On the Power and Limitations of Shallow Quantum Circuits

by

Natalie Parham

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization (Quantum Information)

Waterloo, Ontario, Canada, 2022

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

- Chapters 2 and 3 contain my own summary of previous results.
- Chapter 4 contains new results authored by myself and aided by discussions with David Gosset and Adam Bene Watts.
- Chapter 5 is joint work with Adam Bene Watts, who co-authored this chapter.

Abstract

Constant-depth quantum circuits, or *shallow* quantum circuits, have been shown to exhibit behavior that is uniquely quantum. This thesis explores the power and limitations of constant depth quantum circuits, in particular as they compare to constant-depth *classical* circuits.

We start with a gentle introduction to shallow quantum and classical circuit complexity, and we review the hardness of sampling from the output distribution of a constant-depth quantum circuit. We then give an overview of the shallow circuit advantage from the 1D Magic Square Problem from [BGKT20].

The first novel contribution is an investigation into the limitations of shallow quantum circuits for local optimization problems. We prove that if a shallow quantum circuit's input/output relation is *exactly* that of a local optimization problem, then we can construct a shallow classical circuit that also solves the optimization problem. We also prove an approximate version of this statement.

Finally, we introduce a novel sampling task over an n-bit distribution D_n such that there exists a shallow quantum circuit that takes as input the state $|\mathrm{GHZ}_n\rangle = \frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle)$ and produces a distribution close to D_n whereas, any constant-depth classical circuit with bounded fan-in and $n + n^\delta$ random input bits for some $\delta < 1$, will produce a distribution that is not close to D_n.

Acknowledgements

First and foremost I would like to thank David Gosset, my supervisor. David has been a continual source of interesting technical discussions and valuable advice. I am grateful that David introduced me to shallow circuit complexity and challenged me to prioritize my writing and presentation skills.

I would also like to thank Raymond Laflamme, my co-supervisor, for all of his support, and for curating such a welcoming research group where I found close friendships. I can't stress enough the impact that feeling welcome in this community has had on my motivation and confidence as a researcher.

Next, I would like to acknowledge Adam Bene Watts for being an exceptional mentor and collaborator. As soon as I started working with Adam, his curiosity and enthusiasm were infectious. I am grateful for the many hours spent at the whiteboard with Adam, and the valuable feedback he has provided me on my writing – especially for this thesis.

I appreciate Richard Cleve and Shalev Ben-David for generously offering to read my thesis and giving helpful feedback.

I am very fortunate to have had the support of Satish Rao, who during the final year of my undergrad at U.C. Berkeley, encouraged me to try out theoretical computer science research, despite having a cushy software engineering life waiting for me after graduation. As well as Chinmay Nirkhe, who encouraged me to keep showing up to Umesh Vazirani's graduate quantum computing course, which I was not qualified to enroll in – and for his patience as I spammed him with questions during office hours. I am especially grateful to Chinmay for then giving me my first research project in quantum computation. Had it not been for Satish and Chinmay, I am certain I would not have entertained the idea of pursuing graduate school.

I would like to thank my family: Donna, Hal, Jack, and Kathy for their unconditional and unwavering love. And finally, I would like to thank my partner Martin, and his cat Pizza, for just about everything.

Table of Contents

List of Figures

Chapter 1

Introduction

What can quantum computers do that classical computers cannot? This is the central question guiding much quantum computing research.

Tasks for which quantum algorithms outperform classical algorithms are said to exhibit *quantum advantage*. We note that quantum advantage is a vague term that is used in various contexts. For this reason, it is useful to characterize different types of quantum advantage. In particular, we highlight three desirable properties of a computational problem in the context of quantum advantage.

The first desirable property is that the quantum advantage is *provable*. That is, we can information-theoretically prove that *any* classical algorithm which completes the task requires significantly more resources than the quantum algorithm. The second property is that the computational problem is *useful*, and the third is that the quantum algorithm is *implementable in the near-term*.

Unfortunately, this first property of provability has shown to be quite difficult. Perhaps the most famous quantum algorithm is Shor's factoring algorithm. Shor's algorithm factors an integer in polynomial-time. In contrast, it is widely believed that any classical factoring algorithm requires exponential time. This widespread *belief* of classical hardness, however, has not been proved. Therefore, we consider the complexity separation of factoring as *conditional* on complexity-theoretic conjectures. Furthermore, the quantum algorithm for factoring requires a large-scale quantum computer and is not amenable to near-term, smaller-scale devices.

One model of computation that is more reasonable for near-term implementation, and is the topic of this thesis, is low-depth (or *shallow*) quantum circuits Shallow quantum circuits can be considered as quantum computers that only run for a constant amount of time, but can make small (acting on a constant number of (qu)bits) operations in parallel. Although shallow quantum circuits are relatively simple, they still exhibit uniquely quantum behavior. Assuming complexity-theoretic conjectures, shallow quantum circuits produce output distributions that cannot be efficiently sampled classically [TD02, Aar04] (Section 2.5). Rather than comparing shallow quantum circuits with efficient classical algorithms, if we instead compare them with shallow classical circuits with bounded fan-in, Bravyi, Gosset, and Koenig prove *unconditionally* that there exists a search problem with such a quantum advantage [BGK18]. We detail such an unconditional separation in Chapter 3. Following this result, more search problems were introduced, which also achieve a shallow circuit separation [BGKT20, WKST19, Gal18, CSV21]. These problems, however, are quite contrived and not immediately useful for any natural problem.

In practice, many problems that we care about can be stated as local optimization problems. These are problems with some local objective function[1] $F : \{0,1\}^n \to \mathbb{R}$ that can be written as a sum of smaller (or *local*) terms such that the goal is to find a $y \in \{0,1\}^n$ that minimizes $F(y)$.

Can we achieve a shallow quantum circuit advantage for local optimization problems? The Quantum Approximate Optimization Algorithm (QAOA) introduced by [FGG14] and discussed in Section 2.6, is an algorithm for this type of problem. Moreover, the simplest version of the QAOA can be implemented in constant depth. While there are performance guarantees for low-depth QAOA for certain optimization problems [FH16], they still do not outperform the best-known classical algorithms.

In Chapter 4, we prove limitations of constant depth circuits for local optimization problems. We prove that if a shallow quantum circuit's input/output relation is *exactly* that of a local optimization problem, then we can construct a shallow classical circuit that also solves the optimization problem. We also prove an approximate version of this statement. While this property is rather strict, it still limits the behavior we can expect from shallow quantum circuit optimization algorithms such as the QAOA.

So far, the computational tasks where shallow quantum circuits are proven to outperform classical shallow circuits take the form of *search problems*. In Chapter 5, we consider *sampling problems*. As a step toward finding a sampling task that provides a shallow circuit quantum advantage, we consider the scenario that the shallow quantum circuit is allowed to take the $|GHZ_n\rangle = \frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle)$ state as input. We introduce a distribution D_n over n bits such that for the task of sampling from D_n, shallow quantum circuits with $|GHZ_n\rangle$ as input outperform shallow classical circuits that take uniformly random bits as input.

[1] See Definition 6 for a formal definition of a local objective function.

Chapter 2

Background

This chapter is intended to be a gentle introduction to low-depth, or “shallow” quantum and classical circuits.

2.1 Prelims/Notation

A quantum state on n qubits is a unit vector in $\mathbb{C}^{2^n}$. We use bra-ket (or Dirac) notation: using $|\psi\rangle$ to denote a quantum state, and $\langle\psi|$ its conjugate transpose. We define $|0\rangle = \begin{pmatrix}1 & 0\end{pmatrix}^\top$ and $|1\rangle = \begin{pmatrix}0 & 1\end{pmatrix}^\top$, and for each bitstring $x \in \{0,1\}^n$, we define $|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle$, where $\otimes$ is the tensor product. We refer to $\{|x\rangle : x \in \{0,1\}^n\}$ as the computational basis. Upon measuring state $|\psi\rangle$ in the computational basis, the probability that we get measurement outcome x is $\Pr_\psi[x] = |\langle x|\psi\rangle|^2$. For some bitstring $x \in \{0,1\}^n$ and index $i \in [n]$, we use x_i to denote the ith bit of x. Similarly, for some set of indices $S \subseteq [n]$, we use x_S to denote the substring of x indexed by S. So $x_S = (x_{S_1}, x_{S_2} \ldots, x_{S_{|S|}})$.

Throughout this thesis we will use the single-qubit Pauli operators X, Y, Z the Hadamard H and the Identity I

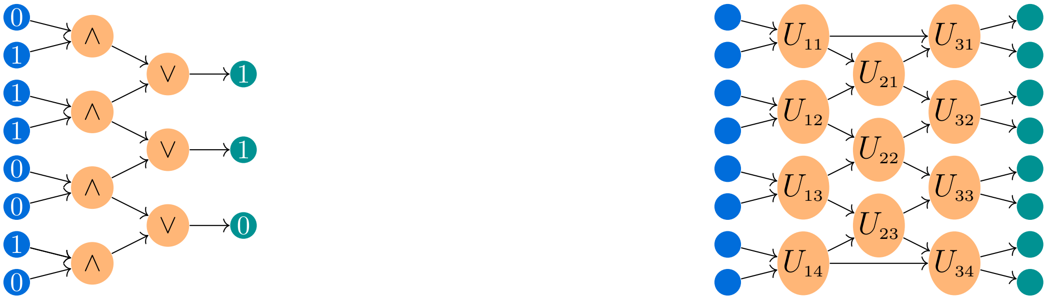
$$X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \quad Z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} \quad Y = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix} \quad H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \quad I = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \tag{2.1}$$

as well as the two-qubit $CNOT$ operator $CNOT = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$. The operator which applies a single qubit operator G to only the ith qubit of an n qubit state, is denoted G_i. For example $X_2 = I \otimes X \otimes I \otimes \cdots \otimes I$.

2.2 Circuits

Classical Circuit A classical circuit with m input bits, and n output bits, is a function $\mathcal{C} : \{0,1\}^m \to \{0,1\}^n$ that is represented by a directed acyclic graph G with m source vertices (no in-degree) and n sink vertices (no out-degree) of in-degree 1. The remaining vertices are *gates*. Each edge is assigned some bit value determined by its starting vertex. Each of the m input vertices are given some bit $\{0,1\}$ which then gets assigned to their outgoing edges. Each gate g with k incoming edges corresponds to some Boolean function $f_g : \{0,1\}^k \to \{0,1\}$ which is evaluated on the string $s \in \{0,1\}^k$ corresponding to the values of its incoming edges. The value $f_g(s)$ is then assigned to each of the outgoing edges of g. Finally, the output vertices, which each have in-degree 1, are assigned the same value as their incoming edge.

The depth of a circuit is the number of gates along the longest path between input and output vertices. A circuit has bounded fan-in k and fan-out k' if each vertex has in-degree at most k and out-degree at most k'. The size of a circuit, denoted $|\mathcal{C}|$ is the number of gates. Figure 2.1a depicts an example of a classical circuit with 8 input bits and 3 output bits.



(a) Classical Circuit with depth 2, fan-in 2 and fan-out 2

(b) Quantum Circuit with depth 3 and fan-in/fan-out 2.

Figure 2.1: Classical and quantum circuits as represented by directed acyclic graphs. Blue vertices represent input (qu)bits, and green vertices represent output (qu)bits. Orange vertices are gates. For clarity these circuits only include gates between nearest neighbors, however, this is not necessary in general.

Quantum Circuit A quantum circuit is also defined on a directed acyclic graph. The input and output are now quantum states, with input/output vertices representing qubits of the state. Each gate is a local unitary that acts on the qubits represented by the incoming edges. A k-qubit *quantum gate* is a $\mathbb{C}^{2^k \times 2^k}$ unitary that acts on a k-qubit state in $\mathbb{C}^{2^k}$. A quantum circuit's fan-in, fan-out, depth, and size are defined in the same way as for classical circuits. Note that a quantum gate has equal fan-in and fan-out, as we cannot copy the output of a gate onto multiple edges. In Figure 2.1b, we display a quantum circuit as a directed acyclic graph, analogous to the classical circuit description.

Equivalently, we can define a quantum circuit on n qubits with bounded fan-in/fan-out k and depth d as of a sequence of d unitaries $U_1 U_2 \dots U_d$ with each $U_i \in \mathbb{C}^{2^n \times 2^n}$ consisting of a tensor product of gates on k or fewer qubits.

For the remainder of this thesis, we assume that the input to the quantum circuit is a computational basis state, and the output is measured in the computational basis. Although we will continue to refer to these directed edges or paths in the circuits, we typically use the standard quantum circuit diagram as shown in Figure 2.2. Unless otherwise specified, a quantum circuit is assumed to have bounded fan-in 2.

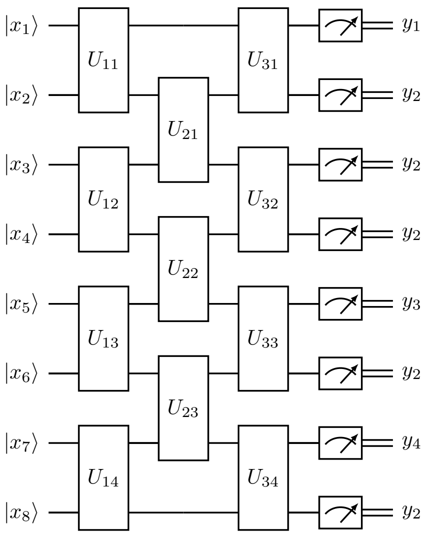


Figure 2.2: Standard quantum circuit diagram of the circuit described by the acyclic graph in Figure 2.1b. Each U_{ij} is a 2-qubit unitary. Time flows left to right. The input is a computational basis state $|x\rangle$ on the left. We apply the first layer of gates $U_{11} \otimes U_{12} \otimes U_{13} \otimes U_{14}$ in the first time step. The output is measured in the computational basis resulting in the classical output string $y \in \{0,1\}^8$. The double lines (or wires) indicate classical information.

2.3 Lightcones

In this section, we introduce lightcones, a useful tool for analyzing the limitations of low-depth circuits.

Definition 1 (Lightcone)**.** *Let x_j be an input (qu)bit to (classical or quantum) circuit $\mathcal{C}$ The forward lightcone $\overrightarrow{\mathcal{L}_{\mathcal{C}}}(x_j)$ is the subset of output (qu)bits that are connected to x_j via a path along the directed edges in the circuit. Similarly, for each output bit y_k, the backwards lightcone $\overleftarrow{\mathcal{L}_{\mathcal{C}}}(y_k)$ is the set of input bits that are connected to y_k. Moreover, for any subset of input bits T, and any subset of outputs S, let $\overrightarrow{\mathcal{L}_{\mathcal{C}}}(T) := \bigcup_{x_j \in T} \overrightarrow{\mathcal{L}_{\mathcal{C}}}(x_j)$ and $\overleftarrow{\mathcal{L}_{\mathcal{C}}}(S) := \bigcup_{y_k \in S} \overleftarrow{\mathcal{L}_{\mathcal{C}}}(y_k)$.*

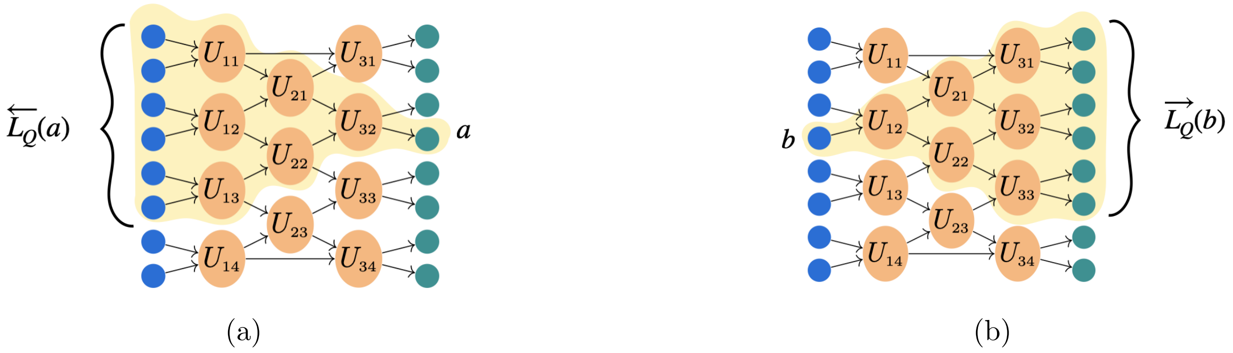
Figure 2.3 depicts a forwards and backwards lightcone of a quantum circuit.

Definition 2 (Blowup)**.** *Let $\mathcal{C}$ be a circuit with inputs I and outputs O, we define the forwards-blowup and backwards-blowup of $\mathcal{C}$ as*

$$B_f := \max_{x_j \in I} |\overrightarrow{\mathcal{L}_{\mathcal{C}}}(x_j)| \qquad\qquad B_b := \max_{y_k \in O} |\overleftarrow{\mathcal{L}_{\mathcal{C}}}(y_k)|$$

respectively. We define the blowup of $\mathcal{C}$ as $B := \max(B_f, B_b)$.

Adding a layer of gates (depth 1) to a circuit with bounded fan-in K and fan-out K', will increase the size of any backwards or forwards lightcone by at most a factor of K and



(a)

(b)

Figure 2.3: Lightcones for quantum circuit $\mathcal{Q}$. The yellow shaded inputs in (a) are the backwards lightcone of output qubit a. The yellow shaded outputs in (b) are the forwards lightcone of input qubit b.

K' respectively. Therefore, for a circuit $\mathcal{C}$ of depth D, bounded fan-in K and fan-out K' the forwards and backwards blowup can be upper bounded as follows

$$\begin{aligned} |\overleftarrow{\mathcal{L}_\mathcal{C}}(y_k)| &\leq K^D && \text{for each output } y_k \\ |\overrightarrow{\mathcal{L}_\mathcal{C}}(x_j)| &\leq K'^D && \text{for each input } x_j. \end{aligned}$$

Thus, $\mathcal{C}$ has blowup at most $\max(K^D, K'^D)$.

We will be focusing on quantum and classical circuits with constant depth $D = O(1)$ and fan-in $K = O(1)$ and therefore constant-sized backwards blowup $B_b = O(1)$. Small backwards blowup limits the ways in which a given output can depend on the circuit's inputs. Understanding these limitations, and how they differ between quantum and classical circuits, helps us characterize the abilities and limitations of shallow circuits.

As shown in Figure 2.4, a particular output bit y_k of a classical circuit $\mathcal{C}$ can be computed by the subcircuit induced by the paths connecting $\overleftarrow{\mathcal{L}_\mathcal{C}}(y_k)$ to y_k. The gates in this subcircuit are sometimes referred to as the gates in the backwards lightcone of y_k. Thus, for a classical circuit with small blowup, each output bit is only dependent on a few of the inputs

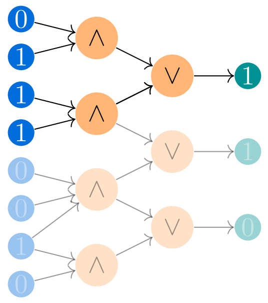


Figure 2.4: Classical subcircuit induced by the backwards lightcone of the first bit.

Do the same limitations hold for quantum circuits with small blowup? While we have defined quantum and classical lightcones in the same way, the distinction between an output's dependence on its backwards lightcone in the quantum versus the classical regime is subtle.

Upon measuring the output of a quantum circuit $\mathcal{C}$ in the computational basis, the resulting string measured will not be fixed by the input, but rather follow a probability distribution that is fixed by the input. For some subset S of measured output bits, rather than being *fixed* by the inputs in their backwards lightcone, the *marginal distribution* of S is fixed by the inputs in $\overleftarrow{\mathcal{L}_{C}}(S)$. This marginal distribution is the same distribution that results from the subcircuit consisting only of gates in the backwards lightcone of S as shown in Figure 2.5. Thus, for both quantum and classical circuits with small blowup, the marginal distribution of any single output bit does not indicate any global properties of the input – however, this is no longer true for quantum circuits once we condition on other outputs.

Lemma 1 (Properties of lightcones in quantum circuits)**.** *Consider a quantum circuit $\mathcal{C}$ acting on n qubits, taking input in the computational basis $|x\rangle$ for $x \in \{0,1\}^n$. Let $y \in \{0,1\}^n$ be string resulting from measuring the circuit output in the computational basis.*

1. *For each subset of output bits $S \subseteq [n]$, the marginal probability distribution of y_S is a function of the inputs in the backwards lightcone of S and is determined by the gates along the path from $\overleftarrow{\mathcal{L}_{\mathcal{C}}}(S)$ to S as shown in Figure 2.5.*

2. *If two subsets of outputs $S, S' \subseteq [n]$ have non-intersecting backwards lightcones $\overleftarrow{\mathcal{L}_{\mathcal{C}}}(S) \cap \overleftarrow{\mathcal{L}_{\mathcal{C}}}(S') = \emptyset$, then the distributions of y_S and $y_{S'}$ are independent.*

To see why (1) implies (2), note that if two subsets $S, S' \subseteq [n]$ have disjoint backwards lightcones, then the subcircuit induced by the gates of $\overleftarrow{\mathcal{L}_{\mathcal{C}}}(S \cup S')$ is disconnected. Therefore this circuit produces the outputs of S and S' independently.

While similar in flavor to the restriction on classical circuits imposed by lightcones, the quantum case is more general. As we will explore in Chapter 3, the less restrictive limitations imposed by lightcones in quantum circuits actually allows us to solve problems with constant-depth quantum circuits that cannot be solved by constant-depth classical circuits [BGK18, BGK18, WKST19].

