Algebraic Methods in Query and Proof Complexity

by

Adrian She

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Mathematics
University of Toronto

Abstract

Algebraic Methods in Query and Proof Complexity

Adrian She
Doctor of Philosophy
Graduate Department of Mathematics
University of Toronto
2024

Algebraic methods have become a powerful tool for analyzing the complexity of various computational models, including low-depth circuits, algebraic proofs, and quantum query algorithms. In particular, the complexity of computing a function in these models is related to whether or not the function admits a low-degree polynomial approximation. In this thesis, we present two novel applications of algebraic methods in computational complexity theory.

In the first part of the thesis, we study unitary property testing, where a quantum algorithm is given query access to a black-box unitary and has to decide whether or not it satisfies some property. In addition to containing the classical query complexity model as a special case, this model also contains "inherently quantum" problems that have no classical analogue. Our main contribution is a generalized polynomial method for analyzing the complexity of unitary property testing problems. By leveraging connections with invariant theory, we apply this method to obtain lower bounds on problems such as determining recurrence times of unitaries, approximating the dimension of a subspace, and approximating the entanglement entropy of a state. We also present a candidate problem towards an oracle separation of QMA and $\mathsf{QMA}(2)$, a long standing open question in quantum complexity theory.

In the second part of the thesis, we study the tensor isomorphism problem (TI), which has recently emerged as having connections to multiple areas of research, including quantum information theory, post-quantum cryptography, and computational algebra. However, the current best upper bound is essentially the brute force algorithm. Being an algebraic problem, the study of tensor isomorphism naturally lends itself to algebraic and semi-algebraic proof systems such as the polynomial calculus (PC) and sum-of-squares (SoS). We show a $\Omega(n)$ lower bound on PC degree or SoS degree for tensor isomorphism and a non-trivial upper bound for testing isomorphism of tensors of bounded rank. Along the way, we also show that PC cannot perform basic linear algebra in sublinear degree, such as comparing the rank of two matrices. We introduce a strictly stronger proof system, called PC + Inv, which enables linear algebra to be done in low degree. We conjecture that even PC + Inv cannot solve TI in polynomial time either, and highlight many other open questions about proof complexity approaches to TI.

Acknowledgements

Doing a PhD is normally considered a difficult task. However, doing a PhD during a global pandemic is a even more difficult task. There are many people to thank for advice, support, and collaboration during this time.

Firstly, thank you to my advisors, Henry Yuen and Toni Pitassi. They are exemplary researchers who have stimulated my interest in computational complexity theory, and particularly the two areas of query and proof complexity explored in this thesis. They are also extraordinarily kind individuals, who provided much-needed support during difficult periods of the PhD.

Some others at the University of Toronto have been an important part of my academic journey. I thank Mrinal Kumar for introducing me to the area of algebraic complexity theory and our collaborations during the first year of my PhD. I thank Shubhangi Saraf for agreeing to serve on my committee and stepping in as necessary to provide additional support. I thank Nathan Wiebe for providing advice about research and graduate studies. I thank Fabian Parsch, Cindy Blois, Camelia Karimianpour, Jason Siefken, and Sarah Mayes-Tang for advice and support about teaching responsibilities, and for helping me develop as an effective teacher over the course of my PhD.

I thank my research collaborators, Joshua Grochow and Nicola Galesi, for the fruitful discussions leading to our joint work on tensor isomorphism, included in the second part of this thesis.

I thank my undergraduate advisor at the University of British Columbia, Steph van Willigenburg, for encouraging me to pursue graduate studies and a research career.

I thank the anonymous reviewers of the papers included in this thesis for their feedback, which helped to improve the quality of the exposition and the results.

I thank my colleagues and friends for their support of my graduate school journey. Thank you for my office mates in the Sandford Fleming theory lab for fun discussions (research or otherwise), free food breaks, and board game nights.

I thank my *kasamas*, both in Vancouver and Toronto, for reminding me that there is more to life than just academics. Thank you especially to Reverend Ariel Siagan for your spiritual guidance and our late-night study sessions at Robarts Library.

I thank Jemima Mersica, Jankie Ramsook, and Ingrid Varga for administrative support during the PhD.

I thank the staff on campus whose work often goes unacknowledged, including the cleaning and cafeteria staff who worked tireless during the pandemic.

I thank the National Sciences and Engineering Council of Canada (NSERC), for the Canada Graduate Scholarship that provided me with financial support during the first three years of my PhD.

Finally, thank you for my family for their unconditional support.

Contents

Chapter 1

Introduction

1.1 Motivation

1.1.1 What is Computational Complexity?

Computational complexity theory aims to classify the relative power of computational resources, such as time or space, for solving mathematical problems. Complexity theory classifies problems into various complexity classes. We give an informal overview of complexity theory in this section. For details, consult a complexity theory textbook such as [AB09].

Some examples of complexity classes include:

- P: The class of problems that can be solved in polynomial time by deterministic algorithms.
- BPP: The class of problems that can be solved in polynomial time by randomized algorithms.
- BQP: The class of problems that can be solved in polynomial time by quantum algorithms.
- NP: The class of problems whose solutions can be verified in polynomial time by a deterministic verifier.
- PSPACE: The class of problems that can be solved assuming that a computer has access to a polynomial amount of memory.

The central questions of computational complexity theory aim to resolve whether or not *efficient* algorithms always exist for certain classes of problems, and to discover the trade-offs between various types of computational resources. Some examples of comparisons that can be asked include:

- P versus NP : Does every problem whose solution can be verified efficiently, also have an efficient algorithm? This seminal problem was introduced by Cook in [Coo71].
- P versus BPP: Is every problem solvable by a polynomial time randomized algorithm also solvable by a polynomial time deterministic algorithm?
- BPP versus BQP: Is every problem solvable by a polynomial time quantum algorithm also solvable by a polynomial time randomized algorithm?

Resolving these questions would contribute significantly to our understanding of computation. On one hand, showing that a problem can be solved by an efficient algorithm in some model could lead to practical applications. However, showing a problem is computationally hard also has interesting consequences.

Firstly, it would give a more complete understanding of various algorithms, including when an algorithm is an *optimal* algorithm for solving a particular problem. Next, complexity theory provides some of the foundations of cryptography. Hardness assumptions are used to show the security of various cryptographic systems [AB09, Chapter 10]. Finally, the quest to prove computational lower bounds has stimulated connections between computational complexity theory and other areas of mathematics. These include combinatorics [Lov17], Fourier analysis [O'D14], and algebra [AB09, Chapter 13, 14]. Establishing computational lower bounds remains a interesting and fundamental challenge in mathematics, and has often lead to surprising and deep results.

The study of computational complexity theory is particularly relevant for the nascent field of quantum computing. There appears some evidence that quantum effects can be powerful computational resource (i.e. it is plausible that $\mathsf{BPP} \neq \mathsf{BQP}$). The discovery of Shor's algorithm [Sho99] for factoring integers provided evidence that there could be problems that could be efficiently solved on a quantum computer, but not any classical device. This is because much of modern cryptography relies on the assumption that factoring integers is hard for deterministic or randomized algorithms. With recent experimental progress in building larger quantum devices and various groups claiming that they have achieved "quantum supremacy" [AAB+19, ZDQ+21], there is a tantalizing possibility that quantum computers could provide speed-ups for practically relevant problems compared to classical devices. Understanding which problems are mostly likely amenable to a quantum speed-ups remains an fascinating open problem.

However, despite decades of progress in computational complexity theory, it remains difficult to prove non-trivial computational lower bounds in general computational models such Turing machines or circuits. This motivates the study of *restricted* computational models, which is a necessary step towards establishing general computational lower bounds. In particular, there appears to be a strong connection between algebraic methods and techniques to prove lower bounds in restricted computational models. We overview this connection in the next section.

1.1.2 Algebraic Methods in Computational Complexity Theory

Algebraic techniques involve converting a problem about computing a Boolean function $f : \{0,1\}^n \to \{0,1\}$ in some model into one about algebraic objects, such as polynomials $p \in \mathbb{F}[x_1, \ldots, x_n]$ for some field $\mathbb{F}$. These techniques have been particularly fruitful for studying restricted models of computation, such as low-depth circuits, algebraic proofs, and quantum query algorithms. We focus on techniques involving the notion of *approximate degree*, which captures the minimum degree necessary for a polynomial p to approximate a function f in some sense. This notion now has many avatars throughout computational complexity theory [BT+22] and depends on the underlying field $\mathbb{F}$.

Approximation over a finite field. Firstly, we can consider the setting where p is a polynomial with coefficients in a finite field with a prime number of elements $\mathbb{F}_p$. In this setting, we say that a polynomial p approximates function f with error ϵ if

$$\Pr_{x \in \{0,1\}^n}[p(x) \neq f(x)] \leq \epsilon$$

where the probability is over a uniformly random chosen input in $\{0,1\}^n$.

This notion of approximation is particularly relevant in circuit complexity, for the study of *constant-depth* circuits. Recall that the *depth* of a circuit is the maximum length of input to output path in the circuit. A family of circuits $\{C_n\}$ has constant-depth if there exists an integer d such that every circuit C_n in the family has depth at most d. We say that a constant-depth circuit family is in $\mathsf{AC}^0[2]$ if the gate set used includes unbounded fan-in AND, OR, NOT, and PARITY gates, where PARITY is the function defined by

$$\mathsf{PARITY}(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i \mod 2.$$

Razborov-Smolensky [Smo87] observed that function f that are computable by small-depth $\mathsf{AC}^0[2]$ circuits have low-degree polynomial approximations over $\mathbb{F}_2$ in the sense described above. This is now known as the Razborov-Smolensky method of approximations. In particular, there are relatively simple functions that do not have polynomially-sized $\mathsf{AC}^0[2]$ circuits, such as the MOD_3 function, defined by

$$\mathsf{MOD}_3(x_1, \ldots, x_n) = \begin{cases} 1 & \sum_{i=1}^{n} x_i \mod 3 = 1 \\ 0 & \text{otherwise} \end{cases}.$$

This is shown by establishing that no low-degree polynomial over $\mathbb{F}_2$ can approximate MOD_3 in the sense described above [AB09, Chapter 13.2]. This was one of the first lower bounds to be shown for the $\mathsf{AC}^0[2]$ circuit class, and remains one of the strongest circuit lower bounds proven to this day.

While some techniques from circuit complexity can sometimes be "lifted" to prove lower bounds in the setting of proof complexity, it is still open to prove lower bounds for $\mathsf{AC}^0[2]$-Frege proofs. As an attempt to generalize the Razborov-Smolensky bounds to the proof complexity setting, proof systems that manipulate polynomials were introduced. These are known as algebraic proof systems. This observation was first used by Beame et al. [BIK+96], which used degree lower bounds in Hilbert's Nullstellensatz to deduce lower bounds for a subsystem of $\mathsf{AC}^0[2]-$Frege. Other algebraic proof systems such as polynomial calculus [CEI96a] were subsequently introduced, and degree lower bounds for these systems have also been studied [Raz98a].

Approximation over the reals. Secondly, we can also consider approximation of Boolean functions by polynomials with real coefficients. In this setting, we say that a polynomial p approximates a function f with error ϵ if pointwise for each $x \in \{0,1\}^n$ we have

$$|p(x) - f(x)| \leq \epsilon.$$

Again, we say that the ϵ-approximate degree $deg_\epsilon(f)$ of a function is the minimum degree of a polynomial p for which a pointwise approximation to error ϵ is possible.

This notion of approximation is also relevant in computational complexity theory, particularly in the query model. The basic example of a query model is the model of deterministic decision trees. A decision tree computes a function $f(x_1, \ldots, x_n)$ by querying variables x_i, branching based on the value of x_i, and outputing a value when a leaf of the tree is reached. The depth of a decision tree is the maximum length of a root-to-leaf path, and is an important measure for the complexity of a decision tree.

Approximation by real polynomials, in the context of classical complexity theory, was introduced

by Nisan and Szegedy as [NS94]. There, they showed that the approximate degree of a function f is polynomially related to its decision tree depth. Furthermore, they show that approximate degree is lower bounded by other complexity measures of f such as its block-sensitivity. In [BDW02], it is also observed that the same holds true when considering randomized decision trees, in addition to deterministic decision trees.

The observation that polynomial degree is a lower bound for query complexity was then adapted for quantum algorithms by Beals, et al. [BBC+01]. Afterwards, it was recognized that the polynomial method is a powerful technique to lower bound the quantum query complexity of a variety of problems (see, e.g., [AS04, BKT18], and the references therein). These lower bound techniques have been used to give insights on the optimality of quantum algorithms and the maximum possible quantum speed-ups achievable in the query model.

1.2 Contributions and Organization of the Results

The contributions of this thesis are to investigate two domains, namely query complexity and proof complexity, where algebraic techniques can be applied to investigate lower bounds. We introduce and develop new algebraic techniques for proving novel computational lower bounds in these domains.

1.2.1 Unitary Property Testing

In the first part of the thesis, based on [SY22], we investigate the quantum query model. The query model of quantum algorithms plays a central role in the theory of quantum computing. In this model, the algorithm queries (in superposition) bits of an unknown input string X, and after some number of queries decides whether X satisfies a property $\mathcal{P}$ or not. We now have an extensive understanding of the query complexity of many problems; we refer the reader to Ambainis's survey [Amb18] for an extensive list of examples.

Although this query model involves quantum algorithms, the task being solved is *classical property testing*, that is, deciding properties of classical strings. This has been very useful for comparing the performance of classical versus quantum algorithms for the same task. In contrast, *quantum property testing* – deciding properties of quantum objects such as states and unitaries – has been been studied much less but has been receiving more attention in recent years [MdW13]. These are examples of "inherently quantum" problems which can be solved on a quantum computer. There has been a recent series of work [Ros23, Kre23] in trying to develop complexity theory for inherently quantum problems, since techniques of classical complexity theory do not immediate generalize to this setting.

In this thesis, we focus on *unitary property testing*, where the goal is to decide whether a unitary U satisfies a property $\mathcal{P}$ by making as few queries to U as possible. This is an inherently quantum problem, and the systematic study of this topic was initiated by Wang [Wan11]. Various aspects have been studied further in [MdW13, CNY22, ACQ22]. Some examples of unitary property testing problems include:

- Approximate dimension: promised that U applies a phase to all states $|\psi\rangle$ in a subspace S of dimension either at w or $2w$, determine the dimension of the subspace. This is analogous to the classical problem of approximating the Hamming weight of an input X. This was studied in [AKKT20].

- Unitary recurrence times: Determining whether $U^t = I$ or $\|U^t - I\| \geq \epsilon$ (promised that one is the case) where t is a fixed integer.
- Hamiltonian properties: Promised that $U = e^{-iH}$ for some Hamiltonian H with bounded spectral norm, determine properties of H, such as whether it is a sum of k-local terms, or the ground space is topologically ordered.
- Unitary subgroup testing: decide whether U belongs to some fixed subgroup of the unitary group (such as the Clifford subgroup). This was studied in [BSW21].
- Entanglement entropy problem: Given access to a unitary $U = I - 2|\psi\rangle\langle\psi|$ for some state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$, decide if the entanglement entropy of the state $|\psi\rangle$ is low or high, promised one is the case.

These examples illustrate the rich variety of unitary property testing problems: some are motivated by well-studied classical problems in computer science (such as junta testing and approximate counting), whereas others are inspired by questions in quantum physics (e.g., identifying quantum chaos, topological order, or entanglement).

We continue explorations of this topic by developing a new lower bound technique based on the polynomial method, in particular approximation by real polynomials. In Chapter 2, we provide background material that is helpful context towards understanding the main results of this work.

- In Section 2.1, we provide background on quantum circuits and the quantum query model.
- In Section 2.2, we provide background on the polynomial method for classical property testing.
- In Section 2.3, we provide background on the quantum complexity classes studied in our work, including the quantum complexity class QMA.

This is followed by Chapter 3 that contains our main results on unitary property testing. Section 3.1 states some lemmas and preliminaries we will need towards proving our main results.

The Generalized Polynomial Method and Applications

In Section 3.2, we prove our main results on the generalized polynomial method for unitary property testing and its applications. We note that we allow queries to both U and U^* as we are interested in lower bounds for the strongest possible query model.

Proposition 1.2.1 (Generalized polynomial method)**.** *The acceptance probability of a quantum algorithm making T queries to a $d \times d$ unitary U and its inverse U^* can be computed by a degree at most $2T$ self-adjoint[1] polynomial $p : \mathbb{C}^{2(d\times d)} \to \mathbb{C}$ evaluated at the matrix entries of U and U^*. Thus, degree lower bounds on such polynomials yields a query lower bound on the algorithm.*

Furthermore, we say that a unitary property $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$ is closed under inversion if $U \in \mathcal{P}_{yes}$ iff $U^* \in \mathcal{P}_{yes}$, and $U \in \mathcal{P}_{no}$ iff $U^* \in \mathcal{P}_{no}$. All properties we will study in this paper will be closed under inversion, and hence the polynomial p satisfies a symmetry under this condition.

[1] A self-adjoint polynomial is unchanged after complex conjugating every variable and every coefficient.

Proposition 1.2.2. *Let $\mathcal{P}$ be an property closed under inversion and suppose there is a T-query quantum algorithm for testing property $\mathcal{P}$. Let p be the polynomial from Proposition 1.2.1 that computes the acceptance probability of the algorithm. Then, we may assume that $p(U, U^*) = p(U^*, U)$.*

Hence, while establishing the existence of p is straightforward, proving lower bounds on its degree is another matter. The standard approach in quantum query complexity is to *symmetrize* p to obtain a related polynomial q whose degree is not too much larger than p, and acts on a much smaller number of variables (ideally a single variable). However, for unitary properties, a symmetrization technique for other properties is less clear. In this direction, we develop symmetrization techniques based on invariant theory. This provides an intriguing connection between unitary property testing and invariant theory, which is a classical area of mathematics.

To connect invariant theory with our Proposition 1.2.1, we prove the following result for testing G-invariant unitary properties. Since we are studying properties of general unitaries, not just boolean strings, we consider symmetries coming from subgroups of the unitary group $\mathrm{U}(d)$. Let $G \subseteq U(d)$ be a compact subgroup equipped with a Haar measure μ (i.e., a measure over G that is invariant under left-multiplication by elements of G).

Definition 1.2.1 (G-invariant property)**.** Let $G \subseteq U(d)$ be a compact group. A d-dimensional unitary property $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$ is G-invariant if for every $g \in G$ we have $g\mathcal{P}_{yes}g^{-1} \subseteq \mathcal{P}_{yes}$ and $g\mathcal{P}_{no}g^{-1} \subseteq \mathcal{P}_{no}$.

For G-invariant properties, we show that an approximating polynomial can always be chosen in the invariant ring $\mathbb{C}[X, X^*]^G_d$, where G acts on polynomials of degree d in the natural way.

Proposition 1.2.3 (Symmeterization for G-invariant properties)**.** *Suppose $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$ is a G-invariant d-dimensional unitary property. If there is a T-query tester for $\mathcal{P}$ that accepts* yes *instances with probability at least a and* no *instances with probability at most b, then there exists a self-adjoint degree-$2T$ polynomial q in the invariant ring $\mathbb{C}[X, X^*]^G_d$ satisfying*

- *If $U \in \mathcal{P}_{yes}$, then $q(U, U^*) \geq a$.*
- *If $U \in \mathcal{P}_{no}$, then $q(U, U^*) \leq b$.*

While Proposition 1.2.3 at first may seem difficult to apply, the invariant ring has been characterized in numerous cases. Depending on the group, the associated invariant ring may have a much simpler description than the full polynomial ring, making it easier to prove degree lower bounds.

We demonstrate the utility of generalized polynomial method by providing applications to several unitary property testing problems. Furthermore, we study lower bounds in both the BQP and QMA settings. A formal definition of the models is provided in Section 2.3. We show query lower bounds in both the QMA and BQP settings, thereby showing that quantum proofs cannot reduce the query complexity significantly for the problems we study.

We note that in the BQP setting, our lower bounds can also be obtained by other methods, such as the "hybrid method" of [BBBV97]. However, it is unclear how to apply this method in the QMA setting, and hence the polynomial method appears necessary to prove non-trivial QMA lower bounds. Furthermore, even in the BQP setting, we believe that the polynomial method provides a clean and simple method to prove lower bounds compared to other methods.

Unitarily Invariant Subspace Properties. As a warmup, consider *subspace properties*, which consist of reflections about a subspace, i.e., $U = I - 2\Pi$ where Π is the projector onto some subspace $S \subseteq \mathbb{C}^d$. We say that U *encodes* the subspace S. An example of a unitarily invariant subspace property is the *Approximate Dimension* problem, which we parametrize by an integer $w \in \{1, 2, \ldots, d\}$. The *yes* instances consist of (unitaries encoding) subspaces of dimension at least $2w$, and the *no* instances consist of subspaces of dimension at most w. This is a quantum generalization of the *Approximate Counting* problem, which is to determine whether the Hamming weight of an input string is at least $2w$ or at most w.

We observe that there is a one-to-one correspondence between symmetric classical properties $\mathcal{S}$ (properties that only depend on the Hamming weight of the input) and unitarily invariant subspace properties $\mathcal{P}$. We make this correspondence precise in the following theorem, which comes from Theorem 3.2.2, a special case of Proposition 1.2.3 for unitarily invariant properties.

Proposition 1.2.4. *Let $\mathcal{P}$ be a unitarily invariant subspace property and let $\mathcal{S}$ be the associated symmetric classical property. The query complexity of distinguishing between* yes *and* no *instances of $\mathcal{P}$ is at least the minimum degree of any polynomial that distinguishes between the* yes *and* no *instances of $\mathcal{S}$.*

Therefore, degree lower bounds on polynomials that decide a classical symmetric property $\mathcal{S}$, automatically yield query complexity lower bounds for the quantum property $\mathcal{P}$. For instance, we obtain the following lower bounds for the Approximate Dimension problem from the corresponding classical results, including the QMA lower bound of Aaronson et al in [AKKT20].

Theorem 1.2.5 (BQP lower bound for Approximate Dimension)**.** *Any tester that decides between whether a unitary encodes a subspace of dimension at least $2w$ or at most w requires $\Omega(\sqrt{\frac{d}{w}})$ queries.*

Theorem 1.2.6 (QMA lower bound for Approximate Dimension)**.** *Suppose there is a T-query algorithm that solves the Approximate Dimension problem (i.e. deciding whether a d-dimensional unitary encodes a subspace of dimension at least $2w$ or at most w) with the help of a m-qubit proof. Then either $m = \Omega(w)$, or $T \geq \Omega(\sqrt{\frac{d}{w}})$.*

Recurrence Time of Unitaries. Not all unitarily invariant properties reduce to classical lower bounds. For instance, we analyze a problem related to the recurrence times of unitaries.

Definition 1.2.2 (Recurrence Time Problem)**.** The (t, ϵ)-Recurrence Time problem is to decide, given oracle access to a unitary U, whether $U^t = I$ (*yes* case) or $\|U^t - I\| \geq \epsilon$ in the spectral norm (*no* case), promised that one is the case.

Note that the instances of this problem are generally not self-adjoint; their eigenvalues can be any complex number on the unit circle. There is no obvious classical analogue of the unitary Recurrence Time problem, and thus it does not seem to naturally reduce to a classical lower bound. We instead employ Theorem 3.2.2 to prove the following lower bound on the Recurrence Time problem in the BQP setting:

Theorem 1.2.7. *Let $\epsilon \leq \frac{1}{2\pi}$. Any quantum query algorithm solving the (t, ϵ)-Recurrence Time problem for d-dimensional unitaries with error ϵ must use $\Omega(\max(\frac{t}{\epsilon}, \sqrt{d}))$ queries.*

We also establish the following upper bound in the BQP setting:

Theorem 1.2.8. *The (t, ϵ)-Recurrence Time problem can be solved using $O(t\sqrt{d}/\epsilon)$ queries.*

Using a similar technique as the BQP lower bound, we also obtain a QMA lower bound for the same problem.

Theorem 1.2.9 (QMA lower bound for the Recurrence Time problem)**.** *Let $\epsilon \leq \frac{1}{2\pi}$. Suppose there is a T-query algorithm that solves the Recurrence Time problem for d-dimensional unitaries with the help of an m-qubit proof. Then either $m \geq \Omega(d)$, or $T \geq \Omega(\max(\sqrt{\frac{d}{m}}, \frac{t}{m}, \frac{1}{\epsilon}))$.*

We leave finding a matching upper bound in the QMA setting as an open problem.

Entanglement Entropy Problem We illustrate further illustrate the connection between quantum query complexity and invariant theory by considering property testing questions related to *entanglement* of quantum states, which is a central concept in quantum information theory. Recall that a state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ is *entangled* if it cannot be written as a tensor product of two states $|\psi_1\rangle \otimes |\psi_2\rangle$ where $|\psi_1\rangle, |\psi_2\rangle \in \mathbb{C}^d$. The *entanglement entropy* of a state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ can be used as a measurement of how entangled the state is.

We now define the Entanglement Entropy problem as the task of distinguishing between high and low entropy states. We let $H_2(|\psi\rangle)$ denote the Rényi two-entropy of state $|\psi\rangle$, defined by $H_2(|\psi\rangle) = -\log \mathrm{Tr}(\rho^2)$ where ρ is the reduced density matrix of the a state $|\psi\rangle$.

Definition 1.2.3 (Entanglement Entropy Problem)**.** Let $0 < a < b \leq \log d$. Given oracle access to a reflection oracle $U = I - 2|\psi\rangle\langle\psi|$ where $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$, decide whether or not the state $|\psi\rangle$ satisfies one of the following two conditions, promised one of the following is the case:

- Low entropy case: $H_2(|\psi\rangle) \leq a$
- High entropy case: $H_2(|\psi\rangle) \geq b$

Since entanglement entropy of a state is an invariant quantity under the local unitary group, we can use our generalized polynomial method and Proposition 1.2.3 to prove a lower bound.

Theorem 1.2.10. *Assume $a \geq 5$. Given parameters $a < b \leq \log d$, any tester must make $\Omega(\exp(a/4))$ queries to distinguish between the low and high entropy cases in the Entanglement Entropy problem.*

We observe that in the limit where $a = O(\log d)$, which is testing if a state is close to maximally entangled or not, this yields a $\Omega(d^{1/4})$ lower bound in terms of the dimension d of the underlying quantum state.

Finally, we also adapt the technique for the BQP lower bound for the Entanglement Entropy problem, to prove a QMA lower bound for the same problem.

Theorem 1.2.11 (QMA lower bound for the Entanglement Entropy problem)**.** *Assume $a \geq 5$ and $a < b \leq \log d$ Suppose there is a T-query algorithm that solves the entanglement entropy problem with the help of an m-qubit witness, then $mT \geq \Omega(\exp(a/4))$.*

We hope that our connection between invariant theory and quantum query complexity can be used as a general framework to prove new lower bounds.

QMA versus QMA(2)

Section 3.3 illustrates the connections between unitary property testing and the complexity class QMA(2). We define the QMA(2) complexity class formally in Section 2.3. Our main contribution in this section is to introduce a problem called the Entangled Subspace problem,. It is a unitary property testing problem solvable by the QMA(2) property testing model but we conjecture that it is not solvable by the QMA property testing model. While we do not obtain a lower bound, we present some observations that may be helpful towards eventually obtaining the desired separation.

In order to define the problem, we first have to define the notion of an ϵ-*completely entangled subspace*. This is a subspace $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$ such that all states $|\theta\rangle \in S$ are ϵ-far in trace distance from any product state $|\psi\rangle \otimes |\phi\rangle$. It is known, via the probabilistic method, that there exist subspaces of dimension $\Omega(d^2)$ that are $\Omega(1)$-completely entangled [HLW06]. We now introduce the Entangled Subspace problem:

Definition 1.2.4 (Entangled Subspace problem)**.** Let $0 \leq a < b < 1$ be constants. The (a, b)-Entangled Subspace problem is to decide, given oracle access to a unitary $U = I - 2\Pi$ where Π is the projector onto a subspace $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$, whether

- (*yes* case) S contains a state $|\theta\rangle$ that is a-close in trace distance to a product state $|\psi\rangle \otimes |\phi\rangle$.
- (*no* case) S is b-completely entangled

promised that one is the case.

We observe that there is in fact a QMA(2) upper bound for the Entangled Subspace problem, which is almost immediate from the definition of QMA(2):

Proposition 1.2.12 (QMA(2) upper bound for the Entangled Subspace problem)**.** *The Entangled Subspace problem can be solved by a* QMA(2) *tester, meaning that the tester receives a proof state in the form* $|\psi\rangle \otimes |\varphi\rangle$ *of* $\operatorname{poly}\log(d)$ *qubits, makes one query to the unitary* U*, and can distinguish between* yes *and* no *cases with constant bias.*

We conjecture the following QMA lower bound on the Entangled Subspace problem.

Conjecture 1. *Any* QMA *tester for the Entangled Subspace problem that makes* T *queries to the oracle and receives an* m*-qubit witness must have either* m *or* T *be superpolynomial in* $\log d$*.*

We observe that the Entangled Subspace property is invariant under the local unitary group: applying local unitaries $g \otimes h$ to a subspace S preserves whether it is a *yes* instance or a *no* instance of the problem. Thus one can hope to prove query lower bounds for the Entangled Subspace problem in both the BQP and QMA setting using our generalized polynomial method and tools from invariant theory. If this conjecture is true, then this would imply the existence of a quantum oracle that separates QMA from QMA(2): the oracle would encode, for each QMA tester, an instance of the Entangled Subspace problem that the tester decides incorrectly.

Average Case Problems. We also propose two *average case* variants of the Entangled Subspace problem, in which the task is to distinguish between two distributions over unitaries U. Let U be a Haar-random matrix on $\mathbb{C}^d \otimes \mathbb{C}^d$. A Haar-random subspace of dimension s is the image of UP_S, where P_S is a projector onto any fixed subspace $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$ of dimension s.

Definition 1.2.5 (Planted Product State Problem)**.** Let $0 < s < d^2$ denote an integer parameter. Consider the following two distributions over subspaces S of $\mathbb{C}^d \otimes \mathbb{C}^d$:

- **No planted state**: S is a Haar-random subspace of dimension s.
- **Has planted state**: S is an $(s+1)$-dimensional subspace chosen by taking the span of a Haar-random s-dimensional subspace with a product state $|\psi\rangle \otimes |\phi\rangle$ for Haar-random $|\psi\rangle, |\phi\rangle$.

The Planted Product State problem is to distinguish, given oracle access to a unitary $U = I - 2\Pi$ encoding a subspace S, whether S was sampled from the **No planted state** distribution (*no* case) or the **Has planted state** distribution (*yes* case), promised that one is the case.

Definition 1.2.6 (Restricted Dimension Counting Problem)**.** Let $0 < t \leq d$ and $0 < r \leq t^2$ denote integer parameters. Consider the following distribution, parameterized by (t, r), over subspaces $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$:

- Sample Haar-random t-dimensional subspaces $R, Q \subseteq \mathbb{C}^d$.
- Sample a Haar-random r-dimensional subspace of $S \subseteq R \otimes Q$.

Let $0 < C_1 < C_2 < 1$ denote constants. The Restricted Dimension Counting problem is to decide, given query access to a unitary $U = I - 2\Pi$ encoding a subspace S, whether S was sampled from either the $(t, C_1 t^2)$ distribution or $(t, C_2 t^2)$ distribution, promised that one is the case.

The relationship between these two average case problems and the Entangled Subspace problem is captured by the following propositions.

Proposition 1.2.13. *If S is sampled from the **Has planted state** distribution of the Planted Product State problem, then it is a* yes *instance of the Entangled Subspace problem. If S is sampled from the **No planted state** distribution with $s = Cd^2$ for some sufficiently small constant $C > 0$, then it is a* no *instance with overwhelming probability.*

Proposition 1.2.14. *There exist constants $0 < C_1 < C_2 < 1$ such that if S is sampled from the $(t, C_1 t^2)$ distribution from the Restricted Dimension Counting problem, it is a* no *instance of the Entangled Subspace problem with overwhelming probability. If it is sampled from the $(t, C_2 t^2)$ distribution, then it is a* yes *instance with overwhelming probability.*

These two propositions are proved using methods from random matrix theory. Proposition 1.2.12 in turn implies that the Planted Product State and Restricted Dimension Counting problems can be solved by a $\mathsf{QMA}(2)$ tester with overwhelming probability.

Our strongest evidence towards showing that the Entangled Subspace problem or one of its average case variants is hard for QMA is a lower bound against QCMA testers. This is the subclass of QMA where the quantum verifier only has access a to a classical proof.

Theorem 1.2.15 (Informal version of Theorem 3.3.13)**.** *Any T-query quantum algorithm solving the Planted Product State problem with the help of an m-bit classical witness must have m or T superpolynomial in* $\log d$.

Finally, Section 3.4 states some open problems about unitary property testing that are potentially interesting for future work.

1.2.2 The Proof Complexity of Tensor Isomorphism

In the second part of the thesis, we investigate the power of *algebraic proof systems* for the tensor isomorphism problem. This section of the thesis is based on [GGPS23].

Tensors are fundamental data structures in linear algebra and throughout various areas of science. The fundamental notion of equivalence between tensors is that of isomorphism: two tensors are isomorphic if one can be transformed into the other by an invertible linear change of basis in each of the corresponding vector spaces. For example, two 2-tensors (=matrices) M, M' are equivalent under this notion if there are invertible matrices X, Y such that $XMY = M'$; similarly, two 3-tensors, represented by 3-way arrays T_{ijk}, T'_{ijk} are isomorphic if there are three invertible matrices X, Y, Z such that

$$\sum_{ijk} X_{ii'} Y_{jj'} Z_{kk'} T_{ijk} = T'_{i'j'k'} \tag{1.1}$$

for all i', j', k'. The problem of (3-)Tensor Isomorphism (TI) is: given two such 3-way arrays, to decide if they are isomorphic.

Tensor isomorphism is a fundamental algebraic question in quantum information theory, post-quantum cryptography, and computational algebra.

In quantum information theory, multipartite quantum states can be represented by tensors. The tensor isomorphism question is then related to entanglement of quantum states. In particular, imagine that a tripartite quantum state $|\psi\rangle \in (\mathbb{C}^d)^{\otimes 3}$ is shared among three parties. The parties are able to perform unitary operations on their own part of the state, and communicate classical information among themselves. The goal of the parties is to produce some state $|\psi\rangle'$ with some non-zero probability by means of these operations. We call states $|\psi\rangle, |\psi\rangle'$ SLOCC-equivalent if such a transformation is possible, where SLOCC stands for stochastic local operations and classical communication. In [DVC00], it was observed that SLOCC-equivalence is equivalent to the condition that the states $|\psi\rangle, |\psi\rangle'$ are isomorphic as tensors. As such, the tensor isomorphism question is closely related to classification of the types of quantum entanglement a quantum state may possess.

In post-quantum cryptography, hardness assumptions related to tensor isomorphism and related problems have been used to develop digital signature schemes among other cryptographic primitives [JQSY19, TDJ+22]. This is because as observed in [JQSY19], tensor isomorphism can be viewed as a type of hidden subgroup problem over the general linear group. Known quantum algorithmic techniques, such as quantum Fourier sampling, cannot solve hidden subgroup problems over the general linear group, in general [DMR10].

In computational algebra, there has been a rich theory being developed around the tensor isomorphism problem. In particular, it was observed that tensor isomorphism has a completeness property in [GQ21b], who showed numerous reduction betweens tensor isomorphism and other problems in computational algebra. In particular, it was shown that tensor isomorphism is equivalent to group isomorphism for p-groups, matrix space isometry and conjugacy, and isomorphism of algebras. This shows that the improvements in developing algorithms for tensor isomorphism could lead to improvements for algorithms in other algebraic domains.

The leading methods to solve tensor isomorphism problems are largely based on Gröbner bases techniques [TDJ+22, FP06]. This is because tensor isomorphism is equivalent to solving a system of polynomial equations, for which Gröbner bases can be used to decide if a solution exists. It is then natural to ask if these algorithms are always efficient for tensor isomorphism problems, or whether

or not there are hard instances. Algebraic proof complexity provides a systematic way to answering these problems. In particular, the *polynomial calculus* proof systems introduced in [CEI96a] provides a method for understanding Gröbner bases algorithms. We initiate the study of algebraic proof complexity approaches to tensor isomorphism in this section of the thesis.

This line is work is also motivated by the more famous *Graph Isomorphism (GI)* problem, where proof complexity plays an important role. Indeed, [GQ21b] observe that the Graph Isomorphism problem reduces to the Tensor Isomorphism problem. Although the Wieslefer-Lehman (WL) algorithm does not, on its own, solve GI in polynomial time [CFI92], it is a key subroutine in many of the best algorithms for GI, both in theory [Bab16] and in practice (see [McK81, MP14]). And the picture that has emerged is that some proof systems for GI are known to be equivalent in power to WL [AM13, BG15], and some lower bounds on proof systems are closely related to lower bounds for WL [SSC14, OWWZ14, BG15]. Versions of WL for groups, and in particular finite p-groups—and hence, by the connection above, tensors over finite fields—have only recently begun to be explored [BGL$^+$19, BS20, BS22, CL22].

In Chapter 4 we provide background on algebraic proof systems that we discuss in this section of the thesis, including Nullstellensatz, polynomial calculus, and sum-of-squares. This is followed by Chapter 5 containing the main results from our study of the tensor isomorphism problem. Section 5.1 discusses preliminaries, including a formal definition of the tensor isomorphism problem and polynomial encodings of various tautologies.

PC for Linear Algebra As a warm up, we discuss PC lower bounds for various linear algebraic principle in Section 5.2. This serves as a warm-up for establishing lower bounds for tensor problems, since isomorphism of 2-tensors is equivalent to deciding whether or not two matrices have the same rank.

Some basic derivations in linear algebra are to relate the ranks of two matrices and to derive $BA = I$ from $AB = I$ (the Inversion Principle, one of the so-called "hard matrix identities" [SC04], only recently shown to have short NC^2-Frege proofs [HT15]). Soltys [Sol01] and Soltys & Cook [SC04] discuss the relationship between these and other standard implications in linear algebra. We show that PC is not strong enough to prove these in low-degree:

Theorem 1.2.16. *The unsatisfiable system of equations $XY = \mathrm{Id}_n$ where X is $n \times r$ and Y is $r \times n$ with $1 \leq r < n$, requires degree $\geq r/2 + 1$ to refute in PC, over any field.*

We refer to this system of equations as the Rank Principle, as refuting them amounts to showing that $\mathrm{rk}\,\mathrm{Id}_n > r$.

Theorem 1.2.17. *Any PC derivation of $BA = I$ from $AB = I$, where A, B are $n \times n$ matrices with $\{0, 1\}$ entries, requires degree $\geq n/2 + 1$, over any field.*

We also observe that the Rank Principle can be derived in low degree from the Inversion Principle.

Upper Bounds for Tensor Isomorphism This is followed by Section 5.3, where we discuss an upper bound for tensor isomorphism of bounded-rank tensors. In particular, we show that isomorphism of bounded-rank tensors can be decided in polynomial time. This uses the degree upper bounds from the Effective Nullstellensatz.

Theorem 1.2.18. *Over any field, the Nullstellensatz degree of refuting isomorphism of two $n \times n \times n$ tensors of tensor rank $\leq r$ is at most $2^{O(r^2)}$. If working over a finite field $\mathbb{F}_q$ and including the equations $x^q - x$, the PC degree is at most $O(qr^2)$.*

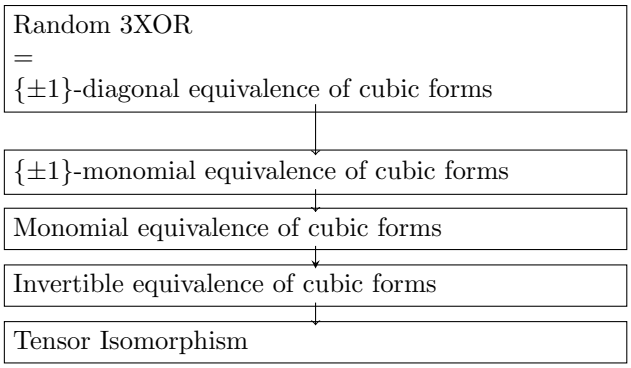

Figure 1.1: Reductions in Section 5.5

In particular, isomorphism of constant-rank tensors can be decided in polynomial time.

We note that the naive degree upper bound from an application of the Effective Nullstellensatz is exponential in the number of variables. For $n \times n \times n$ tensors, this gives an upper bound of $2^{O(n^2)}$ [Som99], and thus, Theorem 1.2.18 gives nontrivial upper bounds all the way up to $r \leq n$.

Lower Bounds for Tensor Isomorphism Finally, in the remaining sections, we present degree lower bounds in polynomial calculus for a large class of instances of tensor isomorphism. Therefore, this gives evidence that Gröbner bases alone cannot solve tensor isomorphism efficiently. This is important for cryptographic applications where evidence of computational hardness can be used to establish security of the a given cryptographic system.

Theorem 1.2.19. *Over any field, there are instances of $n \times n \times n$ Tensor Isomorphism that require PC degree $\Omega(n)$ to refute. Over $\mathbb{R}$, they also require Sum-of-Squares degree $\Omega(n)$ to refute.*

In Section 5.4, we present a reduction from graph isomorphism to the tensor isomorphism equations. The preceding goes by reduction from known lower bounds on PC for Graph Isomorphism [BG15, BG17], but has the disadvantage (from the tensor point of view) that the resulting tensors are quite sparse: in one direction, one of the slices is supported on an $\Omega(n) \times n$ matrix and all the others slices have support size 1.

In a second proof (Section 5.5), we get a polynomially worse lower bound $\Omega(\sqrt[4]{n})$, but with a reduction from Random 3XOR that is more direct. Indeed, we show that 3XOR itself can be viewed as a particular instance of a tensor problem *without* gadgets; gadgets are only then needed to reduce from that tensor problem to Tensor Isomorphism itself. To obtain these results, we show our results a series of low-degree reductions, carried out within the PC proof system. This sequence is illustrated in Figure 1.1.

Our technical contributions in the above theorem are thus three-fold:

1. We show that the known reductions from GI to TI can be carried out in low-degree PC;
2. We realize 3XOR very naturally as a tensor problem; and
3. We give new reductions from 3XOR, through a series of tensor-related problems, to TI, that work as many-one reductions of the decision problems that can be carried out in low-degree PC.

Finally, we discuss open problems in Section 5.6, including a tantalizing conjecture about the power of linear algebra for solving tensor problems.

Part I

Unitary Property Testing

Chapter 2

Quantum Query Complexity

In this section we provide some background on quantum query complexity and quantum complexity theory that will be helpful towards understanding the main results of the thesis.

2.1 Quantum Query Algorithms

We first briefly describe the model of *quantum circuits*, which is the standard model used to describe quantum computations.

A d-dimensional quantum state $|\psi\rangle$ is a unit vector in $\mathbb{C}^d$. If a system contains n qubits, then $d = 2^n$.

A quantum circuit manipulates quantum states by applying unitary operators U, which are operators satisfying the condition that $UU^* = I$ where I is the identity matrix. The unitary operators in a quantum circuit are composed from unitary gates acting only on one or two qubits only.

Recall that in classical computation, a discrete set of logic gates (eg. the AND, OR, and NOT gates) is sufficient to perform any classical computation. Similarly, in quantum computation, a *universal gate set* enables any quantum computation to be performed. In particular, a universal gate set $\mathcal{S}$ guarantees that any unitary operator U can be approximated to arbitrary precision, by applying a some finite sequence of operations from the set $\mathcal{S}$. A common choice of universal gate set of $\mathcal{S} = \{H, CNOT, S, T\}$ where the gates are defined in the following way:

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \quad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\frac{\pi}{4}) \end{bmatrix}.$$

For the proof that the gate set $\mathcal{S}$ is universal, consult [NC10, Chapter 4.5] and [NC10, Appendix 3].

Classical information can be extracted from a quantum circuit by means of a quantum measurement. The probability that a measurement will produce a given outcome is given by Born's rule. For example, suppose the final state of the circuit was

$$|\psi\rangle = \alpha\,|0\rangle\,|\psi_0\rangle + \beta\,|1\rangle\,|\psi_1\rangle\,,$$

for some quantum states $|\psi_0\rangle$ and $|\psi_1\rangle$. If the first qubit was measured in the standard computational

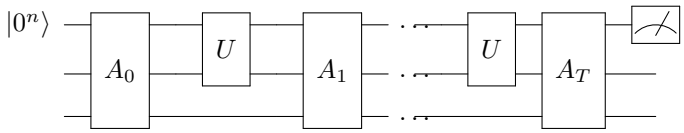

Figure 2.1: A Generic Quantum Query Algorithm

basis, then the circuit would output 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.

Unfortunately, as in classical circuit complexity, there is no known technique that yields general lower bounds on the gate complexity of computing a given unitary operation. However, there are other complexity measures where lower bounds become more tractable. We now turn to a particular model of quantum algorithms, known as quantum query algorithms.

The Quantum Query Model. In the quantum query model, the quantum circuit consists of two quantum registers, namely a d-dimensional query register and an ancilla register of arbitrary size. We further assume that an algorithm has access to a d-dimensional unitary U can be performed in one computational step, known as the query unitary. We may also allow controlled access to the unitary U depending on the state of the algorithm's ancilla qubits. For example, if there is only one control qubit in state $|b\rangle$, then the controlled unitary can be defined by

$$cU |\psi\rangle |b\rangle = \begin{cases} |\psi\rangle |0\rangle & b = 0 \\ U |\psi\rangle |1\rangle & b = 1 \end{cases}.$$

A quantum query algorithm starts off in some fixed initial state, say the all zeros state $|0^n\rangle$ in its registers. The query algorithm then proceeds by applying in succession unitary operators A_i followed by an applications of the query unitary U to the query register. Finally, the resulting quantum state is measured to produce some Boolean output. A generic query algorithm is illustrated in Figure 2.1. The query complexity of the algorithm is the number of times the query unitary U is applied.

The query setting is a natural setting for studying quantum algorithms. Indeed, some of the most commonly applied quantum algorithms are examples of quantum query algorithms. These examples include:

- **Quantum Phase Estimation** [NC10, Chapter 5.2]: In phase estimation, one is given query access to a unitary U and the input is a quantum state $|\psi\rangle$ promised to be an eigenvector of U. The goal of the quantum phase estimation algorithm is to output an estimate $\tilde{\theta}$ for the eigenvalue θ corresponding to $|\psi\rangle$ (i.e. $U |\psi\rangle = e^{2\pi i\theta} |\psi\rangle$), with error $|\tilde{\theta} - \theta| \leq \epsilon$.

  The quantum phase estimation algorithm accomplishes this with $O(\frac{1}{\epsilon})$ queries to U, with constant success probability. It is one of the key routines used in Shor's algorithm for factoring integers [NC10, Chapter 5.4], and estimation of the ground state energy for chemical systems [BBMC20].

- **Grover's Search** [NC10, Chapter 6]: Grover's search algorithm accomplishes the task of black-box search, which is to find an index i of a Boolean string x with $x_i = 1$ if one exists. Query access to the string is given by the unitary U_x, defined by setting $U_x |i\rangle = (-1)^{x_i} |i\rangle$, where $|i\rangle$ is the i^{th} computational basis state. Grover's algorithm uses $O(\sqrt{n})$ queries to U_x to accomplish the search

task, compared to the $\Omega(n)$ queries to the hidden string needed for a deterministic or randomized algorithm.

