\![\widetilde{S_k}]\!]$ and then take the limit as $k \to \infty$. We then have

$$\begin{aligned}
\frac{1}{2}\left\|[\![\widetilde{S_k}]\!]\rho - [\![S_k]\!]\rho\right\|_1 &\leq \frac{1}{2}\left\|[\![\widetilde{S}_{k-1}]\!](\widetilde{\mathcal{E}}(\rho)) - [\![S_{k-1}]\!](\mathcal{E}(\rho))\right\|_1 \\
&\leq \frac{1}{2}\left\|\widetilde{\mathcal{E}}(\rho) - \mathcal{E}(\rho)\right\|_1 + \frac{1}{2}\left\|[\![\widetilde{S}_{k-1}]\!](\mathcal{E}(\rho)) - [\![S_{k-1}]\!](\mathcal{E}(\rho))\right\|_1 \qquad (5.25) \\
&\leq \frac{1}{2}\sum_{i=0}^{k-1}\left\|\widetilde{\mathcal{E}}(\mathcal{E}^i(\rho)) - \mathcal{E}^{i+1}(\rho)\right\| \qquad (5.26) \\
&\leq \epsilon \sum_{i=0}^{k-1} \operatorname{tr}(M_1^\dagger M_1 \mathcal{E}^i(\rho)) \qquad (5.27) \\
&\leq n\epsilon \operatorname{tr}(M_1^\dagger M_1 \rho)\frac{1-a^{\lceil k/n \rceil}}{1-a}. \qquad (5.28)
\end{aligned}$$

The second inequality (5.25) follows from a technique similar to the one used in the proof of the *Sequence* rule. We bound the first term in (5.25) by $\epsilon \operatorname{tr}(M_1 \rho M_1^\dagger)$ by applying the premise $(Q, \lambda) \vdash P_1 \leq \epsilon$ and the induction hypothesis. We will prove the post-measurement state $M_1 \rho M_1^\dagger$ indeed satisfies the precondition (Q, λ) later. Similarly, each term in (5.26) is bounded above by $\epsilon \operatorname{tr}(M_1^\dagger M_1 \mathcal{E}^i(\rho))$ and thus the inequality in (5.27) holds.

To establish the inequality in (5.28), let $b_i := \operatorname{tr}(M_1^\dagger M_1 \mathcal{E}^i(\rho))$. Then the sequence $\{b_k\}_k$ is non-negative and non-increasing. We now prove an upper bound of the series. Since P is (a, n)-bounded, we know that

$$\operatorname{tr}(M_1^\dagger M_1 \mathcal{E}^n(\sigma)) = \operatorname{tr}((\mathcal{E}^*)^n(M_1^\dagger M_1)\sigma) \leq a \operatorname{tr}(M_1^\dagger M_1 \sigma), \text{ where } a < 1 \qquad (5.29)$$

for every state σ, and therefore $b_{i+n} \leq a b_i$ for every i. Since b_k is non-increasing, we

know that

$$\sum_{i=0}^{k-1} b_i = \sum_{m=0}^{\lceil k/n \rceil - 1} (b_{nm} + \ldots + b_{nm+n-1})$$
$$\leq n \sum_{m=0}^{\lceil k/n \rceil - 1} b_{nm} \leq n \sum_{m=0}^{\lceil k/n \rceil - 1} a^m b_0 = \frac{n(1 - a^{\lceil k/n \rceil}) b_0}{1 - a}. \tag{5.30}$$

Thus the inequality in (5.28) holds. Since $b_0 = \operatorname{tr}(M_1^\dagger M_1 \rho) \leq \operatorname{tr}(\rho)$, we have

$$\frac{1}{2} \left\| [\![\widetilde{S_k}]\!]\rho - [\![S_k]\!]\rho \right\|_1 \leq \frac{n\epsilon(1 - a^{\lceil k/n \rceil})}{1 - a} \operatorname{tr}(\rho). \tag{5.31}$$

Taking the limit as $k \to \infty$, we have that $a^{\lceil k/n \rceil} \to 0$, which shows that $\frac{1}{2} \left\| [\![\widetilde{S_k}]\!]\rho - [\![S_k]\!]\rho \right\|_1 \leq \frac{n\epsilon}{1-a} \operatorname{tr}(\rho)$, as desired.