The notion of a lightcone is present in both quantum and classical circuits but has implications with a subtle difference. At the crux of the distinction is that in contrast with classical circuits, the measurement outcome of the output of a quantum circuit is not fixed by the inputs. The reader may wonder, “what about randomized classical circuits?” Each randomized classical circuit (gates are probabilistic) can be equivalently represented by a deterministic classical circuit (the type we have so far been discussing) that takes an additional random bit-string as input. Therefore, we can always assume that a classical circuit is deterministic. This “randomness extraction” technique does not carry over to quantum circuits and is central to the separation in computational power of quantum and classical constant-depth circuits.

To highlight the usefulness of lightcones for lower bounding the depth of both quantum and classical circuits, we provide the following example.

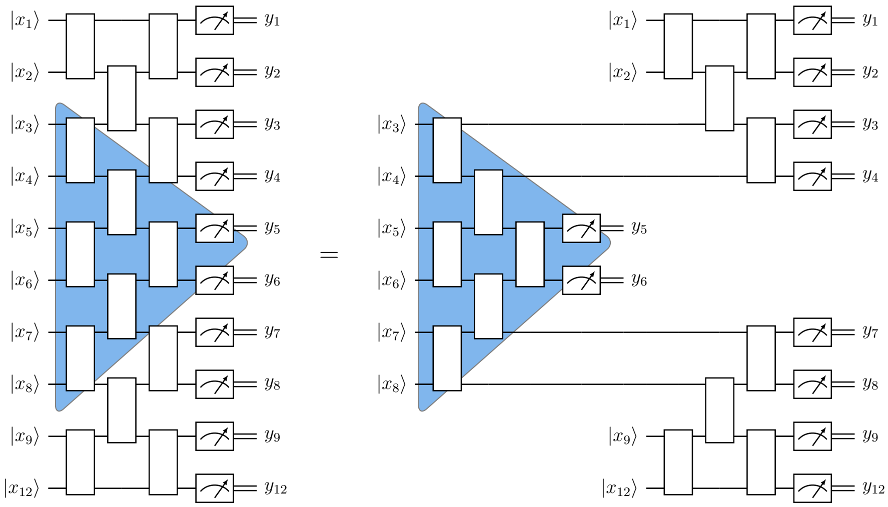


Figure 2.5: The marginal probability distribution of the measurement outcome of outputs y_5 and y_6 from quantum circuit $\mathcal{Q}$ depends only on the inputs in $\overleftarrow{\mathcal{L}_{\mathcal{Q}}}(y_5)$ (highlighted in blue). This figure illustrates how we can "stretch out" our circuit while maintaining the same measurement statistics. Note that in the second circuit diagram, we measure output y_k after only applying the gates in the blue region, acting only on the qubits in $\overleftarrow{\mathcal{L}_{Q}}(y_k)$.

Lower bound for parity using lightcones Consider the function parity_n which indicates whether a given string $x \in \{0,1\}^n$ has an even or odd number of ones:

$$\text{parity}_n(x) := \sum_{i=1}^{n} x_i \mod 2, \qquad\qquad x \in \{0,1\}^n.$$

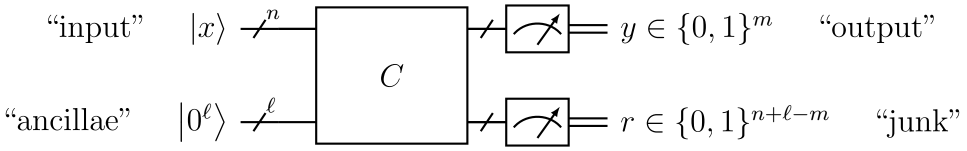
The parity_n function clearly depends on each of its input bits, since flipping any single bit of a string will flip its parity. Therefore, any classical circuit $\mathcal{C}$ that computes parity_n, has backwards-blowup n. Furthermore, if $\mathcal{C}$ has bounded fan-in K and depth D, its backwards-blowup is upper bounded by K^D, therefore $K^D \geq n$ and $D \geq \log_K n$. Therefore, any classical circuit with constant fan-in $K = O(1)$ that computes parity, has depth $\Omega(\log(n))$.

This lower bound also holds for quantum circuits, since the single output bit of the quantum circuit which indicates the parity of the input must have a marginal distribution that also depends on all n inputs.

2.4 Circuit Complexity

This thesis explores the difference in computational power between low-depth quantum and classical circuits. In this section, we define circuit classes and complexity classes involving constant depth circuits, and we categorize different types of computational tasks.

When considering the complexity of some computational task, we usually analyze how the resource requirements (time, space, etc.) of an algorithm for the task grows with the input size. For a given quantum or classical circuit, however, the circuit's input size is fixed. For this reason, we consider *circuit families.* A circuit family $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ is a family of circuits such that for each n, the circuit $\mathcal{C}_n$ takes n classical bits as input. For a quantum circuit that takes n input bits, we allow the quantum circuit to take additional ancillae qubits, so for input $x \in \{0,1\}^n$, the circuit acts on input state $|x\rangle \otimes |0^\ell\rangle$ for some ℓ. Moreover, we may choose to ignore some of the outputs of the circuit (sometimes referred to as "junk") and consider only the remaining bits as the output, as shown below.



Definition 3 ($\mathsf{NC^0}$/$\mathsf{QNC^0}$ circuits)**.** *A family of classical (quantum) circuits $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ is said to be a $\mathsf{NC^0}$ ($\mathsf{QNC^0}$) circuit family if there exists constants k, d, c such that for each $n \in \mathbb{N}$, $\mathcal{C}_n$ has fan-in at most k and depth at most d, and $|\mathcal{C}_n| = O(n^c)$.*

Recall that a quantum circuit has equal fan-in and fan-out, so $\mathsf{QNC^0}$ circuit families have constant-sized fan-out, whereas $\mathsf{NC^0}$ circuit families have unbounded fan-out.

We will often consider a quantum or classical circuit that takes n input bits– implicitly referring to a circuit family. Furthermore, we will say such a circuit is a $\mathsf{QNC^0}$ circuit or an $\mathsf{NC^0}$ circuit rather than stating this for the corresponding circuit family.

While quantum circuits can produce randomness, we can also produce randomness classically by flipping unbiased coins. Thus it is more interesting to consider classical circuits that also are given some random bits as input. $\mathsf{NC^0/rpoly}$ circuits are $\mathsf{NC^0}$ circuits that also can sample from a distribution on a polynomial number of bits that is independent of the input.

Before diving into classifying the computational power of quantum and classical shallow circuits, we first distinguish between different types of computational tasks.

Function: Given n input bits, compute some function $f : \{0,1\}^n \to \{0,1\}^m$, and outputs the result.

Decision Problem: A decision problem on n bits is characterized by a function with a single output bit $f : \{0,1\}^n \to \{0,1\}$. This one output bit indicates the decision: 1="yes", 0="no". Example: the OR function, "output 1 if and only if there is at least one 1 in the input." Often people will refer to a decision problem as a *language* $\mathcal{L}$, which is the set of all 1-inputs (or "yes" instances) $\mathcal{L} := \{x \in \{0,1\}^n : f(x) = 1\}$

Search (Relational) Problem: A search problem (or relational problem), can have many valid outputs for one input. It is characterized by a *relation* $R \subseteq \{0,1\}^n \times \{0,1\}^m$, such that for some input $x \in \{0,1\}^n$, an output $y \in \{0,1\}^m$ is valid if and only if $(x, y) \in R$. Example: "output a string with the same number of ones as the input string." Note that search problems are a generalization of functions/decision problems.

Sampling Problem: A sampling problem is characterized by a distribution $D(x)$ that may depend on some input $x \in \{0,1\}^n$, and is defined over m bits. The task is to output a sample $y \in \{0,1\}^m$ from the distribution $D(x)$. Example: "output uniformly at random an m-bit string with the same number of 1s as x."

The difference in computational power between QNC^0 and NC^0 (or $\mathsf{NC}^0/\mathsf{rpoly}$) circuits is not obvious. On one hand, an NC^0 circuit can fan-out or copy any input polynomially-many times, whereas a quantum circuit has constant-sized fan-out. Thus the simple task of computing the function which maps an n-bit string $(x_1, x_2, \ldots, x_n)$ to one with each bit XOR'ed with the first bit, $(0, x_2 \oplus x_1, \ldots, x_n \oplus x_1)$ can be computed with a NC^0 circuit but not with a QNC^0 circuit, since this requires that the forwards lightcone of x_1 contains all outputs (although this issue is resolved if we allow the quantum circuit to take multiple copies of the input basis qubits $|x_i\rangle$). Moreover, if a QNC^0 circuit computes a function, each output bit can be determined as a function of the inputs in its backwards lightcone, of which there are $O(1)$. So the value of each output bit can be determined with an NC^0 and therefore so can the entire function. Thus, NC^0 is strictly more powerful than QNC^0 for function problems. This is not true, however, for search problems and sampling problems.

Typically, complexity classes are defined by decision problems. However, since any decision problem in QNC^0 is also in NC^0, we will be focusing our attention on other types of problems such as relational and sampling problems. For this reason, we will abuse typical notation and say that some relation R is "in NC^0" (or "in QNC^0") if there exists an NC^0 (or QNC^0) circuit that satisfies the relation (with probability 1).

2.5 Hardness of classically sampling from shallow quantum circuits

The first wave of interest in shallow quantum circuits for quantum advantage was sparked by a line of work on the hardness of sampling from the output distribution of constant depth quantum circuits [TD02, FGHZ05, Aar04, JSB10]. For these results, the hardness proved is *conditional*. That is, they prove that the output distributions of these shallow circuits cannot be sampled by an efficient classical algorithm *if we assume* some conjecture(s) from complexity theory. This result concerns *exact* sampling, as opposed to *approximate* sampling, a more realistic framework which warrants discussion in Section 2.5.2.

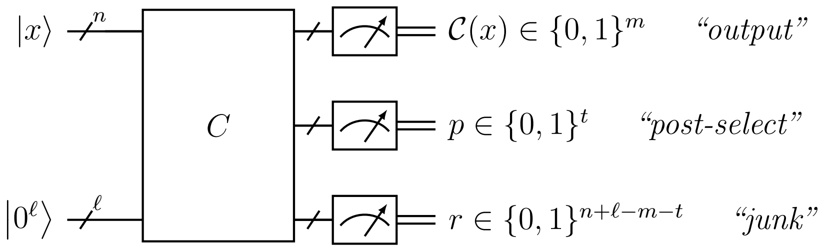
Combining results from [TD02, Aar04], we get the following theorem, whose hardness relies on the polynomial hierarchy (PH) not collapsing:

Theorem 1 ([TD02, Aar04])**.** *If there exists an efficient classical algorithm that can sample from the output distribution of any polynomial-sized quantum circuit with 1- and 2-qubit gates, and depth at most 3, then the polynomial hierarchy collapses to the third level:* $\mathsf{PH} = \Sigma_3$*.*

While we will not further discuss or define the complexity class PH and its third level Σ_3, we note that the collapse of the polynomial hierarchy is widely believed to be unlikely amongst complexity theorists and akin to $\mathsf{NP} = \mathsf{P}$. For more detail on the polynomial hierarchy, we refer the reader to [AB09].

At the core of the proof of Theorem 1 is that we can simulate the output distribution of any post-selected quantum circuit with a post-selected constant-depth quantum circuit.

Definition 4 (Post-selection)**.** *Post-selecting a circuit* $\mathcal{C}$*, refers to altering the output measurement statistics of the circuit by conditioning on the event that a subset of measurements is all 0. Formally, for some* $t \in [0, n + \ell - m]$*, let* $p \in \{0,1\}^t$ *denote the* t *"post-select" bits after the* m *output bits* $\mathcal{C}(x)$*. The output* y *post-selected on* p *follows the distribution* $\Pr[\mathcal{C}(x) = y | p = 0^t]$.



Post-selection is defined in the same way for classical circuits.

Models of computation that allow post-selection are somewhat nonphysical and may seem obscure since we cannot control which probabilistic outcome will occur. However, we will see how they can be used as tools to better understand their non-post-selected counterparts.

The following lemma is helpful for the proof of Theorem 1. We will first show how this Lemma can be used to prove Theorem 1, then we will prove the lemma in the following subsection.

Lemma 2 ([TD02], Compression via post-selection)**.** *For each post-selected quantum circuit* $\mathcal{C}$ *with* n *inputs, and size* $O(\mathsf{poly}(n))$*, there exists a post-selected quantum circuit with depth at most 3, and size* $O(\mathsf{poly}(n))$*, that implements* $\mathcal{C}$*.*

The proof of Lemma 2 is in Section 2.5.1

Definition 5 ($\mathsf{PostBQP}$, $\mathsf{PostBPP}$)**.** *For each decision problem* $\mathcal{L}$*, we say that* $\mathcal{L} \in \mathsf{PostBQP}$ *if and only if there exists a uniform* [1] *family of polynomial-sized quantum circuits* $\{\mathcal{C}_n\}$ *such that for each* $n \in \mathbb{N}$*,* $\mathcal{C}_n$ *has one "output" bit, and a "post-select" register* $p \in \{0,1\}^\ell$ *for some* $\ell(n) \in \{1, \ldots, n + m - 1\}$ *such that for each* $x \in \{0,1\}^n$*:*

$$\begin{aligned} \textit{if } x \in \mathcal{L}: &\quad \Pr\big[\mathcal{C}_n(x) = 1 | p = 0^\ell\big] \geq 2/3 \\ \textit{if } x \notin \mathcal{L}: &\quad \Pr\big[\mathcal{C}_n(x) = 1 | p = 0^\ell\big] \leq 1/3 \end{aligned}$$

PostBPP is defined in the same way, for uniform families of polynomial-sized probabilistic classical circuits $\{\mathcal{C}_n\}$.

Proof of Theorem 1.

Claim 1. *If there exists an efficient probabilistic classical algorithm, that takes as input the description of any depth-3 quantum circuit, and samples from its output distribution, then PostBPP = PostBQP.*

Proof. Since any polynomial-size classical circuit can be simulated by a polynomial-sized quantum circuit, it is always the case that $\mathsf{PostBPP} \subseteq \mathsf{PostBQP}$. So we just need to show the other direction $\mathsf{PostBQP} \subseteq \mathsf{PostBPP}$ under the assumption of efficient classical simulation of depth-3 quantum circuits.

Consider some decision problem $\mathcal{L} \in \mathsf{PostBQP}$. By the definition of PostBQP, we have that there exists a polynomial-sized quantum circuit family $\{\mathcal{C}_n\}$ such that for each n, and each $x \in \{0,1\}^n$,

$$\begin{aligned} \text{if } x \in \mathcal{L}: &\quad \Pr\big[\mathcal{C}_n(x) = 1 | p = 0^{\ell(n)}\big] \geq 2/3 \\ \text{if } x \notin \mathcal{L}: &\quad \Pr\big[\mathcal{C}_n(x) = 1 | p = 0^{\ell(n)}\big] \leq 1/3 \end{aligned}$$

Where p is the value of the post-selected register. By Lemma 2, there exists a uniform family of depth-3 circuits $\{\mathcal{C}'_n\}$ that, with the assistance of post-selection, have the same output statistics as $\{\mathcal{C}_n\}$.

$$\Pr\left[\mathcal{C}'_n(x) | p' = 0^{\ell'(n)}\right] = \Pr\left[\mathcal{C}_n(x) | p = 0^{\ell(n)}\right] \qquad \text{for each } n \in \mathbb{N}, x \in \{0,1\}^n \tag{2.2}$$

Where p' denotes the post-selected register of size $\ell'(n)$ for $\mathcal{C}'_n$.

Let $\mathcal{A}$ be the efficient classical algorithm that samples from depth-3 quantum circuits. The classical description of quantum circuit $\mathcal{C}(x)$ that is taken as input to $\mathcal{A}$ is denoted $\widetilde{C}(x)$. Let $y_{\mathcal{A}} \in \{0,1\}$ and $p_{\mathcal{A}} \in \{0,1\}^{\ell'(n)}$ denote the outputs of our classical algorithm $\mathcal{A}(\widetilde{\mathcal{C}}_n(x))$ corresponding to the output bit of $\mathcal{C}'_n$ and its post-select register p' respectively.

$$\Pr\left[y_{\mathcal{A}} = 1 | p_{\mathcal{A}} = 0^{\ell'}\right] = \Pr\left[\mathcal{C}'_n(x) | p' = 0^{\ell'}\right] = \Pr\left[\mathcal{C}_n(x) | p = 0^{\ell}\right]$$

Thus if we can run $\mathcal{A}(\widetilde{\mathcal{C}}_n(x))$ with post-selection, we will get the same output statistics as $\mathcal{C}_n$. Since $\mathcal{A}$ is a probabilistic, polynomial-time classical algorithm, $\mathcal{L} \in \mathsf{PostBPP}$. [2] □

Finally, with some facts from complexity theory, we may complete our proof. In 2004, Aaronson showed that PostBQP is equivalent to the complexity class PP [Aar04]. Since it had previously been shown by Toda in 1989 that $\mathsf{P}^{\mathsf{PP}} \supseteq \mathsf{PH}$ [Tod89], it follows from Claim 1 that

[1] A circuit family $\{\mathcal{C}_n\}$ is *uniform* if there exists an efficient classical algorithm that takes as input any $n \in \mathbb{N}$ and outputs a description of $\mathcal{C}_n$. The circuit families we construct maintain uniformity but we avoid further discussion of uniformity for the sake of clarity.

[2] The relevant uniform classical circuit family is $\{\mathcal{A} \circ \widetilde{\mathcal{C}_n}\}$, where $\circ$ denotes composition.

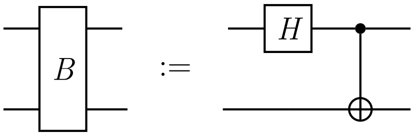
$\mathsf{P}^{\mathsf{PostBQP}} \supseteq \mathsf{PH}$. However, it was shown in 1997 by Han, Hemaspaandra, and Thierauf that $\mathsf{P}^{\mathsf{PostBPP}} \subseteq \Sigma_3$ [HHT97]. Therefore, if $\mathsf{PostBPP} = \mathsf{PostBQP}$, then the polynomial hierarchy collapses to the third level, $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{PostBQP}} = \mathsf{P}^{\mathsf{PostBPP}} \subseteq \Sigma_3$.

□

2.5.1 Proof of Lemma 2: Compression of quantum circuits with post-selection

In this section, we will prove Lemma 2 by constructing a method of compressing quantum circuits into depth-3 with post-selection. While Lemma 2 was first proved in [TD02], this compression scheme follows more closely to the construction in [FGHZ05].

Teleportation We define the *Bell gate* B as the following two-qubit gate:

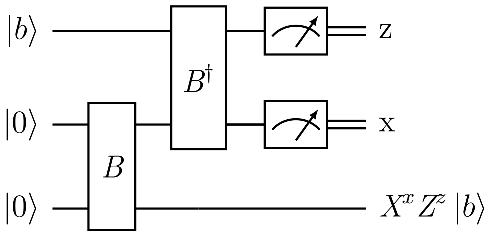


When applied to the $|0\rangle|0\rangle$ state, B prepares the *Bell state* $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

$$|00\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle \xrightarrow{CNOT_{1,2}} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

It can easily be verified that B maps the four computational basis states to the *Bell basis* $\{\frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle), \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle)\}$

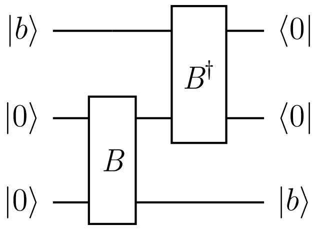
The circuit $B^{\dagger}_{1,2}B_{2,3}$ allows us to "teleport" a single qubit state with some Pauli errors that depend on our measurements.



(2.3)

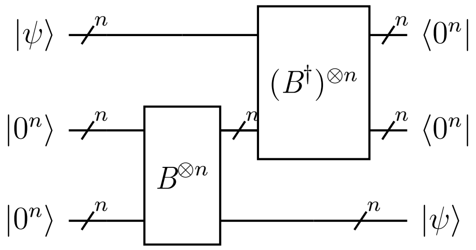
Our measurement outcomes $(z, x) \in \{0, 1\}^2$ on the first two bits are uniformly random, but once measured, the last qubit is $X^x Z^z |b\rangle$, where we use the convention $G^0 = I$, $G^1 = G$ for any gate G.

Post-selection While quantum teleportation is interesting in its own right, we will be interested in the scenario where we allow ourselves to assume the measurement outcomes on the first two bits are both 0. This is called post-selection.



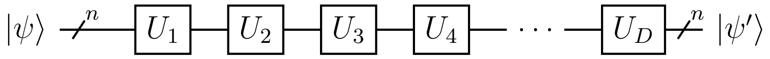
(2.4)

If the first two qubits are measured as 0, as indicated by $\langle 0|$ in the above diagram, then the qubit $|b\rangle$ is teleported to the third qubit without error. This is sometimes referred to as *non-adaptive teleportation* [TD02]. Furthermore, we can teleport an entire n qubit state $|\psi\rangle$ by implementing Equation (2.4) for each qubit.



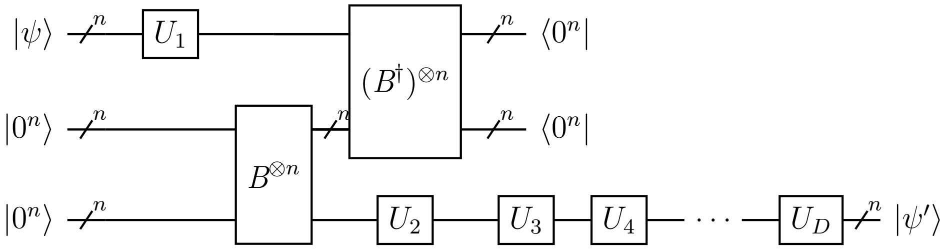
(2.5)

We refer to this as our n-qubit *teleportation gadget*. Suppose we would like to implement a depth D circuit $U_D U_{D-1} \dots U_2 U_1$. Where each U_i can be implemented in depth 1.