As the example of Grover's search illustrates, the quantum query model enables comparison between classical and quantum algorithms. This enables a rigorous study of which problems are mostly likely to benefit from quantum speedups. In particular, a classical query problem can be thought of as computing a Boolean function $f(x_1, \ldots, x_n)$ given only query access to the underlying bits $x_1, \ldots x_n$. In the quantum query model, one enables query access to the bits in superposition through applying the unitary U_x. For classical query tasks, the quantum query model can simulate the deterministic and randomized decision tree models. For a proof sketch, consult [BDW02, Section 3.3].

In the case of total functions, where f is defined for all strings in $\{0,1\}^n$, we now have precise understanding of when quantum speed-ups are possible. In particular, there is at most polynomial speedup between the deterministic and quantum model in this case. The precise exponent was characterized in [ABDK+21].

Theorem 2.1.1 ([ABDK+21])**.** *Let f be a total Boolean function. Suppose $D(f)$ is the number of queries needed for a deterministic decision tree to compute f and $Q(f)$ is the quantum query complexity of f. Then*

$$D(f) \leq O(Q(f)^4).$$

Furthermore, a quartic separation is the best possible due to a construction of [ABB+17].

However, for partial functions, there are now several examples of problems where there is an exponential separation between randomized decision trees and the quantum query model. The first exponential separation between randomized and quantum algorithms in the query model was due to Simon [Sim97]. Simon's problem asks: given query access to a function $f : \{0,1\}^n \to \{0,1\}^n$ with the property that there exists a string s such that $f(x) = f(y)$ iff $x = y$ or $x = y \oplus s$, compute the string s. This is possible with $O(n)$ queries to f in the quantum model, but $\Omega(\sqrt{2^n})$ queries are required with a deterministic or randomized algorithm. Similarly, an exponential lower bound for the black-box order-finding problem in the randomized setting was proven by Cleve [Cle04], although this problem can be efficiently solved in the quantum query model.

2.2 The Polynomial Method

We now turn to lower bound techniques for quantum algorithms using the polynomial method, which is one of the main paradigms for proving lower bounds in quantum query complexity.

The polynomial method is based on the fact that a quantum algorithm making T queries to a boolean input $X = (x_1, \ldots, x_n)$ yields a real polynomial $p : \mathbb{R}^n \to \mathbb{R}$ of degree at most $2T$ such that $p(x_1, \ldots, x_n)$ is equal to the acceptance probability of the algorithm on input X. If the algorithm distinguishes between *yes* and *no* instances with some bias, so does the polynomial p. Thus, lower bounds on the degree of any such distinguishing polynomial directly translates into a lower bound on the bounded-error quantum query complexity for the same task. We now state and prove these observations formally. Recall from the previous section that the unitary defined by $U_x |i\rangle = (-1)^{x_i} |i\rangle$ "hides" the string X.

Theorem 2.2.1. *Suppose $p(x_1, \ldots, x_n)$ is the acceptance probability of a T-query quantum algorithm on input $(x_1, \ldots x_n)$. Then p is a polynomial of degree at most $2T$.*

Furthermore, if the algorithm correctly computes a function f, then

$$|p(x_1, \ldots, x_n) - f(x_1, \ldots, x_n)| \leq \frac{1}{3}$$

on all inputs $X = (x_1, \ldots, x_n)$ in the domain of f.

Proof. We show that if $|\psi_t\rangle$ is the state of the circuit before the t^{th} query, then we can claim that we can decompose $|\psi_t\rangle$ as

$$|\psi_t\rangle = \sum_{b=0}^{1} \sum_{i=1}^{d} \sum_{k=1}^{r} p^t_{b,i,k}(x_1, \ldots, x_n) \, |b, i, k\rangle \, ,$$

where b denotes the control qubit, i denotes the query register, k denotes the ancilla register, and $p^t_{b,i,k}(x_1, \ldots, x_n)$ is a polynomial in the bits $x_1, \ldots, x_n$ of degree at most $t - 1$

We show this by induction on the number of queries t. This is true for $t = 1$ since no queries have been made, so $|\psi_1\rangle$ is a fixed state independent of X. This implies that each $p^t_{b,i,k}(x_1, \ldots x_n)$ is a constant as claimed.

Now, assuming the claim is true for $|\psi_t\rangle$, then applying the controlled oracle cU means that

$$\begin{aligned} cU \, |\psi_t\rangle &= \sum_{i=1}^{d} \sum_{k=1}^{r} p^t_{b,i,k}(x_1, \ldots, x_n) \, |0, i, k\rangle + \sum_{i=1}^{d} \sum_{k=1}^{r} p^t_{b,i,k}(x_1, \ldots, x_n)(-1)^{x_i} \, |1, i, k\rangle \\ &= \sum_{i=1}^{d} \sum_{k=1}^{r} p^t_{b,i,k}(x_1, \ldots, x_n) \, |0, i, k\rangle + \sum_{i=1}^{d} \sum_{k=1}^{r} p^t_{b,i,k}(x_1, \ldots, x_n)(1 - 2x_i) \, |1, i, k\rangle \end{aligned} \tag{2.1}$$

Hence, since $|\psi_{t+1}\rangle = A_{t+1} cU \, |\psi_t\rangle$ for some linear map A_{t+1}, then the amplitudes $p^{t+1}_{b,i,k}(x_1, \ldots, x_n)$ are linear combinations of the polynomials $p^t_{b,i,k}(x_1, \ldots, x_n)$ and $p^t_{b,i,k}(x_1, \ldots, x_n)(1 - 2x_i)$. This implies that since each $p^t_{b,i,k}(X)$ is a polynomial of degree at most t, then each $p^{t+1}_{b,i,k}(X)$ is a polynomial of degree at most $t + 1$.

Thus, the proposition follows, since the acceptance probability of the circuit is given by a sum $\sum_{(b,i,k) \in S} |p^t_{b,i,k}(X)|^2$ over some subset of measurement outcomes at the end of the circuit. This is a real-valued polynomial of degree at most $2t$, and must approximate f if the algorithm computes f correctly on its domain. □

Hence the polynomial method reduces a problem about quantum query complexity, to one about polynomial approximation. This observation has been useful in numerous settings in the quantum query complexity literature. We give a brief example to illustrate the main ideas behind applying the polynomial method.

Example: Black-Box Search To give a simple illustration of the polynomial method, we consider the lower bound for black-box search and show the optimality of Grover's algorithm in the query model. Recall the task of black-box search is to determine whether or not a string $x_1 \ldots x_n$ contains an index i with $x_i = 1$. Hence the function being computed by a black-box search algorithm is an OR of n bits, defined by

$$OR(x_1, \ldots, x_n) = \begin{cases} 0 & x = 0^n \\ 1 & x \neq 0^n \end{cases}.$$

We claim the following polynomial approximation bound for the OR function.

Theorem 2.2.2. *If $p(x_1, \ldots, x_n)$ is a polynomial that approximates the OR_n function pointwise, then the degree of p is at least $\Omega(\sqrt{n})$.*

Hence, combining this bound with the main observation from Theorem 2.2.1, shows that any quantum query algorithm for black-box search must make $\Omega(\sqrt{n})$ queries.

Before proceeding to the proof of theorem, we must discuss a few preliminaries from approximation theory.

In general, it is difficult to study the properties of multivariable polynomials. However, we use the observation that the OR function is a symmetric function, which means that its value depends only on the Hamming weight $|x|$ of the input. Therefore, it would be natural to expect that any approximation to the OR function can be made symmetric in the variables as well, without loss of generality. This symmetry makes the polynomial approximation much easier to analyze, as it reduces a problem about multivariable polynomials to one about univariate polynomials. This is known in the literature as *Minsky-Papert symmeterization* [MP17].

Lemma 2.2.3. *Let $p(x_1, \ldots, x_n)$ be a symmetric polynomial. Then there exists a univariate polynomial q with the property that $q(|x|) = p(x_1, \ldots, x_n)$ for all Boolean inputs $x_i \in \{0, 1\}^n$. and $\deg q \leq \deg p_{sym}$.*

Another ingredient that we need is the *Markov brother's inequality* that relates the degree of a polynomial to the maximum possible magnitude of its derivative.

Lemma 2.2.4 (Markov brother's inequality [Mar89])**.** *Let $p(x)$ be a degree-n real-valued polynomial. If $|p(x)| \leq H$ on the interval $[a, b]$, then for all $x \in [a, b]$,*

$$|p'(x)| \leq \frac{2Hn^2}{b - a}.$$

Combining these two lemmas together, we can obtain the proof of Theorem 2.2.2.

Proof of Theorem 2.2.2. Suppose $p(x_1, \ldots, x_n)$ is a polynomial that approximates the OR function. Then so does the symmetric polynomial

$$p_{sym}(x_1, \ldots, x_n) = \frac{1}{n!} \sum_{\sigma \in S_n} p(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$$

where the sum of the right hand side is over all permutations of n variables. Clearly p_{sym} is a symmetric polynomial, and hence by Lemma 2.2.3 there exists a univariate polynomial q such that $q(|x|) = p_{sym}(x_1, \ldots, x_n)$ on Boolean inputs.

Since p approximates the OR function, so does p_{sym}. Therefore, we deduce that $q(0) \leq \frac{1}{3}$ and $q(i) \geq \frac{2}{3}$ for all integers $i = 1, \ldots, n$. Furthermore, $\deg q \leq \deg p$.

We now proceed in two cases:

Case 1: Suppose $|q(x)| \leq 2$ for all $x \in [0, n]$. Observe that since $q(0) \leq \frac{1}{3}$ and $q(1) \geq \frac{2}{3}$, the mean value theorem guarantees that there is a point $x_0 \in [0, 1]$ with $q'(x_0) \geq \frac{1}{3}$. Therefore, at the point x_0, Markov's inequality and the assumption in this case implies that

$$\frac{1}{3} \leq \frac{4(\deg q)^2}{n}$$

which means $\deg q \geq \Omega(\sqrt{n})$,

Case 2: Otherwise, suppose the maximum of q over $[0, n]$ is $h > 2$. Since $|q(i)| \leq 1$ at all integer points, then the mean value theorem guarantees that there is a point $x_0 \in [0, n]$ with $q'(x_0) \geq \frac{h-1}{1} \geq \frac{h}{2}$ since $h > 2$. Therefore, Markov's inequality and the assumption in this case imply

$$\frac{h}{2} \leq \frac{4h(\deg q)^2}{n}$$

which means that $\deg q \geq \Omega(\sqrt{n})$.

In either case we have $\deg q \geq \Omega(\sqrt{n})$, so $\deg p \geq \deg q \geq \Omega(\sqrt{n})$. Therefore, the claim follows. □

Hence, as Grover's algorithm matches this $O(\sqrt{n})$ query bound for computing the OR function, we conclude that it is in-fact *optimal*, up to constant factors, in the black-box setting. Observe the importance of the symmetrization step in reducing the query lower bound from a multivariable approximation problem to a univariate problem, which will also be used later in the more general unitary property testing setting.

Other Methods for Proving Query Lower Bounds We briefly overview some other methods for proving lower bounds on query quantum complexity. In quantum query complexity, there have been two main paradigms for proving lower bounds for query complexity, namely the polynomial method [BBC+01] and the adversary method [ŠS05]. The methods are generally incomparable, as there are problems where one method is able to prove a tight lower bound but the other cannot. For instance, the collison problem [Kut03] is a case where the polynomial method proves a tight lower bound but the adversary method provably fails to do so. On the other hand, Ambanis [Amb06] constructed an example where the adversary method is provably better than the polynomial method. Furthermore, for evaluation of Boolean functions, the general adversary method characterizes the quantum query complexity up to constant factors [Rei11].

2.3 Quantum Complexity Classes

Another motivation to consider query complexity is that it is a setting where separations between complexity classes can be proven. It is difficult to unconditionally separate complexity classes. Indeed, for the BPP versus BQP problem, showing that $\mathsf{BPP} \neq \mathsf{BQP}$ unconditionally would imply that $\mathsf{P} \neq \mathsf{PSPACE}$, which is beyond what the known techniques of complexity theory can establish. However, in the query setting, the classical query lower bound for Simon's problem in [Sim97] implies that there is an oracle O relative to which $\mathsf{BPP}^O \neq \mathsf{BQP}^O$. This result gives formal evidence of a setting where quantum computation is more powerful than classical computation.

Similarly, the Grover's search lower bound presented in Section 2.2 shows that there is an oracle O relative where $\mathsf{NP}^O \not\subseteq \mathsf{BQP}^O$. This separation was originally presented in [BBBV97]. This gives

evidence that a quantum device cannot give exponential speedups for solving NP-complete problems in the black-box setting. However, polynomial speedups may still be possible.

The quantum complexity classes whose relationship we are most interested in this thesis are complexity classes related to quantum proofs. The power of proofs has been an important question of study in complexity theory. In the classical world, this leads to the definition of the complexity class NP, which has been a central object of study throughout complexity theory. However, the question of defining a "quantum" version of NP is more subtle, and there are several variations. For a full survey of seven possible definitions of "quantum NP", consult the recent survey [Gha23]. However, we describe three variations of complexity classes related to quantum proofs, namely, QMA, QCMA and QMA(2).

QMA The class QMA (Quantum Merlin-Arthur) is the class of problems verifiable on a polynomially-sized quantum circuit given access to a proof state $|\psi\rangle$ with polynomially many qubits. We take the formal definition of QMA from [Wat08, Section V].

Definition 2.3.1 (QMA)**.** A promise problem (A_{yes}, A_{no}) is in QMA if there exists a polynomial-time generated family of quantum circuits $\{C_n\}$ where C_n has $n+p(n)$ input qubits for some polynomial $p(n)$ and one output qubit, such that:

- For all yes instances $x \in A_{yes}$ of length n, there exists a quantum state $|\psi\rangle$ on at most $p(n)$ qubits such that the circuit C_n accepts the pair $(x, |\psi\rangle)$ with probability at least $\frac{2}{3}$.
- For all no instances $x \in A_{no}$ of length n and all quantum states $|\psi\rangle$ of at most $p(n)$ qubits, the circuit C_n accepts the pair $(x, |\psi\rangle)$ with probability at most $\frac{1}{3}$.

As usual in complexity theory, the constants $\frac{2}{3}$ and $\frac{1}{3}$ can be replaced by arbitrary functions $a(n), b(n)$ with $a(n) - b(n) \geq \frac{1}{q(n)}$ for some polynomial q, without changing the definition of the complexity class. See [Wat08, Section V.5] for a proof.

QMA-Complete Problems Like the class NP, the class QMA captures a number of complete problems that are relevant in applications. The canonical complete problem for QMA is the *local Hamiltonian* problem [KKR06]. Let M be a Hermitian $2^n \times 2^n$ matrix. We think of M as an observable acting on a system of n qubits. We say that M is k-local if there exists a subset $S \subseteq [n]$ of k qubits where M can be decomposed as $M = A_S \otimes I$, for some $2^k \times 2^k$ matrix A_S acting on the qubits labelled in S and I is the identity acting on the rest of the qubits.

We can now formally define the k-local Hamiltonian problem. The input to the problem is a set of k-local matrices $H_1, \ldots, H_m$, each with operator norm $||H_i|| \leq 1$ and parameters a, b satisfying $b - a \geq \frac{1}{p(n)}$ for some polynomial $p(n)$. Let $H = \sum_{i=1}^{m} H_i$. Let $\lambda(H)$ denote the smallest eigenvalue of H. The goal is to distinguish between the case where $\lambda(H) \leq a$ or $\lambda(H) \geq b$.

The local Hamiltonian problem is significant for several reasons. Firstly, it was the first problem that was proven to be QMA-complete. Kitaev's circuit-to-Hamiltonian construction used to prove QMA-completeness [KSV02] can be thought of as a quantum analogue of the Cook-Levin construction for showing that Boolean satisfiability is NP-complete. Secondly, the local Hamiltonian problem is a generalization of classical constraint satisfaction problems (CSPs) such as Boolean satisfiability. This motivates trying to generalize existing tools for studying CSPs such as PCP theorems to the more general quantum setting [GHL+15, Section 4.1]. Finally, the local Hamiltonian problem captures the

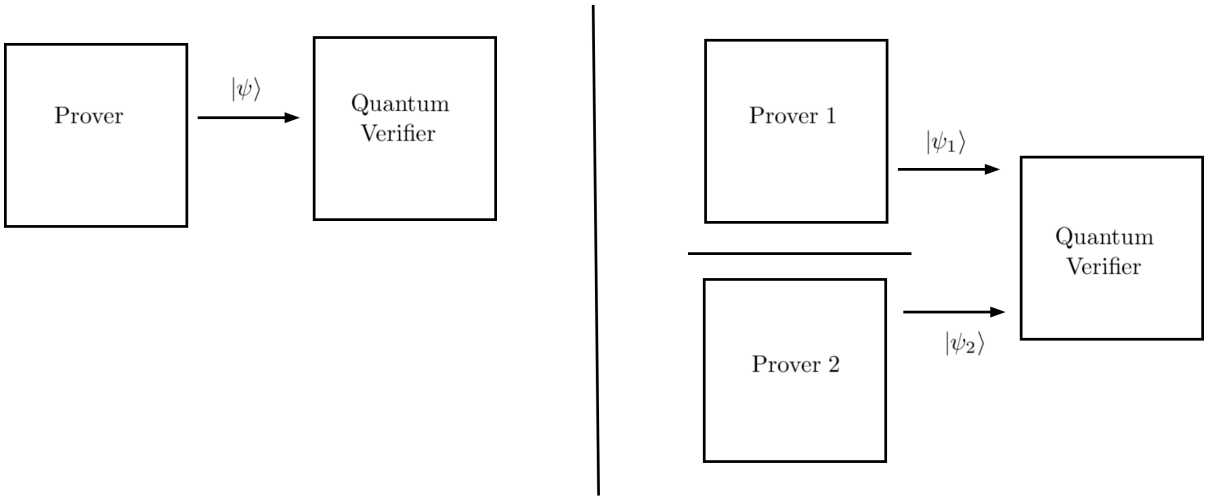

Figure 2.2: QMA versus QMA(2)

complexity of many problems studied in physics. For instance, finding the ground state energy of chemical systems [WLAG13], the Bose-Hubbard model [GHL+15, Section 5.4], and the quantum Heisenberg model [GHL+15, Section 6.1] are all QMA-complete. This suggests that QMA-completeness is useful at classifying the complexity of computational tasks throughout chemistry and physics, and also motivates the study of approximation algorithms for solving these problems.

QCMA By varying the type of proof allowed in quantum algorithm, variants of QMA can be considered.

One such variant is the class QCMA (Quantum-Classical Merlin-Arthur) is the restriction of QMA where the verifier is still a quantum circuit, but the proof state provided is a classical bit string. Aharanov and Naveh [AN02] proposed the QMA versus QCMA problem and conjectured that QCMA = QMA. For example, there are numerous examples of classical descriptions for quantum states, such as tensor networks [GHL+15, Section 4.2], and they seem to be good heuristics for describing the ground states of some local Hamiltonians. Therefore, solving the QMA versus QCMA problem would give insight on whether or not we expect quantum states occurring in nature (i.e. ground states of local Hamiltonian) admit an efficient classical description (eg. if they can be generated by a polynomial-sized quantum circuit).

While the QMA versus QCMA question remains unresolved in generality, we will see later in this section that there are some results separating the two classes in the black-box setting.

QMA(2) Finally, we discuss QMA(2), introduced in [KMY03]. The complexity class QMA(k) is defined as the class of problems verifiable by a polynomial time quantum circuit with access to $k \geq 2$ *un*entangled proofs. That is, for *yes* instances x, there exists a quantum state $|\Psi\rangle$ of the form $|\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle$ where each $|\psi_i\rangle$ is on $poly(n)$ qubits such that the verifier accepts the pair $(x, |\Psi\rangle)$. Otherwise, for *no* instances, the verifier must reject the input x under the promise that the witness is of tensor product form $|\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle$. This models the situation where a verifier is allowed to interact with two unentangled and separated provers. See Figure 2.2 for an illustration of the difference between QMA and QMA(2)

It was shown in [HM13] that for any constant $k > 2$, we have QMA(k) = QMA(2), as any QMA(k) verifier can be simulated by a QMA(2) verifier. Furthermore, they showed that error-reduction is possible for QMA(2).

We note that QMA $\subseteq$ QMA(2) since a QMA(2) verifier could simulate a QMA verifier by using only one of the proofs provided. However, there is evidence that QMA(2) could be more powerful than QMA, for several reasons:

- There are QMA(2) protocols to verify NP-complete problems (eg. graph 3-colouring) using a logarithmic number of qubits, as outlined in [BT09]. If there were a QMA protocol for these problems using a logarithmic number of qubits, then NP $\subseteq$ QMA$_{log}$ = BQP, which is considered unlikely [MW05].

  Otherwise, if there exists a QMA protocol with a sublinear number of qubits for 3-SAT or 3-colouring, then the (classical) Exponential Time Hypothesis is false [ABD+08, CD10].

- There are certain problems in quantum chemistry, specifically the pure state N-representability problem, known to have QMA(2) protocols but not QMA protocols [LCV07].

- There is no test using only local quantum operations and classical communication (LOCC) that can distinguish between product state inputs and entangled inputs, as proven in [HM13]. If such a test existed, then QMA(2) = QMA.

On the other hand, despite many years of study, the only complexity inclusions about QMA(2) known are QMA $\subseteq$ QMA(2) $\subseteq$ NEXP, a vast gap in the complexity-theoretic landscape. A first step towards showing that QMA(2) is indeed more powerful than QMA would be to identify an oracle relative to which QMA(2) is different than QMA. As discussed in Aaronson's survey paper on quantum query complexity in [Aar21], an oracle separation between QMA and QMA(2) has been a notorious open problem in quantum complexity theory. This would already have very interesting consequences in quantum information theory, such as ruling out the existence of disentanglers [ABD+08]. This motivates studying quantum proofs in the black-box, or query model.

Quantum Proofs in the Query Model Quantum proofs can be extended to the query model in a natural way. In addition to a query register and ancilla register, now the quantum query algorithm has access to an additional proof register containing a quantum proof state $|\psi\rangle$. For all oracles U that are "yes" instances, there should be a proof state $|\psi\rangle$ that the algorithm accepts when provided as input. Otherwise, for all "no" instances, the algorithm should reject regardless of what proof state was provided. This provides a query analogue for the complexity class QMA, and the query analogues of QCMA and QMA(2) can be defined similarily by changing the allowed set of proof states.

The polynomial method introduced in Section 2.2 can be generalized to prove lower bounds in the QMA setting. This was applied by Raz to study the QMA query complexity of total Boolean functions [RS04], and Aaronson et al. to study the QMA query complexity of the permutation testing [Aar11] and approximate counting [AKKT20] problems. Aaronson et al.'s lower bounds use the "Guessing Lemma", which is an application of Marriott-Watrous amplification for QMA [MW05]. This lemma will be introduced in Section 3.1.

Finally, we comment on separating QMA and QCMA in the black-box setting, which requires different techniques. In this setting, Aaronson and Kuperberg [AK07] studied a search problem for quantum

states, that is distinguishing between the case where the oracle is the identity matrix $U = I$ versus $U = I - 2|\psi\rangle\langle\psi|$ for some quantum state $|\psi\rangle \in \mathbb{C}^d$. This problem has a one-query algorithm in the QMA setting. However, the main result of [AK07] was to show that $\Omega(\sqrt{\frac{d}{m}})$ queries are necessary when the query algorithm is only provided an m-bit *classical* witness. This bound is also tight, up to constant factors. The techniques are based on results in random matrix techniques, particularly properties of Haar-random quantum states, and a hybrid argument. Furthermore, this query result implies that there exists a quantum oracle relative to which QMA and QCMA are different. We also use Aaronson-Kuperberg's techniques in Section 3.3.

However, Aaronson-Kuperberg's techniques are inherently quantum since they use properties of Haar-random quantum states. It remains open to see if there is a classical oracle separation between QCMA and QMA (i.e. a separation in the setting where the oracle encodes a Boolean string). We note that the work done in [FK15] and [NN22] has made progress on whether or not there exists a classical oracle separation between QCMA and QMA.

Chapter 3

Unitary Property Testing

3.1 Preliminaries

3.1.1 Testers for Unitary Properties

We recall the formal definition for a unitary property tester.

Let $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$ be a d-dimensional unitary property. A quantum algorithm is a *tester* for $\mathcal{P}$ if the following holds:

1. If $U \in \mathcal{P}_{yes}$, then the algorithm makes queries to U and accepts with probability $2/3$.
2. If $U \in \mathcal{P}_{no}$, the algorithm makes queries to U and accepts with probability at most $1/3$.

A quantum algorithm is a QMA *tester* for $\mathcal{P}$ if the following holds:

1. If $U \in \mathcal{P}_{yes}$, then there exists a *quantum proof state* $|\psi\rangle$ such that the algorithm on input $|\psi\rangle$, makes queries to U, and accepts with probability at least $2/3$.
2. If $U \in \mathcal{P}_{no}$, then the algorithm given any proof state as input, making queries to U, accepts with probability at most $1/3$.

As usual in complexity theory, the probabilities $2/3$ and $1/3$ can be set to any constants a and b as long as $a > b$ without changing any of the arguments that follow.

3.1.2 The Guessing Lemma

A simple but useful way to prove query lower bounds on QMA testers is to remove the proof state via the "Guessing Lemma", used in [Aar11, Lemma 5] and [AKKT20, Lemma 17]:

Lemma 3.1.1. *Suppose there is a* QMA *tester for a property* $\mathcal{P}$ *that makes* T *queries and receives an* m*-qubit proof. Then there is a (standard) tester for the* $\mathcal{P}$ *that makes* $O(mT)$ *queries, receives no proof state, and satisfies*

- *For all* $U \in \mathcal{P}_{yes}$ *the tester accepts with probability at least* 2^{-m}.
- *For all* $U \in \mathcal{P}_{no}$*, the tester accepts with probability at most* 2^{-10m}.

3.1.3 Approximation Theory and Laurent Polynomials

When we say that a polynomial is degree-T, we mean that it has degree *at most* T. We say that a polynomial $p(z_1, \ldots, z_d, z_1^*, \ldots, z_d^*)$ with complex coefficients is *self-adjoint* if $p(z_1, \ldots, z_d, z_1^*, \ldots, z_d^*) = p(z_1, \ldots, z_d, z_1^*, \ldots, z_d^*)^*$. We say that p is *symmetric* if applying a permutation $\pi : [d] \to [d]$ to the variables z_i and z_i^* leaves p unchanged.

We record here some useful facts about polynomials, univariate polynomial approximation, and Laurent polynomials.

Lemma 3.1.2 (Fundamental Theorem of Algebra)**.** *A real-valued degree-T univariate polynomial has at most T zeros.*

Lemma 3.1.3 (Markov brother's inequality [Mar89])**.** *Let $p(x)$ be a degree-n real-valued polynomial. If $|p(x)| \leq H$ on the interval $[a, b]$, then for all $x \in [a, b]$,*

$$|p'(x)| \leq \frac{2Hn^2}{b-a}.$$

Lemma 3.1.4 (Paturi's bound [Pat92])**.** *If p is a degree-d real polynomial satisfying $|p(x)| \leq 1$ for all $|x| \leq 1$, then for all $|x| \leq 1 + \mu$ we have*

$$|p(x)| \leq \exp(2d\sqrt{2\mu + \mu^2}).$$

A Laurent polynomial p in variables $x_1, \ldots, x_k$ is a polynomial in the variables $x_1, \ldots, x_k$ and $x_1^{-1}, \ldots, x_k^{-1}$. We say that a univariate Laurent polynomial $p(z)$ is *symmetric* if $p(z) = p(z^{-1})$. The following fact about Laurent polynomials was stated as [AKKT20, Lemma 14].

Lemma 3.1.5. *Suppose $p(z)$ is a symmetric Laurent polynomial. Then there exists a univariate polynomial q such that $p(z) = q(z + \frac{1}{z})$.*

We require a slight modification of Lemma 3.1.5.

Lemma 3.1.6. *If $p(x, x^*)$ is a self-adjoint degree-d polynomial satisfying $p(x, x^*) = p(x^*, x)$, there exists a real-valued univariate polynomial q of degree d with the property that $q(x + x^*) = p(x, x^*)$ when x is restricted to the unit circle.*

Proof. If x is restricted to the unit circle, then $xx^* = 1$. Let $q(x) = p(x, \frac{1}{x})$. We claim that q is a symmetric Laurent polynomial. Since p is self-adjoint and $p(x, x^*) = p(x^*, x)$,

$$q(x) = p\left(x, \frac{1}{x}\right) = p(x, x^*) = p(x^*, x) = p\left(\frac{1}{x}, x\right) = q\left(\frac{1}{x}\right),$$

and q has real coefficients. Hence, by Lemma 3.1.5, there exists a polynomial r such that $p(x, x^*) = q(x) = r(x + \frac{1}{x}) = r(x + x^*)$. □

We will also use the following bounds on the cosine function, which follow from elementary calculus.

Lemma 3.1.7. *For all $|x| \leq 2$, $\frac{x^2}{3} \leq 1 - \cos x \leq \frac{x^2}{2}$.*

Proof. By the Taylor series expansion, $1 - \frac{x^2}{2} \leq \cos x \leq 1 - \frac{x^2}{2} + \frac{x^4}{24}$. Therefore, $1 - \cos x \leq \frac{x^2}{2}$.

For the lower bound, observe that if $|x| \leq 2$, then $x^4 \leq 4x^2$. Therefore, $\cos x \leq 1 - \frac{x^2}{2} + \frac{x^2}{6} = 1 - \frac{x^2}{3}$. Hence, $\frac{x^2}{3} \leq 1 - \cos x$ in this range. □

3.1.4 Invariant Theory

We will use some notions of invariant theory in our work.Invariant theory studies the action of a group G on a polynomial ring $\mathbb{C}[x_1, \ldots, x_n]$. We denote the action of $g \in G$ on $f \in \mathbb{C}[x_1, \ldots, x_n]$ by $g \cdot f$. The ring of invariant polynomials $\mathbb{C}[x_1, \ldots, x_n]^G$ is then the subring of $\mathbb{C}[x_1, \ldots, x_n]$ consisting of polynomials satisfying $g \cdot f = f$ for all $g \in G$, that is $f \in \mathbb{C}[x_1, \ldots, x_n]^G$ is left unchanged by the action of g for all group elements.

There are many natural questions about the invariant ring $\mathbb{C}[x_1, \ldots, x_n]^G$ one can ask, such as construction of a generating set for the invariant ring. For example, one classical example is the action of a permutation $\sigma \in S_n$ acting on a polynomial $p(x_1, \ldots, x_n)$ by permuting the variables by $\sigma \cdot p = p(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$. The invariant ring is known as the ring of symmetric polynomials, for which there are many well-known generating sets. One example of a generating set is the power sum symmetric polynomials given by $p_i = \sum_{j=1}^{n} x_j^i$, which generate the symmetric polynomial ring as an algebra. In other words, for any symmetric polynomial f, there exists a polynomial g for which $f = g(p_1, \ldots, p_n)$. There are similar characterizations of the invariant ring for numerous other group actions.

The group action G we will consider in this work is defined as follows.

Definition 3.1.1 (Invariant rings)**.** Let $G \subseteq U(d)$ be a subgroup of the $d \times d$ unitary group.

Let $\mathbb{C}[X, Y]_d$ be the ring of complex polynomials in matrix variables $X = (x_{i,j})_{1 \leq i,j \leq d}$ and $Y = (y_{i,j})_{1 \leq i,j \leq d}$. Observe that there is an action of G on any $f(X, Y) \in \mathbb{C}[X, Y]$ by simultaneous conjugation $g \cdot f(X, Y) = f(gXg^{-1}, gYg^{-1})$.

The *invariant ring* $\mathbb{C}[X, Y]_d^G$ is the subring of polynomials in $\mathbb{C}[X, Y]_d$ satisfying $g \cdot f = f$ for all $g \in G$.

The general theory of invariant theory guarantees the existence and finiteness of a generating set for the invariant ring $\mathbb{C}[X, Y]^G$ for all compact groups, which includes all finite groups, the unitary group, and the local unitary group.

Definition 3.1.2 (Local Unitary Group)**.** Let $d_1, d_2 \geq 2$. The local unitary group $\mathrm{LU}(d_1, d_2)$ is the subgroup $\mathrm{U}(d_1) \times \mathrm{U}(d_2)$ of $\mathrm{U}(d_1 d_2)$ consisting of all unitaries of the form $g \otimes h$ where $g \in \mathrm{U}(d_1), h \in \mathrm{U}(d_2)$.

3.1.5 Distance between Quantum States

Let ρ and σ be $d \times d$ density matrices.

Definition 3.1.3. The trace distance between ρ and σ is defined as $T(\sigma, \rho) = \frac{1}{2}||\rho - \sigma||_1$. The trace norm for a $d \times d$ Hermitian matrix M is defined as $||M||_1 = \sum_{i=1}^{d} |\lambda_i|$ where λ_i are the eigenvalues of M.

For pure states, which is when $\rho = |\psi_1\rangle \langle\psi_1|$ and $\sigma = |\psi_2\rangle \langle\psi_2|$ are rank one matrices, the trace distance and fidelity satisfy a well-known relation.

Lemma 3.1.8. *Given two pure states $\rho = |\psi_1\rangle \langle\psi_1|$ and $\sigma = |\psi_2\rangle \langle\psi_2|$, the trace distance satisfies $T(\rho, \sigma) = \sqrt{1 - |\langle\psi_1|\psi_2\rangle|^2}$. The quantity $|\langle\psi_1|\psi_2\rangle|^2$ is also known as the fidelity between the states $|\psi_1\rangle$ and $|\psi_2\rangle$.*

3.1.6 Entropy of Quantum States

In Section 3.2.3 we discuss the entanglement entropy problem. We will use the Rényi 2-entropy as our measure of entropy in this problem.

Definition 3.1.4 (Rényi 2-entropy)**.** Given a state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ with reduced density matrix on the first register ρ, the Rényi 2-entropy of $|\psi\rangle$ is defined as $H_2(|\psi\rangle) = -\log \mathrm{Tr}(\rho^2)$.

We note that since $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ is a pure state, it does not matter whether or not the reduced density matrix is taken with respect to the first or second register, since both matrices will have the same set of eigenvalues.

3.2 The Generalized Polynomial Method and Applications

The following Proposition, restated from the introduction, is the foundation for our generalized polynomial method:

Proposition 1.2.1 (Generalized polynomial method)**.** *The acceptance probability of a quantum algorithm making T queries to a $d \times d$ unitary U and its inverse U^* can be computed by a degree at most $2T$ self-adjoint polynomial $p : \mathbb{C}^{2(d\times d)} \to \mathbb{C}$ evaluated at the matrix entries of U and U^*. Thus, degree lower bounds on such polynomials yields a query lower bound on the algorithm.*

Proof. A tester A that queries an oracle U can be written as a product of fixed unitary maps $A_0, A_1, \ldots, A_T$ that don't depend on the oracle U, interleaved with controlled applications of U and its inverse U^* (denoted by cU and cU^* respectively). Let $|\psi_t\rangle$ denote the state of the circuit before the t^{th} query, so that either $|\psi_{t+1}\rangle = A_{t+1}\mathrm{c}U\,|\psi_t\rangle$ or $|\psi_{t+1}\rangle = A_{t+1}\mathrm{c}U^*\,|\psi_t\rangle$. Write

$$|\psi_t\rangle = \sum_{b=0}^{1}\sum_{i=1}^{d}\sum_{k=1}^{r} p^t_{b,i,k}(U,U^*)\,|b,i,k\rangle$$

where b denotes the control qubit, i denotes the query register, k denotes the ancilla register, and $p^t_{b,i,k}(U,U^*)$ is some function of the matrix entries of U and U^*.

We claim that the amplitudes $p^t_{b,i,k}(U,U^*)$ are polynomials in the matrix entries of U and U^* of degree at most $t-1$. To show this, we proceed by induction on the number of queries t. This is clearly true for $t = 1$ since $|\psi_1\rangle$ is some fixed state independent of U, so $p^1_{b,i,k}(U)$ are constants for all b, i, k. Now assume the claim to be true for $|\psi_t\rangle$.

Suppose the t^{th} oracle call is to cU. Then

$$\begin{aligned}\mathrm{c}U\,|\psi_t\rangle &= \sum_{i=1}^{d}\sum_{k=1}^{r} p^t_{0,i,k}(U,U^*)\,|0\rangle\otimes|i\rangle\otimes|k\rangle + \sum_{i=1}^{d}\sum_{k=1}^{r} p^t_{1,i,k}(U,U^*)\,|1\rangle\otimes U\,|i\rangle\otimes|k\rangle \\ &= \sum_{j=1}^{d}\sum_{k=1}^{r} p^t_{0,j,k}(U,U^*)\,|0,j,k\rangle + \sum_{j=1}^{d}\sum_{k=1}^{r}\left(\sum_{i=1}^{d} U_{j,i}\; p^t_{1,i,k}(U,U^*)\right)|1,j,k\rangle\end{aligned}$$

Hence, since $|\psi_{t+1}\rangle = A_{t+1}\mathrm{c}U\,|\psi_t\rangle$ for some linear map A_{t+1}, then $p^{t+1}_{i,k}(U,U^*)$ are linear combinations of the polynomials $p^t_{0,i,k}(U,U^*)$ and $\sum_{i=1}^{d} U_{j,i}\; p^t_{1,i,k}(U,U^*)$. Hence if $p^t_{b,i,k}(U,U^*)$ have degree at most $t-1$, $p^{t+1}_{b,i,k}(X)$ have degree at most t. The same argument holds if the oracle call was to cU^*.

Thus the amplitudes of the final state of the algorithm can be expressed as polynomials of degree at most T. The proposition follows since the acceptance probability of the circuit is given by a measurement of the ancilla qubits and seeing if the string given lies in some set S, which is a sum $p(U,U^*) = \sum_{(b,i,k)\in S} |p^{T+1}_{b,i,k}(U,U^*)|^2$. This is a degree-$2T$ self-adjoint polynomial since each $p^{T+1}_{i,k}(U,U^*)$ has degree at most T. □

We also now show the following, restated from the introduction.

Proposition 1.2.2. *Let $\mathcal{P}$ be an property closed under inversion and suppose there is a T-query quantum algorithm for testing property $\mathcal{P}$. Let p be the polynomial from Proposition 1.2.1 that computes the acceptance probability of the algorithm. Then, we may assume that $p(U,U^*) = p(U^*,U)$.*

Proof. Let p be the polynomial from Proposition 1.2.1 for the property $\mathcal{P}$. Define

$$q(U,U^*) = \frac{p(U,U^*) + p(U^*,U)}{2}.$$

Clearly, $q(U,U^*) = q(U^*,U)$ and the degree of q is no more than the degree of p. Furthermore, since the property $\mathcal{P}$ is closed under inversion, $q(U,U^*) \geq a$ if $p(U,U^*) \geq a$ and $q(U,U^*) \leq b$ if $p(U,U^*) \leq b$. □

Then, when applying Proposition 1.2.1 to a T-query tester for a property $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$, we have that there exists a degree-$2T$ polynomial p such that if $U \in \mathcal{P}_{yes}$, then $p(U,U^*) \geq 2/3$, and if $U \in \mathcal{P}_{no}$, then $p(U,U^*) \leq 1/3$. Furthermore, by Proposition 1.2.2, we can further assume that $p(U,U^*) = p(U^*,U)$ for properties that are closed under inversion. However, proving degree lower bounds on p directly is difficult for general properties $\mathcal{P}$. As mentioned in the introduction, we focus on properties that obey certain symmetries in order to further simplify the polynomial p. For example, the acceptance probability p corresponding to symmetric classical properties of boolean strings can be averaged to a univariate polynomial q using Minsky-Papert symmetrization [BBC+01].

Recall the definition of G-invariant properties and the invariant ring as stated in Definition 1.2.1 and Definition 3.1.1 in the introduction. The following observation states that testers for G-invariant properties give rise to low-degree polynomials in the invariant ring $\mathbb{C}[X,X^*]$ that decide the property:

Proposition 1.2.3 (Symmeterization for G-invariant properties)**.** *Suppose $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$ is a G-invariant d-dimensional unitary property. If there is a T-query tester for $\mathcal{P}$ that accepts* yes *instances with probability at least a and* no *instances with probability at most b, then there exists a self-adjoint degree-$2T$ polynomial q in the invariant ring $\mathbb{C}[X,X^*]^G_d$ satisfying*

- *If $U \in \mathcal{P}_{yes}$, then $q(U,U^*) \geq a$.*
- *If $U \in \mathcal{P}_{no}$, then $q(U,U^*) \leq b$.*

Proof. Let $p(U,U^*)$ be the polynomial from Proposition 1.2.1 corresponding to the tester. Define the function

$$q(U,U^*) = \mathbb{E}_{g\sim\mu}\, p(gUg^{-1}, gU^*g^{-1}) \ .$$

It is clear that $q(U,U^*)$ is a self-adjoint polynomial with degree at most $2T$. Furthermore by construction it belongs to the invariant ring $\mathbb{C}[X,X^*]^G_d$ because for all $h \in G$,

$$q(hUh^{-1}, hU^*h^{-1}) = \mathbb{E}_{g\sim\mu}\, p(hgUg^{-1}h^{-1}, hgU^*g^{-1}h^{-1}) = \mathbb{E}_{g\sim\mu}\, p(gUg^{-1}, gU^*g^{-1}) = q(U,U^*)$$

where the second equality follows from the fact that the Haar measure μ is invariant under left multiplication.

Finally, the stated bounds on the values of $q(U, U^*)$ hold because for all $U \in \mathcal{P}_{yes}$, the unitary gUg^{-1} is also in $\mathcal{P}_{yes}$, and similarly for the *no* instances.

□

3.2.1 Unitarily Invariant Properties

In this paper we focus on two subgroups of the unitary group $\mathrm{U}(d)$. The first is the full unitary group itself. The invariant ring in this case has an extremely simple description. The following result is a special case of a more general theorem due to Procesi [Pro76], who computed the invariant rings of n-tuples of matrices under simultaneous conjugation by the classical groups. As we only need the case of 2 matrices (the unitary U and its adjoint U^*) in this work, we specialize the original result. Firstly, given a permutation $\sigma \in S_n$, define $\mathrm{Tr}_\sigma(A_1, \ldots, A_n) = \prod_{C \in C(\sigma)} \mathrm{Tr}(\prod_{j \in C} A_j)$ where $C(\sigma)$ is the set of disjoint cycles of σ.

Theorem 3.2.1 ([Pro76, Section 11], [KP96, Chapter 4])**.** *Let $\mathbb{C}[X, X^*]_d^G$ be the invariant ring corresponding to the group $G = \mathrm{U}(d)$. Then all homogenous degree-r polynomials $f \in \mathbb{C}[X, X^*]_d^G$ can be written as a linear combination of invariants of the form $\mathrm{Tr}_\sigma(A_1, \ldots, A_r)$, where each $A_i = X$ or X^* and σ is a permutation in S_r. All invariants of degree $\leq r$ are linear combinations of homogenous invariants of degree $\leq r$.*

We therefore get the following result from combining Proposition 1.2.3 and Theorem 3.2.1 for testing a unitarily invariant property.

Theorem 3.2.2 (Symmetrization for unitarily invariant properties)**.** *Let $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$ denote a d-dimensional unitarily invariant property. Suppose there is a T-query quantum algorithm that accepts* yes *instances with probability at least a and* no *instances with probability at most b. Then there exists a degree at most $2T$ symmetric*[1] *self-adjoint polynomial $q(z_1, \ldots, z_d, z_1^*, \ldots, z_d^*)$ satisfying*

- *If $U \in \mathcal{P}_{yes}$ then $q(z_1, \ldots, z_d, z_1^*, \ldots, z_d^*) \geq a$*
- *If $U \in \mathcal{P}_{no}$ then $q(z_1, \ldots, z_d, z_1^*, \ldots, z_d^*) \leq b$*

where $(z_1, \ldots, z_d)$ and $(z_1^*, \ldots, z_d^*)$ are the eigenvalues of U and their complex conjugates, respectively.

Proof. By Proposition 1.2.3, there exists a polynomial p of degree at most $\leq 2T$ in the invariant ring $\mathbb{C}[U, U^*]^G$ with the property that p distinguishes between *yes* and *no* instances. Furthermore, by Theorem 3.2.1, p is a linear combination of polynomials of the form $\mathrm{Tr}_\sigma(A_1, \ldots, A_r)$ where σ is a permutation on $r \leq 2T$ elements and each A_i is U or U^*. Since $UU^* = I$, then each $\mathrm{Tr}_\sigma(A_1, \ldots, A_r)$ is a product of terms of the form $\mathrm{Tr}(U^p)$ or $\mathrm{Tr}((U^*)^q)$ for $p, q \leq r$. Observe that each generator $\mathrm{Tr}(U^p) = \sum_{i=1}^d z_i^p$ is a power sum symmetric polynomial in the eigenvalues z_i of U. Hence, since each term Tr_σ is a polynomial in the eigenvalues $(z_1, \ldots, z_d)$ of U and their conjugates of degree $\leq 2T$, that is symmetric under permutations of $(z_1, \ldots, z_n)$ or $(z_1,^*, \ldots, z_n^*)$, p satisfies the same property since p is a linear combination of the polynomials Tr_σ. □

[1] Here, symmetric means that for all permutations $\pi : [d] \to [d]$, permuting the variables $z_i \to z_{\pi(i)}$ and $z_i^* \to z_{\pi(i)}^*$ leaves the polynomial q unchanged.

We now illustrate some applications of the general theory developed in the previous section. The applications will make crucial use of Theorem 3.2.2 for testing unitarily invariant properties.

3.2.2 Testing Unitarily Invariant Subspace Properties

We first start with the class of unitarily invariant subspace properties. Recall that a subspace property $\mathcal{P}$ is one where all instances are reflections about some subspace, i.e., $U = I - 2\Pi$ where Π is the orthogonal projector onto a subspace $S \subseteq \mathbb{C}^d$ (we say that U encodes the subspace S). Such unitaries have eigenvalues 1 or -1.

We will show that lower bounds for testing $\mathcal{P}$ follow immediately from lower bounds for testing symmetric properties $\mathcal{S}$ of classical strings, which means that the instances of $\mathcal{S}$ are d-bit strings, and the *yes* instances are invariant under permutation of the coordinates (and similarly with the *no* instances).

There is a one-to-one correspondence between unitarily invariant subspace properties and symmetric classical properties:

- Given a unitarily invariant subspace property $\mathcal{P}$, we define $\mathcal{S}_{yes/no} = \{\mathrm{spec}(U) : U \in \mathcal{P}_{yes/no}\}$, where $\mathrm{spec}(U)$ denotes the multiset of eigenvalues of U, interpreted as a d-bit string (with $+1$ mapped to 0 and -1 mapped to 1). The resulting classical property $\mathcal{S}$ is symmetric.
- Given a classical symmetric property $\mathcal{S}$, we define $\mathcal{P}_{yes/no} = \{V^* D_x V : x \in \mathcal{S}_{yes/no}, V \in \mathrm{U}(d)\}$ where D_x is a diagonal matrix with $(-1)^{x_i}$ on the i'th diagonal entry. The resulting unitary property $\mathcal{P}$ is a subspace property and is unitarily invariant.

It is straightforward to see that this correspondence is a bijection.