In order to apply the premise $(Q, \lambda) \vdash P_1 \leq \epsilon$ to states of the form $M_1 \mathcal{E}^i(\rho) M_1^\dagger$, we need to show that $\operatorname{tr}(Q M_1 \mathcal{E}^i(\rho) M_1^\dagger) \geq \lambda \operatorname{tr}(M_1 \mathcal{E}^i(\rho) M_1^\dagger)$ for each i. We can prove this by induction. For the base case $i = 0$, by the precondition $(\lambda M_0^\dagger M_0 + M_1^\dagger Q M_1, \lambda)$ on the input state to the loop, we have $\operatorname{tr}(M_1^\dagger Q M_1 \rho) \geq \lambda \operatorname{tr}(\rho) - \lambda \operatorname{tr}(M_0^\dagger M_0 \rho) = \lambda \operatorname{tr}(M_1^\dagger M_1 \rho)$. Therefore, $\operatorname{tr}(Q M_1 \mathcal{E}^0(\rho) M_1^\dagger) \geq \lambda \operatorname{tr}(M_1 \mathcal{E}^0(\rho) M_1^\dagger)$

For the inductive step, observe that the Hoare triple $\{Q\} P_1 \{R\}$ yields $\operatorname{tr}(R [\![P_1]\!] \sigma) \geq \operatorname{tr}(Q\sigma)$ for $R \equiv \lambda M_0^\dagger M_0 + M_1^\dagger Q M_1$ and all states σ. By the induction hypothesis, we

have

$$\begin{aligned}\operatorname{tr}(R[\![P_1]\!](M_1\mathcal{E}^i(\rho)M_1^\dagger)) &\geq \operatorname{tr}(QM_1\mathcal{E}^i(\rho)M_1^\dagger)\\ &\geq \lambda \operatorname{tr}(M_1\mathcal{E}^i(\rho)M_1^\dagger)\\ &\geq \lambda \operatorname{tr}(\mathcal{E}^{i+1}(\rho)),\end{aligned} \tag{5.32}$$

where the last inequality holds because quantum operations are trace-non-increasing (applied to $[\![P_1]\!]$). Note that $[\![P_1]\!](M_1\mathcal{E}^iM_1^\dagger) = \mathcal{E}^{i+1}$. Now by substituting R, we have

$$\operatorname{tr}(M_1^\dagger QM_1\mathcal{E}^{i+1}(\rho)) \geq \lambda \operatorname{tr}(\mathcal{E}^{i+1}(\rho)) - \lambda \operatorname{tr}(M_0^\dagger M_0\mathcal{E}^{i+1}(\rho)) = \lambda \operatorname{tr}(M_1^\dagger M_1\mathcal{E}^{i+1}(\rho)). \tag{5.33}$$

Or equivalently, $\operatorname{tr}(QM_1\mathcal{E}^{i+1}(\rho)M_1^\dagger) \geq \lambda \operatorname{tr}(M_1\mathcal{E}^{i+1}(\rho)M_1^\dagger)$, which concludes the proof.

9. *While-Unbounded:* the proof is trivial as we use the trivial upper bound 1.

□

5.4.4 Case studies

As a case study, we analysis the robustness of quantum walk on a circle, introduced in Section 5.2.1, with our logic. For more examples, see [36].

The noisy quantum walk on a circle with n points can be written as the following

program:

$$\widetilde{QW}_n \equiv p := |0\rangle;$$

$$c := |L\rangle; \textbf{while } M[p] = 1 \textbf{ do } c :\cong_{p_H,\Phi_H} H[c]; c, p :\cong_{p_S,\Phi_S} S[c, p] \textbf{ done}. \tag{5.34}$$

Now we show that $(I, 0) \vdash \widetilde{QW}_6 \leq 30(\epsilon_H + \epsilon_S)$ where $\epsilon_H = p_H \left\| \Phi_H - H \circ H^\dagger \right\|_\diamond$ and $\epsilon_S = p_S \left\| \Phi_S - S \circ S^\dagger \right\|_\diamond$ are the errors due to noisy application of H and S respectively. First, by the *Sequence* and *Unitary* rules, we bound the error in the loop body by $\epsilon_H + \epsilon_S$. Next, we numerically test increasing values of (a, n) until we find a pair that satisfies (5.15). For this program, we find that the pair $(\frac{5}{6}, 5)$ satisfies the inequality, i.e., $(\mathcal{E}^*)^5(M_1^\dagger M_1) \sqsubseteq \frac{5}{6} M_1^\dagger M_1$ where $\mathcal{E}^* = (M_1^\dagger \circ M_1) \circ [\![c, p := S[c, p]]\!]^* \circ [\![c := H[c]]\!]^*$. This implies that the loop is $(\frac{5}{6}, 5)$-bounded. Note that here, unlike in the previous example, we have computed the (a, n) values numerically. This may be useful in cases where direct deduction of a and n is difficult. Now we can use the *While-Bounded* rule to conclude that the error bound for the loop is $\frac{5(\epsilon_H + \epsilon_S)}{1 - \frac{5}{6}} = 30(\epsilon_H + \epsilon_S)$, and therefore $(I, 0) \vdash (\textbf{while } M[p] = 1 \textbf{ do } \widetilde{P_1} \textbf{ done}) \leq 30(\epsilon_H + \epsilon_S)$ where $\widetilde{P_1}$ is the body of the loop. Two additional applications of the *Sequence* rule conclude the proof.