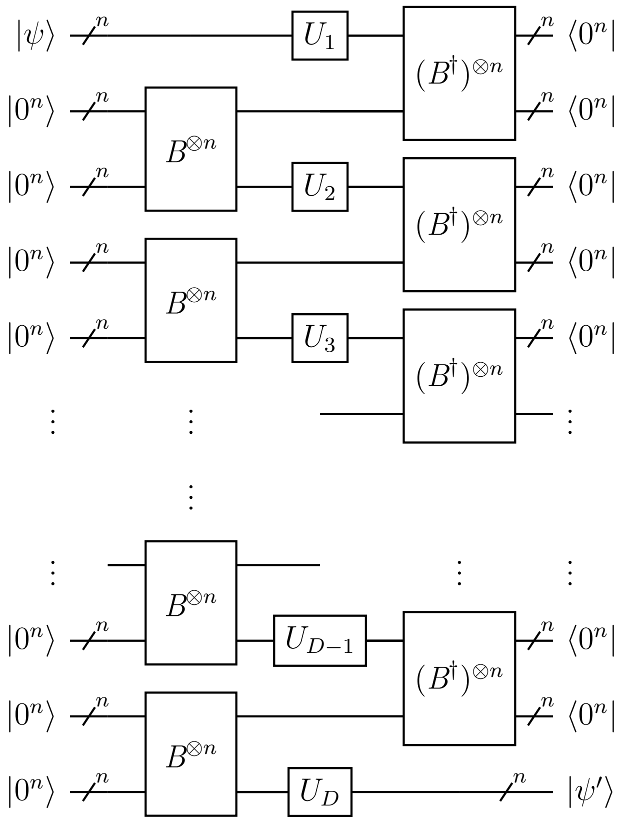
(2.6)

With 2n ancilla qubits in the $|0\rangle$ state, and post-selection, we can teleport the state after the first gate with our n-qubit teleportation gadget as follows.



(2.7)

By iteratively applying our teleportation gadget after each U_i for $i \in \{1, 2, \dots, D\}$, we transform our original circuit into the following one, with $2nD$ ancillae and $2nD$ post-selected bits.



(2.8)

Therefore, for each quantum circuit $\mathcal{C}$ with n inputs, and depth D, we can implement it with a post-selected quantum circuit $\mathcal{C}'$ with depth 3, and size at most $3(n + 2nD)$.

It follows that, for each quantum circuit $\mathcal{C}$ with n inputs and size $O(\mathsf{poly}(n))$, and thus depth $D = O(\mathsf{poly}(n))$, we can implement $\mathcal{C}$ with a depth 3 post-selected quantum circuit of size $O(\mathsf{poly}(n))$.

2.5.2 Exact vs approximate sampling

What does this hardness of sampling result tell us? Even assuming the relevant complexity conjectures are true, can we claim this as a victory for quantum advantage? Note that the result as stated only considers the complexity of *exact* sampling. That is, the sampler must sample from the exact same distribution as the described quantum circuit. But can a quantum computer even do this? Certainly not the ones that currently exist. It is unreasonable to assume that any physical system, quantum or classical, will have *zero* error (interaction with its environment, or control errors). We should instead be considering *approximate* sampling.

Fortunately, the hardness of sampling statement of Theorem 1 can be easily extended to apply to hardness of sampling up to *multiplicative* (or *relative*) error. We say that the classical algorithm $\mathcal{A}$ approximately samples from the same distribution as the circuit $\mathcal{C}$ with multiplicative error $\epsilon \in (0, 1)$ if for each input x and output y

$$|\Pr[\mathcal{A}(x) = y] - \Pr[\mathcal{C}(x) = y]| \leq \epsilon \Pr[\mathcal{C}(x) = y]. \tag{2.9}$$

This notion of error is still quite strict. Alternatively, we might consider *additive* error, which is more realistic considering the error rates expected in near-term quantum devices. We say that $\mathcal{A}$ approximately samples with *additive* (or total variation distance) error $\epsilon \in (0,1)$ if for each input x,

$$\frac{1}{2}\sum_{y} |\Pr[\mathcal{A}(x) = y] - \Pr[\mathcal{C}(x) = y]| \leq \epsilon. \tag{2.10}$$

In contrast to multiplicative error, the hardness of sampling with additive error is less understood though work is ongoing [BMS16, BFLL22, MT20].

2.6 The Quantum Approximate Optimization Algorithm

The hardness of classically sampling from the output distributions of shallow quantum circuits shown in the previous section, provides strong evidence that there exists constant-depth quantum circuit families whose measurement statistics cannot be reproduced by an efficient classical algorithm– however, it does not tell us specifically which circuits are hard to sample, nor if they are *useful* for anything. The Quantum Approximate Optimization Algorithm (QAOA) aims to get us closer to answering both of these questions. The QAOA, introduced by Farhi, Goldstone, and Gutmann [FGG14], is a quantum optimization algorithm that is considered a potential candidate for a *useful* near-term quantum advantage.

The simplest version of the QAOA can be implemented in constant-depth for some problems. Even for QAOA circuits that are constant depth, it has been shown, using similar techniques as in Section 2.5 that the QAOA is hard to simulate classically in general [FH16]. Furthermore, Farhi et al. showed there are certain optimization problems for which the QAOA has performance guarantees [FGG14]. However, there still remains a gap between quantum advantage and usefulness as there is no known optimization problem for which the QAOA outperforms classical methods. In fact, for some problems, it has been shown that even constant-depth classical circuits can outperform constant-depth QAOA [Has19].

Optimization Problems An optimization problem parameterized by input $x \in \{0,1\}^m$ is characterized by an objective (or "cost") function $C_x : \{0,1\}^n \to \mathbb{R}$. The goal is to output a value for $y \in \{0,1\}^n$ that minimizes $C_x(y)$.

In particular, the QAOA is usually considered for *local*-objective functions as these can be implemented in constant-depth.

Definition 6 (Local-optimization problem). *An objective function $C_x(y) = \sum_i C_{x,i}(y)$ is* local *if there exists some $c = O(1)$ such that*

(i) each term $C_{x,i}(y)$ is a function of at most c variables $x_1, \dots, x_m, y_1, \dots, y_n$ (input and output), and

(ii) each output variable $y_1, \dots, y_n$ is involved in at most c terms.

For a physical interpretation, we can consider the objective function C_x as a (classical) Hamiltonian, and each term $C_{x,i}$ as an *interaction* between some variables $x_1, \dots x_m$, $y_1, \dots, y_n$. This definition of *locality* is that for each $i \in [n]$, the variable y_i interacts with a constant number of other variables.

A special case of local-optimization problems are problems defined over graphs of bounded degree.

For example, consider the MAXCUT problem which takes as input some subgraph of a bounded degree graph. For some graph $G = (V, E)$, $\textsc{MaxCut}_G$ takes as input $x \in \{0,1\}^{|E|}$ indicating a subset of edges $E' \subseteq E$, and the task is to output an assignment to the vertices $y \in \{0,1\}^n$ to maximize the number of edges in E' across the cut. The cost function consists of a term for each edge $(u, v) \in E$ such that $C_{(u,v)}(y)$ is set to -1 if $y_u \neq y_v$ and $x_{(u,v)} = 1$, and otherwise, it evaluates to 0.

$$C_{(u,v)}(y) = x_{(u,v)}(y_u + y_v - 2y_u y_v) \qquad \text{for each } (u,v) \in E \tag{2.11}$$

We consider an algorithm to be optimal, if for each $x \in \{0,1\}^m$, it *always* outputs an $y \in \{0,1\}^n$ that achieves the optimal value:

$$C_x^* = \max_y C_x(y). \tag{2.12}$$

Even for the seemingly-simple MAXCUT problem, this task is known to be NP-hard. Thus it is unlikely there exists an efficient algorithm, quantum or classical, to solve this problem exactly. And so, typically, we aim instead to find a good *approximate* solution: a string $y \in \{0,1\}^n$ with a large approximation ratio $C_x(y)/C_x^*$.

We now define the QAOA for a particular local objective function $C(y)$.

The QAOA The local objective function $C(y)$ gives rise to a *cost Hamiltonian* which we will denote as C

$$C\,|y\rangle = C(y)\,|y\rangle \qquad \text{for each } y \in \{0,1\}^n$$

We also define the *mixing Hamiltonian* B as $B := \sum_{j=1}^n X_j$. The QAOA starts the algorithm in the uniform superposition state $|+\rangle^{\otimes n}$ and proceeds by alternating between applying the unitary evolution of C and B.

$$U_C(\gamma) := \exp(i\gamma C) \tag{2.13}$$

$$U_B(\beta) := \exp(i\beta B) = \prod_{j=1}^n \exp(i\beta X_j) \tag{2.14}$$

The mixing unitary $U_B(\beta)$ can be implemented in depth 1 since each of the $\exp(i\beta X_j)$ gates act on different qubits. Furthermore, since C a local objective function, then $U_C(\gamma)$ can also be implemented in constant depth.

For example, the cost Hamiltonian for the MaxCut problem is

$$C = \frac{1}{2} \sum_{(u,v)\in E} 1 - Z_u Z_v \tag{2.15}$$

Since each term commutes, the corresponding unitary (up to a global phase) is

$$U_C(\gamma) = \prod_{(u,v)\in E} \exp\Bigl(-i\frac{\gamma}{2} Z_u Z_v\Bigr) \tag{2.16}$$

Note that the gates $\exp\bigl(-\frac{\gamma}{2} Z_u Z_v\bigr)$ commute, so the order in which we apply them does not matter. Since each edge corresponds to a gate acting on its endpoints, if the graph has degree at most d then the gates can all be applied in depth $d+1$ according to an edge coloring [Viz65].

The QAOA for depth p applies each of these unitaries p times. The algorithm is parameterized by $2p$ variables: $\vec{\gamma} = (\gamma_1, \ldots, \gamma_p)$ and $\vec{\beta} = (\beta_1, \ldots, \beta_p)$ which determine the unitary evolution time. The final state output by the QAOA is

$$\left|\vec{\gamma}, \vec{\beta}\right\rangle := U_B(\beta_p) U_C(\gamma_p) \ldots U_B(\beta_2) U_C(\gamma_2) U_B(\beta_1) U_C(\gamma_1) \left|+\right\rangle^{\otimes n}.$$

With the expected value of the cost function C denoted as

$$\langle C \rangle_{\vec{\gamma},\vec{\beta}} := \left\langle \vec{\gamma}, \vec{\beta} \right| C \left| \vec{\gamma}, \vec{\beta} \right\rangle.$$

Since this thesis is concerned with circuit complexity at constant depth, we will focus on the QAOA at constant depth $p = O(1)$.

Constant depth QAOA for quantum advantage Even with just one iteration of QAOA, $p = 1$, Farhi and Harrow showed, using similar techniques as shown in Section 2.5, that sampling exactly from the output distribution of an arbitrary such $QAOA_{p=1}$ circuit cannot be done classically efficiently unless the polynomial hierarchy collapses. Interestingly though, for any $p = O(1)$, given as input $\vec{\gamma}, \vec{\beta}$ we can efficiently compute the expectation value of the cost function $\left\langle \vec{\gamma}, \vec{\beta} \right| C \left| \vec{\gamma}, \vec{\beta} \right\rangle$. This is because the expectation is the sum of the expectation of each term of C

$$\langle C \rangle_{\vec{\gamma},\vec{\beta}} = \sum_{e\in E} \langle C_e \rangle_{\vec{\gamma},\vec{\beta}} \tag{2.17}$$

The expectation of each term C_e only depends on the marginal distribution of these outputs in e which can be calculated efficiently by simulating the sub-circuit induced by the backwards lightcone of e. Moreover, there are at most $O(\mathsf{poly}(n))$ terms. So although the QAOA requires some parameter optimization of β, γ this can be done classically before running the QAOA.

While Farhi and Harrow's hardness of sampling result tells us that there exist QAOA circuits that are hard to simulate exactly, it does not have implications as to whether these circuits are any good at their corresponding optimization problems.

A more appropriate sampling task to help characterize whether there the QAOA has an *algorithmic* advantage would be sampling from a distribution that is consistent with the output of the QAOA circuit's marginal distributions along the edges. Since this distribution will have the same performance on the related optimization problem.

In Chapter 4, we investigate the limitations of shallow quantum circuits algorithms, such as the QAOA, for solving local optimization problems.

2.7 Shallow circuit separations

In this chapter, we reviewed the proof of [TD02, Aar04] for the hardness of sampling from shallow quantum circuits, which relies on some complexity theoretic assumptions (*conditional*). The rest of this thesis will focus on *unconditional* proofs of separation between shallow classical and quantum circuits – that is, showing that a QNC^0 circuit can solve a problem that NC^0 circuits cannot, without relying on any complexity-theoretic assumptions. In Chapter 3, we review such an unconditional separation from [BGKT20]. In Chapter 5, we investigate whether we can prove a shallow circuit unconditional separation for preparing a specific distribution that does not depend on input.

In this chapter, we also discussed local optimization problems and the QAOA, a shallow quantum circuit algorithm for local optimization problems. In Chapter 4 we investigate whether we can achieve a shallow circuit separation for these problems, proving some limitations of shallow quantum circuits.

Chapter 3

Shallow circuit quantum advantage: the 1D Magic Square Problem

In this chapter, we review previous work by Bravyi, Gosset, Koenig, and Tomamchel [BGKT20].

In 2018 Bravyi, Gosset and Koenig showed the first unconditional separation between shallow quantum and classical circuits. They proved a *relational* separation. That is, there exists a relational problem that can be solved by a shallow quantum circuit (QNC^0), but cannot be solved by a randomized shallow classical circuit ($\mathsf{NC}^0/\mathsf{rpoly}$). This relational problem is called the 2D Hidden Linear Function Problem, or 2D-HLF.

Theorem ([BGK18]). *The relational problem 2D-HLF can be solved exactly by a QNC^0 circuit family, yet no $\mathsf{NC}^0/\mathsf{rpoly}$ circuit family can solve the 2D-HLF with probability greater than $7/8$ on each input.*

Soon after, Bravyi, Gosset, Koenig, and Tomamichel [BGKT20] showed that another problem, the 1D Magic Square Problem (1D-MSP), also has a shallow circuit separation, for which the constant-depth quantum circuit that solves it can be implemented with geometrically local gates in one dimension– that is the circuit only consists of nearest neighbor gates between qubits arranged on a line.

Theorem 2 ([BGKT20]). *For each n the relational problem, 1D-MSP$_n$, can be solved exactly by a QNC^0 circuit with 1D geometrically local gates, yet there exists a set of inputs S_n with $|S_n| = \mathsf{poly}(n)$ such that each $\mathsf{NC}^0/\mathsf{rpoly}$ will not solve the problem with probability at greater than $9/10$ over the uniform choice of input from S_n.*

Furthermore, they show an extension of the 1D-MSP maintains a separation even when the shallow quantum circuit is *noisy*; for which the corresponding quantum circuit can be implemented with geometrically local gates in three dimensions. This chapter, however, will only be concerned with the original 1D Magic Square Problem, and will not further discuss the noise-resilient extension.

Both the 2D Hidden Linear Function and the 1D Magic Square Problem incorporate specific nonlocal games to achieve a circuit separation. While both results employ a similar general framework, this chapter will showcase this technique by detailing the proof for the 1D Magic Square Problem (Theorem 2).

The 1D Magic Square Problem uses a generalization of the magic square game to provide classical lower bounds. The magic square game is a game played with two coordinating players Alice and Bob, who are not allowed to communicate but must each provide answers to some given questions. It is known that if Alice and Bob are allowed shared entanglement, they can win this game with probability 1 whereas if they are only allowed shared randomness, they cannot win with probability greater than 8/9.

The overview of the proof of Theorem 2 in [BGKT20] is as follows: First, they construct a constant-depth 1D quantum circuit and a subset of inputs S_n such that choosing an input randomly from S_n corresponds to randomly choosing two registers $j < k \in [n]$ of the circuit to play a variation of the magic square game. The 1D-MSP is then defined as the input/output relations of this quantum circuit. To prove classical hardness, they then show that for any constant depth classical circuit with bounded fan-in, there are a large fraction of pairs (j, k) of registers (and therefore inputs in S_n) with non-intersecting lightcones. For these inputs, such a circuit implements a classical strategy for the magic square game and therefore is correct with probability at most 8/9.

Before constructing the quantum circuit which defines the 1D-MSP, we first introduce its components. In the rest of this section, we formally define the (generalized) magic square game, as well as an entangled state that can be prepared by a constant-depth 1D circuit. In Section 3.1 we define the 1D-MSP, and the constant-depth 1D quantum circuit that solves it. In Section 3.2 we prove the classical hardness of the 1D-MSP for constant-depth classical circuits with bounded fan-in.

The magic square game The Magic square game is a 2-player nonlocal game with coordinating players Alice and Bob, who are not allowed to communicate. Alice is given $\alpha \in \{1, 2, 3\}$ and outputs $a \in \{+1, -1\}^3$. Bob is given $\beta \in \{1, 2, 3\}$ and outputs $b \in \{+1, -1\}^3$. They "win" the game if Alice's string has odd weight $a_1 a_2 a_3 = -1$, Bob's string is even $b_1 b_2 b_3 = 1$ and $a_\beta = b_\alpha$. This game can also be described in terms of filling out a 3×3 grid array (or square, as the name indicates). Alice is given a column index and Bob is given a row index. Alice and Bob then output an assignment for their row and column respectively. They "win" if Alice's column is odd, Bob's row is even, and they give the same value for their overlapping cell (β, α).

A perfect deterministic classical strategy can be described by some assignment to each cell in the square satisfying the row and column parity constraints. However, this is impossible, since the row constraints tell us that the parity of the entire array will be even, while that of the columns requires an overall odd parity.

Thus, if Alice and Bob always output a column and row satisfying their parity constraints, there will always be some cell where they are not consistent. For example consider the strategy below, where Alice and Bob are consistent on all but the last cell in the array.

-1	-1	1
-1	1	-1
-1	1	?

(3.1)

In order to maintain the parity constraints, Alice will assign $+1$ while Bob will assign -1 to this last cell. In this case, Alice and Bob will win with probability $8/9$ over the uniform choice of $\alpha, \beta \in [3]$, which is in fact an optimal classical strategy. This classical bound of success probability $\leq 8/9$ also holds for randomized classical strategies, since such strategies correspond to a probability distribution over deterministic strategies, even if Alice and Bob have shared randomness.

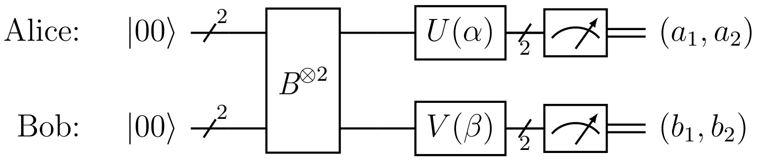
Perhaps surprisingly, if Alice and Bob share two entangled Bell pairs $|\Phi\rangle \otimes |\Phi\rangle = (\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle))^{\otimes 2}$, they can win the Magic Square game with probability 1. A winning quantum strategy now corresponds to filling out the square with observables with eigenvalues in $\{-1, +1\}$ such that the observables of each row and column commute and that the product of each row is $+I$ and the product of each column is $-I$. Such an optimal strategy is shown below.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$
$\beta = 1$	$X \otimes I$	$I \otimes X$	$X \otimes X$
$\beta = 2$	$I \otimes Z$	$Z \otimes I$	$Z \otimes Z$
$\beta = 3$	$-X \otimes Z$	$-Z \otimes X$	$Y \otimes Y$

(3.2)

Alice and Bob will win the game with probability 1 by measuring their states according to these measurement bases.

Equivalently, Alice and Bob can implement this strategy by applying classically controlled gates $U(\alpha), V(\beta)$, that change the basis according to (3.2), then measuring in the computational basis.



(3.3)

Where we use the convention that measuring $|0\rangle$ in the computational basis results in measurement outcome $+1$ and measuring $|1\rangle$ results in measurement outcome -1 (corresponding to measuring in the $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ basis). Alice and Bob then determine their third measurement outcomes from the first two: $a_3 = -a_1 a_2$ and $b_3 = b_1 b_2$. An explicit definition of $U(\alpha)$ and $V(\beta)$ are shown in Figure 3.1.

Next we define a parameterized version of the magic square game, which has the same optimal quantum strategy, but for a different shared state between Alice and Bob.