We now establish the following simple relation between the query complexity of the classical property $\mathcal{S}$ to that of the quantum proprety $\mathcal{P}$:

Proposition 1.2.4. *Let $\mathcal{P}$ be a unitarily invariant subspace property and let $\mathcal{S}$ be the associated symmetric classical property. The query complexity of distinguishing between* yes *and* no *instances of $\mathcal{P}$ is at least the minimum degree of any polynomial that distinguishes between the* yes *and* no *instances of $\mathcal{S}$.*

Proof. The oracles corresponding to subspace properties satisfy $U = U^*$ and $U^2 = I$. Since for all integer j, the traces $\mathrm{Tr}(U^j)$ and $\mathrm{Tr}((U^*)^j)$ are either equal to $\mathrm{Tr}(I) = d$ or $\mathrm{Tr}(U) = \mathrm{Tr}(I - 2\Pi) = d - 2\dim(\Pi)$, Theorem 3.2.1 implies that the acceptance probability can be expressed as a degree-$2T$ polynomial in d and $d - 2\dim(\Pi)$; since d is constant, we can perform a change of variables to obtain a degree-$2T$ univariate polynomial in $\dim(\Pi)$ only. Thus the polynomial q also decides the associated classical symmetric property $\mathcal{S}$, by considering k as the Hamming weight of the associated string $\mathrm{spec}(U)$.

□

We note that one can also prove Proposition 1.2.4 by observing that a T-query tester for a unitarily invariant subspace property $\mathcal{P}$ is also a T-query tester for the associated classical symmetric property $\mathcal{S}$. We now mention some easy applications of Proposition 1.2.4.

Theorem 3.2.3 (Unstructured Search Lower Bound)**.** *Any tester that decides whether an oracle U is a reflection about some quantum state $|\psi\rangle$, i.e., $U = I - 2|\psi\rangle\langle\psi|$, or is the identity $U = I$, must make $\Omega(\sqrt{d})$ queries to U.*

Proof. Define $\mathcal{P}_{yes} = \{I - 2|\psi\rangle\langle\psi| : |\psi\rangle \in \mathbb{C}^d\}$ and $\mathcal{P}_{no} = \{I\}$. Clearly $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$ is a unitarily invariant subspace property. The associated classical property $\mathcal{S}$ consists of *yes* instances that are binary strings of Hamming weight 1 (because the *yes* instances of $\mathcal{P}$ have exactly one -1 eigenvalue) and the *no* instances is the all zeroes string. This is essentially the Grover search problem, it is well-known via the standard polynomial method [BBC+01] that any polynomial deciding $\mathcal{S}$ requires $\Omega(\sqrt{d})$ queries, which implies $\Omega(\sqrt{d})$ query lower bound for property $\mathcal{P}$ by Proposition 1.2.4.

□

We also consider the *Approximate Dimension* problem, which for some integer parameter $0 \leq w \leq d$, distinguish between whether the subspace encoded by the oracle U has dimension at least $2w$ (*yes* instances) or at most w. This is a unitarily invariant subspace property testing problem, as conjugating a reflection $U = I - 2\Pi$ by any unitary V leaves the dimension of the encoded subspace unchanged. This generalizes the classical Approximate Counting problem, which is to determine whether the Hamming weight of an input string is at most w or at least $2w$. Again leveraging the standard polynomial method we obtain the following lower bound:

Theorem 1.2.5 (BQP lower bound for Approximate Dimension)**.** *Any tester that decides between whether a unitary encodes a subspace of dimension at least $2w$ or at most w requires $\Omega(\sqrt{\frac{d}{w}})$ queries.*

Proof. The associated classical property, Approximate Counting, is where the *yes* instances correspond to strings with Hamming weight at least $2w$ and the *no* instances have Hamming weight at most w. By reduction to the Grover search problem, we get that the degree of any polynomial that decides Approximate Counting is at least $\Omega(\sqrt{d/w})$, which by Proposition 1.2.4 is also a lower bound on the number of queries needed to decide the Approximate Dimension problem.

□

We note that by using an appropriate modification of the quantum counting algorithm of Brassard et al. [BHT98], we obtain a matching upper bound.

Proposition 3.2.4. *There exists a tester that using $O(\sqrt{\frac{d}{w}})$ queries and certifies whether or not a unitary $U = I - 2P$ encodes a subspace S of dimension at least $2w$ or at most w.*

Proof. Prepare the maximally entangled state $|\Phi\rangle$ in $\mathbb{C}^d \otimes \mathbb{C}^d$ and observe that $|\Phi\rangle$ can be written as $|\Phi\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |v_i\rangle |\overline{v_i}\rangle$ for any basis $B = \{|v_i\rangle\}$ of $\mathbb{C}^d$. Hence, we can assume that $B = B_1 \cup B_2$ where B_1 is a basis for S and B_2 is a basis for the orthogonal complement $S^\perp$.

Let $s = \dim S$, and $|\Phi_S\rangle = \frac{1}{\sqrt{s}} \sum_{|v_i\rangle \in B_1} |v_i\rangle |\overline{v_i}\rangle$ be maximally entangled over S and $|\Phi_{S^\perp}\rangle = \frac{1}{\sqrt{d-s}} \sum_{|v_i\rangle \in B_2} |v_i\rangle |\overline{v_i}\rangle$ be maximally entangled over $S^\perp$. Let $R = 2|\Phi\rangle\langle\Phi| - 1$ be the reflection around the maximally entangled state. Observe that $|\Phi\rangle \in \text{span}\{|\Phi_S\rangle, |\Phi_{S^\perp}\rangle\}$ and furthermore $(U \otimes I)|\Phi_S\rangle = -|\Phi_S\rangle$, and $(U \otimes I)|\Phi_{S^\perp}\rangle = |\Phi_{S^\perp}\rangle$. Hence, by the analysis of Grover search, the operator $G = R(U \otimes I)$ is a rotation in the plane spanned by $|\Phi_S\rangle$ and $|\Phi_{S^\perp}\rangle$ by an angle of 2θ, where $\sin^2\theta = \frac{s}{d}$. Hence applying phase estimation with the operator G and the state $|\Phi\rangle$ as input, produces an estimate of the angle θ and hence the dimension s since the eigenvalues of G are $e^{\pm 2i\theta}$. By the analysis of phase estimation, at most $O(\sqrt{\frac{d}{w}})$ oracle calls to G can be used to get an estimate $\tilde{s}$ of s satisfying $0.9s \leq \tilde{s} \leq 1.1s$, and hence we can distinguish whether or not $s \geq 2w$ or $s \leq w$ with access to this estimate $\tilde{s}$. □

Aaronson, et al. [AKKT20] showed that having access to a quantum proof does not help reduce the query complexity of the classical Approximate Counting problem, unless the proof state is very large (at

least w qubits). Since a QMA tester for Approximate Dimension is automatically a QMA tester for the Approximate Counting problem, the lower bound proved by [AKKT20] directly gives the following:

Theorem 1.2.6 (QMA lower bound for Approximate Dimension)**.** *Suppose there is a T-query algorithm that solves the Approximate Dimension problem (i.e. deciding whether a d-dimensional unitary encodes a subspace of dimension at least $2w$ or at most w) with the help of a m-qubit proof. Then either $m = \Omega(w)$, or $T \geq \Omega(\sqrt{\frac{d}{w}})$.*

We note that Proposition 3.2.4 shows that Theorem 1.2.6 is tight in the regime where the quantum proof satisfies $m = o(w)$. Otherwise, we conjecture that providing $O(w)$ copies of the mixed state ρ, where ρ is maximally mixed over the the hidden subspace S and performing swap tests to estimate the purity of ρ, suffices to solve the approximate dimension problem. However, this algorithm seems to require an unentanglement guarantee on the witness, which does not immediately show that it is a QMA tester. We leave this investigation to further work.

3.2.3 Recurrence Times of Unitaries

We now turn to analyzing the problem of testing recurrence times of unitaries. This corresponds to analyzing unitarily invariant properties that are not subspace properties. As mentioned in the introduction, one cannot directly use lower bounds on a related classical property testing problem; instead we have to make full use of the generalized polynomial method.

Recall the Recurrence Time problem defined in the introduction:

Definition 1.2.2 (Recurrence Time Problem)**.** The (t, ϵ)-Recurrence Time problem is to decide, given oracle access to a unitary U, whether $U^t = I$ (*yes* case) or $\|U^t - I\| \geq \epsilon$ in the spectral norm (*no* case), promised that one is the case.

Upper Bound. We first present an upper bound on the query complexity of the Recurrence Time Problem.

Theorem 1.2.8. *The (t, ϵ)-Recurrence Time problem can be solved using $O(t\sqrt{d}/\epsilon)$ queries.*

Proof. Fix an integer t. The goal is to determine whether there is an eigenvector $|\psi\rangle$ of U^t such that the phase $e^{2\pi i\varphi}$ associated with $|\psi\rangle$ is more than ϵ far from 1, or, equivalently, whether the phase $e^{2\pi i\theta}$ of $|\psi\rangle$ with respect to U satisfies $2\pi i\theta t$ being more than ϵ away from an integer multiple of $2\pi i$. If there is no such eigenvector, then t is an ϵ-recurrence time for U. To find such an eigenvector, we first prepare the d-dimensional maximally entangled state $|\Phi\rangle = \frac{1}{\sqrt{d}} \sum_j |j\rangle |j\rangle$. Let $\{|\psi_j\rangle\}_j$ denote an eigenbasis for U with associated eigenvalues $\{e^{2\pi i\theta_j}\}_j$; then we have that $|\Phi\rangle$ can be equivalently expressed as

$$|\Phi\rangle = \frac{1}{\sqrt{d}} \sum_j |\psi_j\rangle |\overline{\psi_j}\rangle$$

where $|\overline{\psi_j}\rangle$ denotes the complex conjugate of $|\psi_j\rangle$ with respect to the standard basis. We perform phase estimation on the first register of $|\Phi\rangle$ with respect to U to estimate the phases θ_j up to $\pm\epsilon/8t$ additive error, with success probability at least, say, 99%. The analysis of [NC10, Section 5.2.1] shows that this requires $O(t/\epsilon)$ calls to the unitary U.

The state then has the form

$$|\Phi'\rangle = \frac{1}{\sqrt{d}} \sum_{j,k} \alpha_{j,k} |\widetilde{\theta}_j^{(k)}\rangle |\psi_j\rangle |\overline{\psi_j}\rangle$$

where $\widetilde{\theta}_j^{(k)}$ are the estimates of $|\theta_j\rangle$ from phase estimation, and $\alpha_{j,k}$ are the amplitudes of each of the estimates. As mentioned, the sum of squares of amplitudes $\alpha_{j,k}$ such that the estimate $\widetilde{\theta}_j^{(k)}$ differs from θ_j by more than $\epsilon/8t$ (we call such an estimate $\widetilde{\theta}_j^{(k)}$ *bad*, otherwise it is *good*) is at most 1%.

We now perform amplitude amplification in order to identify whether there is an estimate $|\widetilde{\theta}_j^{(k)}\rangle$ such that

$$\left|e^{2\pi i t\widetilde{\theta}_j^{(k)}} - 1\right| \geq \epsilon/2 \ . \tag{3.1}$$

The amplitude amplification procedure will alternate between applying a phase on the $|\widetilde{\theta}_j^{(k)}\rangle$ states satisfying (3.1), and reflecting about the state $|\Phi'\rangle$. Let P be the projector onto estimates satisfying (3.1) in the first register.

In the *no* case, either phase estimation fails, which occurs with occurs at most 1% probability, or there is an estimate for a phase $|\theta_j^k\rangle$ that is ϵ-far away from 1. We claim that amplitude amplification finds a phase satisfying the condition (3.1) with constant probability. If there is a phase θ_j such that $|e^{2\pi i t\theta_j} - 1| \geq \epsilon$, then the good estimates $\widetilde{\theta}_j^{(k)}$ of θ_j satisfy

$$\begin{aligned}
\left|e^{2\pi i t\widetilde{\theta}_j^{(k)}} - 1\right| &\geq \left|e^{2\pi i t\theta_j} - 1\right| - \left|e^{2\pi i t\widetilde{\theta}_j^{(k)}} - e^{2\pi i t\theta_j}\right| && \text{(triangle inequality)} \\
&\geq \epsilon - 4t\left|\theta_j - \widetilde{\theta}_j^{(k)}\right| && \text{(calculus)} \\
&\geq \epsilon - \epsilon/2 = \epsilon/2.
\end{aligned}$$

Thus when phase estimation succeeds, the initial state satisfies $|P|\Phi'\rangle| \geq \frac{1}{\sqrt{d}}$ and hence when the marked phase is unique, $O(\sqrt{d})$ iterations suffice to boost the probability on the marked phase to constant probability. In the case where there are multiple phases satisfying the condition (3.1), then we run the amplitude amplification algorithm for $\sqrt{d}, \sqrt{\frac{d}{2}}, \sqrt{\frac{d}{4}}, \ldots$, iterations and so on. Since a marked item can be found with $O(\sqrt{\frac{d}{k}})$ iterations if there are k marked items, the binary search procedure terminates in $O(\sqrt{d})$ iterations and finds a phase that is ϵ-far from 1 with constant probability.

Otherwise, in the *yes* case where $U^t = I$, the analysis of [NC10, Section 5.2.1] shows that the phase estimation algorithm produces exact values for the phase register $|\theta_j^k\rangle$ since the phases are integer multiples of 2π. Hence, the initial state in the amplitude amplification algorithm has no overlap with the subspace satisfying (3.1) and hence the final state after amplification remains the same as the initial state up to a global phase. Thus, the algorithm never finds a phase ϵ far from 1.

Hence in $O(\frac{t\sqrt{d}}{\epsilon})$ iterations we are able to distinguish between the *yes* and *no* cases with constant bias.

□

Lower Bound. Using the generalized polynomial method for unitarily invariant properties, we prove the following query lower bound for the Recurrence Time problem.

Theorem 1.2.7. *Let $\epsilon \leq \frac{1}{2\pi}$. Any quantum query algorithm solving the (t,ϵ)-Recurrence Time problem for d-dimensional unitaries with error ϵ must use $\Omega(\max(\frac{t}{\epsilon}, \sqrt{d}))$ queries.*

Before doing this, we introduce a useful symmetrization lemma we use to reduce the number of variables of the polynomial.

Lemma 3.2.5. *Let $q(z_1, \ldots, z_d)$ be the polynomial obtained from Theorem 3.2.2 for the acceptance probability of a T-query algorithm on the Recurrence Time problem.*

Let $D(p, z)$ be a distribution on d-dimensional diagonal unitaries where each diagonal entry is chosen to be equal to $z = e^{i\theta}$ with probability p and otherwise equal to 1 with probability $1-p$. Then the expected value

$$r(p, z) = \mathbb{E}_{(z_1,\ldots,z_d)\sim D(p,z)}[q(z_1, \ldots, z_d)]$$

is a self-adjoint polynomial of degree at most $2 \deg q$.

Proof. Recurrence Time is a unitarily invariant property as $U^t = I$ iff $(VUV^*)^t = I$ and also $\|U^t - I\| \geq \epsilon$ iff $\|(VUV^*)^t - I\| \geq \epsilon$ for any unitaries U and V. Therefore, Theorem 3.2.2 guarantees that the acceptance probability of a T-query algorithm for the problem can be written as a degree $\leq 2T$ self-adjoint polynomial in the eigenvalues of U.

Since q is a self-adjoint polynomial defined on the unit circle, p can be expanded in a basis of binomials $z_I z_J^* + z_J z_I^*$ where $I \subseteq [n], J \subseteq [n]$, $I \cap J = \emptyset$, $z_I = \prod_{i\in I} z_i$ and $z_J = \prod_{j\in J} z_j^*$. The expected value of each binomial under when the eigenvalues are chosen according to $D(p, z)$ is then

$$\sum_{k_1=0}^{|I|}\sum_{k_2=0}^{|J|}\binom{|I|}{k_1}\binom{|J|}{k_2}p^{k_1+k_2}(1-p)^{|I|+|J|-k_1-k_2}[z^{k_1-k_2} + (z^*)^{k_1-k_2}].$$

Hence the expected value $r(p, z) = \mathbb{E}_{(z_1,\ldots,z_d)\sim D(p,z)}[q(z_1, \ldots, z_d)]$ is a polynomial of degree at most $2 \deg q$ with the property that $r(p, z) = r(p, z^*)$. □

We are now ready to prove Theorem 1.2.7.

Proof. By Lemma 3.2.5, if there was a T-query algorithm for the Recurrence Time problem, $r(p, z)$ is a polynomial of degree at most $4T$ that represents the expected probability the algorithm accepts on the distribution $D(p, z)$. We now lower bound the degree of q by lower bounding the degrees of p and z separately.

Firstly we lower bound the degree of p by fixing $z' = \exp(\frac{4\pi i\epsilon}{t})$. For this value of z', $r_1(p) = r(p, z')$ is a real-valued univariate polynomial with the property that $r_1(0) \geq \frac{2}{3}$ (since if $p = 0$ we are given the identity unitary as input).

Otherwise if $p = \frac{2}{d}$, the number of eigenvalues equal to z' is a binomial random variable with d trials and success probability $p = \frac{2}{d}$, which for sufficiently large d is approximately Poisson distributed with mean equal to 2. Hence, for sufficiently large d, the probability that the input is the identity is at most e^{-2}. If not, the input is a *no* instance, since in this case U^t would have an eigenvalue equal to $(z')^t = \exp(4\pi i\epsilon)$, and therefore

$$\|U^t - I\| = \sqrt{2 - 2\cos(4\pi\epsilon)} \geq \sqrt{\frac{2}{3}(4\pi\epsilon)^2} \geq 10\epsilon,$$

by assumption that $|4\pi\epsilon| \leq 2$ and Lemma 3.1.7.

Hence we have that

$$r_1\left(\frac{2}{d}\right) \leq e^{-2} + \frac{1}{3}(1 - e^{-2}) \leq \frac{1}{2}.$$

Therefore, r_1 satisfies the properties that $0 \leq r_1(p) \leq 1$ for all $0 \leq p \leq 1$, $r_1(0) \geq \frac{2}{3}$, and $r_1(\frac{2}{d}) \leq \frac{1}{2}$. By Markov's inequality (Lemma 3.1.3), the inequality

$$\frac{d}{12} \leq 2(\deg r_1)^2,$$

must be satisfied, so $\deg r_1 \geq \Omega(\sqrt{d})$.

Now we lower bound by degree of z by fixing $p = \frac{2}{d}$ and consider the polynomial $r_2(z) = r(p, z)$. Observe that r_2 has the property that $r_2(z^*) = r_2(z)$. Hence Lemma 3.1.6 applies and we can assume $r_2(z) = s_2(z+z^*)$ for some real-valued polynomial s_2 of the same degree. Observe that the s_2 is bounded by one and defined on the interval $[-2, 2]$. Furthermore, for $z_1 = 1$, we have $s_2(z_1 + z_1^*) = r(p, z_1) \geq \frac{2}{3}$, and otherwise for $z_2 = \exp(\frac{4\pi i\epsilon}{t})$, we have from the previous calculation that $s_2(z_2 + z_2^*) = r(p, z_2) \leq \frac{1}{2}$. Since by Lemma 3.1.7 and the assumption that $\epsilon \leq \frac{1}{2\pi}$,

$$|(z_1 + z_1^*) - (z_2 + z_2^*)| = |2 - 2\cos(\frac{4\pi\epsilon}{t})| \leq \frac{16\pi^2\epsilon^2}{t^2},$$

we conclude that the derivative of s_2 must satisfy $|s_2'(x)| \geq \frac{t^2}{96\pi^2\epsilon^2}$. for some point $x \in [2 - 2\cos(\frac{4\pi\epsilon}{t}), 2]$. Hence, by Markov's inequality, we have that the degree of s_2 must satisfy:

$$\frac{t^2}{96\pi^2\epsilon^2} \leq \frac{2(\deg s_2)^2}{4},$$

Hence, $\deg r_2 = \deg s_2 \geq \Omega(\frac{t}{\epsilon})$.

Therefore, combining the two lower bounds implies that there must be monomials in $r(p, z)$ with p-degree at least $\Omega(\sqrt{d})$ and z-degree at least $\Omega(\frac{t}{\epsilon})$. Hence, since $\deg r \leq 2 \deg q \leq 4T$ where T is the query complexity of the algorithm, we obtain $T \geq \Omega(\max(\frac{t}{\epsilon}, \sqrt{d}))$. □

We note that a similar lower bound can also be obtained using hybrid method of [BBBV97]. However, it is unclear whether the hybrid method can be used to obtain lower bounds in the QMA setting. We now modify the previous arguments to show that the Recurrence Time problem remains hard even when the tester receives a quantum proof that is not too large.

Theorem 1.2.9 (QMA lower bound for the Recurrence Time problem)**.** *Let $\epsilon \leq \frac{1}{2\pi}$. Suppose there is a T-query algorithm that solves the Recurrence Time problem for d-dimensional unitaries with the help of an m-qubit proof. Then either $m \geq \Omega(d)$, or $T \geq \Omega(\max(\sqrt{\frac{d}{m}}, \frac{t}{m}, \frac{1}{\epsilon}))$.*

Proof. If $m \geq \Omega(d)$ we are done, so we assume that $m \leq o(d)$.

Let $r(p, z)$ be obtained from Lemma 3.2.5 and Lemma 3.1.1. Again we will lower bound the degree of q by considering the degree of p and z separately.

We first lower bound the degree of p by fixing $z' = \exp(\frac{4\pi i\epsilon}{t})$. For this value of z', $r_1(p) = r(p, z')$ is a real-valued univariate polynomial with the property that $r(0) \geq 2^{-m}$. Otherwise for all $p \geq \frac{10m}{d}$, we have the probability that there is there is no non-trivial eigenvalue is bounded by $(1-p)^d \leq \exp(-dp) \leq \exp(-10m)$, and hence for all $\frac{10m}{d} \leq p \leq 1$, we have

$$r_1(p) \leq \exp(-10m) + 2^{-10m}(1 - \exp(-10m)) \leq 2^{-9m}$$

since this value of z' corresponds to a no instance.

Therefore, if $y_0 = \frac{10m}{d}, y_1 = 1$ the polynomial

$$s_1(x) = 2^{9m} r_1\left(\frac{y_0 - y_1}{2}(x-1) + y_0\right),$$

implies that the polynomial s_1 satisfies $\deg s_1 \leq \deg r_1$, $|s_1(x)| \leq 1$ for all $|x| \leq 1$, and $s(1+\frac{2y_0}{y_1-y_0}) \geq 2^{8m}$. Observe that $\frac{2y_0}{y_1-y_0} = \frac{20m}{d-10m} \leq \frac{40m}{d}$ by our assumption on m. Hence, applying Paturi's Lemma (Lemma 3.1.4) with these conditions and $\mu = \frac{40m}{d}$, we obtain the inequality:

$$2^{8m} \leq \exp(2(\deg s_1)\sqrt{\mu^2 + 2\mu}) \leq \exp(4(\deg s_1)\sqrt{\mu})$$

since by assumption $\mu \leq 2$. Therefore, solving for $\deg s_1$ implies that $\deg r_1 \geq \deg s_1 \geq \Omega(\sqrt{md})$.

Now we lower bound the degree of z by fixing $p = \frac{10m}{d}$ and consider the polynomial $r_2(z) = r(p, z)$. Observe that r_2 has the property that $r_2(z^*) = r_2(z)$. Hence Lemma 3.1.6 applies and we can assume $r_2(z) = s_2(z + z^*)$ for some real-valued polynomial s_2 of the same degree. Observe that since for any integer j, since any unitary whose only eigenvalue is $z = \exp(\frac{2\pi ij}{t})$ is a *yes* instance, we have

$$r_2\left(\exp(\frac{2\pi ij}{t})\right) = s_2\left(2\cos\frac{2\pi j}{t}\right) \geq 2^{-m}.$$

Otherwise, there is at least one point in the interval $x \in (2\cos\frac{2\pi(j+1)}{t}, 2\cos\frac{2\pi j}{t})$, where $s_2(x) \leq 2^{-9m}$, since all unitaries whose only eigenvalue is equal to $z = \exp(\frac{2\pi i(j+1/2)}{t})$ corresponds to a *no* instance, which corresponds to the point $z + z^* = x = 2\cos\frac{2\pi(j+1/2)}{t}$. .

Hence, $s_2(x) - (\frac{2^{-m}+2^{-9m}}{2})$ has at least $\frac{t}{2}$ roots in the interval $[-2, 2]$ since $s_2 - (\frac{2^{-m}+2^{-9m}}{2})$ changes sign at least $\frac{t}{2}$ times, which implies that the degree of s_2 at least $\frac{t}{2}$ by the Fundamental Theorem of Algebra (Lemma 3.1.2). Therefore, since $\deg r_2 = \deg s_2$, the degree of $r_2(z)$ is at least $\Omega(t)$.

Finally, we consider the dependence on the error ϵ. Furthermore, since $s_2(2) \geq 2^{-m}$ and $s_2(x) \leq 2^{-10m}$ for all $y_0 = 2\cos\frac{2\pi(1-\epsilon)}{t} \leq x \leq y_1 = 2\cos\frac{4\pi\epsilon}{t}$, we have that

$$s_3(x) = 2^{10m} s_2\left(\frac{y_1 - y_0}{2}(x-1) + y_1\right),$$

satisfies $|s_3(x)| \leq 1$ for all $|x| \leq 1$, and that when $x = 1 + \frac{2(2-y_1)}{y_1-y_0}$ we have $s_3(x) \geq 2^{9m}$. By Lemma 3.1.7, $y_1 - y_0 \geq \frac{8\pi^2}{3t^2} + O(\epsilon)$ and $2 - y_1 \leq \frac{16\pi^2\epsilon^2}{t^2}$. Hence, we may take $\mu \leq O(\epsilon^2)$ in Paturi's Lemma to conclude that $\deg s_3$ satisfies

$$2^{9m} \leq \exp(4(\deg s_3)\sqrt{\mu}) = \exp(2(\deg s_3)O(\epsilon)),$$

and hence $\deg r_2 = \deg s_2 \geq \deg s_3 \geq \Omega(\frac{m}{\epsilon})$.

Putting these bounds together, we conclude that either $m \geq \Omega(d)$, or otherwise, since $\deg r_1 \leq \deg r$ and $\deg r_2 \leq \deg r$, we have

$$\max(\Omega(\sqrt{md}), \Omega(t), \Omega(\frac{m}{\epsilon})) \leq \deg r \leq 2\deg q \leq O(mT),$$

which was the claimed bound. □

Observe that there is a weaker dependence on ϵ in our QMA lower bound, compared to the BQP lower bound for the Recurrence Time problem. We leave improving this dependence to further work.

We end with some brief observations about the coQMA query complexity of the problem where we provide a certificate for *non*-recurrence. We note that the query complexity of the problem changes significantly in the coQMA setting compared to the QMA setting. Here, a valid certificate is an eigenvector $|\psi\rangle$ of U^t with eigenvalue not equal to $e^{i\theta}$ where $\theta \in [-\epsilon, \epsilon]$, and a quantum phase estimation can be used with $O(\frac{t}{\epsilon})$ queries to compute the the corresponding eigenvalue of $|\psi\rangle$ to $O(\epsilon)$ precision. In particular, there is no dependence on the dimension d. Therefore, in the setting where the recurrence time t is constant, the unitary recurrence problem provides an exponential query complexity separation between QMA and coQMA.

3.2.4 Local Unitary Invariants

Recall from the introduction the definition of the local unitary group.

Definition 3.1.2 (Local Unitary Group)**.** Let $d_1, d_2 \geq 2$. The local unitary group $\mathrm{LU}(d_1, d_2)$ is the subgroup $\mathrm{U}(d_1) \times \mathrm{U}(d_2)$ of $\mathrm{U}(d_1 d_2)$ consisting of all unitaries of the form $g \otimes h$ where $g \in \mathrm{U}(d_1), h \in \mathrm{U}(d_2)$.

As discussed in the introduction, LU-invariance naturally captures the symmetry associated with entanglement properties of states and operators. Proposition 1.2.3 implies that a T-query tester for an LU-invariant property $\mathcal{P}$ gives rise to a degree-$2T$ polynomial q belonging to the invariant ring $\mathbb{C}[X, X^*]^{\mathrm{LU}(d_1,d_2)}$ that decides $\mathcal{P}$, where X, X^* represent the variables and their conjugates of matrices acting on $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$.

As with the full unitary group case, it is possible to characterize the polynomial functions on matrices that are invariant under the local unitary group. The next theorem, due to Procesi [Pro76] and Brauer [Bra37], presents such a characterization.

Theorem 3.2.6 (Generators for LU-invariant polynomials)**.** *Let σ, τ be permutations on k elements and let R_σ, R_τ be the corresponding permutation operators on $(\mathbb{C}^d)^{\otimes k}$. Then the homogenous degree k part of the invariant ring $\mathbb{C}[X]^{\mathrm{LU}(d,d)}$, where X represents the variables of a matrix acting on $\mathbb{C}^d \otimes \mathbb{C}^d$, is in the linear span of the polynomials* $\mathrm{Tr}((R_\sigma \otimes R_\tau)X^{\otimes k})$,[2] *ranging over all permutations $\sigma, \tau \in S_k$.*

As an aside, we note that [BBL13] has provided an interpretation of these invariants in terms of *tensor networks*, which are a visual tool for representing high dimensional tensors. We also note that these invariants have been studied extensively in the pure mathematics and physics literature [QSY20, GRB98, TM+17].

Ultimately, we would like to use Theorem 3.2.6 to prove query complexity lower bounds on LU-invariant properties. However, this characterization of the LU-invariant ring, while explicit, appears less simple to use than Theorem 3.2.1. In the full unitarily invariant case, the generators $\mathrm{Tr}(R_\sigma X^{\otimes k})$ are symmetric polynomials depending only on the cycle structure of σ and the eigenvalues of X. This information is sufficient for us to leverage tools from approximation theory to lower bound the degree the invariant polynomial.

In contrast, it is not so clear how to make use of the quantities $\mathrm{Tr}((R_\sigma \otimes R_\tau)U^{\otimes k})$ for general unitaries U; for example we do not know if the traces can be expressed as a polynomial of some natural quantities (like how the eigenvalues of U are natural linear-algebraic quantities) that capture some entanglement properties of U. However, there is a special case for which we can give a good characterization of the invariant polynomials, which is when X is a projector onto a one-dimensional subspace:

[2] The way the operators should be multiplied is as follows: if the i'th copy of X acts on registers $A_i B_i$, then R_σ permutes the A_i registers and R_τ permutes the B_i registers.

Theorem 3.2.7 ([BBL13, Theorem 22]). *Let $\Pi = |\psi\rangle\langle\psi|$ be the projector onto a bipartite state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$. Let $\rho = \sum_i \lambda_i |v_i\rangle\langle v_i|$ denote the reduced density matrix of ρ on the first subsystem. Let $\sigma, \tau \in S_k$. Then $\mathrm{Tr}((R_\sigma \otimes R_\tau)\Pi^{\otimes k})$ is a symmetric degree-k polynomial in the eigenvalues λ_i of ρ.*

The tuple of eigenvalues $(\lambda_1, \ldots, \lambda_d)$ is called the *entanglement spectrum* of $|\psi\rangle$. This has the following consequence for LU-invariant one-dimensional subspace properties:

Lemma 3.2.8. *Let $\mathcal{P} = (\mathcal{P}_{yes}, \mathcal{P}_{no})$ denote a* LU*-invariant subspace property where the instances consist of reflections $U = I - 2|\psi\rangle\langle\psi|$ for some pure state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$. Suppose there is a T-query tester for $\mathcal{P}$ that accepts* yes *instances with probability at least a and* no *instances with probability at most b. Then there exists a degree-$2T$ symmetric polynomial $p(\lambda_1, \ldots, \lambda_d)$ such that*

- *If $U \in \mathcal{P}_{yes}$, then $p(\lambda_1, \ldots, \lambda_d) \geq a$.*
- *If $U \in \mathcal{P}_{no}$, then $p(\lambda_1, \ldots, \lambda_d) \leq b$.*

Here, p is evaluated at the entanglement spectrum $(\lambda_1, \ldots, \lambda_d)$ of the pure state $|\psi\rangle$ corresponding to U.

Proof. By Proposition 1.2.3 there exists a degree-$2T$ polynomial $q(U, U^*)$ belonging to the invariant ring $\mathbb{C}[X, X^*]^{\mathrm{LU}(d,d)}$ that decides $\mathcal{P}$ with the acceptance probabilities at least a and at most b for *yes* and *no* instances respectively. However since U is self-adjoint this means that q in fact belongs to the invariant ring $\mathbb{C}[X]^{\mathrm{LU}(d,d)}$. Since $U = I - 2|\psi\rangle\langle\psi|$, the polynomial q can be equivalently expressed as a degree-$2T$ function of the projector $|\psi\rangle\langle\psi|$. By Theorem 3.2.6, the polynomial q can be expressed as a linear combination of $\mathrm{Tr}((R_\sigma \otimes R_\tau)|\psi\rangle\langle\psi|^{\otimes k})$ over all permutations σ, τ of k elements with $1 \leq k \leq 2T$. By Theorem 3.2.7, these traces are degree-k symmetric polynomials in the entanglement spectrum of $|\psi\rangle$. Put together, this yields the desired polynomial p. □

3.2.5 The Entanglement Entropy Problem

We use Lemma 3.2.8 to prove lower bounds on an entanglement testing problem. Recall the Entanglement Entropy problem as defined in the introduction:

Definition 3.1.4 (Rényi 2-entropy)**.** Given a state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ with reduced density matrix on the first register ρ, the Rényi 2-entropy of $|\psi\rangle$ is defined as $H_2(|\psi\rangle) = -\log \mathrm{Tr}(\rho^2)$.

Definition 1.2.3 (Entanglement Entropy Problem)**.** Let $0 < a < b \leq \log d$. Given oracle access to a reflection oracle $U = I - 2|\psi\rangle\langle\psi|$ where $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$, decide whether or not the state $|\psi\rangle$ satisfies one of the following two conditions, promised one of the following is the case:

- Low entropy case: $H_2(|\psi\rangle) \leq a$
- High entropy case: $H_2(|\psi\rangle) \geq b$

Recall that the entanglement entropy is invariant under local unitaries, and that the entanglement entropy can be computed by the formula $H_2(|\psi\rangle) = -\log(\sum_{i=1}^d \lambda_i^2)$ where λ_i are the eigenvalues of the reduced density ρ. In particular, if the reduced density ρ of $|\psi\rangle$ was maximally mixed on a subspace of dimension r, then $H_2(|\psi\rangle) = \log r$. With a fairly straightforward application of the polynomial, we can prove the following query lower bound for the entanglement entropy problem.

Theorem 1.2.10. *Assume $a \geq 5$. Given parameters $a < b \leq \log d$, any tester must make $\Omega(\exp(a/4))$ queries to distinguish between the low and high entropy cases in the Entanglement Entropy problem.*

Proof. By properties of the entanglement entropy, we observe that if oracle $O = I - 2\left|\psi\right\rangle\left\langle\psi\right|$ satisfies the low or high entropy condition, so does the oracle $(U \otimes V)O(U^* \otimes V^*)$ for any unitaries U and V. Hence applying Lemma 3.2.8 implies that if there was a T query algorithm to solve the problem, there exists a degree $\leq 2T$ symmetric polynomial $p(\lambda_1, \ldots, \lambda_n)$ which represents the success probability of the algorithm where $(\lambda_1, \ldots, \lambda_n)$ are the eigenvalues of the reduced density ρ of $\left|\psi\right\rangle$.

Let $r \geq 1$ be an integer and consider the success probability of the algorithm on instances where ρ has r eigenvalues equal to $\frac{1}{r}$ (i.e. ρ is maximally mixed on a subspace of dimension r). For this set of eigenvalues Λ_r we have $\lambda_1^i + \cdots + \lambda_n^i = \frac{1}{r^{i-1}}$. Hence, the substitution $q_1(r) = p(\Lambda_r)$ produces a Laurent polynomial with non-positive exponents only. This means that $q_2(r) = q_1(\frac{1}{r})$ is a polynomial satisfying the properties that:

- $q_2(\exp(-a))) \leq \frac{1}{3}$ to satisfy the low entropy case.
- $q_2(\exp(-b)) \geq \frac{2}{3}$ to satisfy the high entropy case
- $0 \leq q_2(i) \leq 1$ at all points $i = \frac{1}{n}$.

Now we bound the degree of q_2. First assume that that $q_2(x)$ is bounded by 2 for all x in the range $x \in [\frac{1}{d}, \exp(-a/2)]$. Then there is a point y where the derivative of q_2 satisfies

$$|q_2'(y)| \geq \frac{\frac{2}{3} - \frac{1}{3}}{\exp(-a) - \exp(-b)} = \frac{\exp(a)}{3(1 - \exp(a-b))}.$$

Since $exp(-a/2) - \frac{1}{d} \geq \exp(-a/2) - \exp(-a) \geq \frac{1}{2}\exp(-a/2)$ by assumption that $a \geq 5$, then Markov's inequality implies that

$$\frac{\exp(a)}{3(1 - \exp(a-b))} \leq \frac{2(\deg q_2)^2}{\exp(-a/2) - \frac{1}{d}} \leq 4\exp(a/2)(\deg q_2)^2.$$

Hence, in this case, $\deg q_2 \geq \Omega(\frac{\exp(a/4)}{\sqrt{1-\exp(a-b)}}) \geq \Omega(\exp(a/4))$.

Otherwise, there exists a point $x \in [\frac{1}{d}, \exp(-a/2)]$ where $q_2(x) = k \geq 2$. In this case, there is a point $\frac{1}{r_1} < y < \frac{1}{r_1+1}$ where the derivative satisfies:

$$|q_2'(y)| \geq \frac{k-1}{\frac{1}{r_1} - \frac{1}{r_1+1}} = \frac{k-1}{\frac{1}{r_1(r_1+1)}} \geq \frac{k}{2}r_1^2 \geq \frac{k}{2}\exp(a).$$

Hence in this case, Markov's inequality implies that

$$\frac{k}{2}\exp(a) \leq \frac{2k(\deg q_2)^2}{\exp(-a/2) - \frac{1}{d}} \leq 4k\exp(a/2)(\deg q_2)^2,$$

which implies that, $\deg q_2 \geq \Omega(\exp(a/4))$. Combining the two cases yields the claimed lower bound. □

Observe the previous bound applies to any local unitarily invariant measure of entanglement entropy with the property that if the reduced density ρ of $\left|\psi\right\rangle$ is maximally mixed on a r-dimensional subspace, then $H_2(\left|\psi\right\rangle) = \log r$. Hence, the query lower bound applies to the von Neumann entropy as well as

the Renyi α-entropy for any $\alpha \neq 1$, as these properties are satisfied by these measures of entanglement entropy as well.

Furthermore, the lower bound also extends to the QMA setting much like for the recurrence problem by using Aaronson's guessing lemma.

Theorem 1.2.11 (QMA lower bound for the Entanglement Entropy problem)**.** *Assume $a \geq 5$ and $a < b \leq \log d$ Suppose there is a T-query algorithm that solves the entanglement entropy problem with the help of an m-qubit witness, then $mT \geq \Omega(\exp(a/4))$.*

Proof. By the proof of Theorem 1.2.10 and the guessing lemma (Lemma 3.1.1), if there was a T-query algorithm using an m-qubit witness that solves the entanglement entropy problem, there exists a polynomial q of degree $O(mT)$ with the property that:

- $q(i) \leq 2^{-10m}$ for all $i = \frac{1}{n}$ and integers $n \leq \exp(a)$.
- $q(i) \geq 2^{-m}$ for all $i = \frac{1}{n}$ and integers $\exp(b) \leq n \leq d$.

In the first case, assume that the maximum of q in the range $[\exp(-a), \exp(-a/2)]$ satisfies $q(x) = k \geq 2^{-9m}$ so that $2^{-10m} \leq \frac{k}{2^{-m}} \leq \frac{k}{2}$. Then, since for that point satisfies $\frac{1}{r+1} \leq k \leq \frac{1}{r}$ for some $r \geq \exp(a/2)$, the derivative of q in that interval satisfies

$$|q'(y)| \geq \frac{k - 2^{-10m}}{\frac{1}{r+1} - \frac{1}{r}} \geq \frac{k}{2} r^2 \geq \frac{k}{2} \exp(a),$$

for some point y in that interval. Hence, by Markov's inequality the degree of q satisfies

$$\frac{k}{2} \exp(a) \leq \frac{2k(\deg q)^2}{\exp(-a/2) - \exp(-a)} \leq 4k \exp(a/2)(\deg q)^2,$$

since $\exp(-a/2) - \exp(-a) \geq \frac{1}{2}\exp(-a/2)$ by assumption. Therefore, $\deg q \geq \Omega(\exp(a/4))$ in this case.

Otherwise, we have that q is bounded by 2^{-9m} in the range $[\exp(-a), \exp(-a/2)]$. Let $y_0 = \exp(-a/2)$ and $y_1 = \exp(-a)$, by rescaling q, using

$$r(x) = 2^{9m} q\left(\frac{y_1 - y_0}{2}(x-1) + y_1\right),$$

r satisfies $|r(x)| \leq 1$ for all $|x| \leq 1$ from the low entropy case. If $y_2 = \exp(-b)$, when $x = 1 + \frac{2(y_2 - y_1)}{y_1 - y_0} = 1 + \frac{2(\exp(-a) - \exp(-b))}{\exp(-a/2) - \exp(-a)}$ we have reached the high entropy case, we have $r(x) \geq 2^{8m}$. Therefore, by Paturi's inequality with $\mu = \frac{2(\exp(-a) - \exp(-b))}{\exp(-a/2) - \exp(-a)} \leq 4\exp(-a/2)$, we obtain

$$2^{8m} \leq \exp(4(\deg r)\sqrt{4\exp(-a/2)}),$$

and hence $\deg q$ satisfies $\deg q \geq \deg r \geq \Omega(m \exp(a/4))$.

Hence recalling by the guessing lemma that $\deg q = O(mT)$ where m is the witness size and T is the query complexity, we have $mT \geq \Omega(\exp(a/4))$ as claimed. □

We now briefly sketch an upper bound in the setting where our property tester has access to proof states and our entanglement entropy measure is the Renyi 2-entropy and in the regime where $a \geq 5$, and $b \geq 2a$. Given two copies of the state $|\psi\rangle$ with reduced density matrix ρ, a swap test can be used to

produce a Bernoulli random variable X with mean μ equal to $\frac{1}{2} - \frac{1}{2}\operatorname{Tr}(\rho^2)$. Furthermore, by using the quantum amplitude estimation algorithm [BHMT02, Ham21, KO22], one can produce an estimation of the mean μ of X to additive error ϵ by an efficient quantum algorithm given $O(\frac{1}{\epsilon})$ samples from X. In particular, $O(\frac{1}{\beta-\alpha})$ samples can be used to distinguish between a Bernoulli random variable with mean at least β or at most α.

Applying these results to our setting, we can solve the entanglement entropy problem if we can distinguish between states $|\psi\rangle$ whose purity satisfies $\operatorname{Tr}(\rho^2) \geq \exp(-b)$ or $\operatorname{Tr}(\rho^2) \leq \exp(-a)$. Hence with $O(\exp(a))$ copies of the state $|\psi\rangle$, a series of swap tests produces $O(\exp(a))$ samples from a Bernoulli random variable with mean equal to $\frac{1}{2} - \frac{1}{2}\operatorname{Tr}(\rho^2)$. As the gap between the means in the yes and no cases satisfies $\beta - \alpha = \exp(-a) - \exp(-b) \geq \frac{1}{2}\exp(-a)$ by assumption, this sample complexity is sufficient to distinguish between the yes and no cases. Overall, this yields an algorithm using $O(\exp(a))$ queries and a proof state with $O(\exp(a))$ qubits.

3.3 The Entangled Subspace Problem and QMA versus QMA(2)

We now turn towards studying an LU-invariant unitary property testing problem that corresponds to a candidate oracle separation between QMA and QMA(2). Recall the definition of completely entangled subspaces and the Entangled Subspace problem from the introduction: an *ϵ-completely entangled subspace* $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$ is such that all states $|\theta\rangle \in S$ are ϵ-far in trace distance from any product state $|\psi\rangle \otimes |\phi\rangle$.

Definition 1.2.4 (Entangled Subspace problem)**.** Let $0 \leq a < b < 1$ be constants. The (a, b)-Entangled Subspace problem is to decide, given oracle access to a unitary $U = I - 2\Pi$ where Π is the projector onto a subspace $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$, whether

- (*yes* case) S contains a state $|\theta\rangle$ that is a-close in trace distance to a product state $|\psi\rangle \otimes |\phi\rangle$.
- (*no* case) S is b-completely entangled

promised that one is the case.

As mentioned earlier, the Entangled Subspace property is LU-invariant: applying local unitaries $g \otimes h$ to a subspace S preserves whether it is a *yes* instance or a *no* instance of the problem.

3.3.1 QMA(2) Upper Bound

We first give a QMA(2) upper bound to the Entangled Subspace problem:

Proposition 1.2.12 (QMA(2) upper bound for the Entangled Subspace problem)**.** *The Entangled Subspace problem can be solved by a* QMA(2) *tester, meaning that the tester receives a proof state in the form $|\psi\rangle \otimes |\varphi\rangle$ of $\operatorname{poly}\log(d)$ qubits, makes one query to the unitary U, and can distinguish between* yes *and* no *cases with constant bias.*

Proof. Consider the verifier illustrated in Figure 3.1 where the provided proof state is the product state $|\psi\rangle \otimes |\phi\rangle$. The controlled-$U$ operation is essentially performing a subspace membership test. The state after the controlled-U operation and the Hadamard on the ancilla qubit can be written as

$$\frac{I+U}{2}\Big(|\psi\rangle \otimes |\phi\rangle\Big) \otimes |0\rangle + \frac{I-U}{2}\Big(|\psi\rangle \otimes |\phi\rangle\Big) \otimes |1\rangle .$$

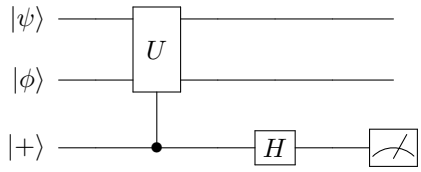

Figure 3.1: The verifier V' in the proof of Lemma 3.3.6.

Since $(I-U)/2 = \Pi$, the acceptance probability of the subspace membership test is

$$\left\|\Pi\Big(\,|\psi\rangle\otimes|\phi\rangle\,\Big)\right\|^2 = |\,\langle\xi|\psi\otimes\phi\rangle\,|^2 = 1 - d(|\xi\rangle\,, |\psi\rangle\otimes|\phi\rangle)^2$$

where $|\xi\rangle$ is the projection of $|\psi\rangle\otimes|\phi\rangle$ on S and d is the trace distance.

In the yes case, there exists a product state that is a-close to a state in S, and hence providing that state as a certificate makes the verifier accept with probability at least $1-a^2$. Otherwise in the no case, all states $|\xi\rangle \in S$ are b-far from product, and hence the verifier accepts with probability no more than $1-b^2$. □

We note that the verifier analyzed in the proof of Proposition 1.2.12 has the property that in the yes case, proof state may not be a symmetric product state (i.e. a state of the form $|\psi\rangle^{\otimes 2}$). We now present a QMA(2) verifier, which we call the *product test verifier*, for the Entangled Subspace Problem, with the additional property that in the yes case there exists a valid proof state that is symmetric. The verifier relies on a procedure known as the *product test*, which was analyzed by Harrow and Montanaro [HM13] and also later in [SW22]. We state the main results about the product test here, specialized to the case of bipartite states.