As an example, consider the program $\widetilde{QW}_6$ where only the Hadamard gate may be faulty, i.e., $p_S = 0$. Say that noisy Hadamard application is characterized by $p_H = 5 \times 10^{-5}$ and $\Phi_H(\rho) = \frac{1}{4}(\rho + X\rho X + Y\rho Y + Z\rho Z)$, the 2-dimensional depolarizing channel. Applying an SDP solver [140, 142, 143] for the calculation of the diamond norm, we find that $\epsilon_H = 5 \times 10^{-5} \times 0.75 = 3.75 \times 10^{-5}$. Therefore the error of $\widetilde{QW}_6$ is $30\epsilon_H = 1.125 \times 10^{-3}$.

Chapter 6: Conclusion

This dissertation included selected topics in quantum cryptography, quantum algorithms and query lower bounds, and formal verification of quantum information processes. Apart from the open questions mentioned in each chapter, we now summarize our results and discuss some future directions that extend our work, as follows.

In Chapter 2, we constructed a protocol which enables a purely classical party to delegate any quantum computation to an untrusted quantum prover non-interactively. Our protocols result from a sequence of significant improvements to the original four-message protocol of Mahadev. We begin by making the first message instance-independent and move it to an offline setup phase. Then we established a parallel repetition theorem for the resulting three-message protocol. This enables an application of the Fiat-Shamir transform, eliminating the public-coin toss and yielding a non-interactive protocol. Furthermore, invoking a commitment scheme, a non-interactive zero-knowledge argument for NP, and a circuit-private classical fully homomorphic encryption scheme, we showed the same task can be performed in zero-knowledge.

In the verification protocols, the honest prover must make one query to the trapdoor claw-free functions or the trapdoor injective functions for each qubit, to generate the response to the verifier. While making such function queries enables the verifiability of the prover's

computation, it introduces a large overhead. Recently, new protocols for demonstrating quantum advantage are proposed, based on various hardness assumptions with an estimation of required resources [10, 11]. In particular, the adaptive hardcore bit property is not required to prove the soundness. It is unclear whether these protocols, when basing on post-quantum assumptions, lead to not only protocols for quantum advantage, but also more efficient protocols for verifying quantum computation. Moreover, the same LWE-based hash functions can also be used to generate certifiable randomness [9]. Is it possible to give a randomness generation protocol using a single near-term device? The analysis of the protocol is based on the adaptive hardcore bit property, which is known to hold with the LWE assumption. It is unclear yet interesting to know whether one can give a randomness expansion protocol from other standard assumptions.

In Chapter 3, we determined the quantum query complexity for the polynomial interpolation problem, which is a fundamental task in numerical analysis and cryptography. In particular, we showed that for a degree-d polynomial of n variables, the query complexity over a finite field $\mathbb{F}_q$ of q elements, complexity numbers $\mathbb{C}$ and real numbers $\mathbb{R}$, the query complexities are $O(\frac{d}{n+d}\binom{n+d}{d})$, $O(\frac{1}{n+1}\binom{n+d}{d})$ and $O(\frac{2}{n+1}\binom{n+d}{d})$ for the success probablities asymptotically approaching 1, except for a few cases in n and d. Furthermore, we showed that for univariate polynomial interpolation, the query complexity is $d/2+1$, and the optimal algorithm is non-adaptive and gate-efficient.

Our model assumes the oracle encodes a function f which is a polynomial, the degree of which is known. Given access to oracles which encode non-polynomial functions, can we give quantum algoithms which output a polynomial that approximates the function? Our current algorithms do not seem robust in this sense, and the algorithm in this setting may

depend on how the closeness is defined.