Definition 7 (Generalized magic square game $MSG(s, t, s', t')$)**.** *The generalized magic square game is parameterized by $s, t, s', t' \in \{-1, +1\}$. Just as in the original magic square game, Alice and Bob are given inputs $\alpha, \beta \in \{1, 2, 3\}$, and each output a string in $\{+1, -1\}^3$ of odd and even weight respectively, however now their intersecting bits a_β, b_α, rather than*

α/β	$U(\alpha)$	$V(\beta)$
0	$I \otimes I$	$I \otimes I$
1	$H \otimes I$	$H \otimes H$
2	$(H \otimes I) \cdot SWAP$	$SWAP$
3	$(H \otimes I) \cdot CNOT$	$(H \otimes H) \cdot CZ \cdot (Z \otimes Z)$

Figure 3.1: Change of basis gates to implement the optimal strategy for the magic square game on input α, β. Applying these gates then measuring in the computational basis is equivalent to measuring in the bases specified in (3.2). $SWAP$ is a swap operator ($SWAP|x_1x_2\rangle = |x_2x_1\rangle$ for each $x_1, x_2 \in \{0,1\}$), and CZ is the controlled-Z gate ($CZ = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z$). While α, β are never set to 0 in the Magic Square Game, we have included this as a possible input as it will be useful later.

being equal must satisfy $a_\beta b_\alpha = f_{\alpha,\beta}(s,s',t,t')$ for $f_{\alpha,\beta}(s,s',t,t')$ defined as follows.

$$f_{\alpha,\beta}(s,t,s',t') := \begin{array}{c|c|c|c} & \alpha = 1 & \alpha = 2 & \alpha = 3 \\ \hline \beta = 1 & s & s & ss' \\ \hline \beta = 2 & t' & t & tt' \\ \hline \beta = 3 & -st' & -s't & ss'tt' \end{array} \tag{3.4}$$

Note that the case $s, s', t, t' = 1$ is the original magic square game.

Suppose that rather than sharing two Bell pairs $|\Phi\rangle \otimes |\Phi\rangle$, Alice and Bob share two *poor man's Bell pairs* $|\Phi_{s,t}\rangle \otimes |\Phi_{s',t'}\rangle$. Where a *poor man's Bell pair* for some $s, t, s', t' \in \{+1, -1\}$ is defined as

$$|\Phi_{s,t}\rangle := (X^{(1-t)/2} Z^{(1-s)/2} \otimes I)\, |\Phi\rangle\,.$$

Claim 2. *If Alice and Bob play the generalized magic square game with parameters (s, t, s', t') following the measurements in Equation (3.2), on the poor man's Bell pairs $|\Phi_{s,t}\rangle \otimes |\Phi_{s',t'}\rangle$, they will win with probability 1.*

Proof. Measuring the state $|\Phi_{s,t}\rangle \otimes |\Phi_{s',t'}\rangle$ in the basis O is equivalent to measuring $|\Phi\rangle \otimes |\Phi\rangle$ in the basis $\tilde{O} := U^\dagger O U$ where

$$U := (X^{(1-t)/2} Z^{(1-s)/2} \otimes I) \otimes (X^{(1-t')/2} Z^{(1-s')/2} \otimes I). \tag{3.5}$$

Note that U only acts nontrivially on Alice's qubits (the first of each pair). Therefore, each of Bob's measurement bases M_B, commute with U and are the same as before: $\tilde{M_B} = U^\dagger M_B U = M_B$. Alice's measurements on her two qubits, however, are conjugated by $U_A = X^{(1-t)/2} Z^{(1-s)/2} \otimes X^{(1-t')/2} Z^{(1-s')/2}$. Since X and Z anticommute, applying a Pauli Z flips X measurement results, and applying an X flips Z measurement results. Therefore, following the measurements from the optimal strategy for the original game Equation (3.2), on the poor man's Bell pairs $|\Phi_{s,t}\rangle \otimes |\Phi_{s',t'}\rangle$ is equivalent to measuring the error-free Bell pair $|\Phi\rangle \otimes |\Phi\rangle$

with Alice's measurements changed to the following:

$\alpha = 1$	$\alpha = 2$	$\alpha = 3$
$sX \otimes I$	$s'I \otimes X$	$ss'X \otimes X$
$t'I \otimes Z$	$tZ \otimes I$	$tt'Z \otimes Z$
$-st'X \otimes Z$	$-s'tZ \otimes X$	$ss'tt'Y \otimes Y$

(3.6)

and Bob's measurements the same as Equation (3.2). Note that the row and parity constraints are still satisfied since the operators in each of Alice's columns commute and multiply to $-I$. Moreover, Alice's measurement for entry (α, β) in the square is equal to that of Bob's times $f_{\alpha,\beta}(s, t, s', t')$, therefore the condition that $a_\beta b_\alpha = f_{\alpha,\beta}(s, t, s', t')$ is always satisfied. Therefore, this corresponds to an optimal strategy to win the generalized magic square game with parameters (s, t, s', t'). □

Poor Man's Bell Pair in 1D Constant Depth To prepare the Bell state $|\Phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ between the first and last of n qubits arranged on a line requires depth $\Omega(\log n)$ if we only have nearest-neighbor gates. This is because the backwards lightcones of the two qubits must intersect. However, we can instead prepare a *poor man's Bell pair* in constant depth. Which is a Bell pair with some Pauli error that is determined by measurements of the qubits in between the pair.

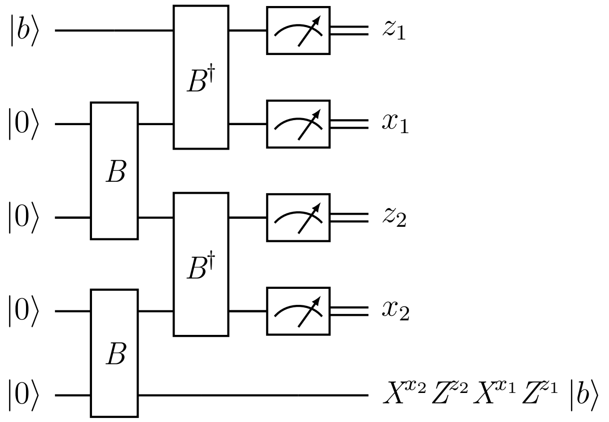
$$|\Phi_{s,t}\rangle = (X^{(1-t)/2} Z^{(1-s)/2} \otimes I) |\Phi\rangle$$

Where $s, t \in \{-1, +1\}$ are uniformly random and depend on measurement outcomes of the qubits in between our pair. This type of state turns out to be a very helpful resource state for proving shallow circuit separations. Although the backwards lightcones of each qubit in the pair may be disjoint (and thus their marginal distributions are independent), by conditioning on the measurements of the intermediate qubits, we can exhibit quantum nonlocality in constant depth!

We now describe two different methods for preparing a poor-man's Bell pair.

Method 1: Teleportation/Entanglement Swapping We can use the teleportation circuit in Equation (2.3) and link multiple such teleportation gadgets similar the circuit compression scheme in Section 2.5.1 but in this instance we will not be utilizing post-selection. Note that when we "teleport" a qubit twice getting measurement outcomes

$x_1, z_1, x_2, z_2 \in \{0,1\}$ as shown below,



(3.7)

the final teleported state will be proportional to $X^{x_1 \oplus x_2} Z^{z_1 \oplus z_2}\,|b\rangle$, since X and Z anticommute. Moreover, we can continue to teleport the state a total of n times with measurement outcomes $(z_1, \ldots, z_n) \in \{0,1\}^n$ and $(x_1, \ldots, x_n) \in \{0,1\}^n$, resulting in a final state $X^x Z^z\,|b\rangle$ where $x = \bigoplus_{i=1}^n x_i$ and $y = \bigoplus_{i=1}^n y_i$.

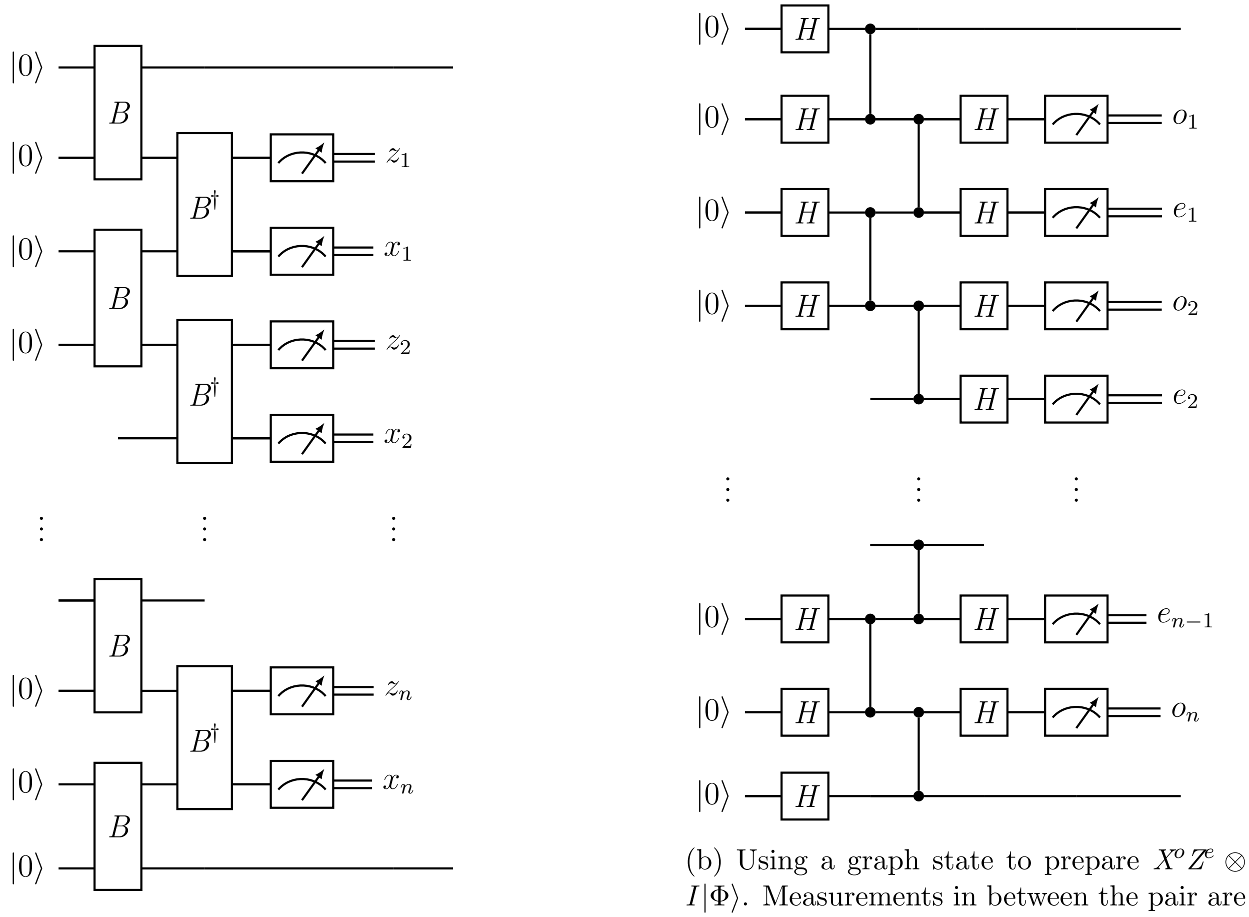
Applying this chain of teleportations to the second qubit of a Bell pair $|\Phi\rangle = B|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, we prepare a poor man's Bell pair as shown in Figure 3.2a. Note that this circuit can be implemented by only using the $CNOT$ and H gates (Since $B = (H \otimes I)CNOT$).

Method 2: Graph State Another method for preparing a poor man's Bell pair on a line is by measuring a graph state. While we have not yet defined graph states, the resulting circuit is shown in Figure 3.2b. This circuit only requires H and CZ gates. We will not go into detail on this method in this section since we will instead utilize the Teleportation-based construction to be consistent with [BGKT20]. Yet it is useful to note that we get similar behavior from a graph state, as this is the method used in [BGK18, WKST19]

Using either the teleportation method or the graph state method shown in Figure 3.2, we can prepare the poor-man's Bell state $|\Phi_{s,t}\rangle$ in 1D in constant depth, with s, t determined by the measurement outcomes of the intermediate qubits. Thus, if we prepare two such states in this way $|\Phi_{s,t}\rangle \otimes |\Phi_{s',t'}\rangle$ and measure them according to the optimal strategy for the (original) magic square game, we will win the generalized magic square game parameterized by (s, t, s', t') exactly. In the next section, we introduce a quantum circuit that does exactly this.

3.1 The 1D Magic Square Problem

The 1D Magic Square Problem is a relational problem corresponding to a variation of playing the magic square game on a line. It can be formally defined by the input-output relation of the quantum circuit shown in Figure 3.3. There are n "Alice" registers (A_i) and n "Bob" registers (B_i), each holding 2 qubits. For each $i \in [n]$, the register A_i takes a (classical) input



(a) Using teleportation, or entanglement swapping to prepare the state $X^x Z^z \otimes I|\Phi\rangle$. Measurements in between the pair are $x_i, z_i \in \{0,1\}$ for $i \in [n]$, and Pauli correction bits are $x = \bigoplus_{i=1}^n x_i$ and $z = \bigoplus_{i=1}^n z_i$.

(b) Using a graph state to prepare $X^o Z^e \otimes I|\Phi\rangle$. Measurements in between the pair are $o_i, \in \{0,1\}$ for $i \in [n]$ and $e_i \in \{0,1\}$ for $i \in [n-1]$, and Pauli correction bits are $o = \bigoplus_{i=1}^n o_i$ and $e = \bigoplus_{i=1}^{n-1} e_i$. The two-qubit gates correspond to CZ gates ($CZ = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z$).

Figure 3.2: Preparation of a poor man's Bell pair in 1D between the first and last qubits on a line.

$\alpha_i \in \{0,1,2,3\}$ and outputs an $x_i \in \{-1,+1\}^2$, similarly B_i takes input $\beta_i \in \{0,1,2,3\}$ and outputs a $y_i \in \{-1,+1\}^2$.

To understand the circuit's behavior, let's step through each layer of gates. The circuit first produces two Bell pairs between A_i and B_i for each $i \in [n]$, followed by a *controlled* $B^\dagger$ between B_i and A_{i+1}, conditioned on $(\alpha_{i+1}, \beta_i) = (0,0)$.

$$B^\dagger_{\alpha_{i+1},\beta_i} := \begin{cases} B^\dagger & \text{if } \alpha_{i+1} = \beta_i = 0 \\ I & \text{otherwise} \end{cases} \qquad \alpha_{i+1}, \beta_i \in \{0,1,2,3\} \tag{3.8}$$

In the case that α_j is the only nonzero $\{\alpha_i\}_i$ and β_k is the only nonzero $\{\beta_i\}_i$, this has the effect of creating poor-man's Bell pairs between A_j and B_k (for $j < k$) via the *teleportation method* in Equation (3.7). The $U(\alpha)$ and $V(\beta)$ gates are the same as those used in the optimal strategy for the magic square game, defined in Figure 3.1, which act as the identity upon input 0. Therefore, in the case where α_j and β_k are the only nonzero inputs for $j < k \in [n]$, the circuit simply prepares two poor man's Bell pairs between A_j and B_k and applies the gates $U(\alpha_j)$ to A_j's state and $V(\beta_k)$ to B_k's state as shown in Figure 3.4.

Finally, we are ready to define the 1D Magic Square Problem.

Definition 8 (1D Magic Square Problem (1D-MSP$_n$))**.** *Given some string $z_{in} \in \{0,1,2,3\}^{2n}$, output some string $z_{out} \in \{0,1\}^{4n}$ such that*

$$\langle z_{out} | \mathcal{C}^{\text{1D-MSP}}_{z_{in}} \left| 0^{4n} \right\rangle > 0 \tag{3.10}$$

Where $\mathcal{C}^{\text{1D-MSP}}_{z_{in}}$ is the circuit (parameterized by z_{in}) in Figure 3.3. For any valid pair (z_{in}, z_{out}) we say that (z_{in}, z_{out}) is "in" 1D-MSP$_n$.

Note that in binary representation, the input to 1D-MSP$_n$ is a $4n$-bit string. By definition, the quantum circuit $\mathcal{C}^{\text{1D-MSP}}_{z_{in}}$ is a QNC^0 circuit that solves every instance of the 1D-MSP$_n$.

Theorem 3. *There exists a QNC^0 circuit with 1D geometrically local gates that solves the 1D Magic Square Problem on all inputs with probability 1.*

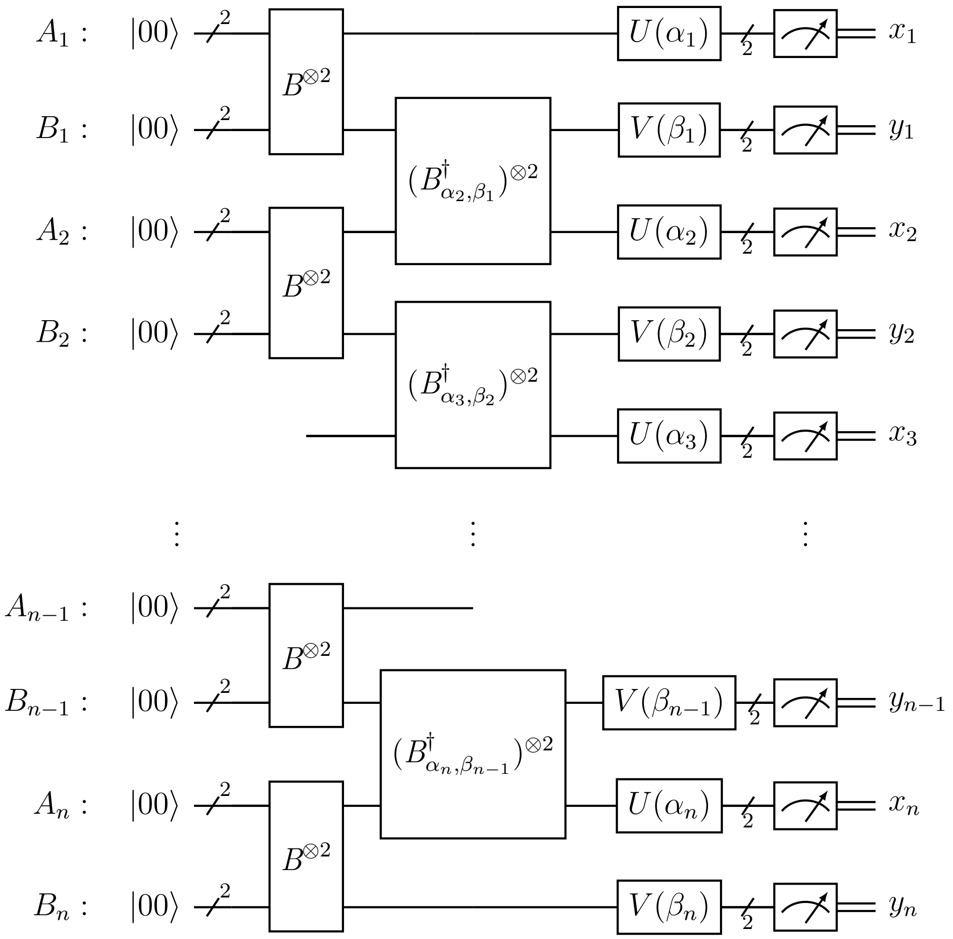
To understand the connection between the 1D-MSP and the (generalized) magic square game, we consider choosing inputs from the set S_n defined below.

Definition 9 ($z_{in}(j,k,\alpha,\beta)$, S_n)**.** *For each $j,k \in [n]$ with $j < k$, and $\alpha, \beta \in \{1,2,3\}$, we define $z_{in}(j,k,\alpha,\beta) := (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n) \in \{0,1,2,3\}^n$ where α_i, β_i are set as*

$$\alpha_i = \begin{cases} \alpha & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad \beta_i = \begin{cases} \beta & \text{if } i = k \\ 0 & \text{if } i \neq j \end{cases} \tag{3.11}$$

We also define the set of all such inputs S_n

$$S_n := \left\{ z_{in}(j,k,\alpha,\beta) : j < k \in n,\ \alpha, \beta \in \{1,2,3\} \right\} \tag{3.12}$$



(3.9)

Figure 3.3: The quantum circuit for the 1D Magic Square Problem, parameterized by $z_{in} = (\alpha_1, \beta_1, \dots \alpha_n, \beta_n) \in \{0, 1, 2, 3\}^n$. The gates $B^\dagger_{\alpha,\beta}$ are defined in Equation (3.8), and $U(\alpha), V(\beta)$ are defined in Figure 3.1.