Definition 3.3.1 (Product test)**.** Let $|\psi\rangle$ be a state in $\mathbb{C}^d\otimes\mathbb{C}^d$. Consider two copies of the $|\psi\rangle^{\otimes 2}$, where the first copy is on registers A_1B_1 and the second copy is on registers A_2B_2. The product test applies the swap test on registers A_1A_2, and another swap test on B_1B_2. The product test accepts iff both swap tests accept.

Observe that if $|\psi\rangle = |\varphi\rangle\otimes|\xi\rangle$, then the product test accepts with probability 1. On the other hand, we have the following bound for the probability an entangled $|\psi\rangle$ will pass the product test.

Theorem 3.3.1 ([SW22, Theorem 8])**.** *Given a state $|\psi\rangle \in \mathbb{C}^d\otimes\mathbb{C}^d$, let*

$$\omega_{|\psi\rangle} = \max\{|\,\langle\psi|\phi_1\otimes\phi_2\rangle\,|^2, |\phi_1\rangle\in\mathbb{C}^d, |\phi_2\rangle\in\mathbb{C}^d\}$$

denote the overlap of $|\psi\rangle$ with the closest product state. Then the probability α that the product test passes satisfies

$$\frac{1}{2}(1+\omega^2_{|\psi\rangle}) \le \alpha \le \frac{1}{3}\omega^2_{|\psi\rangle} + \frac{2}{3}\ .$$

While Theorem 3.3.1 assumes that the input to the product test is symmetric, we note that the product test is also sound against non-symmetric witnesses.

Proposition 3.3.2 ([HM13, Appendix E])**.** *Let $P(|\Phi\rangle)$ be the probability that the product test passes when given a state $|\Phi\rangle$ as input. Then, for any $|\psi_1\rangle\in\mathbb{C}^d\otimes\mathbb{C}^d$ and $|\psi_2\rangle\in\mathbb{C}^d\otimes\mathbb{C}^d$, we have*

$$P(|\psi_1\rangle \otimes |\psi_2\rangle) \leq \frac{P(|\psi_1\rangle^{\otimes 2}) + P(|\psi_2\rangle^{\otimes 2})}{2}.$$

By combining Theorem 3.3.1 and Proposition 3.3.2 we obtain the following result.

Proposition 3.3.3. *Suppose $0 \leq a < b \leq 1$ are constants satisfying satisfy*

$$\frac{1}{2}(1 + (1-a^2)^2) > \frac{1}{3}(1-b^2)^2 + \frac{2}{3}.$$

Then there exists a QMA(2) *verifier for the (a,b)-Entangled Subspace problem with the property that on yes instances with oracle $O = I - 2P_S$ where P_S is a projector onto subspace S, a valid proof state is $|\psi\rangle^{\otimes 2}$ where $|\psi\rangle$ is any state in S that is a-close to product.*

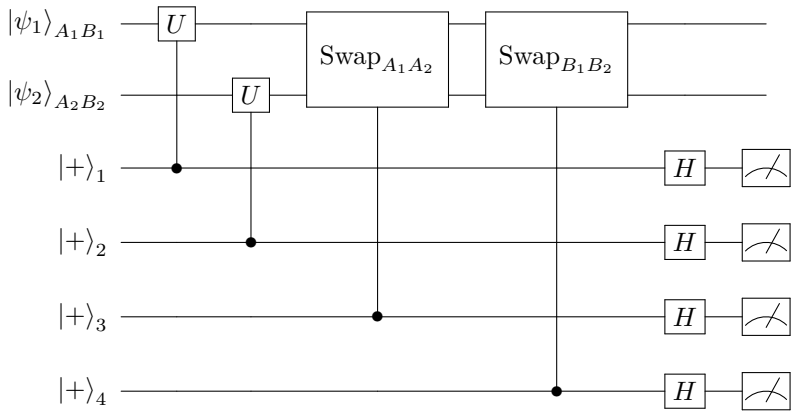

Figure 3.2: QMA(2) product test verifier

Proof. Suppose $|\psi_1\rangle \otimes |\psi_2\rangle$ is given as input to the verifier. From the proof of Proposition 1.2.12, the probability that the first two ancillas accept is $\|P_S|\psi_1\rangle\|\|P_S|\psi_2\rangle\|$. Conditioned on the first two ancillas accepting, the probability the third and fourth ancillas accept is the probability that the product test passes when provided the state $\frac{P_S|\psi_1\rangle}{\|P_S|\psi_1\rangle\|} \otimes \frac{P_S|\psi_2\rangle}{\|P_S|\psi_2\rangle\|}$ as input. Hence, the verifier maximizes its success probability when $|\psi_1\rangle \in S$ and $|\psi_2\rangle \in S$. Furthermore, by Proposition 3.3.2, we can assume that $|\psi_1\rangle = |\psi_2\rangle$ to maximize the verifier's success probability.

By Theorem 3.3.1, in the yes case, there exists a state in $|\psi\rangle \in S$ that is a-close to product, and hence the product test passes with probability at least $\frac{1}{2}(1 + (1-a^2)^2)$ when given $|\psi\rangle^{\otimes 2}$ as input. Otherwise, in the no case, all states in S are b-far from product, and hence the product test passes with probability at most $\frac{1}{3}(1-b^2)^2 + \frac{2}{3}$ in this case on any input $|\psi\rangle^{\otimes 2}$ for $|\psi\rangle \in S$. Therefore, as long as $\frac{1}{2}(1 + (1-a^2)^2) > \frac{1}{3}(1-b^2)^2 + \frac{2}{3}$, there is a bounded gap in the success probability between the two cases. □

To extend the result to an arbitrary gap between a and b, we note that the product test can be further generalized to the situation when input consists of $k \geq 2$ copies of a given state $|\psi\rangle$.

Definition 3.3.2 (k-copy product test)**.** Let $|\psi\rangle$ be a state in $\mathbb{C}^d \otimes \mathbb{C}^d$. Consider $k \geq 2$ copies of the $|\psi\rangle^{\otimes 2}$ where copy i is on registers A_iB_i. The product test is a circuit that performs a projective

measurement $\{P = \Pi_A \otimes \Pi_B, I - P\}$ where Π_A is the projector on the symmetric subspace on registers $A_1, \ldots, A_k$ and Π_B is the projector on the symmetric subspace on registers $B_1, \ldots, B_k$. The success probability of the product test is $\|P |\psi\rangle^{\otimes k}\|^2$.

By the results of [HM13] and [BBD+97], there is an efficient quantum circuit which implements the product test for all constant k. Similarly, we can bound the success probability of the k-copy product test in terms of the overlap with the closest product state, using the proof techniques presented in [SW22].

Theorem 3.3.4. *Let $\omega_{|\psi\rangle}$ be the overlap of $|\psi\rangle$ with the closest product state, as defined in Theorem 3.3.1. For all constant $k \geq 2$, the probability α that the product test passes when given $|\psi\rangle^{\otimes k}$ as input satisfies*

$$\alpha \leq \frac{k-1}{k+1}\omega_{|\psi\rangle}^k + \frac{2}{k+1}.$$

We defer the proof of Theorem 3.3.4 to Appendix A.1. We apply Theorem 3.3.4 to obtain the following result, whose proof is deferred to Appendix A.2.

Theorem 3.3.5. *Let $0 \leq a < b < 1$ be constants. Then there exists a constant $k \geq 2$ sufficiently large such that there is a $\mathsf{SymQMA}(\mathsf{k}+1)$ verifier for the (a, b)-Entangled Subspace problem.*

We recall that $\mathsf{SymQMA}(\mathsf{k})$ is the variant of $\mathsf{QMA}(\mathsf{k})$ where the witness is promised to be a symmetric product state $|\psi\rangle^{\otimes k}$. Since for any constant $k \geq 2$, $\mathsf{SymQMA}(\mathsf{k}) = \mathsf{QMA}(\mathsf{k}) = \mathsf{QMA}(2)$ result of [ABD+08, Lemma 38], this construction provides another proof that the Entangled Subspace Problem is in $\mathsf{QMA}(2)$.

3.3.2 QMA versus QMA(2) for State Property Testing

Next, we turn to constraints on using the Entangled Subspace problem to obtain an oracle separation between QMA and $\mathsf{QMA}(2)$. One might hope that, given the characterization of LU-invariant polynomials for unitaries that encode a one-dimensional subspace (Lemma 3.2.8), we may be able to obtain a QMA versus $\mathsf{QMA}(2)$ separation by proving a lower bound on the Entangled Subspace problem by focusing on one-dimensional subspaces only.

However we show that generally property testing questions concerning states (equivalently, one-dimensional subspaces) are not sufficient to resolve the QMA versus $\mathsf{QMA}(2)$ problem. Therefore proving Conjecture 1 necessarily requires studying problems about the entanglement of higher dimensional subspaces.

Lemma 3.3.6. *Let $\mathcal{P}$ denote a property where the instances are unitaries encoding a one-dimensional subspace (i.e. a pure state): $U = I - 2|\psi\rangle\langle\psi|$ for some state $|\psi\rangle$. Suppose that there is a T-query $\mathsf{QMA}(2)$ tester that decides $\mathcal{P}$, with the condition that a valid proof state for* yes *instances is $|\psi\rangle^{\otimes 2}$. Then there exists a $O(T)$-query QMA tester that also decides $\mathcal{P}$.*

Proof. Let V denote the $\mathsf{QMA}(2)$ verifier that decides $\mathcal{P}$. We construct a QMA verifier V' (depicted in Figure 3.3) that can receive an entangled proof state $|\Phi\rangle$. Label the registers of $|\Phi\rangle$ by $A_1B_1A_2B_2$. The verifier V' performs the subspace membership test on registers A_1B_1 and A_2B_2 separately by calling U controlled on two ancilla qubits initialized in the $|+\rangle$ state.

The verifier V' then applies Hadamard gates to the ancilla and measures. It takes the post-measurement state of registers $A_1B_1A_2B_2$ and runs the original verifier V on them. The new verifier V' accepts if and only if the two ancilla bits accepted and the original verifier V accepted.

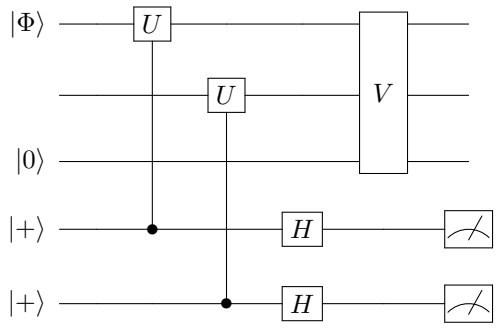

Figure 3.3: The verifier V' in the proof of Lemma 3.3.6.

If U encodes a pure state $|\psi\rangle$ and is a *yes* instance, then by assumption on the original verifier V, we can run V' with proof state $|\Phi\rangle = |\psi\rangle^{\otimes 2}$ and it will accept with the same probability as V.

On the other hand, assume that U is a *no* instance, and let $|\Phi\rangle$ be the (possibly entangled) proof state provided to verifier V'. Decompose $|\Phi\rangle = c_0 |\psi\rangle^{\otimes 2} + c_1 |\xi\rangle$ for some state $|\xi\rangle$ orthogonal to $|\psi\rangle^{\otimes 2}$. Since V' accepts $|\Phi\rangle$ only when the subspace membership tests accept, then V' accepts $|\Phi\rangle$ with probability $|c_0|^2 s$, where s is the probability that V accepts $|\psi\rangle^{\otimes 2}$. Since $|c_0|^2 \leq 1$, then the acceptance probability of V at most s.

Therefore, V' has the same soundness and completeness as V, even allowing for entangled states as input.

□

Hence, combining Lemma 3.3.6 and Proposition 3.3.3, we obtain that there exists a parameter range (a, b) for which there exists a QMA verifier for solving the one-dimensional (a, b)-Entangled Subspace problem. Hence, this theorem is saying that property testing questions related to quantum *states* are insufficient to give an oracle separation between QMA and QMA(2). On the other hand, we obtain a different result in the setting where the hidden subspace is higher-dimensional.

3.3.3 The QMA (Un)soundness of the Product Test Verifier

We now show that when there is no unentanglement guarantee for the proof state, the product test verifier fails to be sound. What this means is that there is a *no* instance U of the Entangled Subspace problem (with different parameters) but also a proof state $|\theta\rangle$ that may be completely entangled across the four registers $A_1B_1A_2B_2$, such that the product test verifier will accept with probability 1 when making queries to U. In other words, the product test verifier can be *fooled* by an entangled proof in the QMA setting.

Proposition 3.3.7. *Let $d \geq 4$. Let $u, v, w, x \in [d]$ be distinct. Let $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$ denote the six-dimensional subspace spanned by $\frac{|uv\rangle+|vu\rangle}{\sqrt{2}}, \frac{|uw\rangle+|wu\rangle}{\sqrt{2}}, \frac{|ux\rangle+|xu\rangle}{\sqrt{2}}, \frac{|vw\rangle+|wv\rangle}{\sqrt{2}}, \frac{|vx\rangle+|xv\rangle}{\sqrt{2}}, \frac{|wx\rangle+|xw\rangle}{\sqrt{2}}$. Let Π be the projector onto S and let $|\psi_{uvwx}\rangle$ be the state*

$$
\begin{aligned}
|\psi_{uvwx}\rangle = \frac{1}{\sqrt{24}}\Big[&(|uv\rangle + |vu\rangle) \otimes (|wx\rangle + |xw\rangle) + (|wx\rangle + |xw\rangle) \otimes (|uv\rangle + |vu\rangle) \\
&+ (|uw\rangle + |wu\rangle) \otimes (|vx\rangle + |xv\rangle) + (|vx\rangle + |xv\rangle) \otimes (|uw\rangle + |wu\rangle) \\
&+ (|ux\rangle + |xu\rangle) \otimes (|vw\rangle + |wv\rangle) + (|vw\rangle + |wv\rangle) \otimes (|ux\rangle + |xu\rangle)\Big] .
\end{aligned}
$$

Then:

1. *S is a $1/4$-completely entangled subspace, with every state $|\varphi\rangle \in S$ having overlap at most $\frac{3}{4}$ with a product state.*

2. *The product test verifier making queries to $U = I - 2\Pi$, accepts the entangled proof state $|\psi_{uvwx}\rangle$ with probability 1 for any $|\phi\rangle \in S$.*

Proof. Assume without loss of generality that $d = 4$. Otherwise apply an isometry $W \otimes W$ to S where $W : \mathrm{Span}(|u\rangle, |v\rangle, |w\rangle, |x\rangle) \to \mathbb{C}^d$ is an isometry, which does not change the magnitude of the closest product state by construction.

Let $|\varphi\rangle \in S$. Observe that S is contained in the symmetric subspace of $\mathbb{C}^d \otimes \mathbb{C}^d$, so by [HMM+08, Lemma 1], the closest product state to $|\varphi\rangle$ can be chosen to be a symmetric state $|\phi\rangle^{\otimes 2}$. Write $|\phi\rangle = \sum_{i=1}^4 \beta_i |i\rangle$. Write $|\psi_{ij}\rangle = \frac{|ij\rangle + |ji\rangle}{\sqrt{2}}$ and $|\psi\rangle = \sum_{1 \le i < j \le 4} \alpha_{ij} |\psi_{ij}\rangle$. Then by the Cauchy-Schwartz inequality and normalization:

$$|\langle \phi^{\otimes 2} | \psi \rangle|^2 = \left| \sum_{1 \le i < j \le 4} \alpha_{ij} \frac{\beta_i \beta_j + \beta_j \beta_i}{\sqrt{2}} \right|^2 \le 2 \sum_{1 \le i < j \le 4} |\alpha_{ij}|^2 \sum_{1 \le i < j \le 4} |\beta_i|^2 |\beta_j|^2 = 2 \sum_{1 \le i < j \le 4} |\beta_i|^2 |\beta_j|^2$$

Since $2\sum_{i<j} |\beta_i|^2 |\beta_j|^2 + \sum_{i=1}^4 |\beta_i|^4 = 1$ and $\sum_{i=1}^4 |\beta_i|^2 = 1$ since $|\phi\rangle$ and $|\phi\rangle^{\otimes 2}$ are normalized, we can conclude that

$$\sum_{i=1}^4 |\beta_i|^4 = \sum_{i=1}^4 |\beta_i|^2 |\beta_i|^2 \ge \frac{1}{4} \sum_{i=1}^4 |\beta_i|^2 = \frac{1}{4}.$$

Therefore, $2\sum_{i<j} |\beta_i|^2 |\beta_j|^2 \le \frac{3}{4}$ is an upper bound for the overlap with a product state for any state $|\varphi\rangle \in S$. This establishes the first item of the proposition statement.

For the second item, let $|\psi_{uvwx}\rangle$ be the proof state given to the product test verifier in Figure A.1. By construction, the membership queries pass with probability one. Furthermore, observe that $|\psi_{uvwx}\rangle$ is symmetric under all permutations of the registers, so $|\psi_{uvwx}\rangle$ passes the 2-copy product test with probability 1. Hence the verifier accepts the state $|\psi_{uvwx}\rangle$ and oracle U with probability one. Hence the verifier is not sound against entangled proofs since $|\psi_{uvwx}\rangle$ was entangled, and S is a $\frac{1}{4}$-completely entangled subspace. □

3.3.4 Average Case Versions of the Entangled Subspace Problem

In this section, we discuss the average case variants of the Entangled Subspace problem, restated from the introduction.

Definition 1.2.5 (Planted Product State Problem)**.** Let $0 < s < d^2$ denote an integer parameter. Consider the following two distributions over subspaces S of $\mathbb{C}^d \otimes \mathbb{C}^d$:

- **No planted state**: S is a Haar-random subspace of dimension s.
- **Has planted state**: S is an $(s+1)$-dimensional subspace chosen by taking the span of a Haar-random s-dimensional subspace with a product state $|\psi\rangle \otimes |\phi\rangle$ for Haar-random $|\psi\rangle, |\phi\rangle$.

The Planted Product State problem is to distinguish, given oracle access to a unitary $U = I - 2\Pi$ encoding a subspace S, whether S was sampled from the **No planted state** distribution (*no* case) or the **Has planted state** distribution (*yes* case), promised that one is the case.

Definition 1.2.6 (Restricted Dimension Counting Problem)**.** Let $0 < t \leq d$ and $0 < r \leq t^2$ denote integer parameters. Consider the following distribution, parameterized by (t, r), over subspaces $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$:

- Sample Haar-random t-dimensional subspaces $R, Q \subseteq \mathbb{C}^d$.
- Sample a Haar-random r-dimensional subspace of $S \subseteq R \otimes Q$.

Let $0 < C_1 < C_2 < 1$ denote constants. The Restricted Dimension Counting problem is to decide, given query access to a unitary $U = I - 2\Pi$ encoding a subspace S, whether S was sampled from either the $(t, C_1 t^2)$ distribution or $(t, C_2 t^2)$ distribution, promised that one is the case.

Combined with the following result about Haar-random subspaces, we obtain our two property testing problems are in fact average case versions of the Entangled Subspace problem as claimed. Informally, the lemma asserts that if $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$ was a Haar-random subspace of small dimension s compared to d^2, then every state in S is entangled with high probability. Otherwise, for sufficiently large s, there exists a state in S that is close to a product state. The proof is based on the techniques of [HLW06] that use Lévy lemma for the Haar measure and some additional observations about the closest product state.

Lemma 3.3.8 (Levy's Lemma [MS86, Led01])**.** *Let $f : \mathbb{S}^k \to \mathbb{R}$ be a function with Lipschitz constant η (with respect to the Euclidean norm) and let $|\psi\rangle \in \mathbb{S}^k$ be chosen uniformly at random from the Haar measure. Then*

$$\Pr\left(\left|f(|\psi\rangle) - \mathbb{E}\, f\right| > \alpha\right) \leq 2\exp\left(-C(k+1)\alpha^2/\eta^2\right)$$

where $C = (9\pi^3 \ln 2)^{-1}$ and $\mathbb{E}\, f$ denotes the average of f over $\mathbb{S}^k$.

Lemma 3.3.9. *Let ω be the overlap of $|\psi\rangle$ with the closest product state as defined in Theorem 3.3.1. If $\rho = \mathrm{Tr}_1(|\psi\rangle\langle\psi|)$ was the reduced density matrix of $|\psi\rangle$, then*

$$\omega^2 \leq \mathrm{Tr}(\rho^2) \leq \omega.$$

Proof. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ be the eigenvalues of ρ. Then [HM13, Lemma 2] shows that $\omega = \lambda_1$. Therefore, since $\sum_{i=1}^d \lambda_i = 1$ and each λ_i is non-negative, we have

$$\lambda_1^2 \leq \mathrm{Tr}(\rho^2) = \sum_{i=1}^d \lambda_i^2 \leq \sum_{i=1}^d \lambda_1 \lambda_i = \lambda_1.$$

□

Theorem 3.3.10. *Let $S \subseteq \mathbb{C}^d \otimes \mathbb{C}^d$ be a Haar-random subspace of dimension s, and let $|\psi\rangle \in S$ and let C be the constant from Levy's Lemma. Then*

1. *For all constant $\delta > 0$, if $d \geq \frac{4}{\delta}$ and $s \leq \frac{C\delta^2}{1024 \log(\frac{40}{\delta})} d^2$, then all states in S have purity at most δ with high probability. In particular,*

$$\Pr_{|\psi\rangle \in S}[\omega_{|\psi\rangle} \geq \sqrt{\delta}] \leq \exp(-O(d^2)).$$

2. *For all constant $\delta > 0$, if $s \geq 2\sqrt{\delta} d^2$, then S contains a state with overlap with the closest product state δ with high probability. In particular,*

$$\Pr_{|\psi\rangle \in S}[\sup_{|\psi\rangle \in S} \omega_{|\psi\rangle} \leq \delta] \leq \exp(-O(d^2)).$$

Proof. To show the first part, let $f(|\psi\rangle) = \mathrm{Tr}(\rho^2)$ denote the purity of the reduced density matrix of $|\psi\rangle$ on the first $\mathbb{C}^d$ factor. The Lipschitz constant of f is 4 by [HLW06, Lemma III.8].

Let P be a fixed projector onto the first s basis states (according to some canonical ordering) of $\mathbb{C}^d \otimes \mathbb{C}^d$, and U is a Haar-random unitary on $\mathbb{C}^d \otimes \mathbb{C}^d$. Let $\mathcal{N}$ denote an ϵ-net for the image of the projector P of size $(5/\epsilon)^{2s}$, which exists by [HLW06, Lemma III.6]. Note that $U\mathcal{N}$ is an ϵ-net for the image of the projector $P_S = UPU^*$, which is a uniformly random subspace of dimension s when U is choosen to be Haar-random.

We want to bound the probability that there exists a state $|\psi\rangle \in S$ whose purity is large.

$$\begin{aligned}
\Pr\Big(\exists\, |\psi\rangle \in S \text{ such that } f(|\psi\rangle) \geq \delta\Big) &\leq \Pr\Big(\exists\, |\varphi\rangle \in U\mathcal{N} \text{ such that } f(|\varphi\rangle) \geq \delta - 4\epsilon\Big) \\
&\leq \sum_{|\phi\rangle \in \mathcal{N}} \Pr\Big(f(U|\phi\rangle) \geq \delta - 4\epsilon\Big) \\
&= \Big(\frac{5}{\epsilon}\Big)^{2s} \cdot \Pr_{|\phi\rangle \sim \mathrm{Haar}(d^2)}\Big(f(|\phi\rangle) \geq \delta - 4\epsilon\Big).
\end{aligned}$$

On average, the purity of a Haar-random state in $\mathbb{C}^d \otimes \mathbb{C}^d$ is $\beta := 2d/(d^2+1)$ by [CN16, Proposition 4.14].

Thus by Levy's Lemma we have that

$$\Pr_{|\phi\rangle \sim \mathrm{Haar}(d^2)}\Big(f(|\phi\rangle) \geq \delta - 4\epsilon\Big) \leq 2\exp\Big(-\frac{C}{16}(d^2+1)(\delta - 4\epsilon - \beta)^2\Big)$$

where C was the constant from Lemma 3.3.8. Now choosing $\epsilon = \frac{\delta}{8}$ and d sufficiently large so that $\frac{1}{d} \leq \frac{2d}{d^2+1} \leq \frac{\delta}{4}$, we have

$$\begin{aligned}
\Pr\Big(\exists\, |\psi\rangle \in S \text{ such that } f(|\psi\rangle) \geq \delta\Big) &\leq 2\Big(\frac{40}{\delta}\Big)^{2s} \exp\Big(-\frac{C}{16}(d^2+1)(\frac{\delta}{4})^2\Big) \\
&= 2\exp\Big(-\frac{C}{256}(d^2+1)\delta^2 + 2s\log(\frac{40}{\delta})\Big)
\end{aligned}$$

by combining the above bounds. Hence the claimed choice of s makes this probability exponentially small in d^2. Furthermore, by Lemma 3.3.9, we have $\Pr[\mathrm{Tr}(\rho^2) \geq \delta] \geq \Pr[\omega_{|\psi\rangle} \geq \sqrt{\delta}]$, and hence the probability that the overlap with a product state is at least $\sqrt{\delta}$ is exponentially small.

To show the second part, fix a product state $|v\rangle = |a\rangle \otimes |b\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$. The overlap between $|v\rangle$ and

the subspace S is captured by the quantity

$$f(U) = \langle v| P_S |v\rangle = \langle v| UPU^* |v\rangle \ .$$

Equivalently, the overlap is

$$f(|\psi\rangle) = \langle \psi| P |\psi\rangle$$

where $|\psi\rangle$ is a Haar-random state since $|v\rangle$ was a fixed vector. The average of $f(|\psi\rangle)$ can be computed as

$$\int f(|\psi\rangle \ \mathrm{d}\,|\psi\rangle = \mathrm{Tr}\left(P \int |\psi\rangle\langle\psi| \ \mathrm{d}\,|\psi\rangle\right) = \frac{1}{d^2}\mathrm{Tr}(P) = \frac{s}{d^2} \ .$$

We compute the Lipschitz constant of f since P is a projector:

$$\begin{aligned}
\sup_{|\psi\rangle,|\varphi\rangle} \frac{|f(|\psi\rangle) - f(|\varphi\rangle)|}{\| \, |\psi\rangle - |\varphi\rangle \, \|} &= \sup_{|\psi\rangle,|\varphi\rangle} \frac{\left| \|P|\psi\rangle\|^2 - \|P|\varphi\rangle\|^2 \right|}{\| \, |\psi\rangle - |\varphi\rangle \, \|} \\
&= \sup_{|\psi\rangle,|\varphi\rangle} \frac{\left| \|P|\psi\rangle\| + \|P|\varphi\rangle\| \right| \cdot \left| \|P|\psi\rangle\| - \|P|\varphi\rangle\| \right|}{\| \, |\psi\rangle - |\varphi\rangle \, \|} \\
&\leq \sup_{|\psi\rangle,|\varphi\rangle} \frac{2\left| \|P|\psi\rangle\| - \|P|\varphi\rangle\| \right|}{\| \, |\psi\rangle - |\varphi\rangle \, \|} \\
&\leq \sup_{|\psi\rangle,|\varphi\rangle} \frac{2\|P(|\psi\rangle - |\varphi\rangle)\|}{\| \, |\psi\rangle - |\varphi\rangle \, \|} \\
&\leq 2 \ .
\end{aligned}$$

We now apply Levy's Lemma to conclude that

$$\Pr\left(f(|\psi\rangle) < \frac{s}{d^2} - \delta\right) \leq 2\exp\left(-\frac{C}{4}(d^2+1)\delta^2\right)$$

Hence as long as $s \geq 2\sqrt{\delta}d^2$, we have for $|w\rangle = P_S |v\rangle$,

$$\begin{aligned}
\Pr[|\langle v|w\rangle|^2 \geq \delta] = \Pr[|\langle v|w\rangle| \geq \sqrt{\delta}] &= 1 - \Pr[|\langle v|w\rangle| \leq \sqrt{\delta}] \\
&\geq 1 - \Pr[|\langle v|w\rangle| \leq \frac{s}{d^2} - \sqrt{\delta}] \\
&\geq 1 - 2\exp\left(-\frac{C\delta}{4}(d^2+1)\right)
\end{aligned}$$

and therefore S contains a state with overlap at least δ with probability 1 - $\exp(-O(d^2))$. □

We note that the above bounds in Theorem 3.3.10 are likely not tight, and finding tight bounds would be an interesting open problem. However, we are also now able to show Proposition 1.2.13 and Proposition 1.2.14 using this result.

Proposition 1.2.13. *If S is sampled from the* ***Has planted state*** *distribution of the Planted Product State problem, then it is a* yes *instance of the Entangled Subspace problem. If S is sampled from the* ***No planted state*** *distribution with $s = Cd^2$ for some sufficiently small constant $C > 0$, then it is a* no *instance with overwhelming probability.*

Proof. Clearly in the yes case, the subspace S contains a product state. Otherwise, given $\epsilon > 0$, choosing $\delta = (1-\epsilon^2)^2$ in Theorem 3.3.10 (1) implies that there is some constant C such that a Haar-random subspace of dimension $s \leq Cd^2$ is ϵ-completely entangled. Hence setting choosing any s in this range gives a *no* instance with probability at least $1 - \exp(-O(d^2))$. □

Proposition 1.2.14. *There exist constants $0 < C_1 < C_2 < 1$ such that if S is sampled from the $(t, C_1 t^2)$ distribution from the Restricted Dimension Counting problem, it is a* no *instance of the Entangled Subspace problem with overwhelming probability. If it is sampled from the $(t, C_2 t^2)$ distribution, then it is a* yes *instance with overwhelming probability.*

Proof. Let $a < b$ be the two parameters in the Entangled Subspace problem where *yes* instances have a state that is a-close to product and otherwise *no* instances are b-completely entangled. Choose C_1 using $\delta = (1-b^2)^2$ from Theorem 3.3.10 (1), and C_2 using $\delta = 1 - a^2$ from Theorem 3.3.10 (2). Then with probability at least $1 - \exp(-O(d^2))$, a Haar-random subspace of dimension $\leq C_1 t^2$ is b-completely entangled, and a subspace of dimension $\geq C_2 t^2$ contains a state that is a-close to a product state. □

Hence, having observed that our average-case problems can be reduced to the Entangled Subspace problem with overwhelming probability, we use our results from Section 3.3.1 to show that they can be solved by a QMA(2) tester with high probability. Furthermore, we conjecture that a lower bound for a QMA tester for the Entangled Subspace problem extends to this average case setting.

3.3.5 Connections to Invariant Theory

Observe that all of our candidate problems, being special cases of the Entangled Subspace problem, have local unitary symmetries. This follows from product states being preserved under local unitary transformations, and the unitary invariance of the trace distance.

This opens up the possibly of using the generalized polynomial method to prove a QMA lower bound for our candidate problems. While these problems are similar in spirit to the entanglement entropy problem that also has a local unitary symmetry introduced in Section 3.2.3, the main barrier to applying the polynomial method in this case is that we do not appear to have a good characterization of the invariant polynomials in Theorem 3.2.6 in the case where P is a projector onto a high-dimensional subspace. While Theorem 3.2.7 characterizes these polynomials in the case where P is a one-dimensional projector, we have seen in the previous section that one-dimensional properties cannot be used to separate QMA and QMA(2). We are not aware of a good characterization of these invariants even in the case where P is a projector onto a two-dimensional subspace. A deeper understanding of these invariants appears necessary to make further progress on these questions.

3.3.6 QCMA Lower Bound for the Entangled Subspace Problem

As described in the previous section, we are not currently able to prove a strong QMA lower bound on the query complexity of the entangled subspace problem. However, using a similar proof strategy as Aaronson and Kuperberg in [AK07], we show a lower bound against QCMA, which is the subclass of QMA of problems verifiable by a polynomial time quantum verifier with a classical proof string.

To present this lower bound, we first recall the definition of a p-uniform measure over quantum states from [AK07].

Definition 3.3.3. Let μ be the Haar measure over n-dimensional sphere $\mathbb{S}^n$. A measure σ is p-uniform if it can be obtained from μ by conditioning on an event A with measure $\mu(A) \geq p$.

Using Lévy's lemma, we can observe the following property of p-uniform measures.

Lemma 3.3.11. *Let $f(|\psi\rangle) : \mathbb{S}^d \to \mathbb{R}$ be a non-negative, Lipschitz function on the sphere bounded by 1. Let $\mathbb{E}_\mu[f]$ be its expectation over the Haar measure. Then if σ is a p-uniform measure, then*

$$\mathbb{E}_\sigma[f] \leq \mathbb{E}_\mu[f] + O\left(\sqrt{\frac{\log \frac{1}{p} + \log d}{d}}\right).$$

Proof. Let $\bar{f} = \mathbb{E}_\mu[f]$, $X = f(|\psi\rangle)$, and $a > 0$. By the definition of p-uniform measure, we have

$$\Pr_\sigma[|X - \bar{f}| \geq a] \leq \frac{1}{p} \Pr_\mu[|X - \bar{f}| \geq a],$$

and by Lévy's lemma there exists a constant C such that,

$$\Pr_\mu[|X - \bar{f}| \geq a] \leq 2\exp(-Cda^2).$$

Hence, for every $a > 0$, we get

$$\mathbb{E}_\sigma[f] \leq (\bar{f} + a) \Pr_\sigma[|X - \bar{f}| \leq a] + \Pr_\sigma[|X - \bar{f}| \geq a] \leq \bar{f} + a + \frac{2}{p}\exp(-Cda^2).$$

To minimize the expectation, we choose $a = \sqrt{\frac{\log \frac{2}{p} + \log d}{Cd}} = O\left(\sqrt{\frac{\log \frac{1}{p} + \log d}{d}}\right)$. This choice of a ensures that $\frac{2}{p}\exp(-Cda^2) = \frac{2}{d} \leq \sqrt{\frac{\log d}{d}} \leq a$ for sufficiently large d. Hence,

$$\mathbb{E}_\sigma[f] \leq \bar{f} + O(a) = \mathbb{E}_\mu[f] + O\left(\sqrt{\frac{\log \frac{1}{p} + \log d}{d}}\right).$$

□

We also require the following observation about the Haar measure.

Lemma 3.3.12. *Let $|\theta\rangle$ be a Haar random state in $\mathbb{C}^d$. For every density matrix ρ on $\mathbb{C}^d \otimes \mathbb{C}^d$, we have*

$$\mathbb{E}_{|\theta\rangle}[\mathrm{Tr}(|\theta\rangle\langle\theta|^{\otimes 2}\rho)] \leq \frac{2}{d(d+1)}.$$

.

Proof. Since $\mathbb{E}[|\theta\rangle\langle\theta|^{\otimes 2}] = \frac{2\Pi}{d(d+1)}$ where Π is the projector onto the symmetric subspace of $\mathbb{C}^d \otimes \mathbb{C}^d$, then for any density matrix ρ :

$$\mathbb{E}_{|\theta\rangle}[\mathrm{Tr}(|\theta\rangle\langle\theta|^{\otimes 2}\rho)] = \frac{2}{d(d+1)}\mathrm{Tr}(\Pi\rho) \leq \frac{2}{d(d+1)}.$$

□

We are now ready to prove the lower bound. In fact, we will prove the lower bound on the average case version of the Entangled Subspace problem, which is the Planted Product State problem, introduced in the previous sections. In this section, we modify the definition of the problem to ensure that the planted product state is always a symmetric state $|\theta\rangle^{\otimes 2}$. However, the modified problem is clearly also a special case of the Entangled Subspace problem.

Theorem 3.3.13. *Any quantum algorithm solving the Planted Product State problem using T queries and an m-bit classical witness must use*

$$T \geq \Omega\left(\sqrt[4]{\frac{d}{m+\log d}}\right)$$

queries to the oracle.

Proof. We apply the hybrid method variant introduced in [AK07]. Let O_1 be the entangled oracle and $O_2 = O_1 - 2|\theta\rangle\langle\theta|^{\otimes 2}$ be the oracle with a hidden product state $|\theta\rangle^{\otimes 2}$, given $|\theta\rangle \in \mathbb{C}^d$.

Suppose we have a quantum algorithm A that solves the Planted Product State problem with T queries with the help of an m-bit classical witness. For each $|\theta\rangle$, fix the string w that maximizes the probability that algorithm accepts O_2. Let $S(w) \subseteq S^d$ be the set of states associated with witness string w. Since $S(w)$ form a partition, then there must be one set $S(w^*)$ with measure at least $\frac{1}{2^m}$.

Let σ be the uniform measure over $S(w^*)$. Hence, fix w^* as the witness in the algorithm, and choose O_2 where $|\theta\rangle$ is selected from σ. We claim that the algorithm still requires a large number of queries T to distinguish between oracles O_1 and O_2 in this case. To establish the lower bound, let $|\psi_t\rangle$ be the result of the algorithm A with oracle O_2 applied t times followed by oracle O_1 applied $T-t$ times. By [AK07], We can bound the difference in Euclidean norm between successive hybrids by:

$$\| \, |\psi_{t+1}\rangle - |\psi_t\rangle \, \|_2 \leq \sqrt{\mathrm{Tr}((O_1 - O_2)^*(O_1 - O_2)\rho_t)} = 2\sqrt{\mathrm{Tr}(|\theta\rangle\langle\theta|^{\otimes 2}\rho_t)},$$

where ρ_t is the marginal state of the query register before the t^{th} query since $O_1 - O_2 = 2|\theta\rangle\langle\theta|^{\otimes 2}$. Hence, the Cauchy-Schwartz inequality implies that over a randomly selected $|\theta\rangle$ from σ:

$$\mathbb{E}_\sigma[\| \, |\psi_{t+1}\rangle - |\psi_t\rangle \, \|_2] \leq 2\sqrt{\mathbb{E}_\sigma[\mathrm{Tr}(|\theta\rangle\langle\theta|^{\otimes 2}\rho_t)]}.$$

Since σ is 2^{-m}-uniform, and the function $f(|\psi\rangle) = \mathrm{Tr}(|\psi\rangle\langle\psi|^{\otimes 2}\rho)$ is a non-negative, bounded, Lipschitz function, then we can bound by Lemma 3.3.11 and Lemma 3.3.12 that:

$$\begin{aligned}
\mathbb{E}_\sigma[\mathrm{Tr}(|\theta\rangle\langle\theta|^{\otimes 2}\rho_t)] &\leq \mathbb{E}_\mu[\mathrm{Tr}(|\theta\rangle\langle\theta|^{\otimes 2}\rho_t)] + O\left(\sqrt{\frac{m+\log d}{d}}\right) \\
&\leq \frac{2}{d(d+1)} + O\left(\sqrt{\frac{m+\log d}{d}}\right) \\
&\leq O\left(\sqrt{\frac{m+\log d}{d}}\right),
\end{aligned}$$

since $\frac{2}{d^2} \leq \sqrt{\frac{\log d}{d}}$ for sufficiently large d. Hence,

$$\mathbb{E}_\sigma[\| |\psi_{t+1}\rangle - |\psi_t\rangle \|_2] \leq 2\sqrt{\mathbb{E}_\sigma[\mathrm{Tr}(|\theta\rangle\langle\theta|^{\otimes 2}\rho_t)]} \leq O\left(\sqrt[4]{\frac{m+\log d}{d}}\right).$$

Hence, if $|\psi_0\rangle$ was the final state where all oracle calls were to O_1, and $|\psi_T\rangle$ was the final state where all oracle calls were to O_2, then the triangle inequality implies that

$$\mathbb{E}_\sigma[\| |\psi_T\rangle - |\psi_0\rangle \|_2] \leq O\left(T\sqrt[4]{\frac{m+\log d}{d}}\right).$$

If the algorithm A correctly distinguishes between the two cases, then $\mathbb{E}_\sigma[\| |\psi_T\rangle - |\psi_0\rangle \|_2] = \Omega(1)$. Hence the number of queries T satisfies

$$T \geq \Omega\left(\sqrt[4]{\frac{d}{m+\log d}}\right)$$

. □

In particular, this bound shows that a polynomial sized classical witness is not sufficient to help a quantum verifier solve the Entangled Subspace Problem efficiently, since any quantum verifier that solves the Entangled Subspace problem can also be used to solve the Planted Product State problem.

3.4 Open Problems

We end by describing some open problems and future directions.

Strong QMA Lower Bounds for the Entangled Subspace Problem. Can one show that any QMA tester for the Entangled Subspace problem requires either a superpolynomial number of queries, or a superpolynomial sized witness? This would yield a (quantum) oracle separation between QMA and QMA(2), and in particular would rule out the existence of so-called "disentanglers" [ABD+08].

Better Query Upper Bounds. Are the bounds proven using the generalized polynomial method tight? In particular, the following gaps remain:

- We have shown that there is a $O(\frac{t\sqrt{d}}{\epsilon})$ upper bound and a $\Omega(\max(\frac{t}{\epsilon}, \sqrt{d}))$ lower bound in the BQP setting for the recurrence problem and used this bound to prove a similar lower bound in the QMA setting. Is there a better lower or upper bound in either the BQP or QMA settings? However, a more sophisticated symmetrization technique may be required to improve the lower bound.
- We expect the BQP lower bound in Theorem 1.2.10 for the entanglement entropy can be improved by using a more creative application of the polynomial method.

We note that [MdW23] proved a tight bound for the recurrence time problem in the BQP setting, and [WZ23] provided an improved lower bound for the entanglement entropy problem in a certain range for the gap. However, both proofs use the adversary method and only consider the BQP setting. It would be interesting to see if there can be any further improvements in the BQP or QMA setting using polynomial method techniques.

Improving Proposition 3.3.7. Is the counterexample of Proposition 3.3.7 tight, in the sense that there are no examples that fool the verifier in dimensions 2, 3, 4, or 5? Otherwise, if there was an example that fools the verifier in dimension 2, this would give additional evidence that the Entangled Subspace problem in low dimensions is already hard for QMA.

Other Applications of the Generalized Polynomial Method. What are other applications of the generalized polynomial method? For instance, Procesi [Pro76] has characterized the invariants of matrix tuples under conjugation by the general linear, unitary, orthogonal, and symplectic groups. Are there natural problems in quantum query complexity that display other, non-unitary symmetries?

A Generalized Dual Polynomial Method? A line of works established tight quantum query lower bounds on classical problems by employing a method of *dual polynomials* [She13, Spa08, BKT18]. The goal of this method is to prove degree lower bounds of acceptance probability polynomials, but instead of symmetrizing the polynomials to obtain a polynomial of one or two variables, one instead takes advantage of *linear programming duality* to prove the degree lower bounds; this involves constructing objects known as dual polynomials. A natural question would be to investigate whether the method of dual polynomials can be extended to prove query lower bounds for unitary property testing.

Communication Complexity Separations. Separations in the query model often imply separations in communication complexity, using "lifting theorems" [GPW17]. Can any of the query separations for unitary property testing be lifted to the communication setting? As observed in [NN22], a separation between QMA and QCMA in the communication complexity setting remains open, although query separations already exist.

Part II

Tensor Isomorphism

Chapter 4

Proof Complexity

In this chapter, we provide background on proof complexity and the various proof systems studied in this section of thesis. Proof complexity studies the minimum length of proofs for a given theorem, which is a fundamental question in logic, computer science, and mathematics.

4.1 Propositional Proof Systems

Proof Systems. We firstly state some basic definitions related to proof systems.

Definition 4.1.1. A proof system for a language L is a polynomial-time algorithm $V(x, p)$ with the property that $x \in L$ if and only if there exists a string p such that V accepts (x, p).

In other words, we think of p as a proof that $x \in L$, and V should be a verification algorithm for the pair (x, p).

Definition 4.1.2. A proof system is p-bounded if for all $x \in L$ of length n, there exists a string p of length $|p| \leq q(n)$ for some polynomial $q(n)$, such that V accepts (x, p).

Recall that UNSAT is the set of unsatisfiable Boolean formulas. A proof system is called a propositional proof system if it is a proof system for UNSAT. We have the following fundamental observation, due to Cook-Reckhow [CR79].

Theorem 4.1.1. NP = *coNP if and only if there is a p-bounded propositional proof system.*

Proof. Recall that since SAT is NP-Complete, then UNSAT is coNP-Complete.

A p-bounded propositional proof system is equivalent to a polynomial-time verification algorithm. Hence, there exists a p-bounded propositional proof system iff UNSAT $\in$ NP. If UNSAT $\in$ NP then since UNSAT is coNP$-$Complete, this implies that coNP $=$ NP. □

Therefore, proof complexity was proposed as a possible program towards resolving the P versus NP problem. If we can show that there is no p-bounded proof system for UNSAT, then NP $\neq$ coNP (which implies P $\neq$ NP). Showing lower bounds against increasingly powerful proof systems can then be viewed as progress towards resulting the P versus NP problem.

We furthermore define p-simulation of proof systems as a way compare proof systems. This is analogous to the notion of polynomial time reductions between languages.

Definition 4.1.3. Given two proof systems P, Q, we say that Q p-simulates P if there exists a polynomial time function f such that P accepts (x, p) if and only if Q accepts $(x, f(p))$.

In other words, if P has a proof of x of length s, then Q has a proof of x of size $poly(s)$, for some polynomial. This is a useful notion to compare the relative power of two proof systems.

Resolution. A canonical example of a propositional proof system is resolution. Resolution aims to certify that a Boolean formula written in conjunctive normal form $C = C_1 \wedge C_2 \cdots \wedge C_m$ is unsatisfiable.

The resolution rule takes in as input two clauses $A \vee x, B \vee \bar{x}$ and derives the clause $A \vee B$. This rule is sound since any assignment of variables satisfying the inputs must also satisfy the output.

Definition 4.1.4 (Resolution)**.** A resolution refutation of an unsatisfiable CNF C is a sequence of clauses $D_1, \ldots, D_l$, where:

- For $1 \leq i \leq m$, $D_i = C_i$ is an input clause.
- For $i \geq m+1$, D_i is equal to the resolution rule applied to D_j and D_k for some $j, k < i$.
- The last clause D_l is the empty clause.

We say that l is the length of the resolution refutation.

Since the resolution rule is sound, if a refutation exists for C, then this is a proof that the original CNF C was unsatisfiable.

Resolution lower bounds have been of great interest since resolution is the basis of numerous SAT-solving algorithms, including the DPLL and CDCL algorithms [BKS03]. As such, resolution lower bounds indicate which instances of SAT will likely be intractable for modern SAT solvers.

The first exponential lower bound for general Resolution is due to Haken [Hak85], which later simplified by Beame and Pitassi [BP96]. These lower bounds are for resolution refutation the pigeonhole principle, which is the unsatisfiable formula encoding the statement that there is no bijection $f : [n+1] \to [n]$, for any natural number n.

Definition 4.1.5 (Pigeonhole Principle)**.** The pigeonhole principle PHP_n^{n+1} is a Boolean formula defined on propositional variables x_{ij} for $1 \leq i \leq n+1, 1 \leq j \leq n$. The clauses of PHP_n^{n+1} are:

- Every pigeon belong to a hole: $\bigvee_{j=1}^{n} x_{ij}$ for each $i = 1, \ldots, n+1$
- No two pigeons occupy the same hole: $\overline{x_{ij}} \vee \overline{x_{i'j}}$ for each $1 \leq i < i' \leq n+1$ and $1 \leq j \leq n$.