In Chapter 4, we studied quantum algorithms that learn properties of a matrix using queries that return its action on an input vector. We showed that for determining various problems, including trace, determinant, rank testing, and solving linear systems, the query complexity is linear in the dimension of the matrix. This implies quantum computers do not provide speedups for these problems. In contrast to the classical setting, we showed the equivalence between models which provide matrix-vector, vector-matrix, and vector-matrix-vector products. This, in turn, implies that quantum computers provide exponential speedups for problems including the parities of rows and deciding if there are two identical columns.

As results over different fields are in general incomparable, it remains open whether one can establish a tight quantum lower bound for the problems we studied over the real numbers. Furthermore, many problems in these models remain unexplored. For example, what are the query complexities of computing matrix norms, e.g., Schatten p-norms, or other functions of the matrix singular values? The models of matrix-vector products and vector-matrix-vector products are closely related to data streaming models which have been extensively studied in various classical settings [111]. Would quantum speedups be possible in quantum analogues of data streaming models?

In Chapter 5, we applied formal methods to reason about the robustness of erroneous execution of quantum processes for local noise models. We provided the syntax and semantics of the quantum while language, extended to include noisy operations. We defined the notion of quantum robustness, which describes the closeness between a noisy output and its ideal correspondence when the input satisfies a property to at least a certain degree. Furthermore,

we defined a proof system for reasoning about quantum robustness. In particular, for noisy while loops, we showed so long as after n iterations, with probability at least $1 - a$, every input state has exited the ideal equivalent, there is a non-trivial upper bound on the final accumulated error.

An interesting question is whether our logic can be adapted to apply in other contexts. For example, is it possible to give a system for estimating the expected running time, circuit depth, or gate counts for quantum programs, when compiled into quantum circuits? Our logic can be seen as a special case that the error rate is the resource to estimate.

Bibliography

[1] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018. arXiv:1801.00862.

[2] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum (NISQ) algorithms. 2021. arXiv:2101.08448.

[3] Kesha Hietala, Robert Rand, Shih-Han Hung, Xiaodi Wu, and Michael Hicks. A verified optimizer for quantum circuits. *Proceedings of the ACM on Programming Languages*, 5(POPL):1–29, 2021. arXiv:1912.02250.

[4] Kesha Hietala, Robert Rand, Shih-Han Hung, Liyi Li, and Michael Hicks. Proving quantum programs correct. 2020. arXiv:2010.01240.

[5] Thomas Vidick. Verifying quantum computations at scale: A cryptographic leash on quantum devices. *Bulletin of the American Mathematical Society*, 57(1):39–76, 2020.

[6] Urmila Mahadev. Classical verification of quantum computations. In *FOCS 2018*, pages 259–267, 2018. arXiv:1804.01082.

[7] Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In *Theory of Cryptography Conference*, pages 153–180. Springer, 2020. arXiv:1911.08101.

[8] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In *Theory of Cryptography Conference*, pages 181–206. Springer, 2020. arXiv:1912.00990.

[9] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In *FOCS 2018*, pages 320–331, 2018. arXiv:1804.00640.

[10] Zvika Brakerski, Venkata Koppula, Umesh Vazirani, and Thomas Vidick. Simpler proofs of quantumness. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. arXiv:2005.04826.

[11] Gregory D. Kahanamoku-Meyer, Soonwon Choi, Umesh V. Vazirani, and Norman Y. Yao. Classically-verifiable quantum advantage from a computational Bell test. 2021. arXiv:2104.00687.

[12] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In *Theory of Cryptography Conference*, pages 92–122. Springer, 2020. arXiv:1910.03551.

[13] Scott Aaronson. Quantum copy-protection and quantum money. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 229–242. IEEE, 2009. arXiv:1110.5353.

[14] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. 2020. arXiv:2004.09674.

[15] Prabhanjan Ananth and Rolando L. La Placa. Secure software leasing. 2020. arXiv:2005.05289.

[16] Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures. 2016. arXiv:1609.09047.

[17] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 41–60, 2012. arXiv:1203.4740.

[18] Mark Zhandry. Quantum lightning never strikes the same state twice. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 408–438. Springer, 2019. `https://eprint.iacr.org/2017/1080`.

[19] Andrea Coladangelo and Or Sattath. A quantum money solution to the blockchain scalability problem. *Quantum*, 4:297, 2020. arXiv:2002.11998.

[20] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. arXiv:quant-ph/9802049.

[21] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997. arXiv:quant-ph/9701001.

[22] Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. arXiv:quant-ph/0002066.

[23] Peter Høyer, 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. arXiv:quant-ph/0611054.

[24] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC '05*, pages 84–93, 2005.

[25] Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private FHE. In *CRYPTO 2014*, pages 536–553, 2014. `https://eprint.iacr.org/2013/307`.