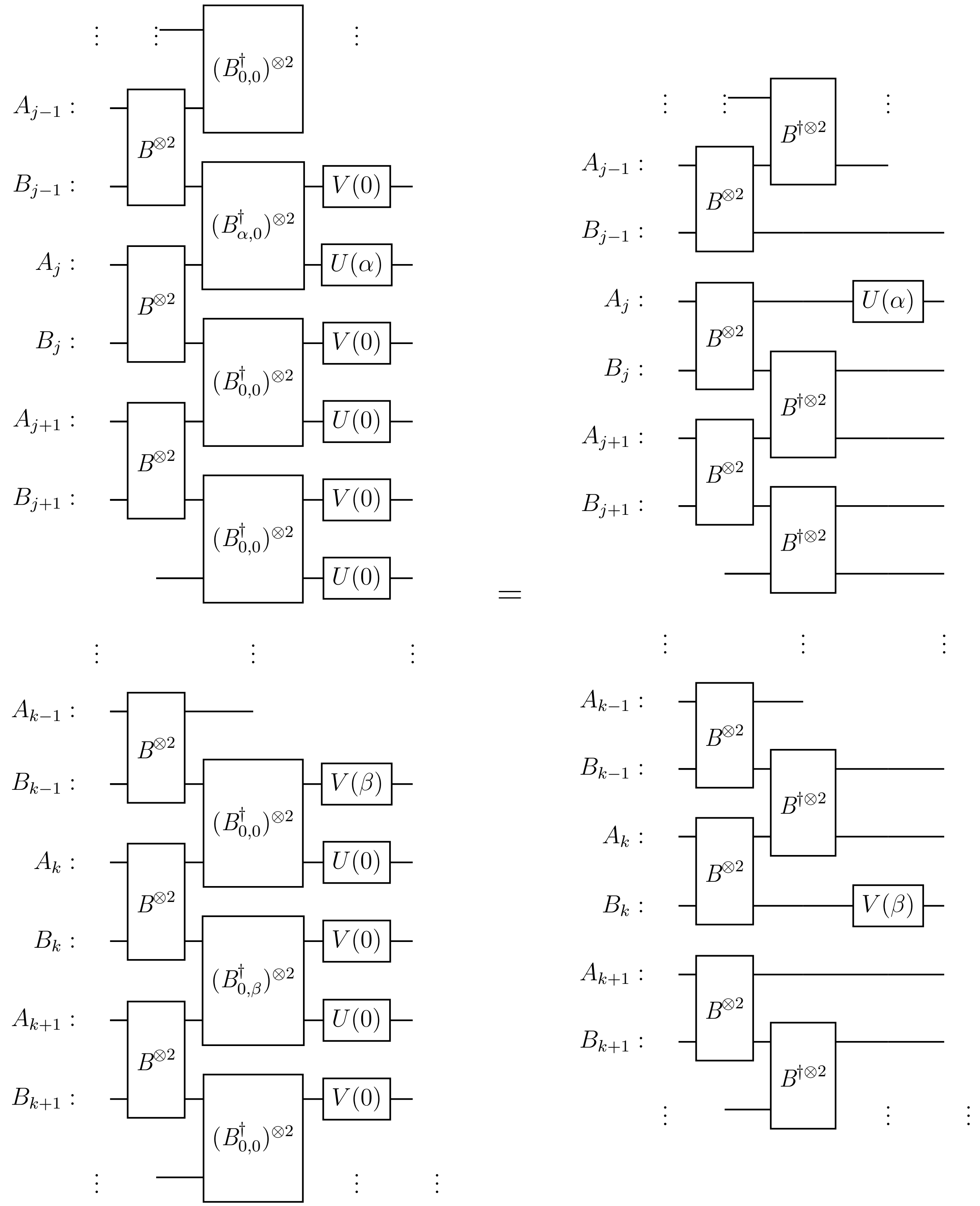


Figure 3.4: Quantum circuit for the 1D Magic Square Problem for input $(\alpha_1, \beta_1, \ldots, \alpha_n, \beta_n)$ with $\alpha_j = \alpha, \beta_k = \beta$ and all other inputs are 0.

For a particular $\alpha, \beta \in \{1,2,3\}$ and $j < k \in [n]$, the circuit $\mathcal{C}_{z_{in}}^{\text{1D-MSP}}$ controlled by input $z_{in}(j,k,\alpha,\beta)$ prepares two poor man's Bell pairs $|\Phi_{s,t}\rangle \otimes |\Phi_{s,t}\rangle$ with *correction bits* $s, t, s', t' \in \{-1,+1\}$ determined by the uniformly random bits measured in between A_j and B_k. Let $a, b \in \{-1,+1\}^3$ denote the strings corresponding to A_j, B_k's outputs with the third value determined by the first two.

$$a = (x_j^1, x_j^2, -x_j^1 x_j^2) \qquad\qquad b = (y_k^1, y_k^2, y_k^1 y_k^2) \tag{3.13}$$

It follows from Claim 2 that (a,b) is a valid output for input (α,β) for the generalized magic square game for (s,t,s',t'). We state this as a lemma below, whose formal proof can be found in [BGKT20].

Lemma (Lemma 3 in [BGKT20])**.** *For each $z_{in}(j,k,\alpha,\beta) \in S_n$, and $z_{out} = (x_1, y_1, \ldots, x_n, y_n)$ such that $(z_{in}, z_{out}) \in$ 1D-MSP$_n$. Then (x_j, y_k) is a valid output for the generalized magic square game on input (α,β) with parameters*

$$s := \prod_{i=j}^{k-1} y_i^1 \qquad s' := \prod_{i=j}^{k-1} y_i^2 \qquad t := \prod_{i=j}^{k-1} x_{i+1}^1 \qquad t' := \prod_{i=j}^{k-1} x_{i+1}^2. \tag{3.14}$$

Where we set $x_j^3 := -x_j^1 x_j^2$, $y_k^3 := y_k^1 y_k^2$.

All that's left to prove the main theorem of this section, Theorem 2 is to show the classical hardness of the 1D-MSP.

3.2 The classical hardness of the 1D Magic Square Problem

Now we will prove that the 1D Magic Square Problem cannot be solved exactly by a probabilistic constant-depth classical circuit with constant fan-in ($\mathsf{NC}^0/\mathsf{rpoly}$). To this end, it is useful to consider the behavior of the classical circuit on the set of inputs S_n, with only one of each α_i and β_i nonzero.

Theorem 4. *Suppose a probabilistic classical circuit $\mathcal{C}$ with bounded fan-in K given a random input from S_n, outputs a valid string for the 1D Magic Square Problem with probability at least $9/10$ over the randomness in the circuit and the choice of input. Then $\mathcal{C}$ has depth $D = \Omega(\log n)$.*

In order to prove Theorem 4, we first show that if the lightcones of the registers A_j and B_k of the 1D Magic Square Problem do not intersect in a certain way, then they essentially cannot communicate any information to each other about α_j and β_k. In this case, outputting a valid string is at least as hard as winning the original magic square game with a classical strategy, which has probability at most $8/9$. Then we show that if we choose j and k randomly, with high probability A_j and B_k have non-intersecting lightcones. Since classical circuits and their lightcones are defined in terms of input and output *bits*, we will denote the binary representation of $\alpha_i, \beta_i \in \{0,1,2,3\}$ as $\hat{\alpha}_i, \hat{\beta}_i \in \{00, 01, 10, 11\}$ for each $i \in [n]$.

Definition 10 ($E_{\mathcal{C}}$). *Consider some classical circuit $\mathcal{C}$ with input $(\hat{\alpha}_1, \hat{\beta}_1, \ldots, \hat{\alpha}_n, \hat{\beta}_n)$ and output $(x_1, x_1, \ldots, x_n, y_n)$, where for each $i \in [n]$, $\hat{\alpha}_i, \hat{\beta}_i, x_i, y_i \in \{0,1\}^2$. Suppose $j < k \in [n]$ are chosen uniformly at random. Define $E_{\mathcal{C}}$ to be the event that all the following are true:*

$$\overrightarrow{\mathcal{L}_{\mathcal{C}}}(\hat{\alpha}_j) \cap \overrightarrow{\mathcal{L}_{\mathcal{C}}}(\hat{\beta}_k) = \emptyset \qquad \overrightarrow{\mathcal{L}_{\mathcal{C}}}(\hat{\alpha}_j) \cap y_k = \emptyset \qquad \overrightarrow{\mathcal{L}_{\mathcal{C}}}(\hat{\beta}_k) \cap x_j = \emptyset. \tag{3.15}$$

In words, $E_{\mathcal{C}}$ is the event that no output bit depends on both α and β, and that the A_j register's output a (or x_j), does not depend on β. Likewise, the B_k register's output b (or y_k) does not depend on α.

Lemma 3 (Lemma 8 in [BGKT20]). *Consider some classical circuit $\mathcal{C}$ and $j < k \in [n]$ such that $E_{\mathcal{C}}$ is true. Then given input $z_{in}(j, k, \alpha, \beta)$, $\mathcal{C}$ will give a valid output to the 1D Magic Square Problem with probability at most $8/9$ over uniformly random choice of $\alpha, \beta \in \{1, 2, 3\}$.*

We provide some proof intuition below, but we refer the reader to [BGKT20] for the formal proof.

Proof Idea *of Lemma 3.* Consider some $j < k$ such that $E_{\mathcal{C}}$ is true. Suppose the input to the circuit is $z_{in}(j, k, \alpha, \beta)$ with $\alpha, \beta \in [3]$ chosen uniformly at random. Since $\overrightarrow{\mathcal{L}_{\mathcal{C}}}(\hat{\alpha}_j) \cap \overrightarrow{\mathcal{L}_{\mathcal{C}}}(\hat{\beta}_k) = \emptyset$, each output bit of the circuit is a function of at most one of α or β. Therefore, we can partition the outputs of the circuits into two separate strings $z_A(\alpha)$ and $z_B(\beta)$. Since $\overrightarrow{\mathcal{L}_{\mathcal{C}}}(\hat{\alpha}_j) \cap y_k = \emptyset$ and $\overrightarrow{\mathcal{L}_{\mathcal{C}}}(\hat{\beta}_k) \cap x_j = \emptyset$ (by definition of $E_{\mathcal{C}}$), it follows that x_j is in the A string and y_k is in the B string. Therefore, these two strings are independent and can be interpreted as two non-communicating players.

The final step in the proof is to show that if the classical circuit outputs a correct value, then this would allow two non-communicating players Alice and Bob with access to $z_A(\alpha)$ and $z_B(\beta)$ respectively to win the magic square game. Since this is a classical strategy, this happens with probability at most 8/9. □

Next, we just need to show that if we choose $j < k$ randomly, the event $E_{\mathcal{C}}$ that A_j and B_k do not have intersecting lightcones as specified by Definition 10 occurs with high probability.

Lemma 4. *For classical circuit $\mathcal{C}$, with depth D and bounded fan in K, the probability of $E_{\mathcal{C}}$ occurring is at least $1 - \frac{24K^{2D}}{n}$ over the random choice of $j < k \in [n]$.*

Proof. First we will bound the probability of $E_{\mathcal{C}}$ occurring when $j, k \in [n]$ are chosen uniformly at random, not necessarily such that $j < k$, we denote this probability as $\Pr_{j,k}[E_{\mathcal{C}}]$. We will then use this to bound the probability over the choice of $j, k \in [n]$ such that $j < k$, denotes $\Pr_{j<k}[E_{\mathcal{C}}]$.

Claim 3. $\Pr_{j,k}[\overrightarrow{\mathcal{L}_{\mathcal{C}}}(\alpha_j) \cap y_k \neq \emptyset], \Pr_{j,k}[\overrightarrow{\mathcal{L}_{\mathcal{C}}}(\beta_k) \cap x_j \neq \emptyset] \leq \frac{2K^D}{n}$

Proof. First, let's consider $\Pr_{j,k}[\overrightarrow{\mathcal{L}_\mathcal{C}}(\alpha_j) \cap y_k \neq \emptyset]$. It is helpful to consider each fixed k.

$$\begin{aligned}
\Pr_{j,k}[\overrightarrow{\mathcal{L}_\mathcal{C}}(\alpha_j) \cap y_k \neq \emptyset] &= \frac{1}{n}\sum_{k=1}^{n} \Pr_j[\overrightarrow{\mathcal{L}_\mathcal{C}}(\alpha_j) \cap y_k \neq \emptyset] && (3.16)\\
&= \frac{1}{n}\sum_{k=1}^{n} \sum_{z \in \overleftarrow{\mathcal{L}_\mathcal{C}}(y_k)} \Pr_j[z \in \alpha_j] && (3.17)\\
& && (3.18)
\end{aligned}$$

Note that for each input bit z for exactly one choice of $j \in [n]$ will z be one of the two bits in α_j. Thus, $\Pr_j[z \in y_k] \leq \frac{1}{n}$ for each input bit z. So we can upper bound the above expression by

$$\begin{aligned}
&\leq \frac{1}{n}\sum_{k=1}^{n} \sum_{z \in \overleftarrow{\mathcal{L}_\mathcal{C}}(y_k)} \frac{1}{n} && (3.19)\\
&\leq \frac{2K^D}{n} && (3.20)
\end{aligned}$$

Where we used the fact that $|\overleftarrow{\mathcal{L}_\mathcal{C}}(y_k)| \leq 2K^D$ for each $k \in [n]$. We can use the same exact reasoning to show $\Pr_{j,k}[\overrightarrow{\mathcal{L}_\mathcal{C}}(\beta_k) \cap x_j \neq \emptyset] \leq \frac{2K^D}{n}$. □

Claim 4. $\Pr_{j,k}[\overrightarrow{\mathcal{L}_\mathcal{C}}(\alpha_j) \cap \overrightarrow{\mathcal{L}_\mathcal{C}}(\beta_k) \neq \emptyset] \leq \frac{2K^{2D}}{n}$

Proof. Again, we will consider each k as fixed.

$$\begin{aligned}
\Pr_{j,k}[\overrightarrow{\mathcal{L}_\mathcal{C}}(\alpha_j) \cap \overrightarrow{\mathcal{L}_\mathcal{C}}(\beta_k) \neq \emptyset] &= \frac{1}{n}\sum_{k=1}^{n} \Pr_j[\overrightarrow{\mathcal{L}_\mathcal{C}}(\alpha_j) \cap \overrightarrow{\mathcal{L}_\mathcal{C}}(\beta_k)] && (3.21)\\
&= \frac{1}{n}\sum_{k=1}^{n} \Pr_j[\alpha_j \cap \overleftarrow{\mathcal{L}_\mathcal{C}}(\overrightarrow{\mathcal{L}_\mathcal{C}}(\beta_k)) \neq \emptyset] && (3.22)\\
&= \frac{1}{n}\sum_{k=1}^{n} \sum_{z \in \overleftarrow{\mathcal{L}_\mathcal{C}}(\overrightarrow{\mathcal{L}_\mathcal{C}}(\beta_k))} \Pr_j[z \in \alpha_j] && (3.23)\\
&\leq \frac{2K^{2D}}{n} && (3.24)
\end{aligned}$$

Where we used that $|\overleftarrow{\mathcal{L}_\mathcal{C}}(\overrightarrow{\mathcal{L}_\mathcal{C}})(\beta_k)| \leq 2K^{2D}$ for each $k \in [n]$, and again that $\Pr_j[z \in \alpha_j] \leq \frac{1}{n}$ for each input bit z over random $j \in [n]$. □

Combining Claims 3 and 4, and using the union bound,

$$\Pr_{j,k}[\neg E_\mathcal{C}] \leq \frac{4K^D}{n} + \frac{2K^{2D}}{n} \qquad (3.25)$$

Finally, we want to convert this bound on the probability of $\neg E_{\mathcal{C}}$ (lightcone intersection) occurring over uniformly random choice of $j, k \in [n]$ to the probability over the uniform choice of $j, k \in [n]$ such that $j < k$. Note that $\Pr_{j<k}[A] = \Pr_{j,k}[A | j < k]$. Therefore, $\Pr_{j<k}[A] = \Pr_{j,k}[A] \cdot \frac{1}{\Pr_{j,k}[j<k]}$. Since there $\binom{n}{2}$ choices of $j < k \in [n]$ and there are n^2 choices for $j, k \in [n]$, it follows that $\Pr_{j,k}[j < k] = \frac{n(n-1)}{2n^2} = \frac{n-1}{2n}$. Combining this with Equation (3.25), it follows that

$$\Pr_{j<k}[\neg E_{\mathcal{C}}] \leq \left(\frac{4K^D}{n} + \frac{2K^{2D}}{n}\right) \cdot \frac{2n}{n-1} \tag{3.26}$$

$$\leq \frac{24K^{2D}}{n} \tag{3.27}$$

Where we used that $K \geq 1$ and $\frac{2n}{n-1} \leq 4$ □

Combining Lemmas 3 and 4, we are ready to prove Theorem 4.

Proof of Theorem 4. From Lemma 3, we have that for a random input z from the set S_n, the probability of success if $E_{\mathcal{C}}$ is true is $\Pr[\text{ success } | E_{\mathcal{C}}] \leq 8/9$, from Lemma 4, we have that $\Pr[E_{\mathcal{C}}] \geq 1 - \frac{24K^{2D}}{n}$. Suppose, as in the theorem statement, that our classical circuit $\mathcal{C}$ succeeds with probability at least $9/10$, then we have that

$$\frac{9}{10} \leq \Pr[\text{ success }] = \Pr[\text{success } | E_{\mathcal{C}}] \cdot \Pr[E_{\mathcal{C}}] + \Pr[\text{ success } | \neg E_{\mathcal{C}}] \cdot \Pr[\neg E_{\mathcal{C}}] \tag{3.28}$$

$$\leq \frac{8}{9} \cdot \Pr[E_{\mathcal{C}}] + \Pr[\text{ success } | \neg E_{\mathcal{C}}] \cdot \frac{24K^{2D}}{n} \tag{3.29}$$

$$\leq \frac{8}{9} + \frac{24K^{2D}}{n} \tag{3.30}$$

$$\frac{1}{90} \leq \frac{24K^{2D}}{n}. \tag{3.31}$$

Rearranging the previous equation and taking a logarithm of each side we lower bound the circuit depth D.

$$D \geq \frac{1}{2}\log_K\left(\frac{n}{90 \cdot 24}\right) \geq \frac{\log_K(0.0004n)}{2} = \frac{\log(0.0004n)}{2\log K} \tag{3.32}$$

This bound is actually tighter by a constant factor than the original bound in [BGKT20], which was $D \geq \frac{\log(0.00001n)}{2\log K}$. □

3.2.1 Near-term prospects of the 1D Magic Square Problem

The 1D Magic Square Problem has key properties that are amenable to a provable quantum advantage that can be implemented on near-term devices. Firstly, the quantum circuit for the 1D Magic Square Problem can be implemented in a 1D architecture (qubits on a line) with only nearest-neighbor interactions. Secondly, Bravyi, Gosset, Koenig, and Tomamichael

[BGKT20] further extend this problem to one that is noise-resilient. That is, even if the quantum circuit is implemented with some errors, as long as the error rate is below some threshold, there is still a separation between the shallow quantum and classical circuit performance. To allow for the presence of noise, the quantum circuit for this new relation makes use of a quantum error-correcting code.

So, can we implement this quantum circuit experimentally and claim quantum advantage? Well, not yet. The 1D Magic Square Problem provides a separation between shallow quantum and classical circuits *as we scale up* n. In order to implement this problem experimentally and claim a quantum advantage, we must at some point pick a particular problem size n. Even for the 1D Magic Square Problem, which is not resilient to noise, we need to choose quite a large value for n for our classical lower bound to kick in.

To illustrate, suppose we want to choose an n such that the probability of success of a classical circuit with depth $D = 3$ and fan in $K = 2$ (matching the quantum circuit), has probability of success at most 9/10 for the 1D Magic Square Problem. By Equation (3.28)

$$\frac{9}{10} \geq \frac{8}{9} + \Pr[\neg E_{\mathcal{C}}]. \tag{3.33}$$

Since $\Pr[\neg E_{\mathcal{C}}]$ depends on n, K, D, we just need to set our n such that Equation (3.33) holds. We can do this by plugging in our tightest bound for $\Pr[\neg E_{\mathcal{C}}]$ from Equation (3.26) and solving for n.

$$\begin{aligned} \frac{9}{10} &\geq \frac{8}{9} + \frac{8K^D + 4K^{2D}}{n-1} && (3.34)\\ n &\geq 90(8K^D + 4K^{2D}) + 1 && (3.35)\\ &= 90(8 \cdot 2^3 + 4 \cdot 2^6) + 1 && (3.36)\\ &= 28801 && (3.37) \end{aligned}$$

If we set $n = 28,801$, then every classical circuit with depth 3 and fan in 2 cannot succeed at the 1D Magic Square Problem with probability better than 9/10 over inputs drawn randomly from S_n. Recall that the quantum circuit for the 1D Magic Square Problem requires $4n$ qubits, so implementing this quantum circuit requires a reliable quantum computer with 115, 204 qubits. And much more if we want to implement error correction!

It should be noted that it is not necessarily true that we *need* this large of a problem instance to see the quantum advantage, but that this is the scale at which we can currently *prove* such a separation according to our best-known classical lower bounds for the 1D Magic Square Problem.

Chapter 4

Limitations of shallow quantum circuits for optimization problems

So far, we have seen that shallow quantum circuits outperform shallow classical circuits at certain relational tasks (Chapter 3), and can produce distributions that are hard to sample from efficiently (Section 2.5), under some strong complexity-theoretic assumptions. However, neither of these hardness results give performance guarantees for any problem which is currently considered *useful*. In this chapter, we explore whether these shallow circuit separations can be extended to more natural problems.

In practice, many problems we care about can be stated as *optimization problems* with local objective (or "cost") functions. Throughout this chapter we use the convention that problems are defined over inputs from the set $\mathcal{X} = \{0,1\}^m$ and outputs from $\mathcal{Y} = \{0,1\}^n$. An optimization problem with input $x \in \mathcal{X}$ and output $y \in \mathcal{Y}$ is defined by its objective function:

$$F(x, y) = \sum_{i \in [t]} F_i(x_{T_i}, y_{S_i}) \tag{4.1}$$

with $t \in \mathbb{N}$, and for some $T_i \subseteq [m]$, $S_i \subseteq [n]$ and $F_i : \{0,1\}^{|T_i|} \times \{0,1\}^{|S_i|} \to \mathbb{R}$ for each $i \in [t]$. Upon given input $x \in \{0,1\}^m$ the goal is to output a $y \in \{0,1\}^n$ that minimizes $F(x, y)$.

$F(x, y)$ defines a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ consisting of all pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $F(x, y)$ is the optimal value for input x, that is $(x, y) \in R$ if $F(x, y) = \min_{y'} F(x, y')$. We then refer to $F(x, y)$ as a *checking function* for R.