Theorem 4.1.2 ([Hak85, BP96])**.** *For sufficiently large n, any Resolution proof of PHP_n^{n+1} requires length at least $2^{n/20}$.*

Frege Systems. Stronger propositional proof systems compared to Resolution have also been studied. In particular, they are able to use general propositional formulas rather than clauses only. These are known as Frege systems. While we do not introduce Frege systems formally, we note that proving lower bounds on general Frege systems remains one of the frontier open problems of proof complexity.

Lower bounds are known for the restriction of Frege systems where each formula appearing in the formula has bounded depth. These are known as AC^0-Frege systems in analogy to the AC^0 circuit class. Exponential lower bounds for the pigeonhole principle are known for AC^0-Frege [PBI93]. However, it still opens an open problem to prove lower bounds for $\mathsf{AC}^0[2]$-Frege systems, although exponential lower bounds for the power of $\mathsf{AC}^0[2]$ circuits are known [Smo87].

4.2 Algebraic Proof Systems

Algebraic proof systems were introduced to make progress on the general problem of proving Frege lower bounds. As observed in [Pit96], the $\mathsf{AC}^0[2]$-Frege system can simulate low-degree algebraic proof systems. Therefore, lower bounds on algebraic proof systems are necessary towards solving the more general problem of proving Frege lower bounds. However, algebraic proof complexity has grown into a field of study in its own right.

Rather than working with propositional formulas, algebraic proof systems aim to prove that a system of polynomial equations has no solution. This is a generalization of the problem of determining whether or a formula is unsatisfiable or not, since the satisfiability of a given propositional formula can be reduced to determining if a system of polynomial equations has a common root.

For instance, the clause $C_1 = x_1 \wedge \bar{x_2} \wedge x_3$ is satisfied if and only if $(1 - x_1)x_2(1 - x_3) = 0$ has a solution where each variable takes on a value in $\{0, 1\}$. Therefore, satisfiability of a Boolean formula is equivalent to solving a system of polynomial equations, by converting a CNF formula into a system of polynomials in this manner. However, algebraic proof complexity is a more general setting since there are systems of polynomial equations that do not come from such translations of propositional formulas.

We now give several examples of algebraic proof systems that will be discussed in this thesis. For the rest of this section, we assume that our polynomials are defined over some field $\mathbb{F}$.

Nullstellensatz. The simplest algebraic proof system is based on Hilbert's Nullstellensatz. The use of the Nullstellensatz in proof complexity was introduced in [BIK+96], who studied the complexity of Nullstellensatz refutations for a modular counting principle. They observed that the Nullstellensatz lower bound implies a lower bound in a certain bounded-depth Frege system.

Definition 4.2.1 (Nullstellensatz)**.** Given an unsatisfiable system of polynomial equations $f_1 = 0, \ldots, f_k = 0$, where each $f_i \in \mathbb{F}[x_1, \ldots, x_n]$ a Nullstellensatz certificate is a set of polynomials $g_i \in \mathbb{F}[x_1, \ldots, x_n]$ such that

$$\sum_{i=1}^{k} f_i g_i = 1.$$

The complexity of a Nullstellensatz proof can be measured by its degree, which is the maximum degree of a polynomial g_i appearing in the proof.

It is clear that if a Nullstellensatz certificate exists, then the original system of polynomial equations cannot satisfied. Thus, the Nullstellensatz proof system is sound.

Nullstellensatz is also complete for systems of polynomial equations coming from translations of unsatisfiable Boolean formulas. A proof of completeness is given in [Pit96]. Therefore, each unsatisfiable Boolean formula on n variables has a Nullstellensatz refutation, where each polynomial g_i is of degree $O(n)$. This is in contrast to the general situation where exponential degree lower bounds for Nullstellensatz refutations are possible [Kol88].

For Boolean systems of equations, several Nullstellensatz lower bounds have been proven. The main technique used is the technique of constructing designs, introduced in [Bus96]. There a sharp correspondence between Nullstellensatz degree and designs in the sense that a degree d design exists for a given system of equations iff a degree d Nullstellensatz refutation does not exist for that system. The design method was applied in the following works to prove Nullstellensatz lower bounds:

- In [BCE+95], an $\Omega(\sqrt{n})$ degree lower bound was proven for the pigeonhole principle. This is applied to provide a separation between TFNP classes in the black-box setting.
- In [BP98], a $\Omega(\log n)$ degree lower bound was proven for the (weak) induction principle. A matching $O(\log n)$ degree upper bound was also proven.
- In [Bus96], an $\Omega(n)$ degree lower bound was proven for the housesitting principle, which is a version of the strong induction principle.

We finally note that Nullstellensatz and Resolution are incomparable as propositional proof systems, in the sense that neither system p-simulates the other.

In one direction, if the field $\mathbb{F}$ is a finite field, then the modular counting principles have $O(1)$-degree Nullstellensatz proofs, but require exponential sized Resolution proofs. In the other direction, the Pebbling formulas, defined in [BOCIP02], are examples of formulas that have polynomial sized Resolution proofs, but require almost maximal degree proofs in Nullstellensatz.

Polynomial Calculus. After Nullstellensatz, the next algebraic proof system to have been studied was the Polynomial Calculus, also known as Gröbner proofs. Polynomial Calculus (PC) can be thought of a "dynamic" version of the Nullstellensatz. It still aims to prove that 1 is in the ideal generated by the original equations, but in an iterative fashion.

Definition 4.2.2 (Polynomial Calculus Rules)**.** There are two rules in polynomial calculus.

- *Linear combinations:* Given polynomials $f = 0$ and $g = 0$, one can derive that any linear combination satisfies $\alpha f + \beta g = 0$ for any $\alpha, \beta \in \mathbb{F}$.
- *Multiplication rule:* Given a polynomial $f = 0$, one can derive that $x_i f = 0$ for any variable x_i.

Definition 4.2.3 (Polynomial Calculus)**.** Given an unsatisfiable system of polynomial equations $f_1 = 0, \ldots, f_k = 0$, where each $f_i \in \mathbb{F}[x_1, \ldots, x_n]$. a polynomial calculation refutation is a sequence of polynomials $p_1, \ldots, p_l$ where:

- For $i = 1, \ldots k, p_i = f_i$.
- For all $i > k$, each p_i is derived from previous polynomials appearing in the proof by either the linear combination or multiplication rule.
- The last polynomial appearing is $p_l = 1$.

The degree of a PC proof is the maximum degree of any polynomial p_i appearing in the proof. The size of a PC proof is the total number of monomials appearing in the proof.

We note that since the polynomial calculus rules are sound, then a PC refutation of a system of equations shows that that the system was unsatisfiable. Since PC can simulate Nullstellensatz, this also shows that every unsatisfiable system of polynomial equations having a Nullstellensatz refutation also has a PC proof.

Example 4.2.1. As an example of a PC proof, we consider the (weak) induction principle on n variables, which is encoded by the equations

$$x_1 = 1 \tag{4.1}$$
$$x_i(1 - x_{i+1}) = 0, 1 \leq i \leq n-1 \tag{4.2}$$
$$x_n = 0 \tag{4.3}$$

A PC refutation of these equations iteratively derives $x_i = 1$ for $1 \leq i \leq n$ in the following manner, which leads to a contradiction with the final equation $x_n = 0$.

If $x_i = 1$ is given, then using the multiplication rule gives $x_i x_{i+1} - x_{i+1} = 0$.

Adding $x_i x_{i+1} - x_{i+1} = 0$ to the axiom $x_i(1 - x_{i+1}) = x_i - x_i x_{i+1} = 0$ yields $x_i - x_{i+1} = 0$.

Adding $x_i - x_{i+1} = 0$ to $x_i - 1 = 0$ gives $x_{i+1} - 1 = 0$.

Overall, this PC proof can be performed in constant degree (degree 2), which is independent of the number of variables.

We note the following results which compare PC to other proof systems:

- Buss-Pitassi [BP98] showed that Nullstellensatz proofs of the induction principle require degree $\Omega(\log n)$. This yields super-constant degree separation between Nullstellensatz and PC proofs, due to the previous example.
- As a generalization of the previous result, there is actually a linear degree separation between Nullstellensatz proofs and PC proofs, as observed in [CEI96a, Bus96]. They showed that the house-sitting principle formulas have $O(1)$ degree PC proofs but require degree $\Omega(n)$ Nullstellensatz proofs.
- [CEI96a] showed that PC proofs can simulate Resolution proofs in the following sense. These results are asymptotically optimal, even for propositional formulas admitting polynomial size resolution proofs, due to the work of Galesi and Lauria [GL10].
  - If there is a tree-like Resolution proof of a CNF with s lines, then there is a PC proof of its polynomial translation of degree $O(\log s)$.
  - For general resolution, if there a Resolution proof of a CNF with s lines, then there is a PC proof for its polynomial translation of degree $O(\sqrt{n} \log s)$.

We also note that in PC, there is a size-degree tradeoff. In particular, proofs with small size can also be made to have small degree. This is formalized in the following result due to Impagliazzo, Pudlak, and Sgall [IPS99].

Theorem 4.2.2. *[IPS99, Theorem 6.2] Suppose P is a set of polynomials in n variables of degree at most d. Then if P has a polynomial calculus refutation with m monomials, then it also has a refutation of maximum degree* $\max(d, O(\sqrt{n \log m}))$.

In particular, if a PC refutation of a set of polynomials of constant degree requires degree d, then then the monomial size of the proof is at least $2^{\Omega(d^2/n)}$. Therefore, any degree bound $d > \sqrt{n}$ implies that the the PC refutation requires super-polynomial size.

Gröbner Bases These results about PC are particularly relevant when discussing the relationship between PC and Gröbner bases, which has been a fundamental idea in computational algebraic geometry. This connection was observed in [CEI96a] in their seminal work that introduced the PC proof system.

We first recall some generalities on Gröbner bases. For details, refer to [CLO13, Chapter 2].

In general, given a set of polynomials $f_1, \ldots, f_n \in \mathbb{F}[x_1, \ldots, x_n]$, it is difficult to determine if a polynomial f lies in the ideal $I = (f_1, \ldots, f_n)$ generated by those polynomials. However, finding a Gröbner basis of I makes the ideal membership problem easier.

Before defining Gröbner bases, we first need to introduce monomial orderings.

Definition 4.2.4. Let $\mathbb{Z}^n_{\geq 0}$ be the set of n-tuples of non-negative integers. A monomial ordering is a relation $\geq$ on $\mathbb{Z}^n_{\geq 0}$ with the properties that:

- $\geq$ is a total order.
- If $\alpha \geq \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$, then $\alpha + \gamma \geq \beta + \gamma$.
- $\geq$ is a well-ordering (i.e. every non-empty subset $S \subseteq \mathbb{Z}^n_{\geq 0}$ has a minimal element.)

A canonical example of a monomial ordering is the lexicographic order. Given $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$, we define $a <_{lex} b$ iff $a_i \leq b_i$ for each index i.

Definition 4.2.5 (Leading Terms)**.** Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial and write $f = \sum_{\alpha \in \mathbb{Z}^n_{\geq 0}} a_\alpha x^\alpha$ coefficientwise. Let $>$ be a monomial ordering.

- The *multi-degree* $md(f)$ of f is the maximum of α appearing in the polynomial f with $a_\alpha \neq 0$, with respect to the ordering $>$.
- The *leading monomial* $LM(f)$ of f is $x^{md(f)}$.
- The *leading term* $LT(f)$ of f is then monomial $a_{md(f)} x^{md(f)}$.

For instance, with respect to the lexicographic ordering with $x > y$, $LT(x^5 y - x y^5) = x^5 y$.

Definition 4.2.6 (Leading Term Ideal)**.** Given an ideal $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$, we define

$$LT(I) = \{LT(f) : f \in I\}$$

and $\langle LT(I) \rangle$ to be the ideal generated by $LT(I)$.

We can now define Gröbner bases.

Definition 4.2.7. Let I be an ideal. A set $\{g_1, \ldots, g_t\} \subseteq I$ is a *Gröbner bases* of I if

$$\langle LT(I) \rangle = \langle LT(g_1), \ldots, LT(g_t) \rangle.$$

Gröbner bases turn out to have very nice properties particularly for computational algebra. For instance, given arbitrary generators $f_1, \ldots, f_n$ of an ideal I, it may not be possible to determine if $f \in I$ by division of f by $f_1, \ldots, f_n$. We do not present the multivariable polynomial divison algorithm in detail here, but for details an interested reader can refer to [CLO13, Chapter 2.3].

Example 4.2.3. [CLO13, Chapter 2.5] We use the lexiographic order with $x > y$. For example, given polynomials $f_1 = x^3 - 2xy$ and $f_2 = x^2y - 2y^2 + x$, we have that $x^2 \in (f_1, f_2)$, since

$$x^2 = x(x^2y - 2y^2 + x) - y(x^3 - 2xy)$$

.

Since $LT(f_1) = x^3$ and $LT(f_2) = x^2y$, the polynomial x^2 does not divide into f_1 or f_2.

However, since a Gröbner basis for $I = (f_1, f_2)$ is given by $G = \{x^3 - 2xy, x^2y - 2y^2 + x, -x^2, -2xy, -2y^2 + x\}$, it is immediately evident from examining the basis that $x^2 \in I$.

Given a Gröbner basis $g_1, \ldots, g_t$ of I, the Gröbner property guarantees that if $f \in I$, then $LT(f)$ can be divided by at least one of the terms of $LT(g_i)$ in the basis. Therefore, division can be carried out to solve the ideal membership problem. This observation is summarized in the following theorem.

Theorem 4.2.4. *Let $G = \{g_1, \ldots, g_t\}$ be a Gröbner basis for I. Then $f \in I$ if and only if the remainder on division of f by G is zero.*

In particular, if $1 \in I$, then 1 must appear as an element in a Gröbner basis G of I.

We now turn to existence of Gröbner bases. The algorithm commonly used to compute Gröbner bases was developed by Buchenberger in his PhD thesis. The proof of correctness of Buchenberger's algorithm is based on the following observation.

Definition 4.2.8. Given polynomials f, g, the S polynomial is defined as

$$S(f, g) = \frac{LCM(LM(f), LM(g))}{LT(f)} f - \frac{LCM(LM(f), LM(g))}{LT(g)} g$$

where LCM denotes the least common multiple.

Given a set G and a polynomial f, we also write $\bar{f}^G$ for the remainder of f upon division by G. Buchenberger's criterion then gives a characterization of Gröbner bases in terms of S-polynomials.

Theorem 4.2.5 (Buchenberger's criterion)**.** *Given a set $G = \{g_1, \ldots, g_t\}$ where each $g_i \in I$, G is a Gröbner basis for an ideal I if and only if*

$$\overline{S(g_i, g_j)}^G = 0$$

for each pair $g_i, g_j \in G$, $i \neq j$.

Buchenberger's criterion than leads to an idea for computing Gröbner bases, which is to repeatedly add polynomials to a given set S of generators defining I, until S is closed under Buchenberger's criterion.

Theorem 4.2.6 (Buchenberger's algorithm)**.** *Given I with generators $F = \{f_1, \ldots, f_s\}$, the Buchenberger's algorithm terminates and computes a Gröbner basis of I.*

A proof of correctness of the algorithm is given in [CLO13, Chapter 2.7].

Polynomial calculus and Gröbner bases We can now discuss the relationship between Gröbner bases and PC proofs. We first need to introduce the notion of a pseudo-ideal, which was a notion introduced in [CEI96a] and also [BGIP99].

Algorithm 1 Buchenberger's Algorithm

$G = F$ ▷ Input: A set $F = \{f_1, \ldots, f_s\}$ generating an ideal I
repeat
  $G' = G$
  for all pairs $(p, q), p \neq q$ in G **do**
    Compute $r = \overline{S(p,q)}^{G}$
    if $r \neq 0$ **then**
      $G \leftarrow G \cup \{r\}$
    end if
  end for
until $G' = G$
return G ▷ Ouput: A Gröbner basis G of I

Definition 4.2.9. A degree d pseudo-ideal V is a subspace of $\mathbb{F}[x_1, \ldots, x_n]$ satisfying the following conditions:

- Every polynomial $p \in V$ has degree at most d.
- If $p \in V$ has degree $\leq d-1$, then $xp \in V$ for any variable x.

The following observation is due to [CEI96a].

Theorem 4.2.7. *Let $I_d(p_1, \ldots, p_k)$ be the set of polynomials having a degree d polynomial calculus proof, starting from $p_1, \ldots, p_k$ as axioms. Then I_d is a degree d pseudo-ideal, and for any pseudo-ideal I containing $p_1, \ldots, p_k$, we have $I_d \subseteq I$.*

We now need a slight variation on the original definition of Gröbner bases.

Definition 4.2.10. Let $G = \{g_1, \ldots, g_t\}$ be a set of polynomials.

A polynomial f has a degree-d representation with respect to G if there are polynomials h_i with $f = \sum_{i=1}^{t} g_i h_i$, and $\deg g_i h_i \leq d$ for each i.

A polynomial f is reducible if the remainder of f upon division by G is zero.

A set G is a degree-d Gröbner basis if all polynomials f with a degree d representation with respect to G are reducible.

Suppose $I = (f_1, \ldots, f_k)$ and G is a degree d Gröbner basis containing $f_1, \ldots, f_k$. Unfortunately, the set of polynomials in an ideal I with a degree d representation is in general not equal to the degree d part of I. However, we note that this is true if the original polynomials $f_1, \ldots, f_k$ are assumed to be homogenous polynomials. This motivated the study of homogenization as a heuristic for PC proofs in [BOCIP02].

However, a degree d Gröbner basis still has some useful computational properties. [CEI96a] observed that a degree-d Gröbner basis can be found using a slight modification of Buchenberger's algorithm, where all S polynomials computed in the algorithm with degree greater than d are ignored. The modified Buchenberger's algorithm is presented here.

Furthermore, [CEI96a] observed that the modified algorithm runs in $n^{O(d)}$ time under the assumption that the input is a set of multilinear polynomials with n variables.

Now suppose that G is a degree-d Gröbner basis containing polynomials $f_1, \ldots, f_k$. If B_G is the set of reducible polynomials with respect to G, then B_G is a degree-d pseudo-ideal. This can be seen since

Algorithm 2 Modified Buchenberger's Algorithm

$G = F$ ▷ Input: A set $F = \{f_1, \ldots, f_s\}$ generating an ideal I
repeat
  $G' = G$
  for all pairs $(p, q), p \neq q$ in G **do**
    if $\deg S(p, q) \leq d$ **then**
      Compute $r = \overline{S(p,q)}^{G}$
      if $r \neq 0$ **then**
        $G \leftarrow G \cup \{r\}$
      end if
    end if
  end for
until $G' = G$
return G ▷ Ouput: A degree-d Gröbner basis G of I

B_G is exactly the set of polynomials with a degree-d representation with respect to G. Combining this observation with Theorem 4.2.7 leads to the following observation.

Theorem 4.2.8. *[CEI96a] Let $I_d(p_1, \ldots, p_k)$ be the set of polynomials with degree d PC proofs starting from initial polynomials $p_1, \ldots, p_k$. Let G be a degree d Gröbner basis containing $p_1, \ldots, p_k$. Then $I_d \subseteq B_G$.*

This implies that if there is a PC proof of degree d for a given set of multilinear polynomials, then the PC proof can be found by computing a degree-d Gröbner basis in time $n^{O(d)}$.

In the converse direction, the time complexity of Buchenberger's algorithm can also be bounded by the size of a smallest PC proof. If polynomials are represented as lists of coefficients, the operations of Buchenberger's algorithm such as S-remainders and polynomial division can be simulated in PC, using the multiplication and linear combination rules. Therefore, this leads to the following observation.

Theorem 4.2.9. *[IPS99] If Buchenberger's algorithm is implemented by storing polynomials as lists of coefficients, then the total runtime of Buchenberger's algorithm for refuting a system of polynomial equations $f_1 = 0, \ldots, f_k = 0$ is at least the size of a PC refutation for the system of equations.*

Such size lower bounds can be obtained using the size-degree tradeoff of Theorem 4.2.2, given sufficiently strong degree lower bounds for a system of polynomial equations.

However, PC lower bounds may not in general give lower bounds on the runtime of other algorithms for computing Gröbner bases, such as the F_4 algorithm [CLO13] that employs linear algebraic techniques for computing a Gröbner basis. Stronger algebraic proof systems would necessary to study the runtime of these algorithms.

Lower Bounds in PC We finally note some important lower bounds for various families of formulas in PC.

- [Raz98a], followed by [IPS99], proved $\Omega(n)$ degree lower bounds for the pigeonhole principle PHP_n^m, for any $m > n$.
- Suppose $\mathbb{F}$ has odd characteristic. [BGIP99] showed $\Omega(n)$ degree lower bounds for the Tseitin principles on a graph on n nodes for PC proofs over $\mathbb{F}$. Informally, the Tseitin principles encode

the principle that a graph must have an even number of vertices of odd degree. They also used a low-degree PC reduction to show that the mod p counting principles cannot be refuted in low degree, assuming the field has characteristic different from p.

- A random k-CNF is generated by picking m clauses uniformly at random out of all $2^k\binom{n}{k}$ possible clauses on n variables. At sufficiently high clause densities $\Delta = \frac{m}{n}$, a random k-CNF is almost surely unsatisfiable. [BSI10] showed that there exists at clause density Δ, degree $\Omega(\frac{n}{\Delta^2})$ degree PC proofs are needed to show a random k-CNF at density Δ is unsatisfiable.

Sum of Squares. We now restrict attention to $\mathbb{F}$ being the real numbers. A semialgebraic proof system is a proof system that can manipulate both polynomial equalities and inequalities. Sum of squares is such an example of a semialgebraic proof system.

We say that a polynomial p is a sum of squares polynomial if there exists polynomials q_i for which $p(x) = \sum_{i=1}^{m} q_i(x)^2$. Observe that sum-of-squares polynomials are always non-negative. This motivates the definition of sum-of-squares proofs.

Definition 4.2.11. Given a set of polynomial equations $f_1 = 0, \ldots, f_m = 0$ and inequalities $h_1 \geq 0, \ldots, h_s \geq 0$, a sum of squares proof that $f \geq 0$ is a polynomial identity:

$$\sum_{i=1}^{m} g_i f_i + p_0 + \sum_{l=1}^{s} p_l h_l = f$$

where g_i are arbitrary polynomials and p_i are sum-of-squares polynomials.

If $f = -1$, we call this a sum-of-squares refutation, since in this case the original system of polynomial equations and inequalities cannot be satisfied.

As usual, the complexity of a proof can be measured by its degree. The degree of a sum-of-squares proof is the maximum degree of a polynomial $g_i f_i$, p_0 or $p_l h_l$ appearing in the proof.

SoS proofs can be a powerful algebraic proof system. [Ber18] studied the relationship between Sum-of-Squares and Polynomial Calculus, and showed that PC proof can be simulated in Sum-of-Squares whenever the set of polynomials contains the Boolean axioms. Furthermore, SoS admits polynomial-sized proofs of the pigeonhole principle [FKP+19, Section 3.2], which has been shown to be a hard example for many other proof systems. Finally, sum-of-squares proofs can be found using semidefinite programming [FKP+19], just as polynomial calculus proofs can be found using the Buchenberger's algorithm.

Lower Bounds for SoS However, SoS also has limitations as a proof system. we review some notable lower bounds for SoS.

- Grigoriev [Gri01] showed $\Omega(n)$ lower bounds for the Tseitin formulas and the parity principle in sum-of-squares.
- Schoenbeck [Sch08] showed $\Omega(n)$ lower bounds for refuting random XOR and CNF formulas in sum-of-squares. This was applied to show tight integrality gaps for various optimization problems for algorithms using the sum-of-squares hierarchy, unless the degree is linear in the number of variables.

Chapter 5

The Proof Complexity of Tensor Isomorphism

5.1 Preliminaries

5.1.1 PC Reductions

We define the notion of a PC reduction between two systems of polynomials.

Definition 5.1.1 (PC reduction between systems of polynomials, cf. [BGIP01, Sec. 3])**.** Let $P(x_1, \ldots, x_n)$ and $Q(y_1, \ldots, y_m)$ be two sets of polynomials over a field $\mathbb{F}$. P is (d_1, d_2)-*reducible* to Q if:

1. For each $i \in [m]$ there is a polynomial $r_i(\mathbf{x})$ of degree at most d_1 (which we think of as defining y_i in terms of the $\mathbf{x}$ variables);
2. There exists a degree d_2 PC derivation of $Q(r_1(\mathbf{x}), \ldots, r_m(\mathbf{x}))$ from polynomials $P(\mathbf{x})$.

Lemma 5.1.1 ([BGIP01, Lem. 1])**.** *If $P(\mathbf{x})$ is (d_1, d_2)-reducible to $Q(\mathbf{y})$ and there is a degree d PC refutation of $Q(\mathbf{y})$, then there is a degree $\max(d_2, d_1 d)$ refutation of $P(\mathbf{x})$.*

In their paper, they typically only applied this to systems of equations which were known to be unsatisfiable (such as PHP and Tseitin tautologies), whereas in our paper we have several situations we want to combine the above notion together with the usual notion of many-one reduction. We encapsulate this in the following definition. We say a decision problem Π is a *polynomial solvability problem* over a field $\mathbb{F}$ if all valid instances of the problem are systems of polynomial equations over $\mathbb{F}$, and the problem is to decide whether such a system of equations has solutions over the algebraic closure $\overline{\mathbb{F}}$. Thus, the difference between multiple polynomial solvability problems is just *which* systems of equations are valid inputs.

Definition 5.1.2 (PC many-one reduction)**.** Let Π_1, Π_2 be two polynomial solvability problems over a field $\mathbb{F}$. We say that Π_1 (d_1, d_2)-*many-one reduces* to Π_2 if there is a polynomial-time many-one reduction ρ from Π_1 to Π_2, such that for all unsatisfiable instances $\mathcal{F}$ of Π_1, $\mathcal{F}$ (d_1, d_2)-reduces to $\rho(\mathcal{F})$. When this occurs with $d_1, d_2 = O(1)$, we write

$$\Pi_1 \leq_m^{PC} \Pi_2.$$

5.1.2 Linear algebra and tensors

Given three vector spaces U, V, W over a field $\mathbb{F}$, a 3-tensor is an element of the vector space $U \otimes V \otimes W$, whose dimension is $(\dim U)(\dim V)(\dim W)$. If e_i is the i-th standard basis vector, then a basis for $U \otimes V \otimes W$ is given by the vectors $\{e_i \otimes e_j \otimes e_k\}$. One may also interpret the symbol $\otimes$ more concretely as the Kronecker product, in which $e_i \otimes e_j \otimes e_k$ represents a 3-way array whose only nonzero entry is in the (i, j, k) position. The vector space of such 3-way arrays (with coordinate-wise addition) is isomorphic to $U \otimes V \otimes W$.

The rank of a tensor $T \in U \otimes V \otimes W$ is the minimum r such that $T = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i$ for some vectors u_i, v_i, w_i.

Two $n \times m \times p$ 3-tensors $T, T' \in U \otimes V \otimes W$ are isomorphic if there exist matrices $X \in \mathrm{GL}(U), Y \in \mathrm{GL}(V), Z \in \mathrm{GL}(W)$ such that $(X, Y, Z) \cdot T = T'$, where the latter is shorthand for the equations

$$\sum_{ijk} X_{ii'} Y_{jj'} Z_{kk'} T_{ijk} = T'_{i'j'k'} \tag{5.1}$$

for each index (i', j', k') of the tensor T'.

If we treat T, T' as given non-isomorphic tensors, then these equations as a system of equations in the $n^2 + m^2 + p^2$ variables $X_{ii'}, Y_{jj'}, Z_{kk'}$. To enforce that these variable matrices are invertible, we furthermore introduce three additional sets of variables X', Y', Z' meant to be the inverse matrices, and include also the equations

$$XX' = X'X = I_n \qquad YY' = Y'Y = I_m \qquad ZZ' = Z'Z = I_p,$$

where I_n denotes the $n \times n$ identity matrix, which is Id_U in any basis. (We could have instead introduced new variables such as δ and the equation $\det(X)\delta = 1$, however, the latter equation is degree n, whereas the above equations all have degree $O(1)$, which is more desirable from the point of view of algebraic proof complexity.)

5.1.3 Polynomial encodings and the inversion principle

Some principles of linear algebra can be formulated as tautologies in propositional logic and therefore also as a set of polynomial equations. In this paper we preliminarily consider two such principles.

Rank Principle. As a first example we consider a set of unsatisfiable polynomials encoding the principle that the product of a $n \times r$ matrix X by a $r \times n$ matrix Y cannot be the identity matrix whenever $r < n$. We consider variables $x_{i,k}, y_{j,k}$ for $i, j \in [n]$ and $k \in [r]$, where $r < n$ to encode X and Y. Then the polynomial encoding is:

$$\mathbb{I}(r, n) := \sum_{k \in [r]} x_{i,k} y_{j,k} - \delta_{i,j} \qquad i, j \in [n]$$

where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise. This set of polynomials is clearly unsatisfiable as long as $r < n$.

Inversion Principle. The second principle encodes the invertibility of a square $n \times n$ matrix A, expressing the tautology that $AB = I \to BA = I$ where A, B are $n \times n$ matrices and I is the identity matrix. Stephen A. Cook suggested this principle as a tautology that may be hard to prove in several proof systems.

Let $a_{i,j}, b_{i,j}$ be formal variables encoding respectively the (i,j)-th entries of A and B. We represent the fact that $AB = I$ as the set of degree 2 polynomials

$$\sum_{k\in[n]} a_{i,k}b_{k,j} - \delta_{i,j} \qquad i,j \in [n],$$

where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise. We denote this set of polynomials by $AB = I$. In Section 5.2, we study the degree complexity of $AB = I \vdash BA = I$, that is of PC derivations of the polynomials $BA = I$ from the polynomials $AB = I$.

In view of the results we obtain in Section 5.2, we consider a *polynomial rule schema* of the form

$$\frac{AB = I}{BA = I}$$

which we call the *Inversion Rule* (INV) meant to be added to PC as an extra rule. We make this slightly more precise here.

A polynomial instantiation τ of the polynomials $AB = I$ is a substitution of polynomials $p_{i,j}, q_{i,j}$ to variables $a_{i,j}$ and $b_{i,j}$. In PC+INV a polynomial p is derivable from a set of polynomials $\mathcal{P}$ if

1. p is an axiom, or $p \in \mathcal{P}$;
2. p is obtained by multiplication or linear combination from previous polynomials in the proof;
3. p is a polynomial among a polynomial instantiation τ of $BA = I$, given that among the polynomials previously derived in the proof there are all the polynomials forming the instantiation τ of $AB = I$.

Pigeonhole Principle. An important role in proving the results in Section 5.2 is played by the well-known *Pigeonhole principle* stating that any function f from $[n]$ to $[r]$ with $r < n$ has a collision, that is there are $i \neq i' \in [n]$ and a $j \in [r]$ such that $f(i) = f(i') = j$. PHP^n_r is the set of polynomials:

$$\sum_{k\in[r]} p_{i,k} - 1, \text{ for } i \in [n], \qquad p_{i,k}p_{j,k}, \text{ for } i \neq j \in [n], k \in [r] \qquad p^2_{ij} - p_{ij}, \text{ for } i \in [n], j \in [r]$$

Razborov [Raz98b] additionally included the "functional equations" (encoding that each pigeon cannot be matched to more than one hole):

$$p_{i,k}p_{i,k'}, \text{ for } i \in [n], k \neq k' \in [r].$$

5.2 Linear algebra warm-up: PC for matrices

Two matrices $M, M' \in U \otimes V$ are isomorphic as tensors if they are equivalent as matrices, meaning under left- and right-multiplication by invertible matrices $X \in \mathrm{GL}(U), Y \in GL(V)$, that is,

$$XMY = M'.$$

Since we want X, Y to be invertible, we also introduce variable matrices X', Y' as before, together with the equations

$$XX' = X'X = \mathrm{Id}_U \qquad YY' = Y'Y = \mathrm{Id}_V .$$

Then by left multiplying our initial matrix equation by Y', we may replace it with the new matrix equation

$$XM = M'Y'.$$

The latter has the advantage of being linear in X and Y', but the quadratic equations $XX' = \mathrm{Id}_U, YY' = \mathrm{Id}_V$ still make even this case not totally obvious.

5.2.1 A trick for PC degree

If our focus is on PC *degree*, we note that the degree of the equations is unchanged if we first left- or right-multiply M, M' by invertible scalar matrices. For example, if we replace M by $\overline{M} = AMB$ with $A, B \in \mathrm{GL}(U)$, then we may replace X by $\overline{X} := XA^{-1}$, Y by $\overline{Y} := B^{-1}Y$. Then we have $\overline{M} \cong M$, so $\overline{M} \cong M'$ iff $M \cong M'$. Furthermore, since the transformation $X \mapsto XA^{-1}$, $Y \mapsto B^{-1}Y$ is linear and invertible, any PC proof that $\overline{M} \ncong M'$ can be transformed by the inverse linear transformation into a PC proof that $M \ncong M'$ of the same degree.

Now, for matrices under this equivalence relation, we have a normal form, namely every matrix M is equivalent to a diagonal matrix with $\mathrm{rk}(M)$ 1s on the diagonal and all the remaining entries 0, that is, $\sum_{i=1}^{\mathrm{rk}(M)} e_i \otimes e_i = I_r \oplus 0$, where the latter 0 denotes a 0 matrix of appropriate size $(n-r) \times (m-r)$. So by using the preceding trick, we may put both M and M' in this form. The two are isomorphic iff $\mathrm{rk}(M) = \mathrm{rk}(M')$, so for PC degree we have now reduced to the case of showing that $I_r \oplus 0$ and $I_{r'} \oplus 0$ are not isomorphic when $r \neq r'$.

Note that, aside from the equations saying X and Y are invertible, this is almost identical to the Rank Principle (see Section 5.1.3). In the rest of this section we will prove PC lower bounds on both the Rank Principle and the Inversion Principle. Here, we show that the addition of these invertibility axioms in fact makes 2TI much easier in PC than the Rank Principle or 3TI.

Proposition 5.2.1. *Let M, M' be two $n \times m$ matrices of ranks r, r' respectively, with $r' > r$. Then, over any field whose characteristic does not divide $r' - r$, the following equations have a degree 3 PC refutation and a degree 4 NS refutation:*

$$XMY^T = M' \qquad XX' = X'X = \mathrm{Id}_n \qquad YY' = Y'Y = \mathrm{Id}_m .$$

For those familiar with the low-degree PC proof of the functional onto-PHP, the following proof is similar.

Proof idea. By the observations in Section 5.2.1, we may assume without loss of generality (from the point of view of PC degree) that $M = \mathrm{Id}_r \oplus 0_{n-r \times m-r}$ and $M' = \mathrm{Id}_{r'} \oplus 0_{n-r' \times m-r'}$.

Write $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}$ where the top-left block X_{11} has size $r' \times r$, and similarly write $Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$ where Y_{11} has size $r' \times r$. In this notation, the matrix equation $XMY^T = M'$ becomes the

equations

$$\begin{aligned} XMY^T &= \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} \mathrm{Id}_r & \\ & 0_{(n-r)\times(m-r)} \end{bmatrix} \begin{bmatrix} Y_{11}^T & Y_{21}^T \\ Y_{12}^T & Y_{22}^T \end{bmatrix} \\ &= \begin{bmatrix} X_{11} & 0 \\ X_{21} & 0 \end{bmatrix} \begin{bmatrix} Y_{11}^T & Y_{21}^T \\ 0 & 0 \end{bmatrix} \\ &= M' = \begin{bmatrix} \mathrm{Id}_{r'} & \\ & 0_{(n-r')\times(m-r')} \end{bmatrix} \end{aligned}$$

which becomes the four matrix equations

$$X_{11}Y_{11}^T = \mathrm{Id}_{r'} \qquad X_{11}Y_{21}^T = 0 \qquad X_{21}Y_{11}^T = 0 \qquad X_{21}Y_{21}^T = 0. \tag{5.2}$$

Note that so far our PC proof hasn't actually done anything—it is all just notation, and all in the same degree we started with (degree 2).

Then, using the equations $XX' = \mathrm{Id}_n$ and $YY' = \mathrm{Id}_m$, we will derive that $Y_{11}^T X_{11} = \mathrm{Id}_r$. Then we derive 1 as

$$\frac{1}{r-r'}\left(\mathrm{Tr}(X_{11}Y_{11}^T - \mathrm{Id}_{r'}) - \mathrm{Tr}(Y_{11}^T X_{11} - \mathrm{Id}_r)\right).$$

The point here is that trace is additive and cyclically invariant, so $\mathrm{Tr}(X_{11}Y_{11}^T) \equiv \mathrm{Tr}(Y_{11}^T X_{11})$, identically as polynomials, so there is no further derivation needed. □

Proof. The proof starts using the first part of the proof idea above, so we continue from (Equation (5.2)) with the notation introduced above. In the remainder of the proof, we will derive $Y_{11}^T X_{11} = \mathrm{Id}_r$. Then the last paragraph of the proof idea will complete the proof.

To derive $Y_{11}^T X_{11} = \mathrm{Id}_r$, we will use the invertibility equations (those involving X' and Y'). Write $X' = \begin{bmatrix} X'_{11} & X'_{12} \\ X'_{21} & X'_{22} \end{bmatrix}$, where X'_{11} has size $r \times r'$ (NB: the size is the "transpose" of the size of X_{11}) and similarly for Y'.

From considering the upper-left $r \times r$ block of the matrix equation $X'X = \mathrm{Id}_n$, we get

$$X'_{11}X_{11} + X'_{12}X_{21} = \mathrm{Id}_r.$$

Right multiplying by Y_{11}^T, we get

$$X'_{11}X_{11}Y_{11}^T + X'_{12}X_{21}Y_{11}^T = Y_{11}^T.$$

But now we can subtract from this X'_{11} times the equation $X_{11}Y_{11}^T = \mathrm{Id}_{r'}$, and also X'_{12} times the equation $X_{21}Y_{11}^T = 0$ to get

$$X'_{11} = Y_{11}^T. \tag{5.3}$$

Similarly, considering the upper-left $r \times r$ block of the matrix equation $Y'Y = \mathrm{Id}_m$, we get $Y'_{11}Y_{11} + Y'_{12}Y_{21} = \mathrm{Id}_r$. For consistency with the notation above, we take the transpose of this entire equation (in PC, this is essentially a null-op—we are just re-arranging how we are viewing a set of $(r')^2$ equations on the page), to get:

$$Y_{11}^T(Y'_{11})^T + Y_{21}^T(Y'_{12})^T = \mathrm{Id}_r.$$

Left multiplying by X_{11}, we get

$$X_{11}Y_{11}^T(Y'_{11})^T + X_{11}Y_{21}^T(Y'_{12})^T = X_{11}.$$

Now, right-multiplying the equation $X_{11}Y_{11}^T = \mathrm{Id}$ by $(Y'_{11})^T$, and right-multiplying the equation $X_{11}Y_{21}^T = 0$ by $(Y'_{12})^T$ and subtracting both of these from the above, we get

$$(Y'_{11})^T = X_{11}. \tag{5.4}$$

Next, we derive $M - X'M'(Y')^T = 0$, as follows: left-multiplying $XMY^T - M'$ by X', and subtract from it $(X'X - I)$ times MY^T, to get $-X'M' + MY^T$. Now right-multiply the latter by $(Y')^T$ and subtract from it M times $(Y^T(Y')^T - I)$, yielding $-X'M'(Y')^T + M$. Now multiply by -1.

Now, from $X'M'(Y')^T = M$, as at the beginning of the proof, we derive that $X'_{11}(Y'_{11})^T = \mathrm{Id}_r$. But above we have derived that $X'_{11} = Y_{11}^T$ and $(Y'_{11})^T = X_{11}$, so from the preceding three equations we get $Y_{11}^T X_{11} = \mathrm{Id}_r$, as claimed. This completes the PC proof.

Let us unroll the PC proof to derive a Nullstellensatz proof (here we underline the use of original equations):

$$r - r' = \mathrm{Tr}(\underline{X_{11}Y_{11}^T - \mathrm{Id}_{r'}}) - \mathrm{Tr}(Y_{11}^T X_{11} - \mathrm{Id}_r)$$

Now we focus on the NS derivation of $Y_{11}^T X_{11} - \mathrm{Id}_r$. Since the trace is linear, and we are focusing on degree, this is without loss of generality. We have:

$$\begin{aligned}
Y_{11}^T X_{11} - \mathrm{Id}_r &= \left(X'_{11}(Y'_{11})^T - \mathrm{Id}_r\right) - (X'_{11} - Y_{11}^T)(Y'_{11})^T - Y_{11}^T\left((Y'_{11})^T - X_{11}\right) \\
&= \Big(-X'\underline{(XMY^T - M')}Y'^T + \underline{(X'X - I)}MY^T(Y')^T + M\underline{(Y^T(Y')^T - I)}\Big)_{11} \\
&\quad + \Big(\underline{(X'_{11}X_{11} + X'_{12}X_{21} - \mathrm{Id}_r)}Y_{11}^T - X'_{11}\underline{(X_{11}Y_{11}^T - \mathrm{Id}_{r'})} - X'_{12}\underline{(X_{21}Y_{11}^T)}\Big)(Y'_{11})^T \\
&\quad - Y_{11}^T\Big(X_{11}\underline{(Y_{11}^T(Y'_{11})^T + Y_{21}^T(Y'_{12})^T - \mathrm{Id}_r)} - \underline{(X_{11}Y_{11}^T - \mathrm{Id})}(Y'_{11})^T - \underline{(X_{11}Y_{21}^T)}(Y'_{12})^T\Big).
\end{aligned}$$

This is visibly degree 4. □

5.2.2 Inversion Principle implies the Rank Principle

Lemma 5.2.2. *If the $r \times r$ Inversion Principle has a degree d PC derivation, then there is a degree $\max\{d, 3\}$ PC refutation of the Rank Principle stating that a rank r matrix is not equivalent (isomorphic) to a rank n matrix, for any $n > r$.*

If the Inversion Principle has a degree d NS derivation, then the Rank Principle has a degree $d + 2$ NS refutation.

Proof. Suppose the $r \times r$ Inversion Principle has a degree-d derivation. Consider the Rank Principle $XY = I_n$ where X is $n \times r$ and Y is $r \times n$, with $n > r$. Write

$$X = \begin{bmatrix} X_0 \\ X_1 \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_0 & Y_1 \end{bmatrix},$$

where X_0, Y_0 are $r \times r$. Then, examining the upper-left $r \times r$ corner of the original equations, we find $X_0 Y_0 = I_r$. As these are square matrices, by assumption in degree d we may then derive that $Y_0 X_0 = I_r$ as well.

Now, multiply both sides of $XY = I_n$ on the left by the matrix $\begin{bmatrix} Y_0 & 0 \\ 0 & I_{n-r} \end{bmatrix}$. The result is then the set of degree-3 equations

$$\begin{bmatrix} Y_0 X_0 \\ X_1 \end{bmatrix} \begin{bmatrix} Y_0 & Y_1 \end{bmatrix} = \begin{bmatrix} Y_0 & 0 \\ 0 & I_{n-r} \end{bmatrix}.$$

Considering the upper-right $r \times (n-r)$ block of these equations, we find the equations $Y_0 X_0 Y_1 = 0$.

But now, from the equation $Y_0 X_0 = I_r$, we may right-multiply by Y_1 to get $Y_0 X_0 Y_1 = Y_1$. Combining with the equation at the end of the last paragraph, we then conclude $Y_1 = 0$.

Finally, consider the lower-right $(n-r) \times (n-r)$ part of the original equation $XY = I_n$, namely, $X_1 Y_1 = I_{n-r}$. We had already derived $Y_1 = 0$, which we can then left-multiply by X_1 to get $X_1 Y_1 = 0$. Considering any diagonal entry of these two equations, we then derive the contradiction $1 = 0$.

To see the NS certificate, we unwrap the above proof. First write $Y_0 X_0 - I_r$ as a linear combination of the equations $X_0 Y_0 - I_r$ with polynomial coefficients, in total degree d. Among our starting equations in the Rank Principle, we have $X_0 Y_1$ and $X_1 Y_1 - I_{n-r}$. Then the following linear combination has degree 2 more than $Y_0 X_0 - I_r$, and derives 1 in any of its diagonal entries:

$$-X_1 Y_0 \underline{X_0 Y_1} + X_1 (Y_0 X_0 - I_r) Y_1 + \underline{(X_1 Y_1 - I_{n-r})}.$$

□

Observation 3. *The $n \times n$ Inversion Principle has a proof of degree $2n + 2$.*

Proof. The idea is to use Laplace expansion. We spell out the details.

We start with $XY = I_n$, where X and Y are $n \times n$ matrices of variables. Left-multiply by Y to get $YXY = Y$, and then right multiply by $Adj(Y)$ (whose entries are the $(n-1) \times (n-1)$ cofactors of Y, hence have degree $n-1$) to get $YXYAdj(Y) = YAdj(Y)$. Now, by Laplace expansion, we have $YAdj(Y) \equiv \det(Y) I_n$, so we get $YX \det(Y) = \det(Y) I_n$.

Next, starting from $XY = I_n$ and expanding out the determinant term-by-term, we derive $\det(XY) = 1$. (Note that here, we are not simply applying the determinant to the matrix $XY - I$, as that would give us the value of the characteristic polynomial evaluated at 1. Instead, we repeatedly use that from $a - b = 0$ and $c - d = 0$ we can derive $ac - bd = 0$ as $\underline{(a-b)}c + b\underline{(c-d)}$. Similarly, we can derive $(a+c)-(b+d) = 0$ as $(a-b)+(c-d)$.) Now, since $\det(XY) \equiv \det(X)\det(Y)$ identically as polynomials, we have derived $\det(X)\det(Y) = 1$ in degree n.

Now, from $YX \det(Y) - \det(Y) I_n$ in the first paragraph, we multiply by $\det(X)$ to get $(YX - I_n)(\det(X)\det(Y))$. From $\det(X)\det(Y) - 1$ in the second paragraph, we multiply by $-(YX - I_n)$ and add to the preceding to get $YX - I_n$, all in degree at most $2n + 2$. □

5.2.3 Lower bound on the Rank Principle (and Inversion Principle) via reduction from PHP

Here we show that the Rank Principle (see Section 5.1.3) requires large PC degree, via a reduction to the Pigeonhole Principle. For the Pigeonhole principle, a tight PC degree lower bound is known:

Theorem 5.2.4 (Razborov [Raz98b])**.** *Any PC refutation of the Functional PHP_r^n requires degree $r/2+1$ over any field.*

We use this to show:

Theorem 5.2.5. *Let $n \in \mathbb{N}$, $n \geq 2$ and $1 \leq r < n$. $\mathbb{I}(r,n)$ (with or without the Boolean axioms) requires degree $r/2+1$ in PC over any field.*

Proof. We prove that PHP_r^n is $(1,2)$-reducible to $\mathbb{I}(r,n)$. First we consider the following degree 1 polynomials defining x and y variables of $\mathbb{I}(r,n)$ in terms of the p variables of PHP_r^n. variables

$$x_{i,k} = y_{i,k} = p_{i,k} \qquad \text{for } i \in [n], k \in [n-1].$$

Second we show a degree 2 PC proof of $\mathbb{I}(r,n)$ from the polynomials defining the PHP_r^n. From PHP axioms $p_{i,k}p_{k,j}$ for $i,j \in [n], i \neq j$, and summing over all $k \in [r]$, we get

$$\sum_{k\in[r]} p_{i,k}p_{k,j},$$

which are exactly the axioms of $\mathbb{I}(r,n)$ for $i \neq j, i,j \in [n]$, after the substitution of variables.