[26] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[27] Daniel M. Kane and Samuel A. Kutin. Quantum interpolation of polynomials. *Quantum Information and Computation*, 11(1):95–103, 2011. arXiv:0909.5683.

[28] David A. Meyer Meyer and James Pommersheim. On the uselessness of quantum queries. *Theoretical Computer Science*, 412(51):7068–7074, 2011. arXiv:1004.1434.

[29] Daniel Copeland and Jamie Pommersheim. Quantum query complexity of symmetric oracle problems. *Quantum*, 5:403, 2021. arXiv:1812.09428.

[30] Andrew M. Childs, Wim van Dam, Shih-Han Hung, and Igor E. Shparlinski. Optimal quantum algorithm for polynomial interpolation. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics*, pages 16:1–16:13, 2016. arXiv:1509.09271.

[31] Jianxin Chen, Andrew M. Childs, and Shih-Han Hung. Quantum algorithm for multivariate polynomial interpolation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209):20170480, 2018. arXiv:1701.03990.

[32] Xiaoming Sun, David P. Woodruff, Guang Yang, and Jialin Zhang. Querying a matrix through matrix-vector products. In *46th International Colloquium on Automata, Languages, and Programming*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019. arXiv:1906.05736.

[33] Mark Braverman, Elad Hazan, Max Simchowitz, and Blake Woodworth. The gradient complexity of linear regression. In *Conference on Learning Theory*, pages 627–647, 2020. arXiv:1911.02212.

[34] Andrew M. Childs, Shih-Han Hung, and Tongyang Li. Quantum Query Complexity with Matrix-Vector Products. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 55:1–55:19, 2021. arXiv:2102.11349.

[35] Michael Carbin, Sasa Misailovic, and Martin C. Rinard. Verifying quantitative reliability for programs that execute on unreliable hardware. In *OOPSLA*, 2013.

[36] Shih-Han Hung, Kesha Hietala, Shaopeng Zhu, Mingsheng Ying, Michael Hicks, and Xiaodi Wu. Quantitative robustness analysis of quantum programs. *Proceedings of the ACM Symposium on Programming Languages*, 3(POPL):1–29, 2019. arXiv:1811.03585.

[37] Ben W. Reichardt, Falk Unger, and Umesh Vazirani. Classical command of quantum systems. *Nature*, 496(7446):456–460, 2013. arXiv:1209.0448.

[38] Matthew McKague. Interactive proofs for BQP via self-tested graph states. *Theory of Computing*, 12(3):1–42, 2016. arXiv:1309.5675.

[39] Alexandru Gheorghiu, Elham Kashefi, and Petros Wallden. Robustness and device independence of verifiable blind quantum computing. *New Journal of Physics*, 17(8):083040, 2015. arXiv:1502.02571.

[40] Michal Hajdušek, Carlos A. Pérez-Delgado, and Joseph F. Fitzsimons. Device-independent verifiable blind quantum computation, 2015. arXiv:1502.02563.

[41] Joseph F. Fitzsimons and Michal Hajdušek. Post hoc verification of quantum computation, 2015. arXiv:1512.04375.

[42] Anand Natarajan and Thomas Vidick. A quantum linearity test for robustly verifying entanglement. In *STOC 2017*, pages 1003–1015, 2017. arXiv:1610.03574.

[43] Andrea Coladangelo, Alex B. Grilo, Stacey Jeffery, and Thomas Vidick. Verifier-on-a-leash: New schemes for verifiable delegated quantum computation, with quasilinear resources. In *EUROCRYPT 2019*, pages 247–277, 2019. arXiv:1708.07359.

[44] Alex B. Grilo. A simple protocol for verifiable delegation of quantum computation in one round. In *ICALP 2019*, pages 28:1–28:13, 2019. arXiv:1711.09585.

[45] Anne Broadbent, Joseph Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *FOCS 2009*, pages 517–526, 2009. arXiv:0807.4154.

[46] Anne Broadbent. How to verify a quantum computation. *Theory of Computing*, 14(11):1–37, 2018. arXiv:1509.09180.

[47] Tomoyuki Morimae and Joseph F. Fitzsimons. Post hoc verification with a single prover, 2016. arXiv:1603.06046.

[48] Dorit Aharonov, Michael Ben-Or, Elad Eban, and Urmila Mahadev. Interactive proofs for quantum computations. 2017. arXiv:1704.04487.

[49] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David N. Cooper, Quynh Dang, Yi-Kai Liu, Carl Frederick Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, and Daniel Smith-Tone. Status report on the first round of the NIST post-quantum cryptography standardization process. 2019.