Definition 11 (Checking function)**.** *$F(x, y)$ is a* checking function *for a relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ if for each $x \in \mathcal{X}, y \in \mathcal{Y}$,*

$$(x, y) \in R \iff F(x, y) = \min_{y' \in \mathcal{Y}} F(x, y') \tag{4.2}$$

We refer to each term $F_i(x_{T_i}, y_{S_i})$ as a *check.*

Definition 12 (Local checking function, local checkability)**.** *We say that $F(x, y)$ is a* local checking function*, and R is* locally checkable *if there exists some $c = O(1)$ such that*

1. *Each* check *is a function of at most c variables*

$$|S_i| + |T_i| \leq c \qquad \text{for each } i \in [t] \tag{4.3}$$

2. *Each (output) y variable is involved in at most c checks*

$$|\{i : j \in S_i\}| \leq c \qquad \text{for each } j \in [n]. \tag{4.4}$$

An example of an optimization problem with an objective function that meets these criteria is the MaxCut problem where we only consider inputs which are a subgraph of some bounded degree graph. For a graph $G = (V, E)$, the MaxCut_G problem takes as input $x \in \{0,1\}^{|E|}$ and the goal is to output a $y \in \{0,1\}^{|V|}$ that minimizes

$$-\sum_{(u,v)\in E} x_{(u,v)}(y_u + y_v - 2y_u y_v). \tag{4.5}$$

To see why this is a *local* checking function, note that each term in this sum involves exactly 1 input variable and 2 output variables. Furthermore, each output variable y_u is involved $\deg(y_u)$ terms.

In this chapter, we investigate whether shallow quantum circuits can outperform shallow classical circuits for local optimization problems. In [Has19], Hastings introduced shallow classical circuit algorithms for certain problems (referred to as *local algorithms*) which outperform, in expected objective value, constant-depth QAOA. In this chapter, we prove a limitation for shallow quantum circuits for local optimization problems in general, using a much stricter performance metric.

We investigate the following question:

Question 1. *Does there exist a locally-checkable relation that can be solved by a shallow quantum circuit (QNC^0), but not by a shallow classical circuit (NC^0, NC^0/rpoly)?*

Why is using shallow quantum circuits to solve locally checkable relations interesting? Firstly, locally checkable relations are a very large class of problems. In fact, many NP-complete problems, such as MaxCut and Maximum Independent Set (on bounded degree graphs), are optimization problems defined by local objective functions. Secondly, there are proposed shallow quantum circuit algorithms for such problems, such as the QAOA. As discussed in Section 2.6, the simplest version of the QAOA can be implemented in constant depth for these local optimization problems.

Finally, the performance of a shallow quantum circuit for a local optimization problem is inherently *robust to noise.* In particular, we consider the error model of applying a *qubit depolarizing channel* $\mathcal{N}_\epsilon$ to the output state of our quantum circuit. The depolarizing channel for an n-qubit state acts independently on each qubit such that for qubit $i \in [n]$, with probability ϵ a uniformly random operator from $\{I, X, Y, X\}$ is applied. For a *single qubit* state ρ_i this has the following behavior.

$$\mathcal{N}_\epsilon : \rho_i \to (1-\epsilon)\rho_i + \epsilon I \tag{4.6}$$

A nice property of the depolarizing channel is that the expectation value of low-weight Pauli operators is approximately preserved. To illustrate this, consider the n-qubit operator $Z(S)$, the tensor product of the Z applied to each of the qubits in $S \subseteq [n]$ and the identity on all $n - |S|$ others.

$$\operatorname{tr}(\mathcal{N}_\epsilon(\rho) Z(S)) = (1-\epsilon)^{|S|} \operatorname{tr}(\rho Z(S)) \tag{4.7}$$

A local objective function $F(x, y)$ for a particular $x \in \mathcal{X}$ can be written as a k-local Hamiltonian *in the Pauli-Z basis* F_x for some $k = O(1)$, such that $F(x, y) = \operatorname{tr}(|y\rangle\langle y| \, F_x)$. Let's consider such a function with no constant terms.

$$F_x = \sum_{\substack{S \subseteq [n]: \\ |S| \leq k, \\ S \neq \emptyset}} \alpha_S Z(S) \qquad\qquad \alpha_S \in \mathbb{R} \text{ for each } S \subseteq [n] : |S| \leq k. \tag{4.8}$$

For example, the MaxCut problem has such a Hamiltonian with $k = 2$. The MaxCut problem for graph $G = (V, E)$, on input $x \in \{0,1\}^{|E|}$, is equivalent to finding a $y \in \mathcal{Y}$ that minimizes $\operatorname{tr}\big(|y\rangle\langle y| \, F_x^{\text{MaxCut}}\big)$ with F_x^{MaxCut} defined as

$$F_x^{\text{MaxCut}} = \sum_{(u,v) \in E : x_{(u,v)} = 1} Z_u Z_v. \tag{4.9}$$

Suppose that we have a quantum algorithm for such a problem that outputs the state $\rho^{(x)}$, achieving expectation value $\langle F_x \rangle = \operatorname{tr}(\rho F_x)$. Now suppose that the quantum state undergoes qubit depolarizing noise with rate ϵ. For each $S \subseteq [n] : |S| \leq k$, the expected value of each term $\langle F_S \rangle_\epsilon = \operatorname{tr}(\alpha_S \mathcal{N}_\epsilon(\rho) Z(S))$ is

$$\langle F_S \rangle_\epsilon = (1-\epsilon)^{|S|} \langle F_S \rangle \approx (1 - |S| \cdot \epsilon) \langle F_S \rangle. \tag{4.10}$$

Therefore, the noise only causes a multiplicative error of $O(k\epsilon)$ to each term of the objective.

$$|\langle F_S \rangle_\epsilon - \langle F_S \rangle| \leq O(k\epsilon) \langle F_S \rangle \tag{4.11}$$

By linearity of expectation, it follows that

$$|\langle F \rangle_\epsilon - \langle F \rangle| \leq O(k\epsilon) \langle F \rangle. \tag{4.12}$$

While calculating the expectation value of some function of the output of a shallow quantum circuit is hard in general, we know that if the function is local, it can be done efficiently. This is because the marginal distribution of each term (and thus the expectation) can be computed efficiently as shown in Section 2.6. Does this promise of function locality make classical simulation easier in general? If a shallow quantum circuit satisfies some relation R, does the classical task of satisfying R become easier if R is locally-checkable?

One might wonder whether any of the known relations used to separate QNC^0 and NC^0 (or even AC^0) are locally checkable. All currently known relational separations between QNC^0 and NC^0 (or even AC^0), use relations that are *characterized* by a QNC^0 circuit [BGK18, BGKT20, WKST19, Gal18, CSV21]. We first consider this class of relations.

For a (classical or quantum) circuit C, let $C(x)$ denote the random variable over $\mathcal{Y}$ of the circuits (measured) output upon given input $x \in \mathcal{X}$.

Definition 13 (*characterizes* relation)**.** *We say that a (quantum or classical) circuit C* characterizes *relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ if for each $x \in \mathcal{X}, y \in \mathcal{Y}$*

$$\Pr[C(x) = y] > 0 \iff (x, y) \in R. \tag{4.13}$$

This is stricter than just *satisfying* the relation.

Definition 14 (*satisfies* relation)**.** *The (quantum or classical) circuit C* satisfies *R if for each $x \in X, y \in \mathcal{Y}$*

$$\Pr[C(x) = y] > 0 \implies (x, y) \in R. \tag{4.14}$$

In addition to requiring that the circuit output is always correct (satisfying R), a circuit that characterizes R requires that every valid solution is output with some nonzero probability.

The 1D Magic Square Problem, discussed in Chapter 3, is an example of a problem that is characterized by a quantum circuit. In fact, the problem itself is defined by the input/output relation of a $\mathsf{QNC^0}$ circuit $C^{\text{1D-MSP}}$ (see Section 3.1).

In this chapter, we show that *quantum advantage is limited in this regime.* First, we show that if $F(x, y)$ is efficiently computable classically (not necessarily local) then we can construct an *efficient* classical algorithm that satisfies R.

Theorem 5. *For each relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ with an efficiently computable checking function $F(x, y)$, if there exists a quantum circuit with fan-in 2 and depth $D = O(\log \log m)$ that characterizes R, then there exists an efficient classical algorithm that satisfies R.*

A direct implication of this theorem is that if a $\mathsf{QNC^0}$ circuit characterizes some search problem that is efficiently verifiable (in NP), then this search problem can be solved by an efficient classical algorithm (in P).

Furthermore, we show that if $F(x, y)$ is a *local* checking function, the classical algorithm can be parallelized into a depth that is exponentially larger than the quantum circuit's depth. Therefore, if the quantum circuit has depth $D = O(1)$, then we can actually solve the relation with an $\mathsf{NC^0}$ circuit.

Theorem 6. *For each locally-checkable relation $R \subseteq \mathcal{X} \times \mathcal{Y}$, if there exists a quantum circuit with depth D and fan-in/out 2 that characterizes R, then there exists a classical circuit with fan-in 2, depth $O(2^{3D})$ and size $O(2^{2^D + 3D})$ that satisfies R.*

Setting $D = O(1)$, we get the following corollary.

Corollary 1. *For each locally-checkable relation $R \subseteq \mathcal{X} \times \mathcal{Y}$, if there exists a $\mathsf{QNC^0}$ circuit that characterizes R, then there exists an $\mathsf{NC^0}$ circuit that satisfies R.*

This corollary tells us that the known relational separations (2D-HLF, 1D-MSP, PHP) from [BGK18, BGKT20, WKST19] between quantum and classical shallow circuits, are *not* locally checkable, since they are each characterized by a QNC^0 circuit. Moreover, Theorem 5 provides an efficient classical algorithm for each of these problems, since they are efficiently verifiable.

Note that the classical circuit in Theorem 8 has an exponential depth blowup compared to the quantum circuit. It remains open whether there exists a locally checkable relation that is characterized by a $\Theta(\log(n))$ depth quantum circuit but cannot be satisfied by a $O(\log(n))$-depth classical circuit.

Note that both of these theorems require the quantum circuit to *always* output a valid solution, leaving no room for error. We strengthen the above two theorems and allow for some probability of invalid outputs so long as each valid output string has a significantly larger probability of being output than the invalid output strings.

Definition 15 (approximately characterizes)**.** *We say that a (quantum or classical) circuit C β-*approximately characterizes *relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ for $\beta \in [0, 1)$ if for each $x \in \mathcal{X}, y \in \mathcal{Y}$*

1. *If $(x, y) \in R$ then $\Pr[Q(x) = y] \geq p$, and*
2. *If $(x, y) \notin R$ then $\Pr[Q(x) = y] \leq \beta p$*

for some $p \in (0, 1]$

We note that this definition is not the most well-motivated performance metric for an approximate optimization algorithm. Ideally, we would use a more natural and less restrictive metric such as the expected objective value, or the overall probability of outputting an optimal solution. However this definition allows us to generalize Theorems 5 and 6 so that the quantum circuit does not necessarily solve the relation *exactly*.

Theorem 7. *For each relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ with an efficiently computable checking function $F(x, y)$, if there exists a quantum circuit with fan-in 2 and depth $D = O(\log \log m)$ that 2^{-2^D}-approximately characterizes R, then there exists an efficient classical algorithm that satisfies R.*

Theorem 8. *For each locally-checkable relation $R \subseteq \mathcal{X} \times \mathcal{Y}$, if there exists a quantum circuit with fan-in 2 and depth $D = O(1)$ that 2^{-2^D}-approximately characterizes R, then there exists a classical circuit that satisfies R, has depth $O(2^{3D})$, fan-in 2, and size $O(2^{2^D+3D})$.*

Corollary 2. *For each locally-checkable relation $R \subseteq \mathcal{X} \times \mathcal{Y}$, if there exists a QNC^0 circuit with depth $D = O(1)$ that 2^{-2^D}-approximately characterizes R, then there exists an NC^0 circuit that satisfies R.*

Intuitively, we can interpret Corollary 2 as saying that we can not expect a quantum advantage for a local optimization problem where the quantum circuit does *exceptionally well*.

The two main insights used to prove these theorems are lightcones, and the locality of the checking function. To highlight how these two techniques work together, we will first make use of lightcones to prove Theorem 7, by constructing an efficient classical algorithm that satisfies a relation with an efficiently computable checking function. This algorithm iteratively updates an input-output pair until arriving at a pair with the desired input. Then we will show that if the checking function is *local*, this algorithm can be parallelized into constant depth.

While we did not answer our Question 1: whether there exists a locally checkable relation with a shallow circuit quantum advantage, we answer it in the negative in the case where the relation is characterized or approximately characterized by a shallow quantum circuit. Furthermore, we conjecture that the answer is also no in the case where our task requires that the classical and quantum circuits satisfies the relation.

Conjecture 1. *For each locally-checkable relation, if there exists a* $\mathsf{QNC^0}$ *circuit that satisfies* R*, then there exists a* $\mathsf{NC^0}$ *circuit that also satisfies* R*.*

Even this conjecture would not rule out the possibility that shallow quantum circuits might outperform shallow classical circuits, or even polynomial-sized circuits, in terms of a different performance metric, such as the expectation value of F, or the probability of outputting the optimal solution. Moreover, our no-go theorems (implicitly) assume that the relation is a *total* relation (that the range is $\mathcal{X} = \{0,1\}^m$), and they break down for partial relations.

These no-go results do, however, rule out the possibility of proving a complexity separation akin to previously known results [BGK18, BGKT20, WKST19] for the task of finding an optimal solution to a local optimization problem. As each of these known separations uses relations that are *characterized* by a $\mathsf{QNC^0}$ circuit. See the definition the 1D-MSP in Section 3.1 in terms of the input/output relation of a constant-depth circuit.

Note that characterizing a relation is a special case of approximately characterizing a relation for any $\beta < 1$. Therefore, Theorems 5 and 6 are special cases of Theorems 7 and 8, so it is sufficient for us to just prove Theorems 7 and 8 – which are proved in the following two sections (Sections 4.1 and 4.2). Following that, in Section 4.3 we prove a stronger version of Theorem 6, with a weaker requirement on the notion of locality for the checking function in the case where there is a $\mathsf{QNC^0}$ circuit that characterizes the relation.

4.1 An efficient algorithm for efficiently checkable relations

In this section we prove Theorem 7, (which itself implies Theorem 5). We construct a classical algorithm that given a goal input $x \in \mathcal{X}$ will iteratively update some initial valid input-output pair $(0^m, y_0)$ by changing a single bit of the input at a time until arriving at a final pair $(x, y) \in R$ for the goal x and some y. We can assume we have access to the pair $(0^m, y_0)$ since such a pair exists, and we are only proving the *existence* of such an algorithm.

For each $i \in [m]$, we will construct a function $\mathsf{update}_i : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ such that for each $(x, y) \in R$, and $(x', y') := \mathsf{update}_i(x, y)$, we have that $(x', y') \in R$ and $x' = x \oplus e_i$, where we use the notation that $x \oplus e_i$ is the string x with the ith bit flipped. Once we have an update_i function for each $i \in [m]$, we can then use them iteratively to get from any valid pair $(x_0, y_0) \in R$ to another (x', y') for any given $x' \in \mathcal{X}$.

Definition 16 (update_i)**.** *For a relation R, we define a subroutine $\mathsf{update}_i : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ for each input index $i \in [m]$. For each $(x, y) \in R$, $\mathsf{update}_i(x, y)$ must produce a (x', y') such that*

(a) x' is x flipped at the ith bit: $x' := x \oplus e_i$, and

(b) $(x', y') \in R$.

For notational clarity in our proofs throughout this chapter, we define $A_i := \overrightarrow{\mathcal{L}_Q}(x_i)$ and $A_i^{\mathsf{c}} := [n] \setminus A_i$ for each $i \in [m]$.

Claim 5. *For each relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ with an efficiently computable checking function, if there exists a quantum circuit Q with blowup B that 2^{-B}-approximately characterizes R, then update_i can be implemented (classically) in time $2^B \cdot \mathsf{poly}(n, m)$.*

Proof. Suppose the quantum circuit Q characterizes relation $R \subseteq \mathcal{X} \times \mathcal{Y}$, and R has checking function $F(x, y)$ which can be computed in time $f(n)$.

The algorithm for update_i is as follows:

Algorithm 1: update_i

```
Input  : $(x, y) \in R$
Output: $(x', y') \in R$ with $x' = x \oplus e_i$.
Let $S := \{z \in \mathcal{Y} : z_{A_i^{\mathsf{c}}} = y_{A_i^{\mathsf{c}}}\}$
$x' \leftarrow x \oplus e_i$
$y' \leftarrow \operatorname{argmin}_{z \in S} F(x', z)$
output $(x', y')$
```

Correctness: This algorithm always meets Item a of the update_i subroutine, by always outputting $x' = x \oplus e_i$ for input $x \in \mathcal{X}$. Therefore, to show correctness, all that remains is to prove that if $(x, y) \in R$, then $\mathsf{update}_i(x, y) = (x', y') \in R$ also. Let $S := \{z \in \mathcal{Y} : z_{A_i^{\mathsf{c}}} = y_{A_i^{\mathsf{c}}}\}$ as defined in Algorithm 1. The marginal distribution of the quantum circuit output at A_i^{c} is independent of the ith input x_i since $\overrightarrow{\mathcal{L}_Q}(x_i) \cap A_i^{\mathsf{c}} = \emptyset$. Therefore we can lower bound the probability that the quantum circuit outputs a string in S.

$$\begin{aligned}
\Pr[Q(x \oplus e_i) \in S] &= \Pr\big[Q(x \oplus e_i)_{A_i^{\mathsf{c}}} = y_{A_i^{\mathsf{c}}}\big] && (4.15)\\
&= \Pr\big[Q(x)_{A_i^{\mathsf{c}}} = y_{A_i^{\mathsf{c}}}\big] && (4.16)\\
&\geq \Pr[Q(x) = y] \geq p && (4.17)
\end{aligned}$$

Now, let $\tilde{y} \in S$ be the element of S with the highest probability

$$\tilde{y} := \text{argmax}_{z \in S} \Pr[Q(x \oplus e_i) = z | Q(x \oplus e_i) \in S] \tag{4.18}$$

Using a simple counting argument, $\Pr[Q(x \oplus e_i) = \tilde{y} | Q(x \oplus e_i) \in S] > \frac{1}{|S|} \geq \frac{1}{2^B}$, since $\Pr[Q(x \oplus e_i) \in S] > 0$. Combining this with Equation (4.17) we can show that $\tilde{y}$ is output with significant probability.

$$\Pr[Q(x \oplus e_i) = \tilde{y}] \geq \Pr[Q(x \oplus e_i) = \tilde{y} | Q(x \oplus e_i) \in S] \cdot \Pr[Q(x \oplus e_i) \in S] \tag{4.19}$$

$$\geq \frac{p}{2^B} \tag{4.20}$$

Since Q 2^{-B}-approximately characterizes R, it follows that $(x \oplus e_i, \tilde{y}) \in R$. Since $\tilde{y}$ is in the set S that our algorithm minimized over, the y' output by our algorithm satisfies

$$F(x \oplus e_i, y') \leq F(x \oplus e_i, \tilde{y}). \tag{4.21}$$

Therefore $(x \oplus e_i, y') \in R$ by definition of the checking function.

Runtime: This algorithm searches over $2^{|A_i|} \leq 2^B$ strings $z \in S$ and evaluates the function $F(x \oplus e_i, z)$ on each. Since F has a runtime of $f(n)$, the total runtime is $O(2^B f(n))$. □

Proof of Theorem 7. Now that we have an efficient update_i function for each $i \in [m]$, we can start with some arbitrary pair in the relation $(x^{(0)}, y^{(0)})$ and iteratively update the pair to get a pair for a desired $x \in \mathcal{X}$ as shown in Algorithm 2. The pair $(x^{(0)}, y^{(0)})$ is hardcoded into our algorithm.

Algorithm 2: Classical algorithm to satisfy relation R

```
Input   : $x \in \mathcal{X}$
Output: $y \in \mathcal{Y} : (x, y) \in R$.
Let $(x^{(0)}, y^{(0)}) \in R$ be some known pair.
Initialize $(\hat{x}, \hat{y}) \leftarrow (x^{(0)}, y^{(0)})$
for $i \in [m]$ do
    if $x_i \neq x'_i$ then
        $(\hat{x}, \hat{y}) \leftarrow \mathsf{update}_i(\hat{x}, \hat{y})$
    end
end
output $y := \hat{y}$
```

Correctness: First note that at each iteration, the $\hat{x}$ variable is only updated at its ith index such that it takes on the value of $\hat{x} \oplus e_i$. Therefore, after the last iteration, $\hat{x} = x$. Since the pair $(\hat{x}, \hat{y})$ at the start is valid $(\hat{x}, \hat{y}) \in R$, and update_i preserves the relation, at the end of each iteration, we still have that $(\hat{x}, \hat{y}) \in R$. Therefore, the algorithms final output y is correct: $(x, y) \in R$.

Runtime: By Claim 5, the update_i subroutine takes $2^B \cdot \mathsf{poly}(n, m)$ time, which we may run within each of the m iterations of Algorithm 2. Thus the runtime of Algorithm 2 is at most $m \cdot 2^B \mathsf{poly}(n, m) = 2^B \cdot \mathsf{poly}(n, m)$ □

4.2 NC^0 circuits for locally-checkable relations

To prove Theorem 8, we will show that Algorithm 2 for satisfying a relation R, can be parallelized into constant depth if R is *locally checkable*. When the checking function is local, we can implement the update_i functions locally, and thus implement many in parallel.

For a local checking function $F(x, y) = \sum_{i\in[t]} F_i(x_{T_i}, y_{S_i})$ we say that two variables share a check if they are acted on by the same checking term. It will be useful to define the *check neighborhoods* of output variables to denote the other variables they share a check with.