For a $i \in [n]$, take the boolean axioms written in the form $p_{i,k}p_{i,k} - p_{i,k}$ and sum them over $k \in [r]$:

$$\sum_{k\in[r]} p_{i,k}p_{i,k} - \sum_{k\in[r]} p_{i,k}$$

Summing this last polynomial with the PHP axiom $\sum_{k\in[r]} p_{i,k} - 1$ we get the polynomial

$$\sum_{k\in[r]} p_{i,k}p_{i,k} - 1,$$

which is the axiom of $\mathbb{I}(r,n)$ for $i = j$ after the substitution of the variables. The proof has degree 2. The result follows immediately from Lemma 5.1.1 and Theorem 5.2.4.

□

Corollary 6. *Any PC proof of $AB = I \vdash BA = I$, where A, B are square $n \times n$ $\{0,1\}$ matrices requires degree $n/2+1$.*

Proof. Follows immediately from Theorem 5.2.5 and Lemma 5.2.2. □

5.3 Upper bound for non-isomorphism of bounded-rank tensors

Theorem 5.3.1. *Over any algebraically closed field, there is a function $f(r) \leq 2^{O(r^2)}$, depending only on r, such that, given two non-isomorphic tensors M, M' of tensor rank $\leq r$, the Nullstellensatz degree of refuting isomorphism is at most $f(r)$.*

If working over a finite field $GF(q)$ and including the equations $x^q - x = 0$ for all variables x, then the PC degree is at most $12qr^2$.

Proof. The proof is based mainly on the so-called inheritance property of tensor rank.

Let $M = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i$ and let $M' = \sum_{i=1}^{r} u'_i \otimes v'_i \otimes w'_i$ be our two tensors of format $n_1 \times n_2 \times n_3$. Let $d_1 = \dim \mathrm{Span}\{u_1, u_2, \ldots, u_r, u'_1, u'_2, \ldots, u'_r\}$, d_2 similarly for the v's and d_3 for the w's. Choose a basis $e_1, e_2, \ldots, e_{n_1}$ for $\mathbb{F}^{n_1}$ such that $\mathrm{Span}\{e_1, \ldots, e_{d_1}\} = \mathrm{Span}\{u_1, \ldots, u_r, u'_1, \ldots, u'_r\}$. Let $f_1, \ldots, f_{n_2}$ be a similar basis for $\mathbb{F}^{n_2}$ (with the first d_2 vectors a basis for $\mathrm{Span}\{v_1, \ldots, v_r, v'_1, \ldots, v'_r\}$), and similarly $g_1, \ldots, g_{n_3}$. Changing everything in sight into the $e_\bullet \otimes f_\bullet \otimes g_\bullet$ basis, we find that M, M' are both supported in the upper-left $d_1 \times d_2 \times d_3$ sub-tensors, with all zeros outside of this. Call the corresponding $d_1 \times d_2 \times d_3$ tensors $\overline{M}, \overline{M}'$. Because all the entries outside this box are zero, it is not difficult to show that $M \cong M'$ iff $\overline{M} \cong \overline{M}'$ (the so-called "Inheritance Theorem,", see, e. g., [Lan12, §3.7.1]); note that isomorphism of $\overline{M}$ with $\overline{M}'$ is via the much smaller group $\mathrm{GL}_{d_1} \times \mathrm{GL}_{d_2} \times \mathrm{GL}_{d_3}$, rather than $\mathrm{GL}_n \times \mathrm{GL}_n \times \mathrm{GL}_n$ (the latter of which is used to determine isomorphism of M with M').

In this basis, isomorphism of $\overline{M}, \overline{M}'$ is solely determined by the upper-left $d_1 \times d_1$ sub-matrix of X, X', the upper-left $d_2 \times d_2$ submatrix of Y, Y', and the upper-left $d_3 \times d_3$ sub-matrix of Z, Z'. So we now only need to deal with equations in $d_1^2 + d_2^2 + d_3^2$ variables. Since each $d_i \leq 2r$, this is at most $12r^2$ variables.

Since we have $\leq 12r^2$ variables, $d_1 d_2 d_3$ cubic equations, and $6n^2$ quadratic equations ($XX' = I = X'X = YY' = \cdots$), over an algebraically closed field Sombra's Effective Nullstellensatz [Som99] implies that the Nullstellensatz degree of refuting our equations is then at most $4 \cdot 3^{\Theta(r^2)}$.

Over a finite field with the extra equations $x^q = x$, we may reduce degrees so that the degree of each variable is never more than q, the size of the field. In this case, the PC degree is at most q times the number of variables, i. e., at most $12qr^2$. □

Remark 2. *For fixed r, testing if an $n \times n \times n$ tensor has rank $\leq r$ can be done in polynomial time, as follows. This will show that the algorithm of Theorem 5.3.1 genuinely solves the decision problem, and not just a promise problem. Given an $n \times n \times n$ tensor T, consider its three $n \times n^2$ flattenings. Use Gaussian elimination to put each such flattening, separately, into reduced row echelon form. If any of these flattenings has rank $> r$, reject. Otherwise, we get from this a list of $3r$ vectors $u_1, \ldots, u_r, v_1, \ldots, v_r, w_1, \ldots, w_r$, such that T lives in the $r \times r \times r$-dimensional space $\mathrm{Span}\{u_1, \ldots, u_r\} \otimes \mathrm{Span}\{v_1, \ldots, v_r\} \otimes \mathrm{Span}\{w_1, \ldots, w_r\}$. Now in this space we can write down the Brent equations [Bre70] for T to have rank $\leq r$, which will be r^3 cubic equations in $3r^2$ variables (Brent's equations [Bre70, (5.06)] were specifically for the matrix multiplication tensor, but analogous equations are easily constructed for arbitrary tensors using the same idea). Since r is constant, these equations may be solved in polynomial time (here we assume that we are either working over a finite field, a finite-degree extension of the rationals—see, for example, Grigoriev [Gri13]—or in the BSS model over an arbitrary field).*

5.4 Lower bound on PC degree for Tensor Isomorphism from Graph Isomorphism

Definition 5.4.1. Given two graphs G, H with adjacency matrices A, B (resp.), the equations for Graph Isomorphism (the same as those used by Berkholz & Grohe [BG15, BG17]) are as follows. Let Z be an $n \times n$ matrix of variables z_{ij} (where the intended interpretation is that $z_{ij} = 1$ iff an isomorphism maps vertex $i \in V(G)$ to vertex $j \in V(H)$). We say that a partial map, which sends $(i, i') \mapsto (j, j')$ is a local isomorphism if (1) $i = i'$ iff $j = j'$ (it's a well-defined map) and (2) $(i, i') \in E(G) \Leftrightarrow (j, j') \in E(H)$. (One may also do Colored Graph Isomorphism and require that the colors match, $c(i) = c(j), c(i') =$

$c(j')$.) Then the equations are:

$z_{ij}^2 - z_{ij}$	$\forall i, j$	All variables $\{0,1\}$-valued
$1 - \sum_i z_{ij}$	$\forall j$	each $j \in V(H)$ is mapped to from exactly one vertex
$1 - \sum_j z_{ij}$	$\forall i$	each $i \in V(G)$ maps to exactly one vertex
$z_{ij} z_{i'j'}$		Whenever $(i,i') \mapsto (j,j')$ is not a local isomorphism.

In this section, we prove a lower bound on PC (and SoS) for TI, by reducing from GI and using the known lower bounds on GI [BG15, BG17]. Specifically, we show

Theorem 5.4.1. *Over any field, there are instances of* Tensor Isomorphism *of size* $O(n) \times O(n) \times O(n)$ *that require PC degree* $\Omega(n)$ *to refute. The same holds over the reals for SoS degree.*

Proof. Berkholz and Grohe [BG15, BG17] show the same statement for n-vertex graphs of bounded vertex degrees, with the same PC/SoS degree bound. In Proposition 5.4.2 we show that GI reduces to Monomial Code Equivalence by a (2,4)-many-one reduction that turns n-vertex, m-edge graphs into $m \times (3m+n)$ matrices. in Proposition 5.4.3 we show that Monomial Code Equivalence reduces to TI by a (2,4)-many-one reduction that turns $k \times N$ matrices into $(k+2N) \times N \times (1+2N)$ tensors. By Lemma 5.1.1, this completes the proof. □

To reduce from GI to TI we use the following intermediate problem. A matrix is *monomial* if it has exactly one nonzero entry in each row and column; equivalently, a monomial matrix is the product of a permutation matrix and an invertible diagonal matrix.

Definition 5.4.2. Monomial Code Equivalence is the problem: given two $k \times n$ matrices C, C', do there exist matrices X, Y such that $XCY^T = C'$ where X is invertible and Y is invertible and monomial? Given two such matrices C, C', the equations for Monomial Code Equivalence are as follows. There are $2(k^2+n^2)$ variables arranged into matrices X, X' (of size $k \times k$) and Y, Y' (of size $n \times n$). The equations are

$$XCY^T = C' \qquad XX' = X'X = \mathrm{Id} \qquad YY' = Y'Y = \mathrm{Id}$$

and

$$y_{ij}y_{ij'}(\forall i \forall j \neq j') \qquad y_{ij}y_{i'j}(\forall i \neq i', \forall j)$$
$$y'_{ij}y'_{ij'}(\forall i \forall j \neq j') \qquad y'_{ij}y'_{i'j}(\forall i \neq i', \forall j)$$

(Note: there are no equations forcing the variables to take on values in $\{0,1\}$.)

Proposition 5.4.2. *The reduction of Petrank & Roth [PR97] from* Graph Isomorphism *to* Linear Code Equivalence *over* $\mathbb{F}_2$ *in fact gives a (2,4)-many-one reduction from* Graph Isomorphism *to* Monomial Code Equivalence *(sic!) over any field.*

Proof. The reduction of Petrank & Roth is as follows: given a simple undirected graph G with n vertices and m edges, let $D(G)$ be its $m \times n$ incidence matrix: $D_{e,v} = 1$ iff $v \in e$ and is 0 otherwise, and let $M(G)$ be the $m \times (3m+n)$ matrix

$$M(G) = \left[\begin{array}{c|c|c|c} I_m & I_m & I_m & D(G) \end{array}\right].$$

Many-one reduction. It was previously shown (over $\mathbb{F}_2$ in [PR97] and over arbitrary fields in [Gro12, Lem. II.4]) that this gives a many-one reduction to Permutational Code Equivalence. Here we observe that the same reduction also gives a reduction to Monomial Code Equivalence. Thus, all that remains to show is that if $M(G)$ and $M(H)$ are monomially equivalent, then G must be isomorphic to H.

In fact, what was shown in [PR97] (over arbitrary fields in [Gro12]) is that, up to permutation and scaling of the rows, $M(G)$ is the unique generator matrix of its code satisfying the following properties: (1) $M(G)$ is $m \times (3m+n)$, (2) each row has Hamming weight ≤ 5, (3) any linear combination that includes two or more rows with nonzero coefficients has Hamming weight ≥ 6.

Now, suppose (X, Y) is a monomial equivalence of the codes $M(G), M(H)$. Then the rowspans of $M(G)Y^T$ and $M(H)$ are the same. Since Y is monomial, if we consider just the supports of the rows of $M(G)Y^T$, up to re-ordering the rows, by the preceding paragraph, those supports must be the same as the supports of the rows of $M(H)$. Thus X must also be monomial. Say $X = DP$ and $Y = EQ$ where D, E are diagonal and P, Q are permutation matrices. Then $PM(G)Q^T$ has the same support as $XM(G)Y^T = M(H)$, and since P and Q are permutation matrices and $M(G)$ and $M(H)$ have all entries in $\{0,1\}$, we must have $PM(G)Q^T = M(H)$. Thus $M(G)$ and $M(H)$ are in fact equivalent by a *permutation* matrix (in place of the monomial matrix Y). Thus, by the fact that $(G, H) \mapsto (M(G), M(H))$ was a reduction to Permutational Code Equivalence, we conclude that $G \cong H$.

Low-degree PC reduction. Let X, X', Y, Y' be the variable matrices in the equations for Monomial Code Equivalence of $M(G), M(H)$, and let Z be the variable matrix in the equations for Graph Isomorphism of G, H. Let $n = |V(G)|, m = |E(G)|$; so, X, X' are of size m, Y, Y' are of size $3m+n$, and Z is of size n.

Let $Z^{(2)}$ denote the $\binom{n}{2} \times \binom{n}{2}$ matrix whose $(\{i,i'\},\{j,j'\})$ entry is $z_{ij}z_{i'j'} + z_{ij'}z_{i'j}$. The idea is that if Z is a map on the vertices, then $Z^{(2)}$ is the corresponding map on the edges; the two terms come from the fact that the edge $\{i,i'\}$ can be mapped to the edge $\{j,j'\}$ either by $(i,i') \mapsto (j,j')$ or by $(i,i') \mapsto (j',j)$. Note that, since Z is a permutation matrix, at most one of these terms is nonzero, and thus $Z^{(2)}$ is also a $\{0,1\}$-matrix (in fact, a permutation matrix). Let $Z^{(2)}_E$ denote the $|E| \times |E|$ submatrix of $Z^{(2)}$ all of whose row indices are $\{i,i'\} \in E(G)$ and all of whose column indices are $\{j,j'\} \in E(H)$. Note also that $(Z^{(2)}_E)^T = (Z^T)^{(2)}_E$, so we use these notations interchangeably for convenience.

Now consider the following substitution:

$$\begin{array}{ll} X \mapsto (Z^{(2)}_E)^T & Y \mapsto (Z^T)^{(2)}_E \oplus (Z^T)^{(2)}_E \oplus (Z^T)^{(2)}_E \oplus (Z^T) \\ X' \mapsto Z^{(2)}_E & Y' \mapsto Z^{(2)}_E \oplus Z^{(2)}_E \oplus Z^{(2)}_E \oplus Z \end{array}$$

After making these substitutions in the equations for Monomial Code Equivalence of $M(G), M(H)$, we get the equations

$$(Z^{(2)}_E)^T Z^{(2)}_E = Z^{(2)}_E (Z^{(2)}_E)^T = \mathrm{Id}_m \qquad (Z^{(2)}_E)^T D(G) Z = D(H) \qquad ZZ^T = Z^T Z = \mathrm{Id}_n \tag{5.5}$$

along with equations saying that Z and $Z^{(2)}_E$ are monomial.

We now show how to derive these equations in low-degree PC from the GI equations.

The monomial equations for Z are part of the GI equations, so there is nothing to do for those.

The monomial equations for $Z^{(2)}_E$ are of the form $(z_{ij}z_{i'j'} + z_{ij'}z_{i'j})(z_{k\ell}z_{k'\ell'} + z_{k\ell'}z_{k'\ell})$ where either

(1) $\{i,i'\} = \{k,k'\}$ and $\{j,j'\} \neq \{\ell,\ell'\}$ or (2) vice versa. We expand out to get

$$z_{ij}z_{i'j'}z_{k\ell}z_{k'\ell'} + z_{ij}z_{i'j'}z_{k\ell'}z_{k'\ell} + z_{ij'}z_{i'j}z_{k\ell}z_{k'\ell'} + z_{ij'}z_{i'j}z_{k\ell'}z_{k'\ell}$$

We show how to get this equation in case (1); case (2) follows similarly, *mutatis mutandis*. In case (1), without loss of generality suppose that $i = k$, $i' = k'$, and $j \notin \{\ell,\ell'\}$. The first two terms are divisible by the GI equations $z_{ij}z_{i\ell}$ (since $i = k$ and $j \neq \ell$), the third term is divisible by $z_{i'j}z_{i'\ell'}$ (since $i' = k'$ and $j \neq \ell'$), and the last term is divisible by $z_{i'j}z_{i'\ell}$ similarly.

Next, the equations $ZZ^T = \mathrm{Id}_n$ are, expanded out,

$$\sum_j z_{ij}z_{ij} - 1 (\forall i) \qquad \sum_j z_{ij}z_{kj} (\forall i \neq k).$$

The first is gotten by linear combination from $1 - \sum_j z_{ij}$ and the Boolean axioms $z_{ij}^2 - z_{ij}$. The second is a linear combination of the monomial axioms $z_{ij}z_{kj}$ (part of the local non-isomorphism axioms). Similarly for $Z^TZ = \mathrm{Id}$, using $1 - \sum_i z_{ij}$ instead.

Next, we expand out the equations $Z_E^{(2)}(Z^T)_E^{(2)} = \mathrm{Id}_m$, to get[1]

$$\sum_{\{j,j'\}\in E(H)} (z_{ij}z_{i'j'} + z_{ij'}z_{i'j})(z_{kj}z_{k'j'} + z_{k'j}z_{kj'}) - \delta_{\{i,i'\},\{k,k'\}} (\forall \{i,i'\},\{k,k'\} \in E(G))$$

Thus, for $\{i,i'\} \neq \{k,k'\}$, we need to derive

$$\sum_{\{j,j'\}\in E(H)} \left(z_{ij}z_{i'j'}z_{kj}z_{k'j'} + z_{ij'}z_{i'j}z_{kj}z_{k'j'} + z_{ij}z_{i'j'}z_{k'j}z_{kj'} + z_{ij'}z_{i'j}z_{k'j}z_{kj'}\right).$$

Without loss of generality, suppose that $i \notin \{k,k'\}$. Then the first two terms of each summand are divisible by the GI equation $z_{ij}z_{kj}$, the third term is divisible by $z_{ij}z_{k'j}$, and the last term is divisible by $z_{ij'}z_{kj'}$. On the other hand, when $\{i,i'\} = \{k,k'\}$, we need to derive

$$-1 + \sum_{\{j,j'\}\in E(H)} \left(z_{ij}^2 z_{i'j'}^2 + 2z_{ij'}z_{i'j}z_{ij}z_{i'j'} + z_{ij'}^2 z_{i'j}^2\right).$$

The middle terms of each summand are divisible by the GI equations $z_{ij'}z_{ij}$. For the first and third terms, we can use the Boolean axioms to remove the squares, and thus we are left to derive

$$-1 + \sum_{\{j,j'\}\in E(H)} (z_{ij}z_{i'j'} + z_{ij'}z_{i'j}) \tag{5.6}$$

We derive this from the GI equations as follows. Consider $(\sum_j z_{ij} - 1)(\sum_{j'} z_{i'j'} - 1) + (\sum_j z_{ij} - 1) + (\sum_{j'} z_{i'j'} - 1)$ and break up the resulting sum according to whether $j = j'$, $\{j,j'\} \in E(H)$ or $\{j,j'\} \notin E(H)$. Then we get

$$\sum_j z_{ij}z_{i'j} + \sum_{j,j':\{j,j'\}\in E(H)} z_{ij}z_{i'j'} + \sum_{j\neq j'\{j,j'\}\notin E(H)} z_{ij}z_{i'j'} - 1$$

[1] We use the notation $\sum_{\{j,j'\}\in E(H)}$ to denote a sum in the index of summation takes on the value $e \in E(H)$ for each edge of H exactly once. Because our edges are undirected, we only use such sums when the summand expression is itself invariant under swapping the roles of j, j'. If so desired, one could equivalently say $\sum_{j<j',\{j,j'\}\in E(H)}$.

Every summand in the first sum is a monomial axiom since $i \neq i'$. Every summand in the third sum is a local non-isomorphism axiom, since $\{i, i'\} \in E(G)$ but $\{j, j'\} \notin E(H)$. Note that every edge $\{j, j'\}$ of $E(H)$ is represented twice in the middle sum: once as (j, j') and once as (j', j). Thus, the above simplifies to

$$\sum_{\{j,j'\} \in E(H)} (z_{ij} z_{i'j'} + z_{ij'} z_{i'j}) - 1,$$

which is what we sought to derive. The derivation of $(Z_E^{(2)})^T Z_E^{(2)} = \mathrm{Id}$ is similar.

Finally, we show how to derive the equation $(Z_E^{(2)})^T D(G) Z = D(H)$ from the equations $ZA(G) = A(H)Z$, where $A(G)$ denotes the adjacency matrix of G, with $A(G)_{ii'} = 1$ iff $\{i, i'\} \in E(G)$. Writing out the equations in indices, we need to derive

$$\sum_{\{i,i'\} \in E(G), k \in V(G)} \left(Z_E^{(2)}\right)_{\{i,i'\},\{j,j'\}} D(G)_{\{i,i'\},k} z_{k\ell} = D(H)_{\{j,j'\},\ell} (\forall \ell \in V(H), \forall \{j, j'\} \in E(H))$$

Using the fact that $D(G)_{\{i,i'\},k} = \delta_{ik} + \delta_{i'k}$ and the definition of $Z^{(2)}$, this is the same as

$$\sum_{\{i,i'\} \in E(G), k \in V(G)} (z_{ij} z_{i'j'} + z_{ij'} z_{i'j}) (\delta_{ik} + \delta_{i'k}) z_{k\ell} = \delta_{j\ell} + \delta_{j'\ell} (\forall \ell \in V(H), \forall \{j, j'\} \in E(H))$$

Thus we need to derive:

$$\sum_{\{i,i'\} \in E(G)} (z_{ij} z_{i'j'} + z_{ij'} z_{i'j}) (z_{i\ell} + z_{i'\ell}) = \begin{cases} 1 & \ell \in \{j, j'\} \\ 0 & \text{otherwise.} \end{cases}$$

Expanding out the summand, we find the four terms

$$z_{ij} z_{i'j'} z_{i\ell} + z_{ij} z_{i'j'} z_{i'\ell} + z_{ij'} z_{i'j} z_{i\ell} + z_{ij'} z_{i'j} z_{i'\ell}.$$

When $\ell \notin \{j, j'\}$, each of these terms is divisible by one of the monomial (local non-isomorphism) axioms, respectively: $z_{ij} z_{i\ell}$, $z_{i'j'} z_{i'\ell}$, $z_{ij'} z_{i\ell}$, and $z_{i'j} z_{i'\ell}$.

Finally, when $\ell \in \{j, j'\}$, without loss of generality suppose that $\ell = j$. Then the only terms that are not divisible by the monomial axioms as above are $z_{ij}^2 z_{i'j'} + z_{ij'} z_{i'j}^2$. Using the Boolean axioms we can easily convert each such summand to $z_{ij} z_{i'j'} + z_{ij'} z_{i'j}$. The derivation of the sum of these over all $\{i, i'\} \in E(G)$ is analogous, *mutatis mutandis*, to the derivation of (Equation (5.6)) above. This completes the proof. □

Proposition 5.4.3. *The many-one reduction from* Monomial Code Equivalence *to* Tensor Isomorphism *from Grochow & Qiao [GQ21a] is in fact a (2, 4)-many-one reduction.*

Proof. We recall the reduction, then prove that it is a low-degree PC reduction. Let M be a $k \times n$ matrix. We build a 3-tensor of size $(k + 2n) \times n \times (1 + 2n)$ as follows. The first frontal slice is $\begin{bmatrix} M \\ 0_{2n \times n} \end{bmatrix}$. The remaining $2n$ slices all have just a single nonzero entry, which serve to place a 2×2 identity matrix "behind and perpendicular" to M, one 2×2 matrix in each column. Let us index these slices by $[n] \times 2$. Then the (i, b) slice has a 1 in entry $(2(i-1) + b, i)$, for all $i \in [n], b \in [2]$. Let us call this tensor $r(M)$. Then the reduction maps M, M' to $r(M), r(M')$.

Let X, X', Y, Y', Z, Z' be the variable matrices for the TI equations for $r(M), r(M')$, and let A, B, A', B' be the variable matrices for Monomial Code Equivalence of M, M' (that is, $AMB^T = M'$, A is invertible, B is monomial and invertible). Consider the substitution:

$$X \mapsto A \oplus (B \otimes I_2) \qquad Y \mapsto B \qquad Z \mapsto 1 \oplus (B' \circ B') \otimes I_2$$

$$X' \mapsto A' \oplus (B' \otimes I_2) \qquad Y' \mapsto B' \qquad Z' \mapsto 1 \oplus (B \circ B) \otimes I_2.$$

As before, $B \circ B$ denotes the Hadamard or entry-wise product. Let us see what the TI equations become under this substitution. We get

$$AMB^T = M' \qquad AA' = A'A = \mathrm{Id} \qquad BB' = B'B = \mathrm{Id} \qquad (B' \circ B')(B \circ B) = (B \circ B)(B' \circ B') = \mathrm{Id}$$

Indeed, notice that the effect of the $B \otimes I_2$ in X and the B in Y is that the row and column locations of the 2×2 matrix gadgets get permuted in the same way, and the gadget get multiplied by the *square* of the nonzero entries of B. These are then multiplied by the $B' \circ B'$ in Z.

Now, we derive these equations from the equations for Monomial Code Equivalence. The first three are already present in the equations for Monomial Code Equivalence. The last one we expand out, to see that we need to derive:

$$\sum_j b_{ij}^2 (b'_{jk})^2 = \delta_{ik} (\forall i, k)$$

Now, for $i \neq k$, we may take the equation $\sum_j b_{ij} b'_{jk}$ and square it, to derive

$$\sum_{j \neq j'} b_{ij} b'_{jk} + b_{ij'} b'_{j'k} + \sum_j b_{ij}^2 b_{j'k}^2.$$

Each term in the first sum is divisible by one of the monomial axioms $b_{ij} b_{ij'}$ since $j \neq j'$, and the second sum is what we wanted to derive.

Finally, for $i = k$, we square the equation $\sum_j b_{ij} b'_{ji} - 1$ and add to it $2\left(\sum_j b_{ij} b'_{ji} - 1\right)$. We then proceed to cancel terms with the monomial axioms as above, and end up with $\sum_j b_{ij}^2 (b'_{ji})^2 - 1$, as desired. □

5.5 Lower bound on PC degree for Tensor Isomorphism from Random 3XOR

We get a lower bound on PC refutations for Tensor Isomorphism through the following series of low-degree PC many-one reductions (Definition 5.1.2):

$$\begin{aligned}
&\text{Random 3-XOR} \leq_m^{PC} \{\pm 1\}\text{-Monomial Equivalence of} && (5.7)\\
&\qquad \{\pm 1\}\text{-Multilinear Noncommutative Cubic Forms} && (5.8)\\
&\leq_m^{PC} \text{Monomial Equivalence of } \{\pm 1\} \text{ Noncommutative Cubic Forms} && (5.9)\\
&\leq_m^{PC} \text{Equivalence of } \{\pm 1\} \text{ Noncommutative Cubic Forms} && (5.10)\\
&\leq_m^{PC} \text{Tensor Isomorphism} && (5.11)
\end{aligned}$$

We then appeal to the following PC lower bound on Random 3-XOR:

Theorem 5.5.1 (Ben-Sasson & Impagliazzo [BI99, Thm. 3.3 & Lem. 4.7])**.** *Let $\mathbb{F}$ be any field of characteristic $\neq 2$. A random 3-XOR instance with clause density $\Delta = m/n$ requires PC degree $\Omega(n/\Delta^2)$ to refute, with probability $1 - o(1)$.*

This allows us to prove:

Theorem 5.5.2. *Over any field of characteristic $\neq 2$, there is a random distribution of instances of $n \times n \times n$ Tensor Isomorphism—which assigns nonzero probability to at least $2^{\Omega(\sqrt[4]{n}) \log n}$ different instances—whose associated equations require PC degree $\Omega(\sqrt[4]{n})$ to refute, with probability $1 - o(1)$.*

Note that such instances have $N = 6n^2$ variables, so this is really only an $\Omega(\sqrt[8]{N})$ lower bound relative to the number of variables.

In the following subsections we recall the definitions of the above problems and their associated systems of polynomial equations, and we give the reductions in the order listed above.

The first two reductions are gadget constructions of linear size; the proof of correctness for the first uses the fact that random hypergraphs have no automorphisms, while the second is fairly straightforward. (Equation (5.10)) uses a gadget from Grochow & Qiao [GQ21b], albeit for a new application, and shows that the reduction using this gadget also yields a low-degree PC reduction. (Equation (5.11)) is based on two lemmas, which show that the many-one reduction for this problem in fact also gives a low-degree PC reduction.

Remark 3. *Both of the latter two reductions have a quadratic size increase, so while we get a nearly-linear lower bound on PC degree for refutations of* Monomial Equivalence of Noncommutative Cubic Forms*, we only get a $\Omega(\sqrt{n})$ degree lower bound* Equivalence of Noncommutative Cubic Forms *and a $\Omega(\sqrt[4]{n})$ degree lower bound on* Tensor Isomorphism*. If the gadget sizes of these latter two reductions could be improved to linear, we would get a similarly near-linear lower bound (linear in the side length, still $\sqrt{N}$ relative to the number of variables) on PC refutations for* Tensor Isomorphism *as well. As many of the reductions in [FGS19, GQ21b] are of a similar flavor to the ones we consider here, we believe that they can all be proven in low-degree PC, so we expect the main obstacle to such an improvement is the size of the constructions themselves.*

5.5.1 From Random 3-XOR to $\{\pm 1\}$-multilinear noncommutative cubic forms

Definition 5.5.1. A random 3-XOR instance with n variables and m clauses is obtained by sampling m clauses independently and uniformly from the set of all $2\binom{n}{3}$ parity constraints on 3 variables. Each parity constraint is encoded by an equation of the form $x_i x_j x_k = \pm 1$, and the Boolean constraints are encoded by $x_i^2 = 1$.

By a $\{\pm 1\}$-monomial matrix, we mean a monomial matrix in which all nonzero entries are one of ± 1. $\{\pm 1\}$-Monomial Equivalence of Noncommutative Cubic Forms is the problem of deciding, given two noncommutative cubic forms f, f' in n variables $x_1, \ldots, x_n$ with all nonzero coefficients ± 1, whether there is a permutation $\pi \in S_n$ and signs $e_i \in \{\pm 1\}$ such that $f(e_1 x_{\pi(1)}, \ldots, e_2 x_{\pi(2)}, \ldots, e_n x_{\pi(n)}) = f'(\vec{x})$. Equivalently, if we represent a noncommutative cubic form f by the 3-way array T_{ijk} such that $f(\vec{y}) = \sum_{i,j,k \in [n]} T_{ijk} y_i y_j y_k$, the problem here asks whether there is a $\{\pm 1\}$-monomial matrix A such that $(A, A, A) \cdot T = T'$, that is, whether $T'_{i'j'k'} = \sum_{ijk} a_{ii'} a_{jj'} a_{kk'} T_{ijk}$ for all $i', j', k' \in [n]$.

Definition 5.5.2. We define the systems of equations associated to several variations of Equivalence of Noncommutative Cubic Forms.

1. Given two $n \times n \times n$ 3-way arrays T, T', the system of equations for Equivalence of Noncommutative Cubic Forms is the following system of equations in $2n^2$ variables. Let A, A' be $n \times n$ matrices of independent variables a_{ij}, a'_{ij}, respectively.

$$\begin{aligned} (A, A, A) \cdot T &= T' && (A \text{ is an equivalence}) \\ AA' &= A'A = \mathrm{Id} && (A \text{ is invertible with } A^{-1} = A') \end{aligned}$$

2. The system of equations for Monomial Equivalence of Noncommutative Cubic Forms includes the preceding equations, as well as:

$$\begin{aligned} a_{ij}a_{ij'} &= 0 \quad \forall i \forall j \neq j' && (\text{at most one nonzero per row}) \\ a_{ij}a_{i'j} &= 0 \quad \forall j \forall i \neq i' && (\text{at most one nonzero per column}) \end{aligned}$$

3. The system of equations for $\{\pm 1\}$-Monomial Equivalence of Noncommutative Cubic Forms includes all the preceding equations, as well as

$$a_{ij}(a_{ij} + 1)(a_{ij} - 1) = 0 \quad \forall i, j \in [n] \quad (\text{all entries in } \{0, \pm 1\})$$

4. A noncommutative cubic form $\sum_{ijk} T_{ijk} x_i x_j x_k$ is *multilinear* if all nonzero terms T_{ijk} have i, j, k distinct (that is, $|\{i, j, k\}| = 3$). The system of equations for $\{\pm 1\}$-Monomial Equivalence of Adjective Noncommutative Cubic Forms is the same as the above, with the restriction that T and T' both satisfy Adjective (e. g., multilinear, nonzero entries in $\{\pm 1\}$, etc.).

Theorem 5.5.4. *There is a linear-size (1,3)-reduction from* Random 3-XOR *instances on n variables with m clauses, where $10^4 n \leq m \leq \binom{n}{3}/10^{12}$, to* $\{\pm 1\}$-Monomial Equivalence of $\{\pm 1\}$ Multilinear Noncommutative Cubic Forms*, over any ring R of characteristic $\neq 2$.*

The reduction is always a (1,3)-reduction, but we only show the resulting system of equations for $\{\pm 1\}$-Monomial Equivalence of Noncommutative Cubic Forms is unsatisfiable with probability $1 - o(1)$ when the 3-XOR instance is chosen randomly with the parameters specified in the theorem. (It is possible that it is always unsatisfiable when the input 3-XOR instance is, but our proof does not answer this question.)

Proof idea. We build multilinear noncommutative cubic forms from the 3-XOR instance such that they are equivalent by a $\{\pm 1\}$ diagonal matrix iff the 3-XOR instance is satisfiable: an equation $x_i x_j x_k = \pm 1$ corresponds to setting $T_{ijk} = 1, T'_{ijk} = \pm 1$ in this construction. The noncommutative cubic forms are multilinear because the construction of the random 3XOR instance ensures that each XOR clause contains 3 distinct variables. In fact, the equations for $\{\pm 1\}$-diagonal equivalence of the correspondence noncommutative cubic forms will turn out to be identically the same as the equations for the 3-XOR instance.

Next, for random instances chosen with the stated parameters, the 3-way arrays T, T' are the adjacency hyper-matrices of a 3-uniform hypergraph that has no nontrivial automorphisms by [OWWZ14,

Lemma 6.9]; this is why we needed to restrict the parameter range for m as we did. Because the hypergraphs have no nontrivial automorphisms, any monomial equivalence of the corresponding cubic forms must in fact be diagonal, thus letting us further reduce to $\{\pm 1\}$-monomial equivalence. □

Proof. We are given a system of 3-XOR equations, which we'll denote $x_{i_\ell} x_{j_\ell} x_{k_\ell} = s_\ell$ for $\ell = 1, \ldots, m$, where $i_\ell \le j_\ell \le k_\ell \in [n]$ are indices of variables and $s_\ell \in \{\pm 1\}$ for all ℓ. It also includes the equations $x_i^2 = 1$ for all $i = 1, \ldots, n$.

Step 1: Reduce from random 3-XOR to $\{\pm 1\}$-diagonal equivalence of noncommutative cubic forms. From the above system of equations, we now construct two $n \times n \times n$ 3-way arrays T, T'. For the original equations $x_{i_\ell} x_{j_\ell} x_{k_\ell} = s_\ell$ ($\ell = 1, \ldots, m$), and for any $a_\ell \in \{\pm 1\}$ of our choice (we may set all $a_\ell = 1$ if we wish, but this additional flexibility may be useful in other settings) we set

$$T_{i_\ell, j_\ell, k_\ell} = a_\ell \text{ and } T'_{i_\ell, j_\ell, k_\ell} = s_\ell a_\ell.$$

All other entries of T and T' are set to zero.

We start with a warmup lemma, to see that this part of the construction already has a desirable property. By a "$\{\pm 1\}$ diagonal isomorphism" of two non-commutative cubic forms, we mean a diagonal matrix X whose diagonal entries are all one of ± 1 such that X gives an equivalence between T, T'.

Lemma 5.5.5. *Notation as in the paragraph above. There is a bijection between the solutions to the 3-XOR instance and the $\{\pm 1\}$ diagonal isomorphisms of the noncommutative cubic forms defined by T, T'.*

Proof. Suppose $\mathbf{x}$ is a solution to the 3-XOR instance. Let $X = \mathrm{diag}(x_1, \ldots, x_n)$ be the diagonal matrix with $\mathbf{x}$ on the diagonal. We claim that X is an equivalence between the noncommutative cubic forms represented by T, T', or the same, that (X, X, X) is an isomorphism of the tensors T, T'. Note that for any diagonal matrices X, Y, Z, we have $((X, Y, Z) \cdot T)_{ijk} = x_i y_j z_k T_{ijk}$. In particular, the action of diagonal matrices does not change which entries of T are zero or nonzero, it merely scales the nonzero entries. Since T, T' have the same support by construction, it is necessary and sufficient to handle the nonzero entries. By the construction above, there are precisely m such nonzero entries, one for each cubic equation in the 3-XOR instance. For each $\ell = 1, \ldots, m$, we have

$$\begin{aligned} ((X, X, X) \cdot T)_{i_\ell j_\ell k_\ell} &= x_{i_\ell} x_{j_\ell} x_{k_\ell} T_{i_\ell j_\ell k_\ell} \\ &= s_\ell T_{i_\ell j_\ell k_\ell} \\ &= T'_{i_\ell j_\ell k_\ell}. \end{aligned}$$

In the other direction, if $X = \mathrm{diag}(\mathbf{x})$ is a diagonal matrix whose diagonal entries are in $\{\pm 1\}$ giving an isomorphism of the noncommutative cubic forms, then we have

$$x_{i_\ell} x_{j_\ell} x_{k_\ell} = T_{i_\ell j_\ell k_\ell} T'_{i_\ell j_\ell k_\ell} = s_\ell$$

for $\ell = 1, \ldots, m$. (Here we have pulled $T_{i_\ell, j_\ell, k_\ell}$ across the equals sign because every term in the above equation is ± 1.) This concludes the proof of the lemma. □

We thus consider the equations corresponding to $\{\pm 1\}$-diagonal equivalence of T, T': there are n variables x_i ($i = 1, \ldots, n$). Let X denote the diagonal matrix with $\mathbf{x}$ on the diagonal. Then the

equations are

$$X^2 = \mathrm{Id} \qquad (X, X, X) \cdot T = T'. \tag{5.12}$$

By Lemma 5.5.5, we have that the original 3XOR instance is satisfiable iff (Equation (5.12)) is satisfiable. We claim furthermore that there is (1,3)-reduction from the 3XOR equations to this system of equations. In fact, as the proof of the preceding lemma shows, they are actually *the same set of equations*! So there is nothing more to show.

Step 2: Reduce from $\{\pm 1\}$-diagonal equivalence to $\{\pm 1\}$-monomial equivalence. We claim that there is a $(1, 3)$-reduction from (Equation (5.12)) to the the equations for $\{\pm 1\}$-monomial equivalence, see (Definition 5.5.2). The variable substitution is given by

$$a_{ij} = a'_{ij} \mapsto \begin{cases} 0 & i \neq j \\ x_i & i = j. \end{cases}$$

Under this substitution:

- The equivalence condition $(A, A, A) \cdot T = T'$ becomes exactly the original equivalence condition $(X, X, X) \cdot T = T'$.
- The invertibility equations $AA' = A'A = \mathrm{Id}$ become $XX = \mathrm{Id}$
- The row and column equations both become $0 = 0$, since at least one of the two a_{ij} variables occurring will not be on the diagonal, hence will become 0 after substitution.
- The equation $a_{ij}(a_{ij} + 1)(a_{ij} - 1) = 0$ becomes $x(x^2 - 1)$ for the appropriate variable $x \in \mathbf{x}$. This is derivable from the original equation $x^2 - 1$ by multiplication by x.

Lastly, we show that the system of equations in Definition 5.5.2(3) for $\{\pm 1\}$-monomial equivalence is satisfiable iff the original 3-XOR instance was. Since we showed above that that $\{\pm 1\}$-diagonal equivalence equations are satisfiable iff the original 3-XOR instance was, we show the equisolvability of (Equation (5.12)) and the equations of Definition 5.5.2(3).

Since diagonal matrices are monomial, any solution to (Equation (5.12)) is a solution to the equations of Definition 5.5.2(3).

Conversely, suppose the equations of Definition 5.5.2(3) are solvable. Then there is a $\{\pm 1\}$-monomial matrix X given an equivalence between T and T'; we may write $X = DP$ where D is diagonal and P is a permutation matrix. Now, as the original 3-XOR instance was chosen uniformly at random, the support of T (the positions of its nonzero entries) is precisely a uniformly random 3-uniform hypergraph G. As T, T' have the same support by construction, we find that P must be an automorphism of G. But by [OWWZ14, Lemma 6.9], uniformly random such hypergraphs have no nontrivial automorphisms with probability $1 - o(1)$. Thus $P = I$ and X must in fact be diagonal, hence a solution to (Equation (5.12)). □

Remark 6. *We may avoid the heavy hammer of [OWWZ14, Lemma 6.9] by "rigidifying" (in the sense of removing automorphisms) the system of 3-XOR equations before constructing the 3-way arrays as follows. The construction corresponds to a standard graph-theoretic gadget for removing automorphisms. Add new variables z and y_{ij} for $i = 1, \ldots, n$ and $j = 1, \ldots, n + i$, as well as the equations $x_i y_{ij} z = 1$ for all i, j, as well as $y_{ij}^2 = 1$ and $z^2 = 1$. The downside of this construction is that it quadratically increases*

the number of variables, which would result in a further quadratic loss in our lower bounds on Tensor Isomorphism.

5.5.2 From $\{\pm 1\}$-monomial equivalence to (unrestricted) monomial equivalence

Theorem 5.5.7. *There is a linear-size (2, 6)-many-one reduction from*

$\{\pm 1\}$-Monomial Equivalence of $\{\pm 1\}$ Multilinear Noncommutative Cubic Forms

to

Monomial Equivalence of $\{\pm 1\}$ Noncommutative Cubic Forms,

over any ring R of characteristic $\neq 2$ such that $\{\pm 1\}$ are the only square roots of 1.

Furthermore, the reduction r has the property that, given any two $\{\pm 1\}$ multilinear noncommutative cubic forms f, f', any monomial equivalence between $r(f)$ and $r(f')$ must have all its nonzero entries sixth roots of unity, and this can be derived by a degree-6 PC proof.

Remark 8. *We note the difference between a reduction to $\sqrt[6]{1}$-*Monomial Equivalence *and a reduction to* Monomial Equivalence *with the property stated in the theorem. In the former case, the problem being reduced to only accepts $\sqrt[6]{1}$-monomial matrices as solutions (and then the goal of the reduction is to introduce gadgets to get this down to $\{\pm 1\}$). In the latter case, the problem being reduced to allows arbitrary monomial matrices as solutions, but the gadgets enforce that, on the reduced instances, any such monomial matrix must in fact have its nonzero entries being sixth roots of unity.*

Proof. Let T be an $n \times n \times n$ 3-way array representing a multilinear noncommutative cubic form with all nonzero entries in ± 1. We extend T to $r(T)$ of size $2n \times 2n \times 2n$, by setting

$$\begin{aligned} r(T)_{ijk} &= T_{ijk} && i,j,k \in [n] \\ r(T)_{i,i,n+i} &= 1 && i \in [n] \\ r(T)_{n+i,n+i,n+i} &= 1 && i \in [n] \end{aligned}$$

and all other entries of $r(T)$ set to zero.

Many-one reduction. We first show that the map $(T, T') \mapsto (r(T), r(T'))$ is a many-one reduction. Suppose T, T' are $\{\pm 1\}$-monomially equivalent by a matrix X, where $X = DP$ with $D = \mathrm{diag}(x_1, \ldots, x_n)$ a diagonal matrix with $x_i \in \{\pm 1\}$ for all i, and P is a permutation matrix. Let π denote the permutation corresponding to P; that is, $P_{i,\pi(i)} = 1$ for all $i \in [n]$. Then we claim the $2n \times 2n$ matrix $X \oplus P = \begin{bmatrix} X & 0 \\ 0 & P \end{bmatrix}$ is a monomial equivalence of $r(T)$ with $r(T')$. Since $X \oplus P$ is block-diagonal, the upper-left X certainly sends the upper-left $n \times n \times n$ sub-array of $r(T)$ (which is just T) to that of $r(T')$ (which is just T'). So the only thing to check is what happens to the positions at indices greater than n.

Let $X' = X \oplus P$. We have

$$\begin{aligned} ((X', X', X') \cdot r(T))_{i,i,n+i} &= r(T)_{\pi(i),\pi(i),n+\pi(i)} (X'_{i,\pi(i)})^2 X'_{n+i,n+\pi(i)} \\ &= r(T)_{\pi(i),\pi(i),n+\pi(i)} (X_{i,\pi(i)})^2 P_{i,\pi(i)} \\ &= 1 = r(T')_{i,i,n+i}. \end{aligned}$$

Similarly, we have:

$$((X', X', X') \cdot r(T))_{n+i,n+i,n+i} = r(T)_{n+\pi(i),n+\pi(i),n+\pi(i)} P^3_{i,\pi(i)} = 1 = r(T')_{n+i,n+i,n+i}$$

Because X' is monomial, it is easy to see that the zeros of $r(T)$ are sent to zeros of $r(T')$. Thus X' is a monomial equivalence of $r(T)$ with $r(T')$.

Conversely, suppose $r(T)$ and $r(T')$ are equivalent by a monomial matrix $Y = DP$, with D diagonal and P a permutation matrix corresponding to permutation $\pi \in S_{2n}$. We will show that this implies that T and T' are equivalent by a $\{\pm 1\}$ monomial matrix. Since T is multilinear, we have $T_{i,i,i} = r(T)_{i,i,i} = 0$. Since $r(T)_{n+j,n+j,n+j} = 1$ for all $j \in [n]$, the permutation π cannot send any element $> n$ to any element $\leq n$. Thus P is block-diagonal, say $P = \begin{bmatrix} P_1 & 0_n \\ 0_n & P_2 \end{bmatrix}$. Let π_1 (resp., π_2) be the permutation of $[n]$ corresponding to P_1 (resp., P_2).

Next, we claim $P_1 = P_2$. By considering the positions at indices $(i, i, n+i)$, we have:

$$((P, P, P) \cdot r(T))_{i,i,n+i} = r(T)_{\pi_1(i),\pi_1(i),n+\pi_2(i)}$$

But the latter is equal to the corresponding position in $r(T')$, which is 1 iff $\pi_1(i) = \pi_2(i)$. Since this holds for all i, we have $\pi_1 = \pi_2$, and thus $P_1 = P_2$.

Finally, we *do not* claim that the diagonal entries y_i themselves must be in ± 1. Rather, we will show that they are all sixth roots of unity. Then cubing them will yield a new $n \times n$ matrix D' all of whose diagonal entries are ± 1 such that $D'P_1$ is a ± 1-monomial equivalence of T with T'.

From the positions $(n+i, n+i, n+i)$, we have

$$\begin{aligned} 1 &= r(T')_{n+\pi_1(i),n+\pi_1(i),n+\pi_1(i)} \\ &= ((Y, Y, Y) \cdot r(T))_{n+i,n+i,n+i} \\ &= y^3_{n+i}. \end{aligned}$$

But then, considering the positions $(i, i, n+i)$, we similarly get that $y_i^2 y_{n+i} = 1$. Cubing the latter equation, we get $y_i^6 y_{n+i}^3 = 1$. But as we already have $y_{n+i}^3 = 1$, this gives us $y_i^6 = 1$ by a degree-6 PC proof, as claimed in the "furthermore."