[50] Ivan Damgård. On Σ-protocols. *Lecture Notes, University of Aarhus, Department for Computer Science*, 2002.

[51] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *STOC 1988*, page 103–112, 1988.

[52] Scott Aaronson. BQP and the polynomial hierarchy. In *STOC 2010*, pages 141–150, 2010. arXiv:0910.4698.

[53] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, pages 41–69, 2011. arXiv:1008.0931.

[54] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO 1986*, pages 186–194, 1986.

[55] Johan Håstad, Rafael Pass, Douglas Wikström, and Krzysztof Pietrzak. An efficient parallel repetition theorem. In *TCC 2010*, pages 1–18, 2010.

[56] Iftach Haitner. A parallel repetition theorem for any interactive argument. In *FOCS 2009*, pages 241–250, 2009.

[57] Itay Berman, Iftach Haitner, and Eliad Tsfadia. A tight parallel repetition theorem for partially simulatable interactive arguments via smooth KL-divergence, 2019. `https://eprint.iacr.org/2019/393`.

[58] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS 1993*, pages 62–73, 1993.

[59] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[60] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *FOCS 1997*, pages 374–383, 1997.

[61] John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009. arXiv:quant-ph/0511020.

[62] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *FOCS 2014*, pages 474–483, 2014. `https://eprint.iacr.org/2014/296`.

[63] Andrea Coladangelo, Thomas Vidick, and Tina Zhang. Non-interactive zero-knowledge arguments for QMA, with preprocessing, 2019. arXiv:1911.07546.

[64] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. In *CRYPTO 2019*, pages 356–383, 2019. `https://eprint.iacr.org/2019/190`.

[65] Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In *CRYPTO 2019*, 2019. `https://eprint.iacr.org/2019/262`.

[66] Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. In *FOCS 2016*, pages 31–40, 2016. arXiv:1604.02804.

[67] Thomas Vidick and Tina Zhang. Classical zero-knowledge arguments for quantum computations, 2019. arXiv:1902.05217.

[68] Anne Broadbent and Alex B. Grilo. Zero-knowledge for QMA from locally simulatable proofs, 2019. arXiv:1911.07782.

[69] Roy Radian and Or Sattath. Semi-quantum money, 2019. arXiv:1908.08889.

[70] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors, 2019. `https://eprint.iacr.org/2019/158`.

[71] Alexandru Gheorghiu and Thomas Vidick. Computationally-secure and composable remote state preparation, 2019. arXiv:1904.06320.

[72] Alex Bredariol Grilo, William Slofstra, and Henry Yuen. Perfect zero knowledge for quantum multiprover interactive proofs. In *FOCS 2019*, pages 611–635, 2019. arXiv:1905.11280.

[73] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds, 2019. `https://eprint.iacr.org/2019/1279`.

[74] Alexei Yu. Kitaev, Alexander H. Shen, and Mikhail N. Vyalyi. *Classical and quantum computation*. American Mathematical Society, 2002.

[75] Jacob D. Biamonte and Peter J. Love. Realizable Hamiltonians for universal adiabatic quantum computers. *Physical Review A*, 78(1):012352, 2008. arXiv:0704.1287.

[76] Toby Cubitt and Ashley Montanaro. Complexity classification of local Hamiltonian problems. *SIAM Journal on Computing*, 45(2):268–316, 2016. arXiv:1311.3161.

[77] Thomas Vidick and John Watrous. Quantum proofs. *Foundations and Trends in Theoretical Computer Science*, 11(1-2):1–215, 2016. arXiv:1610.01664.

[78] Mark Zhandry. How to construct quantum random functions. In *FOCS 2012*, pages 679–687, 2012. `https://eprint.iacr.org/2012/182`.

[79] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In *Proceedings of Eurocrypt*, pages 592–608. 2013. `https://eprint.iacr.org/2012/606`.

[80] Dave Bacon, Andrew M. Childs, and Wim van Dam. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 469–478. 2005. arXiv:quant-ph/0504083.

[81] Andrew M. Childs and Wim van Dam. Quantum algorithm for a generalized hidden shift problem. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms*, pages 1225–1234. 2007. arXiv:quant-ph/0507190.

[82] Thomas Decker, Jan Draisma, and Pawel Wocjan. Efficient quantum algorithm for identifying hidden polynomials. *Quantum Information and Computation*, 9(3):215–230, 2009. arXiv:0706.1219.