Definition 17 (Output Check Neighborhood N_F). *For each output $k \in [n]$, we define its* output check neighborhood *as $N_F(y_k) := \bigcup_{i\in[t]:y_k\in S_i} S_i$. Similarly, we define the output check neighborhood of any subset of outputs $S \subseteq [n]$ as $N_F(S) := \bigcup_{i\in[t]:S_i\cap S\neq\emptyset} S_i$.*

Definition 18 (Input Check Neighborhood, N_F^{in}). *For each output $k \in [n]$, we define its input check neighborhood as $N_F^{in}(y_k) := \bigcup_{i\in[t]:y_k\in S_i} T_i$. Similarly, for each subset of output bits $S \subseteq [m]$ we define $N_F^{in}(S) := \bigcup_{i\in[t]:S_i\cap S\neq\emptyset} T_i$.*

We now provide a parameterized definition of checking function locality.

Definition 19 ((K, L)-local checking function). *A checking function is (K, L)-local if*

$$|N_F^{in}(S)| \le K|S| \qquad\qquad |N_F(S)| \le L|S| \qquad\qquad \textit{for each } S \subseteq [n] \tag{4.22}$$

Note that our notion so far of *local checking functions* is equivalent to $(O(1), O(1))$-local checking functions.

In the process of upper bounding the circuit complexity of our classical algorithms, we will prove their *function locality*. Slightly overloading "locality", we say that a function $f : \{0,1\}^m \to \{0,1\}^n$ is ℓ-local if each output bit $f(z)_i$ can be written as a function of only ℓ of the inputs $f_i : \{0,1\}^\ell \to \{0,1\}$. In Appendix A, we show how to upper bound circuit complexity from function locality.

Theorem 9 (Theorem 8 restated). *For each relation R with a (K, L)-local checking function, if there exists a quantum circuit Q with blowup B that 2^{-B}-approximately characterizes R, then there exists a $(B(2K+L))^{(B^2L)}$-local function that satisfies R which can be implemented by a classical circuit with fan-in 2, depth $O(B^3L(2K + L))$, and size $O(B^3L(2K + L) \cdot 2^{B(2K+L)})$.*

Proof. Recall that for a particular quantum circuit Q, we use the notation $A_i = \overrightarrow{\mathcal{L}_Q}(x_i)$ and $A_i^{\mathsf{c}} = [n] \setminus A_i$. Additionally, we will define the *boundary* of A_i as $\delta(A_i) = N_F(A_i) \setminus A_i$.

Claim 6. *For each $i \in [m]$, $\mathsf{update}_i(x, y)$ can be implemented to only flip the ith bit of x, and update y_{A_i} as a function of $x_{N_F^{in}(A_i)}$ and $y_{\delta(A_i)}$.*

Proof. The local algorithm for update_i is the same as Algorithm 1, except for the step $y' \leftarrow \mathrm{argmin}_{z\in S} F(x \oplus e_i, z)$ can be implemented without actually computing the entire checking function $F(x, z)$, but instead just the sum of the terms that act on the y bits that we are changing: y_{A_i}.

$$\operatorname{argmin}_{z \in S} \sum_{i \in [t] : S_i \cap A_i \neq \emptyset} F_i(x'_{T_i}, z_{S_i}) \tag{4.23}$$

This is equivalent to choosing a y' that minimizes the entire checking function because we are simply ignoring the terms whose values are fixed to the same value for each $z \in S$. Therefore, the correctness of the algorithm follows from the correctness of Algorithm 1 proved in Claim 5. Calculating the sum of these local terms depends only on the variables $x_{N_F^{in}(A_i)}$ and $y_{N_F(\delta(A_i))}$. Clearly, the update on x only flips x_i. Moreover, we are only updating y at its bits in A_i, as desired. □

We generalize the update_i subroutine from the previous section to update multiple inputs as update_V for some subset of inputs $V \subseteq [m]$.

Definition 20 (update_V)**.** *For some subset of input indices $V \subseteq [m]$, we define the subroutine $\mathsf{update}_V : \mathcal{X} \times \mathcal{Y} \times \{0,1\}^{|V|}$. For each $(x, y) \in R$ and $v \in \{0,1\}^{|V|}$, update_i must produce an (x', y') such that the following are true:*

(a) x' is flipped at the indices in V according to v.

$$x'_V = x_V \oplus v, \qquad x'_{V^c} = x_{V^c} \tag{4.24}$$

(b) $(x', y') \in R$.

Clearly for any $V \subseteq [m]$, we can implement update_V by implementing each update_i in series for each $i \in V$. However, we will show, that for certain choices of V we can parallelize these updates.

Claim 7. *For each $V \subseteq [m]$ if*

$$N_F(\overrightarrow{\mathcal{L}_Q}(a)) \cap \overrightarrow{\mathcal{L}_Q}(b) = \emptyset \qquad\qquad \textit{for each } a \neq b \in V \tag{4.25}$$

then update_V can be implemented with a $B(2K + L)$-local function.

Proof. Let $V \subset [m]$ be some subset of inputs satisfying Equation (4.25), with some ordering $V = \{V_1, V_2, \ldots, V_m\}$. Consider the implementation of $\mathsf{update}_V(x, y, v)$ shown in Algorithm 3 which iteratively updates with update_i for each $i \in V$ with $v_i = 1$.

We will denote the initial values for (x, y) as $(x^{(0)}, y^{(0)})$, and for each $i \in \{1, 2, \ldots, |V|\}$, let $(x^{(i)}, y^{(i)})$ denote the value of (x, y) after the ith iteration (after update_{V_i}).

Note that for each $i \in V$, each update_i updates disjoint variables. This is because each update_i changes only the bits x_i and y_{A_i} (by Claim 6), and $A_i \cap A_j = \emptyset$ for each $i \neq j \in V$ by Equation (4.25). Therefore it is sufficient to show that each $\mathsf{update}_i(x, y)$ can be implemented as a $B(2K + L)$-local function of the initial inputs $(x^{(0)}, y^{(0)}, v)$.

Algorithm 3: $\mathsf{update}_V(x, y, v)$

```
for $j \in \{1, \ldots, |V|\}$ do
    if $v_j = 1$ then
        $(x, y) \leftarrow \mathsf{update}_{V_j}(x, y)$
    end
end
Output $(x, y)$
```

By Claim 6, for each $i \in [m]$, the update step $(x, y) \leftarrow \mathsf{update}_i(x, y)$ can be replaced by just flipping x_i and applying a local update function for y, $\mathsf{localupdate}_i : \{0,1\}^{|N_F^{in}(A_i)|} \times \{0,1\}^{|\delta(A_i)|} \to \{0,1\}^{|A_i|}$ such that

$$(x, y) \leftarrow \mathsf{update}_i(x, y) \quad = \quad \begin{cases} y_{A_i} & \leftarrow \mathsf{localupdate}_i(x_{N_F^{in}(A_i)}, y_{\delta(A_i)}) \\ x_i & \leftarrow 1 - x_i \end{cases}. \tag{4.26}$$

Next, we will show that each $\mathsf{localupdate}_i$ depends (still locally) only on the original inputs $(x^{(0)}, y^{(0)}, v)$, and therefore can be implemented independently. To do this, it is sufficient to show that $x_{N_F^{in}(A_i)}$ and $y_{\delta(A_i)}$ depend locally on $(x^{(0)}, y^{(0)}, v)$.

Claim 8. *$y^{(j)}_{\delta(A_i)} = y^{(0)}_{\delta(A_i)}$ for each $i, j \in [|V|]$.*

Proof. Note that for each $i \in V$ the $\delta(A_i)$ bits of y do not change throughout our update algorithm. This is because only the y bits of $\cup_{i \in V} A_i$ change and for each $a \neq b \in V$, $\delta(A_a) \cap A_b \subseteq N_F(A_a) \cap A_b = \emptyset$ and $\delta(A_i) \cap A_i = \emptyset$ for each $i \in V$. That is $y^{(j)}_{\delta(A_i)} = y^{(0)}_{\delta(A_i)}$ for each $i, j \in [|V|]$. □

Next, we will show that for each step of the algorithm $j \in [|V|]$, $x_{N_F^{in}(A_j)}$ is determined by at most $|N_F^{in}(A_j)|$ bits of each $x^{(0)}$ and v.

Let $\hat{v} \in \{0,1\}^m$ denote the change in x determined by the input v, that is $\hat{v} := x^{(0)} \oplus x^{(|V|)}$.

$$\hat{v}_{V_i} = v_i \quad \text{for } i \in [|V|], \qquad \text{and} \qquad (\hat{v}(j))_i = 0 \quad \text{for } i \in [m] \setminus V \tag{4.27}$$

Furthermore, for each $j \in [|V|]$, let $\hat{v}(j) := x^{(j)} \oplus x^{(0)}$. Note that $\hat{v}(j)$ is now a function of $v_1, v_2 \ldots v_j$.

$$(\hat{v}(j))_{V_i} = \begin{cases} v_i & \text{if } i \leq j \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i \in [|V|], \qquad \text{and} \qquad (\hat{v}(j))_i = 0 \quad \text{for } i \in [m] \setminus V \tag{4.28}$$

So for each $j \in [|V|]$, we can write $x^{(j)}$ as $x^{(0)} \oplus \hat{v}(j)$.

So the local update step can be rewritten as a local function of $(x^{(0)}, y^{(0)}, \hat{v})$.

$$(x, y) \leftarrow \mathsf{update}_i(x, y) \quad = \quad \begin{cases} y_{A_i} & \leftarrow \mathsf{localupdate}_i(x^{(0)}_{N_F^{in}(A_i)} \oplus \hat{v}(j)_{N_F^{in}(A_i)}, y^{(0)}_{\delta(A_i)}) \\ x_i & \leftarrow 1 - x_i \end{cases}. \tag{4.29}$$

Note that each of these $\mathsf{localupdate}_i$ is a function of at most $2|N_F^{in}(A_i)|+|\delta(A_i)| \leq 2B(K+L)$ bits of $(x^{(0)}, y^{(0)}, v)$. Therefore, each output bit of update_V is also a function of at most $2B(K+L)$ bits of $(x^{(0)}, y^{(0)}, v)$. □

Claim 9. *There exists a partition of the input bits into $k \leq B^2L$ subsets $[n] = V^{(1)} \uplus V^{(2)} \uplus \cdots \uplus V^{(k)}$ such that for each $i \in [k]$*

$$N_F(\overrightarrow{\mathcal{L}_Q}(x_a)) \cap \overrightarrow{\mathcal{L}_Q}(x_b) = \emptyset \qquad\qquad \textit{for each } a \neq b \in V^{(i)} \tag{4.30}$$

Proof. Consider the graph G with each vertex representing an outut $V(G) = [n]$, and edges defined by the negation of Equation (4.25).

$$E(G) = \{(a,b) : N_F(\overrightarrow{\mathcal{L}_Q}(x_b)) \cap \overrightarrow{\mathcal{L}_Q}(x_a) \neq \emptyset, a \neq b \in [n]\} \tag{4.31}$$

Note that this condition is symmetric:

$$N_F(\overrightarrow{\mathcal{L}_Q}(x_b)) \cap \overrightarrow{\mathcal{L}_Q}(x_a) \neq \emptyset \iff N_F(\overrightarrow{\mathcal{L}_Q}(x_a)) \cap \overrightarrow{\mathcal{L}_Q}(x_b) \neq \emptyset.$$

By the definition of our edges $E(G)$, any independent set $I \subseteq V(G)$ (no edges between any vertices in I) satisfies Equation (4.25). Any proper k-vertex coloring forms a partition of the vertices into k independent sets. Note that G has maximum degree at most B^2L-1 because $|\overleftarrow{\mathcal{L}_Q}(N_F(\overrightarrow{\mathcal{L}_Q}(x_a)))| \leq B^2L$ for each $a \in [n]$, and $x_a \in \overleftarrow{\mathcal{L}_Q}(N_F(\overrightarrow{\mathcal{L}_Q}(x_a)))$. Therefore, there exists a k-vertex coloring for $k \leq B^2L$ [Viz65]. Let $V^{(1)}, V^{(2)}, \ldots V^{(k)}$ be the color classes for such a coloring. □

Finally, we can describe our algorithm which satisfies R:

Algorithm 4: Local algorithm for a locally-checkable relation

```
Input  : $x \in \mathcal{X}$
Output: $y \in \mathcal{Y} : (x,y) \in R$.
Let $(x^{(0)}, y^{(0)}) \in R$ be some known pair.
Let $V^{(1)}, V^{(2)}, \ldots, V^{(k)} \subseteq [m]$ be the partition of $[m]$ into $k \leq B^2L$ sets from Claim 9
Initialize $(\hat{x}, \hat{y}) \leftarrow (x^{(0)}, y^{(0)})$
for $i \in [k]$ do
    if $x_i \neq x'_i$ then
        $(\hat{x}, \hat{y}) \leftarrow \mathsf{update}_{V^{(k)}}(\hat{x}, \hat{y})$
    end
end
output $y := \hat{y}$
```

Combining Claims 7 and 9, the function implemented by Algorithm 4 which iteratively applies $\mathsf{update}_{V^{(i)}}$ for each $i \in [k]$ has locality $(B(2K+L))^k \leq (B(2K+L))^{B^2L}$. Moreover, for each $i \in [k]$, since $\mathsf{update}_{V^{(i)}}$ has locality $B(2K+L)$, it can be implemented by a classical circuit with fan-in 2, depth $O(B(2K+L))$ and size $O(B(2K+L) \cdot 2^{B(2K+L)})$ by Lemma 11 in Appendix A. Therefore, iteratively applying $\mathsf{update}_{V^{(i)}}$ for each $i \in [k]$ can be implemented by a classical circuit with fan-in 2, depth $O(k \cdot (B(2K+L)) = O(B^3L(2K+L))$ and size $O(k \cdot B(2K+L) \cdot 2^{B(2K+L)}) = O(B^3L(2K+L) \cdot 2^{B(2K+L)})$.

□

4.3 Removing input locality for relations characterized by shallow quantum circuits

We actually prove a result that is slightly stronger than Theorem 6. In the case where a quantum circuit characterizes a relation, our theorem statement only requires that the checking function is local in the output, rather than the output *and* the input.

Theorem 10. *Suppose the relation R has a checking function that is L-local in output ($|N_F(S)| \leq L|S|$ for each $S \subseteq [n]$). If there exists a quantum circuit with blowup B that characterizes R, then there exists a $(BL(2B+1))^{B^2L}$-local function that satisfies R which can be implemented by a classical circuit with fan-in 2, depth $O(B^3L(2K+L))$, and size $O(B^3L(2K+L)\cdot 2^{B(2K+L)})$.*

In the case where a relation R is characterized by some quantum circuit with blowup B, we can upper bound the output locality of the checking function for R with its input locality and B.

Lemma 5. *Suppose the relation R has a quantum circuit with blowup B that characterizes R. If R has a checking function that is L local in output, then R has a (BL, L)-local checking function.*

Proof. Let $F(x,y) = \sum_{i\in[t]} F_i(x, y_{S_i})$ be the L-local checking function for R, with $S_i \subseteq [n]$. We define a new relation $R' \subseteq \mathcal{X} \times \mathcal{Y}$ such that $(x,y) \in R$ if y is in the marginal support of $Q(x)$ for each S_i.

$$R' = \{(x,y) : x \in \mathcal{X}, y_{S_i} \in \mathrm{supp}(Q(x)_{S_i}) \text{ for each } i \in [t]\}$$

Which has a corresponding checking function $F'(x,y) = \sum_{i\in[t]} F'_i(x, y_{S_i})$. Where check i verifies that $y_{S_i} \in \mathrm{supp}(Q(x)_{S_i})$.

$$F'_i(x, y_{S_i}) = \begin{cases} 0 & \text{if } y_{S_i} \in \mathrm{supp}(Q(x)_{S_i}) \\ 1 & \text{if } y_{S_i} \notin \mathrm{supp}(Q(x)_{S_i}) \end{cases}$$

Note that $\mathrm{supp}(Q(x)_{S_i})$ is determined by the backwards lightcone of S_i for each $i \in [t]$. Setting $T_i = \overleftarrow{\mathcal{L}}[Q](S_i)$, we can rewrite $F'(x,y)$ as $F'(x,y) = \sum_{i\in[t]} F'_i(x_{T_i}, y_{S_i})$. Thus, R' is (BL, L)-local.

Next we will prove that $R' = R$. Since $R = \{(x,y) : x \in \mathcal{X}, y \in \mathrm{supp}(Q(x))\}$, clearly $R \subseteq R'$. To prove the other direction, we consider the expected value of $F(x,y)$. For each

$x \in \mathcal{X}$, let $\mathcal{Y}_x := \min_{y \in \mathcal{Y}} F(x,y)$.

$$\begin{aligned}
\mathcal{Y}_x &= F(x,y) && \text{for each } (x,y) \in R \\
&= \mathbb{E}[F(x,y)] && \text{for each } x \in \mathcal{X}, y \sim Q(x) \\
&= \sum_{i \in [t]} \mathbb{E}[F_i(x, y_{S_i})] && \text{for each } x \in \mathcal{X}, y \sim Q(x) \\
&= \sum_{i \in [t]} \mathbb{E}_{y_{S_i} \sim Q(x)_{S_i}}[F_i(x, y_{S_i})] && \text{for each } x \in \mathcal{X} \\
&= \mathbb{E}[F(x,y)] && \text{for each } x \in \mathcal{X}, y \sim_S Q(x)
\end{aligned}$$

Where $y \sim_S Q(x)$ denotes that $y \sim \widetilde{Q(x)}$, where $\widetilde{Q(x)}$ is a distribution over $\mathcal{Y}$ such that $\widetilde{Q(x)}_{S_i} \sim Q(x)_{S_i}$ for each $i \in [t]$. Since $F(x,y) \geq \mathcal{Y}_x$ for each $x \in \mathcal{X}, y \in \mathcal{Y}$, there is no probability mass of $F(x,y)$ below its expectation: $\Pr_{y \sim_S Q(x)}[F(x,y) < \mathbb{E}[\mathcal{Y}_x]] = 0$. Therefore, there is also no probability above the expectation: $\Pr_{y \sim_S Q(x)}[F(x,y) > \mathbb{E}[\mathcal{Y}_x]] = 0$. It follows that

$$\begin{aligned}
Pr[F(x,y) = \mathcal{Y}_x] = 1 && \text{for each } x \in \mathcal{X}, y \sim_S Q(x) \\
F(x,y) = \mathcal{Y}_x && \text{for each } (x,y) \in R'
\end{aligned}$$

Therefore $R' \subseteq R$, so $R' = R$ as desired. □

Proof of Theorem 10. By Lemma 5, R has a (BL, L)-local checking function. Plugging this into Theorem 9 with $K = BL$, there exists a $(BL(2B+1))^{B^2L}$-local function that satisfies R. □

Chapter 5

The Sampling Power of Shallow Quantum Circuits with GHZ Input

This chapter is joint work with and coauthored by Adam Bene Watts.

This chapter studies the sampling power of constant depth quantum circuits with bounded fan-in (QNC^0). We have already seen in Section 2.5 that sampling exactly from the output distributions of QNC^0 circuits for any input is hard in general for an *efficient* classical algorithm (assuming the PH does not collapse). And in Chapter 3 we saw that there exist search problems that can be solved by QNC^0 circuits but not constant-depth bounded fan-in classical (NC^0) circuits. However, perhaps counterintuitively, neither of these results directly provide a *specific* distribution, which does not depend on any input, that can be produced by a QNC^0 circuit but not an NC^0 circuit. The question of whether such a distribution exists was proposed in [BGK18].

Answering this question is the basis of ongoing research with Adam Bene Watts. In this chapter, we prove a sampling separation for a weaker model. We consider the sampling power of a QNC^0 circuit that takes as input an n-qubit GHZ state: $|\mathrm{GHZ}_n\rangle = \frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle)$. We construct a sampling task for which QNC^0 circuits with a GHZ state as input outperform NC^0 circuits that take uniformly random bits as input. Before making this statement more rigorous, we introduce our sampling task.

Sampling task We consider the task of sampling from the distribution $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$, where X is sampled uniformly from $\{0,1\}^{n-1}$. The functions majmod_p and parity are defined as

$$\mathrm{majmod}_p(x) := \begin{cases} 0 & \text{if } \sum_i x_i \le p/2 \pmod{p} \\ 1 & \text{otherwise} \end{cases} \qquad \mathrm{parity}(x) := \sum_i x_i \mod 2 \qquad (5.1)$$

for $x \in \{0,1\}^{n-1}$ and prime p. The function majmod_p (majority mod p) was introduced by Viola, where he showed the hardness of sampling from the distribution $(X, \mathrm{majmod}_p(X))$ for classical circuits with bounded fan-in and unbounded fan-out (NC_0 circuits) [Vio12]. The suggestion to look into Viola's work on classical lower bounds for distributions was also proposed in [BGK18].

In contrast with the hardness of sampling result [TD02, Aar04] discussed in Section 2.5, in this chapter we consider separations for *approximate* sampling tasks (additive error) rather than *exact* sampling. As discussed in Section 2.5.2, approximate sampling is a more reasonable task since in practice we do not expect quantum computers to sample exactly either (there will always be *some* noise).

The measure we use to define how close a sampling algorithm is to a distribution is *total variation distance*.

Definition 21 (Total variation distance, Δ)**.** *The total variation distance (or statistical distance) between two distributions D, D' over $\{0,1\}^m$ is*

$$\Delta(D, D') := \max_{T \subseteq \{0,1\}^m} \Big| \Pr[D \in T] - \Pr[D' \in T] \Big| = \frac{1}{2} \sum_{a \in \{0,1\}^m} \Big| \Pr[D = a] - \Pr[D' = a] \Big| \tag{5.2}$$

A sampling algorithm $\mathcal{A}$ is said *to approximately sample* from the distribution D with *additive error* ϵ, if $\Delta(\mathcal{A}, D) \leq \epsilon$. We consider a rather loose requirement for our sampling task, that we just need to approximately sample with *constant* ($< 1/2$) additive error. We are now ready to state the main result of this chapter.