Now we use the fact that T, T' have all entries in $\{0, \pm 1\}$. Thus, each nonzero entry of $r(T)$ in the front-upper-left block (corresponding to T) gives us an equation of the form $y_i y_j y_k T_{ijk} = T'_{\pi_1(i),\pi_1(j),\pi_1(k)}$. Since the nonzero entries of T, T' are ± 1, this is thus an equation of the form $y_i y_j y_k = \pm 1$. If we cube both sides of this equation, we get $y_i^3 y_j^3 y_k^3 = \pm 1$. But since we established above that $y_i^6 = 1$ for all i, we have that $y_i^3 \in \{\pm 1\}$ for all i. Thus, defining $x_i := y_i^3$ for $i = 1, \ldots, n$, we have $x_i \in \{\pm 1\}$ and letting $D' = \mathrm{diag}(x_1, \ldots, x_n)$, we have $D'P_1$ is a $\{\pm 1\}$-monomial equivalence from T to T'.

Low-degree PC reduction. We claim that the system of equations for $\{\pm 1\}$ monomial equivalence of T and T' is (2,6)-reducible to the system of equations for monomial equivalence of $r(T)$ and $r(T')$. Let X, X' be the $n \times n$ variable matrices for the equations for for $\{\pm 1\}$-monomial equivalence of the original tensors T and T', and let Y, Y' be the $2n \times 2n$ matrices for the equations for monomial equivalence of

$r(T), r(T')$. The PC reduction is defined by the following substitution:

$$\begin{aligned} y_{ij} &\mapsto x_{ij} && i,j \in [n] \\ y_{n+i,n+j} &\mapsto x_{ij}^2 && i,j \in [n] \\ y_{i,n+j}, y_{n+i,j} &\mapsto 0 && i,j \in [n], \end{aligned}$$

and similarly for the y' variables being substituted by the x' variables. That is, we have

$$Y \mapsto \begin{bmatrix} X & 0_n \\ 0_n & X \circ X \end{bmatrix} \qquad Y' \mapsto \begin{bmatrix} X' & 0_n \\ 0_n & X' \circ X' \end{bmatrix},$$

where $X \circ X$ denotes the entrywise (aka Hadamard) product with itself, that is $(X \circ X)_{ij} = x_{ij}^2$. The reason to use $X \circ X$ here is that if X is $\{\pm 1\}$-valued and monomial, then $X \circ X$ is the permutation matrix with the same support as X; that is, this substitution is essentially the same as the one used in the proof above for the many-one reduction.

Now, taking advantage of the block structure in the substitution above and the block structure in $r(T), r(T')$, let us see what our equations become after substitution, and how to derive them from the equations for T, T'. This will complete the proof.

1. The set of equations $(Y, Y, Y) \cdot r(T) = r(T')$ becomes the set of equations $(X, X, X) \cdot T = T'$ (by examining the front-upper-left corner), as well as the equations

$$\sum_{i,j,k \in [2n]} y_{ii'} y_{jj'} y_{k,k'} r(T)_{ijk} = \begin{cases} 1 & i' = j' = k' - n \text{ or } i' = j' = k' > n \\ 0 & \text{otherwise.} \end{cases}$$

   We deal with the three cases ($i' = j' = k' - n$, $i' = j' = k' > n$, or neither of these) separately.

   (a) Suppose $i' = j' = k' - n$. In this case, $y_{ii'}$ is only nonzero for $i \in [n]$, and similarly for $y_{jj'}$, while $y_{kk'}$ is only nonzero for $k > n$. Thus the substituted equation becomes

$$\sum_{i,j,k \in [n]} y_{ii'} y_{ji'} y_{n+k,n+i'} r(T)_{i,j,n+k} = \sum_{i,j,k \in [n]} x_{ii'} x_{ji'} x_{k,i'}^2 r(T)_{i,j,n+k} = 1$$

   Now, the only positions in $r(T)$ of the form $(i, j, n+k)$ with $i, j, k \in [n]$ that are nonzero are those of the form $(i, i, n+i)$, so the preceding equation simplifies further to

$$\sum_{i \in [n]} x_{ii'} x_{ii'} x_{ii'}^2 = 1$$

   i.e.,

$$\sum_{i \in [n]} x_{ii'}^4 = 1. \tag{5.13}$$

   We will now show how to derive (Equation (5.13)) from the equations for $\{\pm 1\}$-monomial equivalence of for T, T' (Crefdef:equations for equivalence). From the $\{0, \pm 1\}$ equation in Definition 5.5.2(3), if we multiply by $x_{ii'}$, we get

$$x_{ii'}^2 (x_{ii'}^2 - 1), \tag{5.14}$$

i.e., the usual Boolean equation but for $x_{ii'}^2$ rather than $x_{ii'}$ itself. Next, from $x_{ii'}x_{i''i'}$ with $i \neq i''$, we may square this to get

$$x_{ii'}^2 x_{i''i'}^2. \tag{5.15}$$

and we similarly get $(x'_{i'i})^2(x'_{i'i''})^2$ when $i \neq i''$.

Lastly, from the equation $XX' = \mathrm{Id}$ and multiplying by $\sum_{i\in[n]} x_{ii'}x'_{i'i} + 1$, we obtain

$$(\sum_{i\in[n]} x_{ii'}x'_{i'i}+1)(\sum_{i\in[n]} x_{ii'}x'_{i'i}-1) = \sum_{i\in[n]} x_{ii'}^2 x_{i'i}^2 + \sum_{i,j\in[n]i\neq j} x_{ii'}x'_{i'i}x_{ji'}x'_{i'j} - 1 = \sum_{i\in[n]} x_{ii'}^2 x_{i'i}^2 - 1, \tag{5.16}$$

where we observed that from the axioms that $x_{ii'}x_{ji'} = 0$ for $i \neq j$ we may derive in degree 4 that the middle term $\sum_{i,j\in[n]i\neq j} x_{ii'}x_{ji'}x'_{i'i}x'_{i'j} = 0$.

Now, (Equation (5.14))–(Equation (5.16)) are a degree-2 substitution instance of the equations in Lemma 5.5.9 with $c = 2, d = 1$. Thus, by Lemma 5.5.9, we can derive (Equation (5.13)) from these in degree 6.

(b) Suppose $i' = j' = k' > n$. In this case, the substitution makes all of $y_{ii'}, y_{jj'}, y_{kk'}$ equal to zero unless $i, j, k > n$. Thus we may write the equation, after substitution, as

$$\begin{aligned}\sum_{i,j,k\in[n]} y_{n+i,i'}y_{n+j,i}y_{n+k,i}r(T)_{n+i,n+j,n+k} &= \sum_{i,j,k\in[n]} x_{i,i'-n}^2 x_{j,i'-n}^2 x_{k,i'-n}^2 r(T)_{n+i,n+j,n+k} \\ &= r(T')_{i',i',i'} = 1.\end{aligned}$$

However, because the only entries $r(T)_{n+i,n+j,n+k}$ that are nonzero are those in which $i = j = k$, this simplifies further to:

$$\sum_{i\in[n]} x_{i,i'-n}^6 = 1.$$

This is a degree-2 substitution instance of Lemma 5.5.9 with $c = 3, d = 1$, so it can be derived in degree 6 from the equations derived in part (a).

(c) Suppose neither of the previous two cases hold. The derivation will depend on which of i', j', k' lie in $[n]$ versus $\{n+1, \ldots, 2n\}$.

i. When all are in $[n]$, we are in the front-upper-left corner of the tensor, and we exactly get the equations $(X, X, X) \cdot T = T'$.

ii. When all three of i', j', k' are $> n$, the only nonzero entries of $r(T)$ are of the form $r(T)_{n+i,n+i,n+i}$, so the equation becomes

$$\sum_{i\in[n]} x_{i,i'-n}^2 x_{i,j'-n}^2 x_{i,k'-n}^2 = 0.$$

Since we have assumed $|\{i', j', k'\}| > 1$, there are at least two distinct indices among them, and thus each term in this sum is a multiple of one of our $x_{ij}x_{ij'}$ axioms with $j \neq j'$.

iii. Next, suppose instead that $i', j' \in [n], k' > n$. In this case, the only nonzero entries of Y

after substitution are those with $i, j \in [n], k > n$. Thus the equation becomes

$$\sum_{i,j,k \in [n]} x_{ii'} x_{jj'} x^2_{k,k'-n} r(T)_{i,j,n+k} = 0$$

However, the only nonzero entries of $r(T)$ in which the first two coordinates are $\leq n$ and the third is $n+k$ are those of the form $i = j = k$, so the preceding becomes

$$\sum_{i \in [n]} x_{ii'} x_{ij'} x^2_{ik'-n} = 0.$$

Since we do not have $i' = j' = k' - n$ (as that was covered in a previous case), at least two of the column indices differ, and thus each term of this sum is divisible by one of the axioms of the form $x_{ij} x_{ij'}$ with $j \neq j'$.

iv. In all other cases, the corresponding entries of $r(T)$ are all zero, so the equation reduces to $0 = 0$.

2. The equations $YY' = Y'Y = \mathrm{Id}$ become $XX' = X'X = \mathrm{Id}$ and $(X \circ X)(X' \circ X') = (X' \circ X')(X \circ X) = \mathrm{Id}$. The first of these is one of our original equations, so it remains to derive the second. We show how to derive $(X \circ X)(X' \circ X') = \mathrm{Id}$; the other is similar. For clarity, let us write it out using indices:

$$\sum_j x^2_{ij} (x'_{jk})^2 - \delta_{ik} = 0 \qquad \forall i, k \in [n] \tag{5.17}$$

Starting from the equation $\sum_j x_{ij} x'_{jk} - \delta_{ik} = 0$, we multiply by $\sum_j x_{ij} x'_{jk}$, to get

$$\sum_j x^2_{ij} (x'_{jk})^2 + \sum_{j \neq j'} x_{ij} x'_{jk} x_{ij'} x_{j'k} - \delta_{ik} \sum_j x_{ij} x'_{jk}.$$

Note that every term in the middle summation here is divisible by some $x_{ij} x_{ij'}$ with $j \neq j'$, which is one of our equations, so we may cancel off those terms using those equations in degree 4. If $i \neq k$, then we are done. If $i = k$, then we add in our equation $\sum_j x_{ij} x'_{jk} - 1$ to get Item 2.

3. The equations $y_{ij} y_{ij'} = 0$ for $j \neq j'$ become 0 after substitution unless i, j, j' are either all in $[n]$ or all in $\{n+1, \ldots, 2n\}$. In the former case, the substituted equation is $x_{ij} x_{ij'} = 0$, which is already one of the original equations. In the latter case, the equation becomes $x^2_{ij} x^2_{ij'} = 0$; but this is easily derivable from $x_{ij} x_{ij'}$ by multiplying it by itself (degree 4). The equations saying there is at most one entry per column of Y are derived from those for X similarly.

This covers all the equations for monomial equivalence of $r(T), r(T')$, and thus we are done. □

Lemma 5.5.9. *For any integers $d \geq 1, c \geq 1$, from the equations*

$$x_i (x_i^d - 1) (\forall i) \qquad x_i x_j (\forall i \neq j) \qquad \sum_{i=1}^{n} x_i y_i - 1$$

there is a degree-$\max\{d+2, cd\}$ *PC derivation (over any ring R) of*

$$\sum_{i\in[n]} x_i^{cd} - 1$$

Although in the proof above we only used the $d = 1$ and $c = 2, 3$, we will later have occasion to use this lemma with larger values of d and c, which is why we phrase it in this level of generality.

Proof. First we show it for $c = 1$, then derive the general case from that.

Let $S = \sum_{i\in[n]} x_i^d$, $D = \sum_{i\in[n]} x_i y_i$. Our first goal is to derive $S - 1$. For each $i = 1, \ldots, n$, we can derive $x_i y_i(S - 1)$ in degree $d + 2$ as follows:

$$\begin{aligned} x_i y_i(S-1) &= x_i^{d+1} y_i + y_i \sum_{j\neq i} x_i x_j^d - x_i y_i \\ &= y_i(x_i^{d+1} - x_i) + y_i \sum_{j\neq i} x_i x_j^d = \underline{x_i(x_i^d - 1)} y_i + y_i \sum_{j\neq i} \underline{x_i x_j} x_j^{d-1}, \end{aligned}$$

where we have underlined the use of the axioms.

Summing up the preceding for all i, we derive $DS - D$ in degree $d + 2$. Finally, we multiply the starting equation $D - 1$ by S to get $SD - S$, also in degree $d + 2$. Then we have

$$(DS - D) - (SD - S) + (D - 1) = S - 1 = \sum_i x_i^d - 1,$$

as desired.

For $c > 1$, we then sum the preceding with $\sum_{i\in[n]} (x_i^{(c-1)d-1} + x_i^{(c-2)d-1} + \cdots + x_i^{d-1})\underline{(x_i^{d+1} - x_i)} = \sum_{i\in[n]} x_i^{cd} - x_i^d$, which has degree cd. □

5.5.3 From monomial equivalence to general equivalence of noncommutative cubic forms

Theorem 5.5.10. *There is a quadratic-size many-one reduction from*

Monomial Equivalence of Noncommutative Cubic Forms

to

Equivalence of Noncommutative Cubic Forms,

over any field.

If furthermore the input cubic forms f, f' have the property that any monomial equivalence between them must have its nonzero scalars being d-th roots of unity, and the latter can be derived by PC in degree $d + 1$, then the reduction above is a $(d, 2d)$-many-one reduction.

Proof. Let f be a noncommutative cubic form in variables $u_1, \ldots, u_n$. Then $r(f)$ will be a new noncommutative cubic form, in $n + 2n(n + 1)$ variables $u_1, \ldots, u_n, v_{11}, v_{12}, \ldots, v_{n,n+1}, w_{11}, w_{12}, \ldots, w_{n,n+1}$, which is $r(f) = f + \sum_{i\in[n],j\in[n+1]} u_i v_{ij} w_{ij}$. In terms of the underlying three-way arrays, if we have $f = \sum_{i,j,k\in[n]} T_{ijk} u_i u_j u_k$, then we use $r(T)$ to denote the array underlying $r(f)$, which can be described as follows. The 3-way array $r(T)$ will have size $N \times N \times N$ where $N = n + 2n(n + 1)$. Let T_i denote

the i-th frontal slice of T_i, that is, T_i is the matrix such that $(T_i)_{jk} = T_{ijk}$. For $i = 1, \ldots, n$, the frontal slices of $r(T)$ will be defined as:

$$r(T)_i = \left(\begin{array}{c|c|c} T_i & & \\ \hline & \begin{array}{ccccc} 0_{n+1} & & & & \\ & 0_{n+1} & & & \\ & & \ddots & & \\ & & & 0_{n+1} & \\ & & & & \ddots \\ & & & & & 0_{n+1} \end{array} & \begin{array}{ccccc} 0_{n+1} & & & & \\ & 0_{n+1} & & & \\ & & \ddots & & \\ & & & I_{n+1} & \\ & & & & \ddots \\ & & & & & 0_{n+1} \end{array} \\ \hline & \begin{array}{ccccc} 0_{n+1} & & & & \\ & 0_{n+1} & & & \\ & & \ddots & & \\ & & & 0_{n+1} & \\ & & & & \ddots \\ & & & & & 0_{n+1} \end{array} & \begin{array}{ccccc} 0_{n+1} & & & & \\ & 0_{n+1} & & & \\ & & \ddots & & \\ & & & 0_{n+1} & \\ & & & & \ddots \\ & & & & & 0_{n+1} \end{array} \end{array}\right),$$

where the I_{n+1} occurs in the i-th $(n+1) \times (n+1)$ block of its region. That is, the lower-right $2n(n+1) \times 2n(n+1)$ sub-matrix is the Kronecker product $E_{i,n+i} \otimes I_{n+1}$, where $E_{i,n+i}$ is the $2n \times 2n$ matrix with a 1 in position $(i, n+i)$ and zeros everywhere else. For the slices $i = n+1, \ldots, N$ we will have $r(T)_i = 0$.

Our main claim is that the map $(T, T') \mapsto (r(T), r(T)')$ is the reduction claimed in the theorem.

Many-one reduction. Suppose $X \cdot f = f'$ with X monomial. Write $X = PD$ with D diagonal and P a permutation matrix corresponding to the permutation $\pi \in S_n$. Then we claim that

$$Y = X \oplus ((PD^{-1}) \otimes I_{n+1}) \oplus (P \otimes I_{n+1})$$

is an equivalence between $r(f)$ and $r(f')$, where here we assume our variables are ordered as above. For we have

$$\begin{aligned} Y \cdot r(f) &= \sum_{ijk \in [n]} T_{ijk}(Yu_i)(Yu_j)(Yu_k) + \sum_{i \in [n], j \in [n+1]} (Yu_i)(Yv_{ij})(Yw_{ij}) \\ &= \sum_{ijk \in [n]} T_{ijk}(Xu_i)(Xu_j)(Xu_k) + \sum_{i \in [n], j \in [n+1]} (Xu_i)(PD^{-1}v_{ij})(Pw_{ij}) \\ &= X \cdot f + \sum_{i \in [n], j \in [n+1]} D_{ii} u_{\pi(i)} (D_{ii}^{-1} v_{\pi(i),j}) w_{\pi(i),j} \\ &= f' + \sum_{i \in [n], j \in [n+1]} u_{\pi(i)} v_{\pi(i),j} w_{\pi(i),j} \\ &= r(f'). \end{aligned}$$

The final inequality here follows from the fact that π is a permutation, so the final sum includes all terms of the form $u_i v_{ij} w_{ij}$, just listed in a different order than originally.

Conversely, suppose $Y \cdot r(f) = r(f')$ for an arbitrary invertible $N \times N$ matrix Y. To find an equivalence between f and f', here we find it more useful to take the viewpoint of the 3-way arrays $r(T)$ and $r(T')$ corresponding to $r(f)$ and $r(f')$, respectively.

The way Y acts on the 3-way array $r(T)$ is to first take linear combinations of the frontal slices, say by replacing the i-th slice with $\sum_{j\in[N]} Y_{ij} r(T)_j$ (corresponding to the action of Y on the third variable in each monomial), and then to take each slice S and replace it by YSY^t (the left multiplication corresponds to the action on the first variable in each monomial, and the right multiplication corresponds to the action on the second variable in each monomial). As this latter transformation preserves the rank of each slice, we will use the ranks of linear combinations of the slices to reason about properties of Y.

Claim 1: Y is a block-diagonal sum of an $n \times n$ matrix X and a $2n(n+1) \times 2n(n+1)$ matrix.

Proof of claim 1. First we show that Y is block-triangular. To see this, note that since the last $2n(n+1)$ slices are zero, the action of Y by taking linear combinations of slices cannot send any of the first n slices to the last $2n(n+1)$ slices. That is, Y has the form $Y = \begin{bmatrix} X & Z \\ 0 & W \end{bmatrix}$ where X is $n \times n$ and W is $2n(n+1) \times 2n(n+1)$. It remains to show that Z must be zero.

Since Y is block-diagonal and invertible, we have that X and W are each invertible.

Let R be the tensor gotten from $r(T)$ by having Y act by taking linear combinations of the slices. That is, the i-th frontal slices of R is $R_i = \sum_{j\in[N]} Y_{ij} r(T)_j$. Since each slice $r(T)_i$ has its support in the upper-left $n \times n$ sub-matrix and the middle-right $n(n+1) \times n(n+1)$ sub-matrix, so does each slice R_i. Write

$$R_i = \begin{bmatrix} R_i^{(1,1)} & 0 & 0 \\ 0 & 0 & R_i^{(2,2)} \\ 0 & 0_{n(n+1)} & 0 \end{bmatrix},$$

where $R_i^{(1,1)}$ is $n \times n$ and $R_i^{(2,2)}$ is $n(n+1) \times n(n+1)$.

Now consider the action of Y that sends R_i to $YR_iY^t = r(T')_i$. We now break up Y further into blocks commensurate with how we wrote R_i above; write

$$Y = \begin{bmatrix} X & A & B \\ 0 & C & D \\ 0 & E & F \end{bmatrix} \qquad Z = \begin{bmatrix} A & B \end{bmatrix} \qquad W = \begin{bmatrix} C & D \\ E & F \end{bmatrix},$$

where A, B are $n \times n(n+1)$, and C, D, E, F are each $n(n+1) \times n(n+1)$. Then we have:

$$\begin{aligned} YR_iY^t &= \begin{bmatrix} X & A & B \\ 0 & C & D \\ 0 & E & F \end{bmatrix} \begin{bmatrix} R_i^{(1,1)} & 0 & 0 \\ 0 & 0 & R_i^{(2,2)} \\ 0 & 0_{n(n+1)} & 0 \end{bmatrix} \begin{bmatrix} X^t & 0 & 0 \\ A^t & C^t & E^t \\ B^t & D^t & F^t \end{bmatrix} \\ &= \begin{bmatrix} XR_i^{(1,1)} & 0 & AR_i^{(2,2)} \\ 0 & 0 & CR_i^{(2,2)} \\ 0 & 0 & ER_i^{(2,2)} \end{bmatrix} \begin{bmatrix} X^t & 0 & 0 \\ A^t & C^t & E^t \\ B^t & D^t & F^t \end{bmatrix} \\ &= \begin{bmatrix} XR_i^{(1,1)}X^t + AR_i^{(2,2)}B^t & AR_i^{(2,2)}D^t & AR_i^{(2,2)}F^t \\ CR_i^{(2,2)}B^t & * & * \\ ER_i^{(2,2)}B^t & * & * \end{bmatrix}, \end{aligned}$$

where we have put $*$'s in positions we won't need in the argument.

Next, since each of the first n slices of $r(T')$ must be of this form, and those slices have zeros in each block except the $(1,1)$ and $(2,3)$ blocks, by considering the blocks $(1,2),(1,3),(2,1),(3,1)$ we must have

$$AR_i^{(2,2)}D^t = 0 \qquad AR_i^{(2,2)}F^t = 0 \qquad CR_i^{(2,2)}B^t = 0 \qquad ER_i^{(2,2)}B^t = 0.$$

For reasons that will become clear below, we combine these into the two equations

$$AR_i^{(2,2)}\begin{bmatrix} D^t & F^t \end{bmatrix} = 0 \qquad \begin{bmatrix} C \\ E \end{bmatrix} R_i^{(2,2)}B^t = 0.$$

Note that the $n(n+1) \times 2n(n+1)$ matrices $\begin{bmatrix} D^t & F^t \end{bmatrix}$ and $\begin{bmatrix} C^t & E^t \end{bmatrix}$ must both be full rank, since otherwise $W = \begin{bmatrix} C & D \\ E & F \end{bmatrix}$ would not be invertible.

The sum of the (2,3) blocks (of size $n(n+1) \times n(n+1)$) of the first n slices of $r(T)$ is precisely the identity matrix $I_{n(n+1)}$. Thus, the linear span of these blocks contains an invertible matrix in it. Since Y is invertible, that linear span is the same as the linear span of the blocks $\{R_i^{(2,2)} : i \in [n]\}$. Thus the latter contains a full-rank matrix, say $\sum_{i=1}^n \alpha_i R_i^{(2,2)}$. But since we have $AR_i^{(2,2)}\begin{bmatrix} D^t & F^t \end{bmatrix} = 0$ for all i, we may left multiply by A and right-multiply by $\begin{bmatrix} D^t & F^t \end{bmatrix}$ to get $A\left(\sum_{i=1}^n \alpha_i R_i^{(2,2}\right)\begin{bmatrix} D^t & F^t \end{bmatrix} = \sum_{i=1}^n \alpha_i AR_i^{(2,2)}\begin{bmatrix} D^t & F^t \end{bmatrix} = 0$. But now we have that $\sum \alpha_i R_i^{(2,2)}$ is invertible, and $\begin{bmatrix} D^t & F^t \end{bmatrix}$ has full rank $n(n+1)$, so their product also has full rank $n(n+1)$. But then we have that A times a full rank matrix is equal to 0, hence A must be zero. The same argument, *mutatis mutandis*, using the equation $\begin{bmatrix} C \\ E \end{bmatrix} R_i^{(2,2)}B^t = 0$, gives us that $B = 0$. Hence Y is block-diagonal as claimed. □

Next, we use properties of the ranks of the slices coming from the I_{n+1} gadgets to show that X must in fact be monomial.

Claim 2: $Y = \begin{bmatrix} X & 0 \\ 0 & W \end{bmatrix}$ where X is monomial.

Proof. In both $r(T)$ and $r(T')$, any linear combination consisting of k of the first n slices (with nonzero coefficients) has rank in the range $[k(n+1), k(n+1)+n]$, for any $k = 0, \ldots, n$. The lower bound can be seen by noting that any such linear combination is block-diagonal with k copies of I_{n+1} on the block diagonal of the $(2,3)$ block. The upper bound comes from the fact that these are the only nonzero blocks in the lower-right $2n(n+1) \times 2n(n+1)$ sub-matrix, and the only other nonzero entries are in the $n \times n$ upper-left sub-matrix, which has rank at most n because of its size.

Using notation from the proof of the preceding claim, since $YR_iY^t = r(T')_i$, and the latter has rank in the range $[n+1, 2n+1]$, R_i must also have rank in the same range. But this is only possible if R_i is a linear combination of precisely one of the first n slices of $r(T)$. Thus, X is monomial. □

From claim 2, we thus have that there is a permutation $\pi \in S_n$ and nonzero scalars $d_1, \ldots, d_n$ such that $R_i = d_i r(T)_{\pi(i)}$ for all $i = 1, \ldots, n$, where $X = DP$ with D the diagonal matrix with diagonal entries d_i and P the permutation matrix corresponding to π. Finally, in the proof of claim 1, we saw that the upper-left block of YR_iY^t was $XR_i^{(1,1)}X^t + AR_i^{(2,2)}B^t$, and then learned that $A = B = 0$.

Putting these together, and recalling that the upper-left block of $r(T)_i$ is T_i, we thus get

$$(DP)d_iT_{\pi(i)}(DP)^t = T'_i$$

for all i. In other words, X is a monomial equivalence from T to T' (hence, from f to f'). This completes the proof that the construction gives a many-one reduction.

Low-degree PC reduction. To prove the "furthermore", suppose that the pair of cubic forms f, f' has the property that any monomial equivalence between them must have its nonzero entries being d-th roots of unity, for some $d \geq 1$, and that this can be derived—more specifically, the equations $y_{ij}^{d+1} - y_{ij}$ and similarly for y'_{ij}—in degree $d+1$.

Let Y, Y' be the variable matrices for (general) equivalence of $r(f), r(f')$; let X, X' be the variable matrices for monomial equivalence of f, f'. Consider the substitution

$$Y \mapsto \begin{bmatrix} X & 0 & \\ 0 & X^{\circ(d-1)} \otimes I_{n+1} & \\ 0 & 0 & X^{\circ d} \otimes I_{n+1} \end{bmatrix} \qquad Y' \mapsto \begin{bmatrix} X' & 0 & \\ 0 & (X')^{\circ(d-1)} \otimes I_{n-1} & \\ 0 & 0 & (X')^{\circ d} \otimes I_{n+1} \end{bmatrix}, \tag{5.18}$$

where $X^{\circ(d-1)}$ denotes the $(d-1)$-fold Hadamard product $X \circ X \circ \cdots \circ X$, namely, $(X^{\circ(d-1)})_{ij} = x_{ij}^{d-1}$. We will show that the equations for equivalence of $r(f), r(f')$, after this substitution, can be derived from the equations for monomial equivalence of f, f' in low-degree PC.

(Note that the substitutions above correspond precisely to the forward direction of the many-one reduction, in which $X \oplus (D^{-1}P \otimes I_{n+1}) \oplus (P \otimes I_{n+1})$ served as an equivalence. For, once we have $x_{ij}^{d+1} - x_{ij}$, we have $X^{\circ(d-1)} = D^{d-1}P = D^{-1}P$, and $X^{\circ d} = D^d P = P$.)

Recall that these equations are $Y \cdot r(f) = r(f')$ and $YY' = Y'Y = \mathrm{Id}$. The latter equations are easier to handle so we begin with those. They become $X^{\circ c}(X')^{\circ c} = (X')^{\circ c}X^{\circ c} = \mathrm{Id}$ for $c \in \{1, d-1, d\}$. For $c = 1$, these are some of our starting equations. For $c > 1$, this is similar to the argument in Theorem 5.5.7, iterated, resulting in a proof of degree $2c$ for any c—in this case, $2d$.

Now to the equation(s) $Y \cdot r(f) = r(f')$. After substitution, these become

$$\sum_{i,j,k\in[n]} T_{ijk}(Xu_i)(Xu_j)(Xu_k) + \sum_{i\in[n],j\in[n+1]} (Xu_i)(X^{\circ(d-1)}v_{ij})(X^{\circ d}w_{ij}) = \sum_{ijk} T'_{ijk}u_iu_ju_k + \sum_{ij} u_iv_{ij}w_{ij}. \tag{5.19}$$

Focusing on the first summations on both sides of the equation, we see these are precisely the equations $X \cdot f = f'$. After subtracting these off, we now deal with the remaining terms.

We have

$$\begin{aligned}
\sum_{ij} u_iv_{ij}w_{ij} &= \sum_{i\in[n],j\in[n+1]} (Xu_i)(X^{\circ(d-1)}v_{ij})(X^{\circ d}w_{ij}) \\
&= \sum_{i\in[n],j\in[n+1]} \left(\sum_{k\in[n]} x_{k,i}u_k\right)\left(\sum_{\ell\in[n]} x_{\ell,i}^{d-1}v_{\ell,j}\right)\left(\sum_{h\in[n]} x_{h,i}^{d}w_{h,j}\right) \\
&= \sum_{k,\ell\in[n],j\in[n+1]} u_kv_{\ell,j}w_{\ell,j}\left(\sum_{i\in[n]} x_{k,i}x_{\ell,i}^{d-1}x_{\ell,i}^{d}\right) + \sum_{\substack{k,\ell,h\in[n],j\in[n+1] \\ \ell\neq h}} u_kv_{\ell,j}w_{\ell',j}\left(\sum_{i\in[n]} x_{k,i}x_{\ell,i}^{d-1}x_{h,i}^{d}\right)
\end{aligned}$$

This becomes the system of equations

$$\begin{aligned} \delta_{k,\ell} &= \textstyle\sum_{i\in[n]} x_{k,i}x_{\ell,i}^{d-1}x_{\ell,i}^{d} && (\forall k,\ell\in[n]) \\ 0 &= \textstyle\sum_{i\in[n]} x_{k,i}x_{\ell,i}^{d-1}x_{h,i}^{d} && (\forall k,\ell,h\in[n],\ell\neq h). \end{aligned}$$

(Note that technically we should quantify over all $j \in [n+1]$, but j plays no role in these equations—it just serves to repeat the same equation $n+1$ times. This corresponds to the fact that the lower-right part of our matrices have the form $* \otimes I_{n+1}$.)

When $k \neq \ell$, every term in the first equation is a degree-$2d$ multiple of the monomial axiom $x_{k,i}x_{\ell,i}$. Similarly, every term in the second set of equations is a degree-$2d$ multiple of the monomial axiom $x_{\ell,i}x_{h,i}$. Thus all that remains is the first equation when $k = \ell$, namely, $1 = \sum_{i\in[n]} x_{k,i}x_{k,i}^{d-1}x_{k,i}^{d}$. This is derived in Lemma 5.5.9, with $c = 2$ in degree $2d$ (since $d > 1$, we have $\max\{2d, d+2\} = 2d$). This completes the proof that we have a $(d, 2d)$-reduction. □

Remark 11. *There is a slightly simpler and smaller many-one reduction, namely* $f \mapsto f + \sum_{i\in[n],j\in[n+1]} u_i v_{ij}^2$. *However, in using that reduction, the witness for the forward direction becomes* $X \oplus (D^{-1/2}P \otimes I_{n+1})$. *This square root introduces a square into the equations that made it difficult to show that it was also a PC reduction. The reduction above fixes this issue.*

5.5.4 From cubic forms to tensors

Our reductions here are those from Futorny–Grochow–Sergeichuk [FGS19, Cor. 3.4 and Thm. 2.1]. The many-one property follows from the results there. We prove that each of these reductions is in fact also a low-degree PC reduction between the corresponding polynomial solvability problems. They reduce first to a problem we call Block Tensor Isomorphism, and then from there to Tensor Isomorphism, so we begin by introducing the former problem and its associated equations.

Definition 5.5.3 (see Futorny–Grochow–Sergeichuk [FGS19])**.** A block $n \times m \times p$ 3-way array is a 3-way array together with a partition of its index sets $\{1,\dots,n\} = \{1,\dots,n_1\} \sqcup \{n_1+1, n_1+2, \dots, n_1+n_2\} \sqcup \cdots \sqcup \{\sum_{i=1}^{N-1} n_i + 1, \dots, n\}$, and similarly for the other two directions. Two block 3-way arrays are said to be *conformally partitioned* if they have the same size and the same partitions of their index sets. Two conformally partitioned 3-way arrays T, T' with block sizes as above are *block-isomorphic* (called "block-equivalent" in [FGS19]) if there exist invertible matrices $S_{11},\dots,S_{1,N}, S_{21},\dots,S_{2M}, S_{31},\dots,S_{3P}$, where $S_{1,I}$ is of size $n_I \times n_I$, $S_{2,J}$ is of size $m_J \times m_J$, and $S_{2,K}$ is of size $p_K \times p_K$, such that the block-diagonal matrices give an isomorphism of tensors:

$$(S_{11} \oplus S_{12} \oplus \cdots \oplus S_{1N}, S_{21} \oplus \cdots \oplus S_{2M}, S_{31} \oplus \cdots \oplus S_{3P}) \cdot T = T'.$$

Given two block 3-way arrays T, T' as above, the equations for Block Tensor Isomorphism are as folllows. There are $2(\sum_{I\in[N]} n_i + \sum_{J\in[M]} m_J + \sum_{K\in[P]} p_K)$ variables arranged into $2(N+M+P)$ square matrices X_I, X_I' (of size $n_I \times n_I$), Y_J, Y_J' (of size $m_J \times m_J$), and Z_K, Z_K' (of size $p_K \times p_K$). Then the equations are:

$$(X_1 \oplus \cdots \oplus X_N, Y_1 \oplus \cdots \oplus Y_M, Z_1 \oplus \cdots \oplus Z_P) \cdot T = T'$$

$$X_I X_I' = X_I' X_I = \mathrm{Id} (\forall I \in [N]) \qquad Y_J Y_J' = Y_J' Y_J = \mathrm{Id} (\forall J \in [M]) \qquad Z_K Z_K' = Z_K' Z_K = \mathrm{Id} (\forall K \in [P])$$

Lemma 5.5.12. *The many-one reduction from*

Equivalence of Noncommutative Cubic Forms to Block Tensor Isomorphism

in [FGS19, Cor. 3.4] is in fact a linear-size (1,3)-many-one reduction.

Proof. Given a noncommutative cubic form f in n variables, $f = \sum_{i,j,k\in[n]} T_{ijk}u_iu_ju_k$, we recall the block tensor $r(T)$ from [FGS19, Cor. 3.4]. It is partitioned into $2 \times 3 \times 3$ many blocks, with the rows being partitioned into $n, 1$, the columns into $n, n, 1$, and the depths also into $n, n, 1$; thus its total size is $(n+1) \times (2n+1) \times (2n+1)$. Let E_{ijk} denote the tensor of this size whose only nonzero entry is a 1 in position (i, j, k). Then we define

$$r(T) = T + \sum_{i\in[n]} (E_{i,n+i,2n+1} + E_{i,2n+1,n+i} + E_{n+1,i,n+i} + E_{n+1,n+i,i}) + E_{n+1,2n+1,2n+1}$$

If you wanted to think of this as part of the tensor corresponding to a cubic form, that cubic form would have $n+1$ new variables $v_1, \ldots, v_n, z$, and the form would be:

$$r(f) := f + \sum_{i\in[n]} (u_iv_iz + u_izv_i + zu_iv_i + zv_iu_i) + z^3.$$

(This doesn't quite line up with the above description of a tensor, as the tensor corresponding to $r(f)$ would necessarily have all 3 side lengths the same, $2n+1$. However, there are n of the $2n+1$ rows in that tensor that are entirely zero, namely, the rows corresponding to those monomials that begin with a v_i.)

The equations for block isomorphism of $r(T)$ and $r(T')$ have the following variable matrices X, X' are $n \times n$, x, x' are 1×1, Y_1, Y_1', Y_2, Y_2' are $n \times n$, y, y' are 1×1, Z_1, Z_1', Z_2, Z_2' are $n \times n$, and z, z' are 1×1. Let U, U' be the $n \times n$ variable matrices for the equations for equivalence of the noncommutative cubic forms f, f'. We consider the following substitution:

$$X, Y_1, Z_1, Y_2', Z_2' \mapsto U \qquad X', Y_1', Z_1', Y_2, Z_2 \mapsto U' \qquad x, x', y, y', z, z' \mapsto 1.$$

Under this substitution, the equations for block isomorphism of $r(T), r(T')$ become

$$\begin{aligned}(U,U,U)\cdot T + \sum_{i\in[n]} &((U,U',1)\cdot E_{i,n+i,2n+1} + (U,1,U')\cdot E_{i,2n+1,n+i}\\ &+(1,U,U')\cdot E_{n+1,i,n+i} + (1,U',U)\cdot E_{n+1,n+i,i} + (1,1,1)\cdot E_{n+1,2n+1,2n+1})\\ =&T' + \sum_{i\in[n]} (E_{i,n+i,2n+1} + E_{i,2n+1,n+i} + E_{n+1,i,n+i} + E_{n+1,n+i,i}) + E_{n+1,2n+1,2n+1}\end{aligned}$$

Now, because each summand inside the big sum corresponds to an identity matrix in a block (e.g. $\sum_{i\in[n]} E_{i,n+i,2n+1}$ is an identity matrix in rows $\{1, \ldots, n\}$, columns $\{n+1, \ldots, 2n\}$, and depth $2n+1$), the above equations give us many instances of $UU' = \mathrm{Id}$ and $U'U = \mathrm{Id}$, which is one of our starting equations. We also get the equation $1 = 1$, and lastly, $(U, U, U) \cdot T = T'$, which is another one of our starting equations. Thus the equations we get here are in fact precisely the same as the equations we started with. As these are cubic equations and the substitutions were linear, it is a (1,3)-PC reduction. □

Lemma 5.5.13. *When the number of blocks is $O(1)$, the many-one reduction from*

Block Tensor Isomorphism

to

Tensor Isomorphism

in [FGS19, Thm. 2.1] is in fact a quadratic-size (1,3)-many-one reduction.

Note that the output of the reduction of Lemma 5.5.12 has $2 \times 3 \times 3$ many blocks, so the restriction to $O(1)$ many blocks in Lemma 5.5.13 presents no obstacle to our goal.

Proof. The key is to show how to effectively remove the partition in one of the three directions; then that reduction can be applied three times in the three separate directions. Let T, T' be block tensors of size $n \times m \times p$, with $N \times M \times P$ many blocks. The construction of [FGS19, Lem. 2.2] shows how to construct from this a block tensor of quadratic size with $N \times M \times 1$ many blocks. We recall the construction here and show that it is a (1,3)-PC reduction.

Let $p_1, \ldots, p_P$ denote the sizes of the parts of the partition in the third direction. Let $r = \min\{n, m\} + 1$—this will govern the rank of the identity matrix gadgets we add. Let $s = \sum_{K=1}^{P} 2^{K-1} r$ and $t = \sum_{K=1}^{P} 2^{K-1} r p_K$. Then the output tensor will have size $(n+s) \times (m+t) \times p$. (Note that, since $P = O(1)$, we have that s is linearly bounded in n, m and t is quadratic as a function of n, m, p.) Let $T_1, \ldots, T_p$ be the frontal slices of T. The i-th slice of $r(T)$ will be as follows. Suppose i is in the K-th block, and write $i = i_0 + \sum_{k=1}^{K} p_k$ with $1 \leq i_0 \leq p_{K+1}$. Write the slices T_i as $T_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$, where A_i is $n_1 \times m_1$—representing the first part of the partition of T into rows and columns, and D_i represents all the other parts. Then we construct:

$$r(T)_i := \left[\begin{array}{ccc:ccc:c:c|c} 0 & \cdots & 0 & & & & & & & \\ \hdashline & \ddots & & & & & & & & \\ \hdashline & & & 0 & \cdots & I_{2^{K-1}r} \cdots 0 & & & & \\ \hdashline & & & & & & \ddots & & & \\ \hdashline & & & & & & & 0 \cdots 0 & & \\ \hdashline & & & & & & & & A_i & B_i \\ \hline & & & & & & & & C_i & D_i \end{array}\right],$$

where the $I_{2^{K-1}r}$ is in the i_0-th position within the K-th block-row and block-column as indicated by the dashed lines. Here the dashed lines *do not* represent additional parts of the partition, they are just for visual clarity. The solid lines indicate the first part of the new partition into rows and columns. The rows of C_i and D_i are partitioned into blocks the same as they were originally in T_i, and the columns of B_i and D_i are partitioned into parts in the same way as they were originally in T_i. That is, the entire big gadget in the upper-left gets prepended to the first parts of the row and column partitions. This is the many-one reduction.

Let $X_1, \ldots, X_N$, $Y_1, \ldots, Y_M$, and Z be variable matrices (with associated primes matrices X_1', etc.), with sizes as follows:

- X_1 has size $(s + n_1) \times (s + n_1)$
- X_I for $I \geq 2$ has size $n_I \times n_I$

- Y_1 has size $(t+m_1)\times(t+m_1)$
- Y_J for $J\geq 2$ has size $m_J\times m_J$
- Z has size $p\times p$.

We start from the equations for Block Isomorphism (but now where there is only one block in the third direction), namely

$$X_I X_I' = X_I' X_I = \mathrm{Id} \qquad Y_J Y_J' = Y_J' Y_J = \mathrm{Id} \qquad ZZ' = Z'Z = \mathrm{Id}$$

and

$$(X_1 \oplus \cdots \oplus X_N, Y_1 \oplus \cdots \oplus Y_N, Z)\cdot r(T) = r(T').$$

We make the following substitution (with the same substitutions, *mutatis mutandis*, for the primed variables):

- $X_1 \mapsto I_s \oplus \hat{X}_1$, where $\hat{X}_1$ is a matrix of variables of size $n_1\times n_1$.
- For $I\geq 2$, X_I maps to itself.
- $Y_1 \mapsto I_t \oplus \hat{Y}_1$, where $\hat{Y}_1$ is a matrix of variables of size $m_1\times m_1$.
- For $J\geq 2$, Y_J maps to itself.
- Z maps to a block matrix $Z_1\oplus\cdots\oplus Z_P$, where for each $K\in[P]$, we have Z_K is a $p_K\times p_K$ matrix of variables.

Under these substitutions, the equations for Block Isomorphism of $r(T), r(T')$ become precisely the original equations for Block Isomorphism of T, T', together with equations of the form $I_s E_i I_t = E_i$, where E_i is the $s\times t$ gadget matrix in the upper-left in the i-th slice. Thus we get a $(1,3)$-reduction.

Finally, this is then repeated in the other two directions to reduce the number of blocks in all three directions to one, thus giving an instance of Tensor Isomorphism. □

5.5.5 Putting it all together

Finally, we combine all the above to prove Theorem 5.5.2.

Proof of Theorem 5.5.2. Let $m = cn$ with $c\geq 10^4$. By Theorem 5.5.1, random 3XOR instances with clause density c require PC degree $\Omega(n/c^2)=\Omega(n)$ (in our case) to refute. The number of instances that the random distribution assigns nonzero probability is $\binom{2\binom{n}{3}}{m} \sim \binom{n^3}{cn} \geq n^{3cn}/(cn)^{cn} = c^{2cn\log n - cn} \geq c^{\Omega(n\log n)}$.

By Theorem 5.5.4, there is a (1,3)-many-one reduction from those instances to $\{\pm 1\}$-Monomial Equivalence of $\{\pm 1\}$ Multilinear Noncommutative Cubic Forms, where the number of variables in the cubic form is the same as the number of variables in the 3XOR instance. By Theorem 5.5.7 there is then a (2,6)-many-one reduction to Monomial Equivalence of $\{\pm 1\}$ Noncommutative Cubic Forms, where the number of variables in the output cubic form is linear in the original number of variables, and such that the output forms have the property that any monomial equivalence between them has all its nonzero entries being 6-th roots of unity. This thus satisfies the hypothesis of Theorem 5.5.10 with $d=6$, so there is a (6,12)-many-one reduction to Equivalence of Noncommutative

Cubic Forms, where the output has a quadratic number of variables compared to the input. Finally, combining Lemma 5.5.12 and Lemma 5.5.13, we get a (1,3) reduction from Equivalence of Noncommutative Cubic Forms to Tensor Isomorphism, which further increases the size quadratically. In total, the size increases multiply, yielding a quartic size increase. The substitution degrees multiply and the derivation degrees we take the max, yielding a (12,12)-many-one reduction from Random 3XOR to Tensor Isomorphism on tensors of size $O(n^4) \times O(n^4) \times O(n^4)$. By Lemma 5.1.1, any PC refutation of these Tensor Isomorphism instances requires degree $\Omega(n)$. □

5.5.6 Lower Bound in Sum-of-Squares

We note that our lower bound for tensor isomorphism also applies to the stronger Sum-of-Squares proof system. This is due to the fact that there is lower bound for random 3XOR in Sum-of-Squares, as shown by Grigoriev [Gri01] and independently by Schoenbeck [Sch08], which makes the dependence on the clause density explicit.

Theorem 5.5.14 ([Sch08, Theorem 12])**.** *A random 3-XOR instance with clause density $\Delta = m/n = dn^{\epsilon}$, for all sufficiently large constants d, requires SoS degree $\Omega(n^{1-\epsilon})$ to refute, with probability $1 - o(1)$.*

In particular, this is a linear $\Omega(n)$ lower bound in the case of constant clause density ($\epsilon = 0$), which matches the PC lower bound of Theorem 5.5.1.

As we observe all of our reductions go through in Sum-of-Squares, since Sum-of-Squares simulates PC over the reals due to Berkholz [Ber18]. Furthermore, this simulation preserves degrees of proofs up to a constant factor.

Theorem 5.5.15 ([Ber18, Theorem 1.1])**.** *Assume that a system of equations $\mathcal{F}$ contains the Boolean axioms $x^2 - x = 0$. If a system of polynomial equations $\mathcal{F}$ over the reals has a PC refutation of degree d and size s, it also has a sum-of-squares refutation of degree $2d$ and size $poly(s)$.*

It is observed in [BGL23] that the simulation of [Ber18] also works when the Boolean axioms are encoded as $x^2 = 1$, or when the domain is the set of k-roots of unity over $\mathbb{C}$.

We formally state how [Ber18, Theorem 1.1] implies a lower bound on the SoS degree of Tensor Isomorphism. This is inspired by [Ber18, Corollary 2.2] that discusses a proof system called the Positivestellensatz Calculus, which is a proof system that extends both PC and SoS.

Lemma 5.5.16. *Let $\mathcal{P}$ be a system of polynomial equations containing the Boolean axioms. Suppose there is a (d_1, d_2)-PC reduction from $\mathcal{P}$ to $\mathcal{Q}$. If $\mathcal{Q}$ has a degree d SoS refutation, then $\mathcal{P}$ has a degree $\max(2d_1 d, 2d_2)$ SoS refutation.*

Proof. Suppose $\mathcal{P} = \{p_1(x_1, \ldots, x_m), \ldots, p_k(x_1, \ldots, x_m)\}$ and $\mathcal{Q} = \{q_1(y_1, \ldots, y_n), \ldots, q_l(y_1, \ldots, y_n)\}$.

Suppose $\mathcal{Q}$ has a degree d SoS refutation. Then there exists a polynomial identity of the form

$$\sum_{i=1}^{l} f_i(y_1, \ldots, y_n) q_i(y_1, \ldots, y_n) + p_0(y_1, \ldots, y_n) = -1$$

where p_0 is a sum-of-squares polynomial.