[83] Jaikumar Radhakrishnan, Pranab Sen, and S. Venkatesh. The quantum complexity of set membership. *Algorithmica*, pages 462–479, 2002. arXiv:quant-ph/0007021.

[84] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Limit on the speed of quantum computation in determining parity. *Physical Review Letters*, 81(24):5442–5444, 1998. arXiv:quant-ph/9802045.

[85] Wim van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science*, pages 362–367. 1998. arXiv:quant-ph/9805006.

[86] Ashley Montanaro. The quantum query complexity of learning multilinear polynomials. *Information Processing Letters*, 112(11):438–442, 2012. arXiv:1105.3310.

[87] Johan Håstad. Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644–654, 1990.

[88] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6):45:1–45:39, 2013. arXiv:0911.1393.

[89] Yaroslav Shitov. How hard is the tensor rank?, 2016. arXiv:1611.01559.

[90] Alessandra Bernardi, Grigoriy Blekherman, and Giorgio Ottaviani. On real typical ranks, 2015. arXiv:1512.01853.

[91] Grigoriy Blekherman and Zach Teitler. On maximum, typical and generic ranks. *Mathematische Annalen*, 362(3-4):1021–1031, 2015. arXiv:1402.2371.

[92] Joe Harris. *Algebraic Geometry: A First Course*, volume 133 of *Graduate Texts in Mathematics*. Springer, 1992.

[93] Arnold Knopfmacher and John Knopfmacher. Counting polynomials with a given number of zeros in a finite field. *Linear and Multilinear Algebra*, 26(4):287–292, 1990.

[94] Gerald M. Pitstick, João R. Cruz, and Robert J. Mulholland. A novel interpretation of Prony's method. *Proceedings of the IEEE*, 76(8):1052–1053, 1988.

[95] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2013.

[96] Lars Hörmander. *The Analysis of Linear Partial Differential Operators: Distribution Theory and Fourier Analysis*. Springer, 1983.

[97] F. Palatini. Sulle varietà algebriche per le quali sono di dimensione minore dell'ordinario, senza riempire lo spazio ambiente, una o alcune delle variet'a formate da spazi seganti. *Atti. Accad. Torino*, 44:362–374, 1909.

[98] Bjorn Adlandsvik. Joins and higher secant varieties. *Mathematica Scandinavica*, 61:213–222, 1987.

[99] James Alexander and André Hirschowitz. Polynomial interpolation in several variables. *Journal of Algebraic Geometry*, 4(2):201–222, 1995.

[100] Robin Hartshorne. *Algebraic Geometry*, volume 52 of *Graduate Texts in Mathematics*. Springer, 1977.

[101] Pierre Comon and Giorgio Ottaviani. On the typical rank of real binary forms. *Linear and multilinear algebra*, 60(6):657–667, 2012. arXiv:0909.4865.

[102] Antonio Causa and Riccardo Re. On the maximum rank of a real binary form. *Annali di Matematica Pura ed Applicata*, 190(1):55–59, 2011. arXiv:1006.5127.

[103] Anthony V. Geramita. Inverse systems of fat points: Waring's problem, secant varieties of Veronese varieties and parameter spaces for Gorenstein ideals. In *The Curves Seminar at Queen's*, volume 10, pages 2–114, 1996.

[104] Joseph M. Landsberg and Zach Teitler. On the ranks and border ranks of symmetric tensors. *Foundations of Computational Mathematics*, 10(3):339–366, 2010. arXiv:0901.0487.

[105] Andrzej Białynicki-Birula and Andrzej Schinzel. Representations of multivariate polynomials by sums of univariate polynomials in linear forms. *Colloquium Mathematicum*, 112(2):201–233, 2008.

[106] Joachim Jelisiejew. An upper bound for the Waring rank of a form, 2013. arXiv:1305.6957.

[107] Edoardo Ballico and Alessandro De Paris. Generic power sum decompositions and bounds for the waring rank, 2013. arXiv:1312.3494.

[108] Serge Lang and André Weil. Number of points of varieties in finite fields. *American Journal of Mathematics*, 76(4):819–827, 1954.

[109] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[110] J. Niel de Beaudrap, Richard Cleve, and John Watrous. Sharp quantum vs. classical query complexity separations. *Algorithmica*, 34:449–461, 2002. arXiv:quant-ph/0011065.

[111] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014. arXiv:1411.4357.

[112] Cyrus Rashtchian, David P. Woodruff, and Hanlin Zhu. Vector-matrix-vector queries for solving linear algebra, statistics, and graph problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. arXiv:2006.14015.

[113] Troy Lee, Miklos Santha, and Shengyu Zhang. Quantum algorithms for graph problems with cut queries. In *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms*, pages 939–958. SIAM, 2021. arXiv:2007.08285.

[114] Ashley Montanaro and Changpeng Shao. Quantum algorithms for learning graphs and beyond, 2020. arXiv:2011.08611.

[115] Joran van Apeldoorn and Sander Gribling. Simon's problem for linear functions, 2018. arXiv:1810.12030.

[116] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv:0811.3171.

[117] Andrew M. Childs. Equation solving by simulation. *Nature Physics*, 5:861, 2009.

[118] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291, 2015.

[119] Simon Apers and Ronald de Wolf. Quantum speedup for graph sparsification, cut approximation and laplacian solving. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science*, pages 637–648. IEEE, 2020. arXiv:1911.07306.

[120] Scott Aaronson. Quantum lower bound for the collision problem. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 635–642, 2002. arXiv:quant-ph/0111102.

[121] Pascal Koiran, Vincent Nesme, and Natacha Portier. The quantum query complexity of the abelian hidden subgroup problem. *Theoretical Computer Science*, 380(1-2):115–126, 2007.

[122] Stephen P. Jordan. Fast quantum algorithm for numerical gradient estimation. *Physical Review Letters*, 95(5):050501, 2005. arXiv:quant-ph/0405146.

[123] Ankit Garg, Robin Kothari, Praneeth Netrapalli, and Suhail Sherif. No quantum speedup over gradient descent for non-smooth convex optimization. In *12th Innovations in Theoretical Computer Science Conference (to appear)*, 2021. arXiv:2010.01801.

[124] Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, and Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. *Quantum*, 4:221, 2020. arXiv:1809.01731.

[125] Joran van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Convex optimization using quantum oracles. *Quantum*, 4:220, 2020. arXiv:1809.00643.

[126] Jean-Pierre Serre. *Linear Representations of Finite Groups*, volume 42 of *Graduate Texts in Mathematics*. Springer, 1977.

[127] Andries E. Brouwer, Arjeh M. Cohen, and Arnold Neumaier. *Distance-Regular Graphs*. Springer-Verlag, 1989.

[128] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. *Proceedings of Symposia in Applied Mathematics: Quantum Information Science and Its Contributions to Mathematics*, 68, 2010. arXiv:0904.2557.

[129] Yangjia Li and Mingsheng Ying. Algorithmic analysis of termination problems for quantum programs. In *POPL*, 2018.

[130] Michael Carbin, Deokhwan Kim, Sasa Misailovic, and Martin C. Rinard. Proving acceptability properties of relaxed nondeterministic approximate programs. In *PLDI*, 2012.

[131] Mingsheng Ying. *Foundations of Quantum Programming*. Morgan Kaufmann, 2016.

[132] Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani. Quantum walks on graphs. In *STOC*, 2001. arXiv:quant-ph/0012090.

[133] Iulia Georgescu, Sahel Ashhab, and Franco Nori. Quantum simulation. *Reviews of Modern Physics*, 86(1), 2014. arXiv:1308.6253.

[134] Mingsheng Ying, Shenggang Ying, and Xiaodi Wu. Invariants of quantum programs: Characterisations and generation. In *POPL*, 2017.

[135] Mingsheng Ying. Floyd–Hoare logic for quantum programs. *ACM Transactions on Programming Languages and Systems*, 33(6), 2011.

[136] Ellie D'Hondt and Prakash Panangaden. Quantum weakest preconditions. *Mathematical Structures in Computer Science*, 16(3), 2006. arXiv:quant-ph/0501157.

[137] Michael A. Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[138] Barbara M. Terhal. Quantum error correction for quantum memories. *Reviews of Modern Physics*, 87(2), 2015. arXiv:1302.3428.

[139] John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018.

[140] John Watrous. Semidefinite programs for completely bounded norms. *Theory of Computing*, 5(11), 2009. arXiv:0901.4709.

[141] Runzhou Tao, Yunong Shi, Jianan Yao, John Hui, Frederic T. Chong, and Ronghui Gu. Gleipnir: Toward practical error analysis for quantum programs (extended version). 2021. arXiv:2104.06349.

[142] Marcus P. da Silva. matlab-diamond-norm: A MATLAB function for computing the diamond norm (completely bounded induced 1-norm on linear superoperators), 2015.

[143] John Watrous. Simpler semidefinite programs for completely bounded norms. *Chicago Journal of Theoretical Computer Science*, 2013(8), 2013. arXiv:1207.5726.