From the PC reduction, there exists polynomials $y_i = r_i(x_1, \ldots, x_m)$ for each $1 \leq i \leq n$, and a degree d_2 derivation of polynomials $q_i(r_1, \ldots, r_n)$ from the set of polynomials $\mathcal{P}$.

We claim that $-1 - p_0(r_1, \ldots, r_n)$ can be derived from $\mathcal{P}$ in PC in degree $\max(d_2, d_1 d)$. This is because by our PC reduction, polynomials $q_i(r_1, \ldots, r_n)$ can be derived from $\mathcal{P}$ in degree d_2. Afterwards, since the polynomials $f_i(r_1, \ldots, r_n)$ have degree at most $d_1 d$, then $\sum_{i=1}^{l} f_i(r_1, \ldots, r_n) q_i(r_1, \ldots, r_n) = -1 - p_0(r_1, \ldots, r_n)$ can be derived in degree $d_1 d$.

Now, by [Ber18], since $-1 - p_0(r_1, \ldots, r_n)$ has a degree $\max(d_2, d_1 d)$ PC derivation from $\mathcal{P}$, and $\mathcal{P}$ contains the Boolean axioms, we obtain a degree $\max(2d_2, 2d_1 d)$ SoS certificate

$$-(-1 - p_0(r_1, \ldots, r_n))^2 = \sum_{i=1}^{k} a_i(x_1, \ldots, x_m) f_i(x_1, \ldots, x_m) + h_0(x_1, \ldots, x_m)$$

where $h_0(x)$ is a SoS polynomial. Since the left hand side is equal to $-1 - 2p_0 - p_0^2$, we obtain by rearranging that

$$-1 = \sum_{i=1}^{k} a_i(x) f_i(x) + h_0(x) + 2p_0(r_1, \ldots, r_n) + p_0(r_1, \ldots, r_n)^2.$$

Observe that since p_0 is an SoS polynomial of degree $\leq d$, then $p_0(r_1, \ldots, r_n)$ and $p_0(r_1, \ldots, r_n)^2$ are also SoS polynomials, of degree $\leq 2d_1 d$. Hence overall, this provides a SoS certificate that $\mathcal{P}$ is unsatisfiable, of degree $\leq \max(2d_2, 2d_1 d)$. □

Hence, observing that the original system of polynomial equations in Random 3XOR or Graph Isomorphism contain the Boolean axioms, by combining Theorem 5.5.14, Lemma 5.5.16 and the PC reductions used to prove Theorem 5.5.2, we obtain the following lower bound for tensor isomorphism in Sum-of-Squares.

Theorem 5.5.17. *Over the real numbers, there is a distribution on $n \times n \times n$ Tensor Isomorphism whose associated equations require SoS degree $\Omega(\sqrt[4]{n})$ to refute with probability 1 - o(1).*

5.6 Open Questions

Lower Bounds in PC+Inv Although it remains open whether the Inversion Principle is "complete" for linear-algebraic reasoning (see [Sol01, SC04]), we introduce the proof system PC+Inv in an attempt to capture some linear-algebraic reasoning that seems potentially useful for TI. PC+Inv has all the same derivation rules as PC, but in addition, for any square matrices A, B (whose entries may themselves be polynomials—that is, we allow substitution instances), we have the rule

$$\frac{AB = I}{BA = I}.$$

where the antecedent represents the set of n^2 equations corresponding to $AB = I$, and similarly the consequent denotes the set of n^2 equations $BA = I$ (see Section 5.1.3 for more details). Degree is still measured in the usual way, but this rule lets us "cut out" the high-degree proof that would usually be required to derive $BA = I$ from $AB = I$.

We now formalize our intuition that linear algebra should not suffice to solve TI efficiently in the following:

Conjecture 2. Tensor Isomorphism *for $n \times n \times n$ tensors requires degree $\Omega(n)$ in PC+Inv, over any field.*

Despite the conjecture, we do not yet know how to prove lower bounds on PC+Inv for any unsatisfiable system of equations, let alone those coming from TI. Mod p counting principles (for p different from the characteristic of the field) strike us as potentially interesting instances to examine for PC+Inv lower bounds, before tackling a harder problem like TI.

Degree Beyond Conjecture 2, we highlight several more questions we find interesting about the algebraic proof complexity of Tensor Isomorphism.

Open Question 5.6.1. What is the correct value for the PC degree of rank-r Tensor Isomorphism?

Note that by using the reductions from Section 5.5, we can produce (random) $r \times r \times r$ tensors that require PC degree $\Omega(r^{1/4})$ to refute. However, the number of variables is $6r^2$, this lower bound is only $\Omega(N^{1/8})$ where N is the number of variables. Since their rank could be as large as $R = \Theta(r^2)$ (and indeed, very likely is), the upper bound we get from Theorem 5.3.1 is only $2^{O(r^4)}$ (without the $x^q - x$ axioms) or $O(r^4)$ (with the $x^q - x$ axioms, with $q = O(1)$). Even in the latter case, this leaves a polynomial gap between the lower and upper bounds (without those the gap is exponential).

We note that the upper bound in Theorem 5.3.1 without the $x^q - x$ equations already applies to the weaker Nullstellensatz proof system. Is there a polynomial upper bound on PC degree—as a function of rank—without the $x^q - x$ axioms?

Size In the presence of the Boolean axioms, there is a size-degree tradeoff for PC (or even PCR—a system with the same degree bounds as PC, but is stronger when measuring size by number of monomials or number of symbols) [CEI96b, ABRW04]. This implies that in the presence of the Boolean axioms, a good degree lower bound implies a good size lower bound. But TI does not have the Boolean axioms.

Open Question 5.6.2. Get lower and upper bounds on the *size* of PC proofs for Tensor (Non-)Isomorphism. Are there subexponential size upper bounds, despite the polynomial degree lower bounds?

Other matrix problems While many different tensor-related problems are all equivalent to TI, in the case of matrices, we have three genuinely different problems: matrix equivalence (2-TI), matrix conjugacy, and matrix congruence. Conjugacy is determined by the Rational Normal Form or Jordan Normal Form, while congruence depends on the field (e.g., over algebraically closed fields it only depends on rank, over $\mathbb{R}$ it depends on the signature, and over finite fields it depends on whether the determinant is a square or not).

Open Question 5.6.3. What is the PC complexity (size, degree, etc.) of matrix conjugacy? Of matrix congruence?

More precisely, for conjugacy we have in mind the system of equations:

$$XM = M'X \qquad XX' = X'X = I,$$

and for congruence the system of equations:

$$XMX^T = M' \qquad XX' = X'X = I.$$

Bounded border rank Not only can testing a tensor for bounded rank can be done in polynomial time (Remark 2), testing a tensor for bounded *border*-rank can also be done in polynomial time (see, e. g., [Gro13]), by evaluating a polynomial number of easy-to-evaluate equations. While several partial results are available, the gap for what is known about the ratio between rank and border rank is quite large: there are 3-tensors known whose rank approaches 3 times their border rank [Zui17], but the currently known upper bound is Lehmkuhl and Lickteig [LL89], who show that for tensors of border rank b, the ratio of rank to border rank is at most $c^{\Theta(nb)}$. See the Zuiddam's introduction [Zui17] for more details.

Open Question 5.6.4. What is the PC degree of testing isomorphism of tensors of bounded border-rank? Can such tests be done (by any method) in polynomial time?

Relating different reductions from Graph Isomorphism While we chose a particular reduction from GI to TI for the lower bound in Section 5.4, we are aware of several others, including:

- GI to Permutational Code Equivalence [PR97, Luk93, Miy96], then to Matrix Lie Algebra Conjugacy [Gro12], then to TI [FGS19];
- GI to Semisimple Matrix Lie Algebra Conjugacy [Gro12], and then to TI [FGS19];
- GI to Alternating Matrix Space Isometry [GQ21b, HQ21], then to TI [FGS19];
- GI to Algebra Isomorphism [Gri81, AS05], then to TI [FGS19].

We believe all of these can be realized as low-degree PC reduction as well. In the first arXiv version of [GQ21b], they asked which of these might be equivalent in some sense (though there the final target was Alternating Matrix Space Isometry, another TI-complete problem, rather than TI itself). Here we make this question slightly more precise, in terms of PC reductions:

Open Question 5.6.5. Which, if any, of the reductions above from Graph Isomorphism to Tensor Isomorphism are equivalent under low-degree PC?

Bibliography

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[Aar11] Scott Aaronson. Impossibility of succinct quantum proofs for collision-freeness. *arXiv preprint arXiv:1101.0403*, 2011.

[Aar21] Scott Aaronson. Open problems related to quantum query complexity. *ACM Transactions on Quantum Computing*, 2(4):1–9, 2021.

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[ABB+17] Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in query complexity based on pointer functions. *Journal of the ACM (JACM)*, 64(5):1–24, 2017.

[ABD+08] Scott Aaronson, Salman Beigi, Andrew Drucker, Bill Fefferman, and Peter Shor. The power of unentanglement. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 223–236. IEEE, 2008.

[ABDK+21] Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of Huang's sensitivity theorem. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1330–1342, 2021.

[ABRW04] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM J. Comput.*, 34(1):67–88, 2004.

[ACQ22] Dorit Aharonov, Jordan Cotler, and Xiao-Liang Qi. Quantum algorithmic measurement. *Nature communications*, 13(1):1–9, 2022.

[AK07] Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. *Theory of Computing*, 3:129–157, 2007.

[AKKT20] Scott Aaronson, Robin Kothari, William Kretschmer, and Justin Thaler. Quantum lower bounds for approximate counting via Laurent polynomials. In *35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[AM13] Albert Atserias and Elitza N. Maneva. Sherali–Adams relaxations and indistinguishability in counting logics. *SIAM J. Comput.*, 42(1):112–137, 2013.

[Amb06] Andris Ambainis. Polynomial degree vs. quantum query complexity. *Journal of Computer and System Sciences*, 72(2):220–238, 2006.

[Amb18] Andris Ambainis. Understanding quantum algorithms via query complexity. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3265–3285. World Scientific, 2018.

[AN02] Dorit Aharonov and Tomer Naveh. Quantum NP-a survey. *arXiv preprint quant-ph/0210077*, 2002.

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM (JACM)*, 51(4):595–605, 2004.

[AS05] Manindra Agrawal and Nitin Saxena. Automorphisms of finite rings and applications to complexity of problems. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Proceedings*, pages 1–17, 2005.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *STOC'16—Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 684–697. ACM, New York, 2016.

[BBBV97] 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.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM (JACM)*, 48(4):778–797, 2001.

[BBD+97] Adriano Barenco, Andre Berthiaume, David Deutsch, Artur Ekert, Richard Jozsa, and Chiara Macchiavello. Stabilization of quantum computations by symmetrization. *SIAM Journal on Computing*, 26(5):1541–1557, 1997.

[BBL13] Jacob Biamonte, Ville Bergholm, and Marco Lanzagorta. Tensor network methods for invariant theory. *Journal of Physics A: Mathematical and Theoretical*, 46(47):475301, 2013.

[BBMC20] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22):12685–12717, 2020.

[BCE+95] Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 303–314, 1995.

[BDW02] Harry Buhrman and Ronald De Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[Ber18] Christoph Berkholz. The relation between polynomial calculus, Sherali-Adams, and sum-of-squares proofs. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[BG15] Christoph Berkholz and Martin Grohe. Limitations of algebraic approaches to graph isomorphism testing. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 155–166. Springer, 2015.

[BG17] Christoph Berkholz and Martin Grohe. Linear diophantine equations, group CSPs, and graph isomorphism. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 327–339. SIAM, 2017. Preprint arXiv:1607.04287 [cs.CC].

[BGIP99] Sam Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 547–556, 1999.

[BGIP01] Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. Syst. Sci.*, 62(2):267–289, 2001.

[BGL+19] Peter A. Brooksbank, Joshua A. Grochow, Yinan Li, Youming Qiao, and James B. Wilson. Incorporating Weisfeiler–Leman into algorithms for group isomorphism. arXiv:1905.02518 [cs.CC], 2019.

[BGL23] Ilario Bonacina, Nicola Galesi, and Massimo Lauria. On vanishing sums of roots of unity in polynomial calculus and sum-of-squares. *computational complexity*, 32(2):12, 2023.

[BHMT02] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

[BHT98] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum counting. In *International Colloquium on Automata, Languages, and Programming*, pages 820–831. Springer, 1998.

[BI99] Eli Ben-Sasson and Russell Impagliazzo. Random CNF's are hard for the Polynomial Calculus. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 415–421. IEEE Computer Society, 1999. (Journal version in *Comput. Complex.* 2010, doi:10.1007/s00037-010-0293-1).

[BIK+96] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proceedings of the London Mathematical Society*, 3(1):1–26, 1996.

[BKS03] Paul Beame, Henry Kautz, and Ashish Sabharwal. Understanding the power of clause learning. In *IJCAI*, pages 1194–1201. Citeseer, 2003.

[BKT18] Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 297–310, 2018.

[BOCIP02] Joshua Buresh-Oppenheim, Matthew Clegg, Russell Impagliazzo, and Toniann Pitassi. Homogenization and the polynomial calculus. *Computational complexity*, 11:91–108, 2002.

[BP96] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 274–282. IEEE, 1996.

[BP98] Samuel R Buss and Toniann Pitassi. Good degree bounds on Nullstellensatz refutations of the induction principle. *Journal of computer and system sciences*, 57(2):162–171, 1998.

[Bra37] Richard Brauer. On algebras which are connected with the semisimple continuous groups. *Annals of Mathematics*, pages 857–872, 1937.

[Bre70] R. P. Brent. Algorithms for matrix multiplication. Stanford Computer Science Dept. Tech. Report STAN-CS-70-157, available online at https://apps.dtic.mil/sti/pdfs/AD0705509.pdf, 1970.

[BS20] Jendrik Brachter and Pascal Schweitzer. On the Weisfeiler–Leman dimension of finite groups. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 287–300. ACM, 2020.

[BS22] Jendrik Brachter and Pascal Schweitzer. A systematic study of isomorphism invariants of finite groups via the Weisfeiler–Leman dimension. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPIcs*, pages 27:1–27:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[BSI10] Eli Ben-Sasson and Russell Impagliazzo. Random CNF's are hard for the polynomial calculus. *Computational Complexity*, 19:501–519, 2010.

[BSW21] Zvika Brakerski, Devika Sharma, and Guy Weissenberg. Unitary subgroup testing. *arXiv preprint arXiv:2104.03591*, 2021.

[BT09] Hugue Blier and Alain Tapp. All languages in NP have very short quantum proofs. In *2009 Third International Conference on Quantum, Nano and Micro Technologies*, pages 34–37, 2009.

[BT+22] Mark Bun, Justin Thaler, et al. Approximate degree in classical and quantum computing. *Foundations and Trends® in Theoretical Computer Science*, 15(3-4):229–423, 2022.

[Bus96] Samuel R Buss. Lower bounds on Nullstellensatz proofs via designs. *Proof complexity and feasible arithmetics*, 39:59–71, 1996.

[CD10] Jing Chen and Andrew Drucker. Short multi-prover quantum proofs for SAT without entangled measurements. *arXiv preprint arXiv:1011.0716*, 2010.

[CEI96a] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 174–183, 1996.

[CEI96b] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 174–183. ACM, New York, 1996.

[CFI92] Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

[CL22] Nathaniel A. Collins and Michael Levet. Count-free Weisfeiler–Leman and group isomorphism. arXiv:2212.11247 [cs.DS], 2022.

[Cle04] Richard Cleve. The query complexity of order-finding. *Information and Computation*, 192(2):162–171, 2004.

[CLO13] David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[CN16] Benoit Collins and Ion Nechita. Random matrix techniques in quantum information theory. *Journal of Mathematical Physics*, 57(1):015215, 2016.

[CNY22] Thomas Chen, Shivam Nadimpalli, and Henry Yuen. Testing and learning quantum juntas nearly optimally. *arXiv preprint arXiv:2207.05898*, 2022.

[Coo71] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, 1971.

[CR79] Stephen A Cook and Robert A Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

[DMR10] Hang Dinh, Cristopher Moore, and Alexander Russell. The Mceliece cryptosystem resists quantum Fourier sampling attacks. *arXiv preprint arXiv:1008.2390*, 2010.

[DVC00] Wolfgang Dür, Guifre Vidal, and J Ignacio Cirac. Three qubits can be entangled in two inequivalent ways. *Physical Review A*, 62(6):062314, 2000.

[FGS19] Vyacheslav Futorny, Joshua A. Grochow, and Vladimir V. Sergeichuk. Wildness for tensors. *Linear Algebra Appl.*, 566:212–244, 2019.

[FK15] Bill Fefferman and Shelby Kimmel. Quantum vs classical proofs and subset verification. *arXiv preprint arXiv:1510.06750*, 2015.

[FKP+19] Noah Fleming, Pravesh Kothari, Toniann Pitassi, et al. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, 2019.

[FP06] Jean-Charles Faugère and Ludovic Perret. Polynomial equivalence problems: Algorithmic and theoretical aspects. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 30–47. Springer, 2006.

[GGPS23] Nicola Galesi, Joshua A Grochow, Toniann Pitassi, and Adrian She. On the algebraic proof complexity of tensor isomorphism. *arXiv preprint arXiv:2305.19320*, 2023.

[Gha23] Sevag Gharibian. The 7 faces of quantum NP. *arXiv preprint arXiv:2310.18010*, 2023.

[GHL+15] Sevag Gharibian, Yichen Huang, Zeph Landau, Seung Woo Shin, et al. Quantum hamiltonian complexity. *Foundations and Trends® in Theoretical Computer Science*, 10(3):159–282, 2015.

[GL10] Nicola Galesi and Massimo Lauria. Optimality of size-degree tradeoffs for polynomial calculus. *ACM Transactions on Computational Logic (TOCL)*, 12(1):1–22, 2010.

[GPW17] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 132–143. IEEE, 2017.

[GQ21a] Joshua A. Grochow and Youming Qiao. On p-group isomorphism: Search-to-decision, counting-to-decision, and nilpotency class reductions via tensors. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 16:1–16:38. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[GQ21b] Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials I: Tensor Isomorphism-Completeness. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 31:1–31:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[GRB98] Markus Grassl, Martin Rötteler, and Thomas Beth. Computing local invariants of quantum-bit systems. *Physical Review A*, 58(3):1833, 1998.

[Gri81] D. Ju. Grigoriev. Complexity of "wild" matrix problems and of the isomorphism of algebras and graphs. *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, 105:10–17, 198, 1981. Theoretical applications of the methods of mathematical logic, III.

[Gri01] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1-2):613–622, 2001.

[Gri13] Dima Grigoriev. Polynomial complexity of solving systems of few algebraic equations with small degrees. In Vladimir P. Gerdt, Wolfram Koepf, Ernst W. Mayr, and Evgenii V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing - 15th International Workshop, CASC 2013, Berlin, Germany, September 9-13, 2013. Proceedings*, volume 8136 of *Lecture Notes in Computer Science*, pages 136–139. Springer, 2013.

[Gro12] Joshua A. Grochow. Matrix Lie algebra isomorphism. In *IEEE Conference on Computational Complexity (CCC12)*, pages 203–213, 2012. Also available as arXiv:1112.2012 [cs.CC] and ECCC Technical Report TR11-168.

[Gro13] Joshua A. Grochow. Answer to "deciding bound on tensor rank for a fixed value". CSTheory StackExchange, https://cstheory.stackexchange.com/a/19518/129, 2013.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical computer science*, 39:297–308, 1985.

[Ham21] Yassine Hamoudi. Quantum Sub-Gaussian mean estimator. *arXiv preprint arXiv:2108.12172*, 2021.

[HLW06] Patrick Hayden, Debbie W. Leung, and Andreas Winter. Aspects of generic entanglement. *Communications in Mathematical Physics*, 265(1):95–117, Mar 2006.

[HM13] Aram W Harrow and Ashley Montanaro. Testing product states, quantum Merlin-Arthur games and tensor optimization. *Journal of the ACM (JACM)*, 60(1):3, 2013.

[HMM+08] Masahito Hayashi, Damian Markham, Mio Murao, Masaki Owari, and Shashank Virmani. Entanglement of multiparty-stabilizer, symmetric, and antisymmetric states. *Physical Review A*, 77(1):012104, 2008.

[HQ21] Xiaoyu He and Youming Qiao. On the Baer-Lovász-Tutte construction of groups from graphs: isomorphism types and homomorphism notions. *European J. Combin.*, 98:Paper No. 103404, 12, 2021.

[HT15] Pavel Hrubes and Iddo Tzameret. Short proofs for the determinant identities. *SIAM J. Comput.*, 44(2):340–383, 2015.

[IPS99] Russell Impagliazzo, Pavel Pudlák, and Jiri Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8:127–144, 1999.

[JQSY19] Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In *Theory of Cryptography Conference*, pages 251–281. Springer, 2019.

[KKR06] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *Siam journal on computing*, 35(5):1070–1097, 2006.

[KMY03] Hirotada Kobayashi, Keiji Matsumoto, and Tomoyuki Yamakami. Quantum Merlin-Arthur proof systems: Are multiple Merlins more helpful to Arthur? In *Algorithms and Computation: 14th International Symposium, ISAAC 2003, Kyoto, Japan, December 15-17, 2003. Proceedings 14*, pages 189–198. Springer, 2003.

[KO22] Robin Kothari and Ryan O'Donnell. Mean estimation when you have the source code; or, quantum Monte Carlo methods. *arXiv preprint arXiv:2208.07544*, 2022.

[Kol88] János Kollár. Sharp effective Nullstellensatz. *Journal of the American Mathematical Society*, 1(4):963–975, 1988.

[KP96] Hanspeter Kraft and Claudio Procesi. Classical invariant theory, a primer. *Lecture Notes. Preliminary version*, 1996.

[Kre23] William Werner Kretschmer. *Inherently quantum lower bounds on computational complexity*. PhD thesis, 2023.

[KSV02] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

[Kut03] Samuel Kutin. A quantum lower bound for the collision problem. *arXiv preprint quant-ph/0304162*, 2003.

[Lan12] J. M. Landsberg. *Tensors: geometry and applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2012.

[LCV07] Yi-Kai Liu, Matthias Christandl, and Frank Verstraete. Quantum computational complexity of the n-Representability Problem: QMA complete. *Physical Review Letters*, 98(11), Mar 2007.

[Led01] Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2001.

[LL89] Thomas Lehmkuhl and Thomas Lickteig. On the order of approximation in approximative triadic decompositions of tensors. *Theoret. Comput. Sci.*, 66(1):1–14, 1989.

[Lov17] Shachar Lovett. Additive combinatorics and its applications in theoretical computer science. *Theory of Computing*, pages 1–55, 2017.

[Luk93] Eugene M. Luks. Permutation groups and polynomial-time computation. In *Groups and computation (New Brunswick, NJ, 1991)*, volume 11 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 139–175. Amer. Math. Soc., Providence, RI, 1993.

[Mar89] AA Markov. Sur une question posée par Mendeleieff. *IAN*, 62:1–24, 1889.

[McK81] Brendan D. McKay. Practical graph isomorphism. *Congr. Numer.*, 30:45–87, 1981.

[MdW13] Ashley Montanaro and Ronald de Wolf. A survey of quantum property testing. *arXiv preprint arXiv:1310.2035*, 2013.

[MdW23] Nikhil S Mande and Ronald de Wolf. Tight bounds for quantum phase estimation and related problems. *arXiv preprint arXiv:2305.04908*, 2023.

[Miy96] Takunari Miyazaki. Luks's reduction of graph isomorphism to code equivalence. Comment to E. W. Clark, https://groups.google.com/forum/#!msg/sci.math.research/puZxGj9HXKI/CeyH2yyyNFUJ, 1996.

[MP14] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *J. Symbolic Comput.*, 60:94–112, 2014.

[MP17] Marvin Minsky and Seymour Papert. *Perceptrons*. MIT press, 2017.

[MS86] Vitali D Milman and Gideon Schechtman. *Asymptotic Theory of Finite Dimensional Normed Spaced*, volume 1200. Springer Berlin, 1986.

[MW05] Chris Marriott and John Watrous. Quantum Arthur–Merlin games. *Computational Complexity*, 14(2):122–152, 2005.

[NC10] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[NN22] Anand Natarajan and Chinmay Nirkhe. A classical oracle separation between QMA and QCMA. *arXiv preprint arXiv:2210.15380*, 2022.

[NS94] Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomials. *Computational complexity*, 4(4):301–313, 1994.

[O'D14] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[OWWZ14] Ryan O'Donnell, John Wright, Chenggang Wu, and Yuan Zhou. Hardness of robust graph isomorphism, Lasserre gaps, and asymmetry of random graphs. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1659–1677. SIAM, 2014. Preprint available as arXiv:1401.2436 [cs.CC].

[Pat92] Ramamohan Paturi. On the degree of polynomials that approximate symmetric Boolean functions (preliminary version). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, page 468–474, New York, NY, USA, 1992. Association for Computing Machinery.

[PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational complexity*, 3:97–140, 1993.

[Pit96] Toniann Pitassi. Algebraic propositional proof systems. *Descriptive complexity and finite models*, 31:215–244, 1996.

[PR97] Erez Petrank and Ron M. Roth. Is code equivalence easy to decide? *IEEE Trans. Inf. Theory*, 43(5):1602–1604, 1997.

[Pro76] C Procesi. The invariant theory of $n \times n$ matrices. *Advances in Mathematics*, 19(3):306–381, 1976.

[QSY20] Youming Qiao, Xiaoming Sun, and Nengkun Yu. Local equivalence of multipartite entanglement. *IEEE Journal on Selected Areas in Communications*, 38(3):568–574, 2020.

[Raz98a] Alexander A Razborov. Lower bounds for the polynomial calculus. *computational complexity*, 7:291–324, 1998.

[Raz98b] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Comput. Complex.*, 7(4):291–324, 1998.

[Rei11] Ben W Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 560–569. SIAM, 2011.

[Ros23] Gregory Rosenthal. *Quantum State and Unitary Complexity*. PhD thesis, University of Toronto (Canada), 2023.

[RS04] Ran Raz and Amir Shpilka. On the power of quantum proofs. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 260–274. IEEE, 2004.

[SC04] Michael Soltys and Stephen Cook. The proof complexity of linear algebra. *Ann. Pure Appl. Logic*, 130(1-3):277–323, 2004.

[Sch08] Grant Schoenebeck. Linear level lasserre lower bounds for certain k-CSPs. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 593–602. IEEE, 2008.

[She13] Alexander A Sherstov. The intersection of two halfspaces has high threshold degree. *SIAM Journal on Computing*, 42(6):2329–2374, 2013.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[Sim97] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 77–82, 1987.

[Sol01] Michael Soltys. *The complexity of derivations of matrix identities.* PhD thesis, University of Toronto, 2001. Availalble on ECCC at https://eccc.weizmann.ac.il/resources/pdf/soltys.pdf.

[Som99] Martín Sombra. A sparse effective Nullstellensatz. *Adv. in Appl. Math.*, 22(2):271–295, 1999.

[Spa08] Robert Spalek. A dual polynomial for OR. *arXiv preprint arXiv:0803.4516*, 2008.

[ŠS05] Robert Špalek and Mario Szegedy. All quantum adversary methods are equivalent. In *International Colloquium on Automata, Languages, and Programming*, pages 1299–1311. Springer, 2005.

[SSC14] Aaron Snook, Grant Schoenebeck, and Paolo Codenotti. Graph Isomorphism and the Lasserre hierarchy. arXiv:1401.0758 [cs.CC], 2014.

[SW22] Mehdi Soleimanifar and John Wright. Testing matrix product states. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1679–1701. SIAM, 2022.

[SY22] Adrian She and Henry Yuen. Unitary property testing lower bounds by polynomials. *arXiv preprint arXiv:2210.05885*, 2022.

[TDJ+22] Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022,*

Proceedings, Part III, volume 13277 of *Lecture Notes in Computer Science*, pages 582–612. Springer, 2022.

[TM+17] Jacob Turner, Jason Morton, et al. A complete set of invariants for LU-equivalence of density operators. *SIGMA. Symmetry, Integrability and Geometry: Methods and Applications*, 13:028, 2017.

[Wan11] Guoming Wang. Property testing of unitary operators. *Physical Review A*, 84(5):052328, 2011.

[Wat08] John Watrous. Quantum computational complexity. *arXiv preprint arXiv:0804.3401*, 2008.

[WLAG13] James Daniel Whitfield, Peter John Love, and Alán Aspuru-Guzik. Computational complexity in electronic structure. *Physical Chemistry Chemical Physics*, 15(2):397–411, 2013.

[WZ23] Qisheng Wang and Zhicheng Zhang. Quantum lower bounds by sample-to-query lifting. *arXiv preprint arXiv:2308.01794*, 2023.

[ZDQ+21] Han-Sen Zhong, Yu-Hao Deng, Jian Qin, Hui Wang, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Dian Wu, Si-Qiu Gong, Hao Su, et al. Phase-programmable gaussian boson sampling using stimulated squeezed light. *Physical review letters*, 127(18):180502, 2021.

[Zui17] Jeroen Zuiddam. A note on the gap between rank and border rank. *Linear Algebra Appl.*, 525:33–44, 2017.

Appendix A

Deferred Proofs from Part I

A.1 A Generalized Product Test Analysis

Our goal in this section is to prove Theorem 3.3.4. Let $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ be a bipartite state. Recall that the k-copy product test uses as input k-copies of the state $|\psi\rangle^{\otimes k}$, where each copy is on registers A_i and B_i, and then performs the measurement $\{P = \Pi_A \otimes \Pi_B, I - P\}$, where Π_A is the projection onto the symmetric subspace among registers $A_1, \ldots, A_k$ and Π_B is the projection onto the symmetric subspace among registers $B_1, \ldots, B_k$. Recall from Theorem 3.3.1 that we defined

$$\omega_{|\psi\rangle} = \max_{|\phi_1\rangle, |\phi_2\rangle} \{|\langle\psi|\phi_1 \otimes \phi_2\rangle|^2, |\phi_1\rangle, |\phi_2\rangle \in \mathbb{C}^d\}$$

be the overlap with the closest product state. In particular, from Lemma 3.3.9, we have that $\omega_{|\psi\rangle} = \lambda_1$ where λ_1 is the largest eigenvalue of the reduced density matrix ρ of $|\psi\rangle\langle\psi|$.

We establish the following bound on the performance of the product test for any constant $k \geq 2$, using the techniques of [SW22].

Theorem 3.3.4. *Let $\omega_{|\psi\rangle}$ be the overlap of $|\psi\rangle$ with the closest product state, as defined in Theorem 3.3.1. For all constant $k \geq 2$, the probability α that the product test passes when given $|\psi\rangle^{\otimes k}$ as input satisfies*

$$\alpha \leq \frac{k-1}{k+1}\omega_{|\psi\rangle}^k + \frac{2}{k+1}.$$

Before proceeding with the proof, we fix some notation we will use throughout the rest of this section. Let $[n] = \{1, \ldots, n\}$, $\lambda = (\lambda_1, \ldots, \lambda_m)$ be a list of real numbers and $\alpha = (\alpha_1, \ldots, \alpha_l)$ be a list of non-negative integers. Whenever well-defined, let $\lambda_\alpha = \prod_{i=1}^l \lambda_{\alpha_i}$ and $\lambda^\alpha = \prod_{i=1}^m \lambda_i^{\alpha_i}$. Similarly if $a = \{a_1, \ldots, a_n\}$ is a set of vectors, then $|a\rangle_\alpha = |a_{\alpha_1}\rangle \otimes \cdots \otimes |a_{\alpha_l}\rangle$. Next, for the list α, let $n(\alpha)$ be the number of non-zero entries in α and type(α) be the list $(\beta_1, \ldots, \beta_j)$ where β_j is the number of times integer j appears in the list α. We will also write $\alpha \vdash k$ if $\alpha = (\alpha_1, \ldots, \alpha_l)$ is a list with $\sum_{i=1}^l \alpha_i = k$, and $\binom{k}{\alpha}$ for the multinomial coefficient $\frac{k!}{\prod_{i=1}^l \alpha_i!}$.

We first establish the *exact* probability the product test passes given the state $|\psi\rangle^{\otimes k}$ as input.

Lemma A.1.1. *Let $h_k(x_1, \ldots, x_d) = \sum_{\substack{\beta_1+\cdots+\beta_d=k \\ \beta_i \geq 0}} \prod_{i=1}^d x_i^{\beta_i}$ be the homogenous symmetric polynomial of degree k in d variables. Then the probability that the k-copy product state passes when run on state*

$|\psi\rangle^{\otimes k}$ *is equal to* $h_k(\lambda_1, \ldots, \lambda_d)$*, where* $\lambda_1, \ldots, \lambda_d$ *are the eigenvalues of the reduced density matrix* ρ *of* $|\psi\rangle\langle\psi|$.

Proof. Let $k \geq 2$ be given. Consider the Schmidt decomposition of $|\phi\rangle = \sum_{i=1}^{d} \sqrt{\lambda_i}\,|a_i\rangle\,|b_i\rangle$ across the two subsystems, ordered so that $\lambda_j \geq \lambda_{j+1}$ for every index $1 \leq j < d$. Letting $\Lambda = (\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n})$ be the list of Schmidt coefficients, then

$$|\phi\rangle^{\otimes k} = \sum_{\substack{\alpha=(\alpha_1,\ldots,\alpha_k)\\ \alpha_i\in[d]}} \Lambda_\alpha\,|a\rangle_\alpha\,|b\rangle_\alpha\,. \tag{A.1}$$

Now, note that if α, α' are two sequences with the same type β, then $\Lambda_\alpha = \Lambda_{\alpha'} = \Lambda^\beta$. So rewrite Equation A.1, combining sequences of the same type.

$$|\phi\rangle^{\otimes k} = \sum_{\substack{\beta=(\beta_1,\ldots,\beta_d)\\ \beta_1+\cdots+\beta_d=k}} \Lambda^\beta \sum_{\substack{\alpha=(\alpha_1,\ldots,\alpha_k)\\ \text{type}(\alpha)=\beta}} |a\rangle_\alpha\,|b\rangle_\alpha\,. \tag{A.2}$$

We now consider the action of the projection Π_A on the state $|\phi\rangle^{\otimes k}$. Write the projection as $\Pi_A = \frac{1}{k!}\sum_{\sigma\in S_k} P_\sigma$ where P_σ permutes vectors in $(\mathbb{C}^d)^k$. Note that if α and α' have the same type $\beta = (\beta_1, \ldots, \beta_d)$, there is some permutation P_σ for which $P_\sigma\,|a\rangle_\alpha = |a\rangle_{\alpha'}$ and that there is a subgroup $S_\beta = S_{\beta_1} \times \cdots \times S_{\beta_d} \leq S_k$ of permutations fixing $|a\rangle_\alpha$ of size $\prod_{i=1}^{d} \beta_i!$. Hence, for every $|a\rangle_\alpha$,

$$\Pi_A\,|a\rangle_\alpha = \frac{1}{k!}\sum_{\text{cosets } \sigma S_\beta} |S_\beta| P_\sigma\,|a\rangle_\alpha = \frac{1}{\binom{k}{\beta}} \sum_{\substack{\text{all } \alpha'=(\alpha_1,\ldots,\alpha_k)\\ \text{type}(\alpha)=\alpha'}} |a\rangle_{\alpha'}\,.$$

Therefore,

$$\begin{aligned}(\Pi_A \otimes I)\,|\phi\rangle^{\otimes k} &= \sum_{\substack{\beta=(\beta_1,\ldots,\beta_d)\\ \beta_1+\cdots+\beta_d=k}} \frac{\Lambda^\beta}{\binom{k}{\beta}} \sum_{\substack{\alpha=(\alpha_1,\ldots,\alpha_k)\\ \text{type}(\alpha)=\beta}} \sum_{\substack{\alpha'=(\alpha_1,\ldots,\alpha_k)\\ \text{type}(\alpha')=\beta}} |a\rangle_{\alpha'}\,|b\rangle_\alpha\,. \\ &= \sum_{\substack{\beta=(\beta_1,\ldots,\beta_d)\\ \beta_1+\cdots+\beta_d=k}} \Lambda^\beta \left(\frac{\displaystyle\sum_{\substack{\alpha=(\alpha_1,\ldots,\alpha_k)\\ \text{type}(\alpha)=\beta}} |a\rangle_\alpha}{\sqrt{\binom{k}{\beta}}}\right)\left(\frac{\displaystyle\sum_{\substack{\alpha=(\alpha_1,\ldots,\alpha_k)\\ \text{type}(\alpha)=\beta}} |b\rangle_\alpha}{\sqrt{\binom{k}{\beta}}}\right)\end{aligned} \tag{A.3}$$

We will write $|a\rangle^\beta = \left(\dfrac{\displaystyle\sum_{\substack{\alpha=(\alpha_1,\ldots,\alpha_k)\\ \text{type}(\alpha)=\beta}} |a\rangle_\alpha}{\sqrt{\binom{k}{\beta}}}\right)$ and note that the set of all $\{|a\rangle^\beta\}$ over all $\beta \vdash k$ is an orthonormal basis for the symmetric subspace $(\text{Sym}^k\mathbb{C}^d)$. Hence the probability that the product test applied to the first subsystem passes is $\mu = \sum_{\beta\vdash k}(\Lambda^\beta)^2$, and conditioned on this, the post-measurement mixed state is $|a\rangle^\beta\,|b\rangle^\beta$ with probability $\frac{(\Lambda^\beta)^2}{\mu}$. At this point, the second subsystem becomes a symmetric state, and hence the projection onto the B registers passes with probability one. Hence, we obtain that the probability that the product test passes is

$$\mu = \sum_{\beta \vdash k} (\Lambda^\beta)^2 = \sum_{\substack{\beta_1 + \cdots + \beta_d = k \\ \beta_i \geq 0}} \prod_{i=1}^{d} \lambda_i^{\beta_i} = h_k(\lambda_1, \ldots, \lambda_d). \tag{A.4}$$

□

Proof of Theorem 3.3.4. Firstly, we claim that

$$\frac{2}{k+1}\lambda_1^k + \sum_{\beta \vdash k, \beta_1 < k} (\Lambda^\beta)^2 \leq \frac{2}{k+1}. \tag{A.5}$$

From Equation (A.2), we observe that

$$1 = \sum_{\beta \vdash k} \binom{k}{\beta} (\Lambda^\beta)^2 = \lambda_1^k + \sum_{\beta \vdash k. \beta_1 < k} \binom{k}{\beta} (\Lambda^\beta)^2. \tag{A.6}$$

Therefore, we can divide the sum into three cases depending on the length $n(\beta)$ and the the value of β_1,

$$\begin{aligned}
&\frac{2}{k+1}\lambda_1^k + \sum_{\beta \vdash k, \beta_1 < k} (\Lambda^\beta)^2 = \frac{2}{k+1}\left[1 - \sum_{\beta \vdash k, \beta_1 < k} \binom{k}{\beta} (\Lambda^\beta)^2\right] + \sum_{\beta \vdash k, \beta_1 < k} (\Lambda^\beta)^2 \\
&= \frac{2}{k+1} + \sum_{\beta \vdash k, \beta_1 < k} \left[1 - \frac{2}{k+1}\binom{k}{\beta}\right] (\Lambda^\beta)^2 \\
&= \frac{2}{k+1} + \sum_{n(\beta)=1, \beta_1 < k} \left[1 - \frac{2}{k+1}\binom{k}{\beta}\right] (\Lambda^\beta)^2 + \sum_{n(\beta)=2, \beta_1 = k-1} \left[1 - \frac{2}{k+1}\binom{k}{\beta}\right] (\Lambda^\beta)^2 \\
&+ \sum_{\beta \vdash k, \beta_1 < k-1, n(\beta) \geq 2} \left[1 - \frac{2}{k+1}\binom{k}{\beta}\right] (\Lambda^\beta)^2.
\end{aligned} \tag{A.7}$$

Observe firstly that if $n(\beta) \geq 2$, $\binom{k}{\beta} \geq \binom{k}{\beta_i} \geq k$ since $\beta_i \leq k-1$ is the largest entry in the list β. Hence in all these cases $\left[1 - \frac{2}{k+1}\binom{k}{\beta}\right] (\Lambda^\beta)^2 \leq 0$ since $1 - \frac{2}{k+1}\binom{k}{\beta} < 0$. This shows that the third case is always negative.

Next, to bound the first and second cases:

$$\begin{aligned}
&\sum_{n(\beta)=1, \beta_1 < k} \left[1 - \frac{2}{k+1}\binom{k}{\beta}\right] (\Lambda^\beta)^2 + \sum_{n(\beta)=2, \beta_1 = k-1} \left[1 - \frac{2}{k+1}\binom{k}{\beta}\right] (\Lambda^\beta)^2 \\
&= (1 - \frac{2}{k+1}) \sum_{i=2}^{d} \lambda_i^k + [1 - \frac{2k}{k+1}] \sum_{i=2}^{d} \lambda_1^{k-1}\lambda_i \\
&= \frac{k-1}{k+1} \sum_{i=2}^{d} \lambda_i^k - \frac{k-1}{k+1} \sum_{i=2}^{d} \lambda_1^{k-1}\lambda_i = \frac{k-1}{k+1} [\sum_{i=2}^{d} (\lambda_i(\lambda_i^{k-1} - \lambda_1^{k-1})] \leq 0
\end{aligned} \tag{A.8}$$

since each λ_i is positive and λ_1 is the greatest of all of the Schmidt coefficients. Therefore, we have established Equation (A.5) since we have shown all of the sums in Equation (A.7) are negative.

Equation (A.5) and Lemma A.1.1 implies our bound since the probability that the product test

passes is

$$h_k(\lambda_1,\ldots,\lambda_d) = \sum_{\beta\vdash k}(\Lambda^\beta)^2 = \lambda_1^k + \sum_{\beta\vdash k,\beta_1<k}(\Lambda^\beta)^2 \\ \leq \lambda_1^k + \frac{2}{k+1} - \frac{2}{k+1}\lambda_1^k = \frac{k-1}{k+1}\lambda_1^k + \frac{2}{k+1} = \frac{k-1}{k+1}\omega_{|\psi\rangle}^k + \frac{2}{k+1}. \tag{A.9}$$

where we have used the fact that for bipartite states, $\omega_{|\psi\rangle} = \lambda_1$ from Lemma 3.3.9.

□

A.2 A SymQMA Verifier for the Entangled Subspace Problem

We now ready to apply Theorem 3.3.4 to prove Theorem 3.3.5.

Theorem 3.3.5. *Let $0 \leq a < b < 1$ be constants. Then there exists a constant $k \geq 2$ sufficiently large such that there is a* SymQMA(k + 1) *verifier for the (a,b)-Entangled Subspace problem.*

The verifier V_{k+1} has $k+1$ proof registers $|\psi\rangle^{\otimes k+1}$. The k-copy product test is performed on registers 1 to k. Finally, the circuit performs a controlled U operation on the $k+1$(st) proof state. The verifier accepts if and only if the product test passes and the ancilla qubit for the controlled U measures to be 1. The verifier V_3 is depicted in Figure A.1.

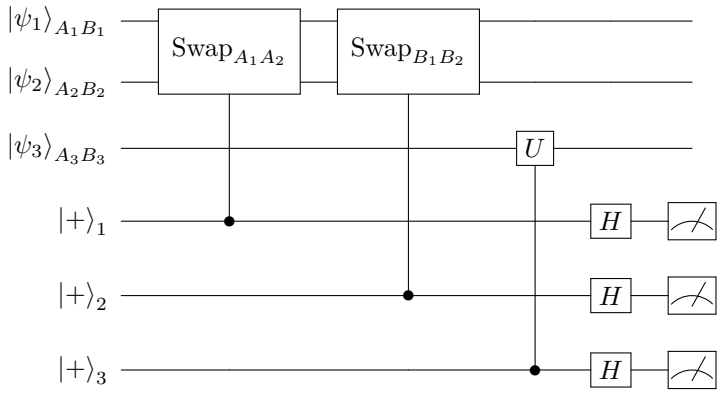

Figure A.1: SymQMA(3) verifier

Proof. By assumption there exists a constant ϵ such that $b^2 - a^2 = \epsilon > 0$. Choose k sufficiently large so that $\frac{k-1}{k+1}(1-\frac{\epsilon^2}{4})^k + \frac{2}{k+1} \leq 1 - b^2$. Such k exists if $\epsilon > 0$ and $0 \leq b < 1$. We claim that the verifier V_{k+1} suffices to distinguishes between the two cases.

Suppose we are given a *yes* instance U of the Entangled Subspace problem. This means that U encodes a subspace S containing a state $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ that is a-close to a product state $|\varphi\rangle \otimes |\xi\rangle$ in trace distance. Suppose we run the verifier on the following proof state: $|\theta\rangle^{\otimes(k+1)}$ where $|\theta\rangle = |\varphi\rangle \otimes |\xi\rangle$. Clearly, this proof will pass the product test with probability 1 for any $k \geq 2$. Furthermore, the acceptance probability of the subspace membership test is $\left\|\Pi\Big(|\varphi\rangle \otimes |\xi\rangle\Big)\right\|^2 = |\langle\psi|\varphi\otimes\xi\rangle|^2 \geq 1 - a^2$, by assumption that S contains a state that is a-close to product.

Now suppose U is a *no* instance, and suppose for contradiction there exists a proof $|\psi\rangle^{\otimes(k+1)}$ that is accepted by the verifier with probability greater than $1 - \frac{\epsilon}{2} - a^2$. Then in particular this means that

the product test with $|\psi\rangle^{\otimes k}$ accepts with probability at least $1 - \frac{\epsilon}{2} - a^2$. Hence, by Theorem 3.3.4, if the witness causes the verifier to accept with probability at least $1 - \frac{\epsilon}{2} - a^2$, then by the choice of k, there exists a product state $|\theta\rangle = |\varphi\rangle \otimes |\xi\rangle$ such that $|\langle\phi|\theta\rangle|^2 \geq 1 - \frac{\epsilon^2}{4}$ since $b^2 - a^2 = \epsilon$ implies that $1 - \frac{\epsilon}{2} - a^2 \geq 1 - b^2$.

Let P_S be the projector onto the hidden subspace S. Since the definition of trace distance implies that:

$$|\operatorname{Tr}(P_S(|\psi\rangle\langle\psi| - |\theta\rangle\langle\theta|))| \leq \| |\phi\rangle\langle\phi| - |\theta\rangle\langle\theta| \| \leq \sqrt{1 - (1 - \frac{\epsilon^2}{4})} = \frac{\epsilon}{2}.$$

Then since $\operatorname{Tr}(P_S |\psi\rangle\langle\psi|) > 1 - \frac{\epsilon}{2} - a^2$, by assumption we have

$$\operatorname{Tr}(P_S |\theta\rangle\langle\theta|) > 1 - \epsilon - a^2 = 1 - (\epsilon + a^2) = 1 - b^2.$$

which is a contradiction since S was b-completely entangled. Therefore, the verifier must accept with probability at most $1 - \frac{\epsilon}{2} - a^2$ in the no case. Hence, there is a gap of at least $1 - a^2 - (1 - \frac{\epsilon}{2} - a^2) = \frac{\epsilon}{2}$ in distinguishing between the yes and no cases. Thus, there is a constant gap between the acceptance probabilities of the *yes* and *no* instances, showing that the (a, b)-Entangled Subspace problem is in SymQMA(k+1). □