Theorem 11. *For any prime p, define the distribution $D_{n,p} := (X, majmod_p(X) \oplus parity(X))$ with X sampled uniformly from $\{0,1\}^{n-1}$. For each $\delta < 1$, sufficiently large n, and prime $p \in [n^{1/\delta}, n^{1/3}]$:*

- *There exists a constant-depth quantum circuit with fan-in 2 which takes $|GHZ\rangle_n$ as input, and produces an output distribution with total variation distance $\frac{1}{2} - \Omega(1)$ from $D_{n,p}$.*
- *Each classical circuit with fan-in 2 and depth $\Omega(\log\log(n))$ which takes $n + n^\delta$ random bits as input, has total variation distance $\frac{1}{2} - O(1/\log(n))$ from $D_{n,p}$.*

In Section 5.1 we construct the quantum circuit that takes as input a GHZ state and produces an output that is close to the distribution $(X, \text{majmod}_p(X) \oplus \text{parity}(X))$. To this end, we first consider a pseudo-quantum circuit defined with non-unitary gates that gives us the desired behavior. Then we replace these non-unitary gates with actual unitaries, and show that we preserve our desired output statistics. In Section 5.2 we prove the hardness of sampling from $(X, \text{majmod}_p(X) \oplus \text{parity}(X))$ with a constant-depth classical circuit with bounded fan-in. This proof follows closely to Viola's proof of the hardness of sampling $(X, \text{majmod}_p(X))$ [Vio12].

5.1 Sampling from $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$ using a GHZ state

In this section we consider quantum circuits with access to an n-qubit GHZ state as input. We show these circuits can produce samples from the distribution $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$, where X is a uniformly random bitstring of length $n-1$. We will prove this result in two steps – in Section 5.1.1 we give a "quantum-like" circuit which samples from the correct distribution but includes non-unitary single qubit operations. In Section 5.1.2 we show how to replace those non-unitary operations with actual unitaries. Before beginning these proofs we review some details about GHZ states.

Review of GHZ States An n-qubit GHZ state is defined to be the state

$$|\mathrm{GHZ}_n\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n} \right). \tag{5.3}$$

It is a well-known fact that applying a Hadamard transform to each qubit of a GHZ state produces a uniform superposition over bitstrings with even Hamming weight:

$$H^{\otimes n} |\mathrm{GHZ}_n\rangle = 2^{-n/2} \sum_{e \in E_n} |e\rangle \tag{5.4}$$

where E_n is the set containing all even parity n-bit strings. We can equivalently describe this state as coherent superposition of $n-1$ random bits and a final bit whose value equals the parity of the $n-1$ other bits. We can compute the parity of the first $n-1$ bits onto the final qubit by preparing the $|0\rangle$ state and flipping it once for each of the first $n-1$ qubits that are in the $|1\rangle$ state. So,

$$H^{\otimes n} |\mathrm{GHZ}_n\rangle = \left(\prod_{i=1}^{n-1} \mathrm{CNOT}_{i,n} \right) |+\rangle^{\otimes n-1} \otimes |0\rangle. \tag{5.5}$$

where $\mathrm{CNOT}_{i,j}$ denotes a CNOT gate controlled on qubit i and applied to qubit j. Equation (5.5) will be our starting point for designing circuits which use the GHZ state as a resource state.

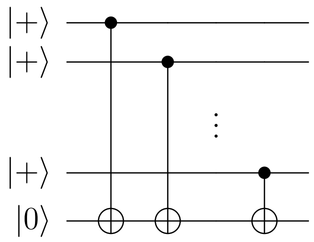


Figure 5.1: A circuit constructing the state $H^{\otimes n} |\mathrm{GHZ}_n\rangle$, as described in Equation (5.5).

5.1.1 Sampling with non-unitary operations

We now consider constant depth quantum circuits augmented with specific single qubit non-unitary "gates" A_θ, which we will soon define. We show these circuits can sample (approximately) from the distribution $(X, \text{majority}(X) + \text{parity}(X))$. While this model is obviously non-physical, introducing it allows us to isolate some key ideas which we will reuse in the fully quantum circuit developed in the next section.

First, for each $\theta \in \mathbb{R}$, define the (non-unitary) matrix A_θ to be the two qubit matrix which acts on the computational basis states as

$$A_\theta |0\rangle = |0\rangle \tag{5.6}$$
$$A_\theta |1\rangle = \exp(i\theta X) |1\rangle \tag{5.7}$$

When drawing circuit diagrams in this section we sometimes include A_θ gates, and understand that they represent the matrix A acting on the qubits indicated. We also sometimes drawn $A_\theta^\dagger$ gates, which represent the adjoint of the matrix A_θ acting on the qubits indicated.

We now prove the following useful circuit identity.

Lemma 6. *For any one-qubit state $|\psi\rangle$ and computational basis state $|x\rangle$ with $x \in \{0,1\}$, we have*

$$\langle x|_2 \left(A_\theta^\dagger\right)_2 CNOT_{2,1} |\psi\rangle_1 |+\rangle_2 = \frac{1}{\sqrt{2}} \exp(i(\theta + \pi/2)xX_1) |\psi\rangle_1 \tag{5.8}$$

Proof. Direct computation gives

$$\begin{aligned}\langle x|_2 \left(A_\theta^\dagger\right)_2 \mathrm{CNOT}_{2,1} |\psi\rangle_1 |+\rangle_2 &= \langle x|_2 \exp(i\theta x X_2)\mathrm{CNOT}_{2,1} |\psi\rangle_1 |+\rangle_2 && (5.9)\\ &= \langle x|_2 \,\mathrm{CNOT}_{2,1} \exp(i\theta x X_1 X_2) |\psi\rangle_1 |+\rangle_2 && (5.10)\\ &= \langle x|_2 \,\mathrm{CNOT}_{2,1} \exp(i\theta x X_1) |\psi\rangle_1 |+\rangle_2 && (5.11)\\ &= \exp(i(\theta + \pi/2)xX_1) |\psi\rangle_1 \langle x|+\rangle_2 && (5.12)\\ &= \frac{1}{\sqrt{2}} \exp(i(\theta + \pi/2)xX_1) |\psi\rangle_1 && (5.13)\end{aligned}$$

where we used on the first line that

$$A_\theta |x\rangle = \exp(i\theta X x) |x\rangle \tag{5.14}$$

by definition, the commutation relation[1]

$$X_2 \mathrm{CNOT}_{2,1} = \mathrm{CNOT}_{2,1} X_1 X_2 \tag{5.15}$$
$$\implies \exp(i\theta X_2)\mathrm{CNOT}_{2,1} = \mathrm{CNOT}_{2,1} \exp(i\theta X_1 X_2) \tag{5.16}$$

on the second line, that $|+\rangle$ is a 1-eigenstate of the X operator on the third line, and then the definition of the CNOT gate and the $|+\rangle$ state on the final two lines. Figure 5.2 gives a diagrammatic version of this proof. □

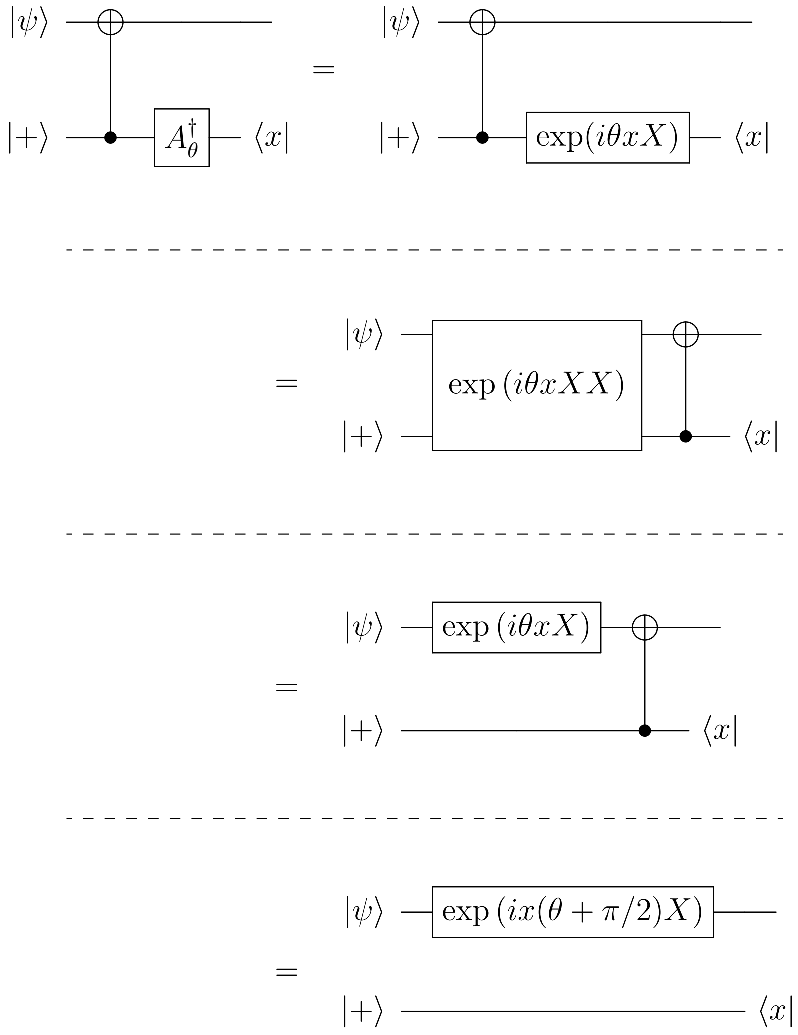

$|\psi\rangle$
$|+\rangle$
$A_\theta^\dagger$
$\langle x|$
$=$
$\exp(i\theta x X)$
$\exp\left(i\theta x X X\right)$
$\exp\left(i\theta x X\right)$
$\exp\left(i x(\theta + \pi/2) X\right)$

Figure 5.2: A diagrammatic proof of Lemma 6. The equivalence between each line is explained in the proof of the lemma.

We now prove the main result of this section, and construct a constant depth circuit with a GHZ state as input and A_θ gates which samples approximately from the distribution $(X, \mathrm{majmod}_p(X))$ for any p. The construction builds on Lemma 6 as well as the observations about the GHZ state discussed in Section 5.1.

Theorem 12. *For each prime number p there is a constant depth circuit consisting of one and two-qubit unitary gates and A_θ operations which takes a GHZ state as input and produces an output which, when measured in the computational basis, produces a distribution with total variation distance $\frac{1}{2} - \Omega(1)$ from the distribution $(X, majmod_p(X) \oplus parity(X))$ for sufficiently large n.*

[1] To prove the implication, use the standard decomposition $\exp(i\theta X) = \cos(\theta) + i\sin(\theta)X$, then commute the resulting terms.

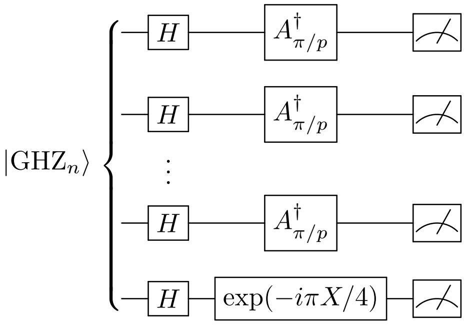


Figure 5.3: Constant depth circuit producing approximate samples from the distribution $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$.

Proof. We first describe the circuit which, when measured in the computational basis, produces a distribution close to $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$. Fix $\theta = \pi/p$. The circuit takes as input a GHZ state, applies a Hadamard transform to each qubit of the state, then applies a $A^\dagger_\theta$ operation to the first $n-1$ qubits in the GHZ state and a $\exp(-i\pi X/4)$ rotation to the final qubit. This circuit is indicated diagrammatically in Figure 5.3.

To prove this circuit samples (approximately) from the correct distribution we write the (unnormalized) output state of the circuit conditioned on first $n-1$ qubits of the circuit being measured in computational basis state $|x\rangle = |x_1\rangle \otimes |x_2\rangle \otimes ... \otimes |x_{n-1}\rangle$ as:

$$\langle x|_{1...n-1} \left(\left(A^\dagger_\theta\right)^{\otimes n-1} \otimes \exp(-i\pi X/4) \right) H^{\otimes n} |\mathrm{GHZ}_n\rangle$$

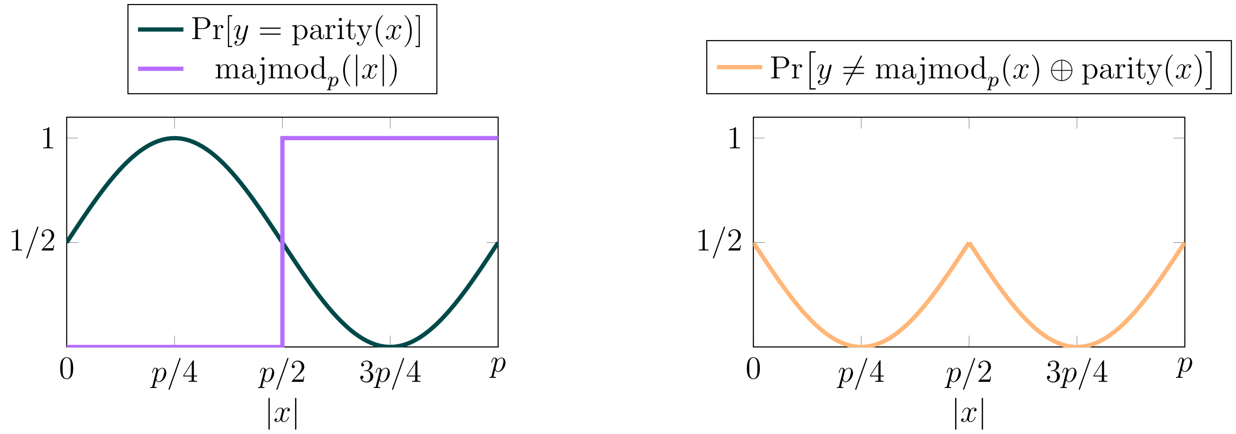
$$= \langle x|_{1...n-1} \left(\left(A^\dagger_\theta\right)^{\otimes n-1} \otimes \exp(-i\pi X/4) \right) \left(\prod_{i=1}^{n-1} \mathrm{CNOT}_{i,n} \right) |+\rangle^{\otimes n-1} \otimes |0\rangle \tag{5.17}$$

$$= \prod_{i=1}^{n-1} \langle x_i| A^\dagger_\theta \, (\mathrm{CNOT}_{i,n}) |+\rangle_i \otimes \exp(-i\pi X/4) \, |0\rangle_n \tag{5.18}$$

$$= 2^{-(n-1)/2} \exp\left(iX \left(-\frac{\pi}{4} + \sum_{i=1}^{n-1} x_i \left(\theta + \frac{\pi}{2} \right) \right) \right) |0\rangle_n \tag{5.19}$$

where we used Equation (5.5) on the first line, reordered terms on the second (noting that $\exp(i\pi X/4)_n$ commutes with $\mathrm{CNOT}_{i,n}$ for any $i \in [n-1]$), and then used Lemma 6 on the third. A diagrammatic version of this analysis is given in Figure 5.5.

Now, tracing over the final qubit we see the probability of the first $n-1$ qubits being measured in any computational basis state $|x\rangle$ is $2^{-(n-1)}$ so the measurement of the first $n-1$ bits produces a uniformly random bit string, as desired. Additionally, conditioning on



(a) Inverse correlation of $\Pr[Y_x = \mathrm{parity}(x)]$ and $\mathrm{majmod}_p(x)$

(b) Probability that Y_x is incorrect, $f(|x|)$

Figure 5.4: Plots displaying the correlation of Y_x and $\mathrm{majmod}_p(x) \oplus \mathrm{parity}(x)$ where Y_x is the last bit output by the circuit in Figure 5.3 conditioned on the first $n-1$ measurements resulting in string $x \in \{0,1\}^{n-1}$.

bit string $x = x_1 x_2 ... x_{n-1}$ being measured, we see the state of the n-th qubit is

$$\exp\Big(iX\left(-\frac{\pi}{4} + |x|\left(\theta + \frac{\pi}{2}\right)\right)\Big)\left|0\right\rangle_n \tag{5.20}$$

$$\propto \exp\Big(iX\left(-\frac{\pi}{4} + \theta|x|\right)\Big)\left|\mathrm{parity}(x)\right\rangle_n \tag{5.21}$$

$$= \cos\Big(-\frac{\pi}{4} + \theta|x|\Big)\left|\mathrm{parity}(x)\right\rangle_n + i\sin\Big(-\frac{\pi}{4} + \theta|x|\Big)\left|1 \oplus \mathrm{parity}(x)\right\rangle_n . \tag{5.22}$$

Where $|x| = \sum_{i=1}^{n-1} x_i$ denotes the hamming weight of x. Let Y_x denote the random variable that is the measurement of the last output bit of the quantum circuit conditioned on the measurement of the first $n-1$ bits being x. When measured in the computational basis, the probability that the measurement Y_x on the nth qubit is $\mathrm{parity}(x)$ is

$$\Pr[Y_x = \mathrm{parity}(x)] = \cos^2\left(-\frac{\pi}{4} + \frac{\pi}{p}|x|\right). \tag{5.23}$$

Where we set $\theta = \frac{\pi}{p}$. Figure 5.4 displays the inverse correlation between $\Pr[Y_x = \mathrm{parity}(x)]$ and $\mathrm{majmod}_p(x)$. We refer to a measured string $(x, y) \in \{0,1\}^{n-1} \times \{0,1\}$ as *correct* if $y = \mathrm{majmod}_p(x) \oplus \mathrm{parity}(x)$. The probability that the last bit Y_x is correct is

$$\Pr\big[Y_x = \mathrm{majmod}_p(x) \oplus \mathrm{parity}(p)\big] = \begin{cases} \cos^2\left(-\frac{\pi}{4} + \frac{\pi}{p}|x|\right), & |x| \le p/2 \\ \sin^2\left(-\frac{\pi}{4} + \frac{\pi}{p}|x|\right), & |x| > p/2 \end{cases}. \tag{5.24}$$

Claim 10. *The probability that the quantum circuit output $(X, Y) \in \{0,1\}^{n-1} \times \{0,1\}$ is incorrect is at most*

$$\Pr\big[Y \neq \mathit{majmod}_p(X) \oplus \mathit{parity}(X)\big] \le \frac{1}{2} - \frac{1}{2\pi} + O(p^{3/2}\exp(-n/p^2)). \tag{5.25}$$

Proof. The probability that the circuit output is invalid is

$$\Pr\big[Y \neq \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X)\big] = \sum_{k=0}^{p-1} \Pr\big[Y \neq \mathrm{majmod}_p \oplus \mathrm{parity}(x)\big||X| = k\big] \cdot \Pr[|X| = k] \tag{5.26}$$

Let $f(k)$ be the probability that our output measurement is incorrect given that the hamming weight of the first n bits have hamming weight k.

$$f(k) := \Pr\big[Y \neq \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X)\big||X| = k\big] \tag{5.27}$$

It follows from Equation (5.24), that

$$f(k) = \begin{cases} \sin^2\left(-\frac{\pi}{4} + \frac{\pi}{p}k\right), & k \leq p/2 \mod p \\ \cos^2\left(-\frac{\pi}{4} + \frac{\pi}{p}k\right), & k > p/2 \mod p \end{cases} \tag{5.28}$$

which is plotted in Figure 5.4b. Let δ be the total variation distance between $|X| \mod p$ and U_p, the uniform distribution over $\{0, 1, \ldots, p-1\}$. Then $\Pr[|X| = k \mod p] \leq \frac{1}{p} + \delta$. We can upper bound Equation (5.26), as

$$\begin{aligned} \Pr\big[Y \neq \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X)\big] &\leq \left(\frac{1}{p} + \delta\right) \sum_{k=0}^{p-1} f(k) && (5.29) \\ &= \left(\frac{1}{p} + \delta\right) \left(\frac{1}{2} + 2 \sum_{k=1}^{(p-1)/2} f(k)\right) && (5.30) \\ &= \left(\frac{1}{p} + \delta\right) \left(\frac{1}{2} + 2 \int_{1/2}^{p/2} f(k)\right) dk && (5.31) \end{aligned}$$

Where in the second line we use the fact that $f(k)$ is symmetric about $p/2$, so $\sum_{k=1}^{(p-1)/2} f(k) = \sum_{k=(p+1)/2}^{p-1} f(k)$. In the third line we used that $f(k)$ is convex over $(0, p/2)$, and therefore $\sum_{i=1}^{(p-1)/2} f(k)$ is a (midpoint-Riemann sum) overapproximation of $\int_{1/2}^{p/2} f(k)$. Next we evaluate the integral.

$$\begin{aligned} \int_{1/2}^{p/2} f(k)\, dk &= \int_0^{p/2} \sin^2\left(-\frac{\pi}{4} + \frac{\pi}{p}k\right) dk && (5.32) \\ &= \int_0^{p/2} \frac{1}{2}\left(1 + \cos\left(\frac{2\pi}{p}k + \frac{\pi}{2}\right)\right) dk && (5.33) \\ &= \frac{1}{2}\left(k + \frac{p}{2\pi}\sin\left(\frac{2\pi}{p}k + \frac{\pi}{2}\right)\right)\Bigg|_0^{p/2} && (5.34) \\ &= \frac{p}{4}\left(1 - \frac{2}{\pi}\right) && (5.35) \end{aligned}$$

Combining this with Equation (5.31), we get the probability we measure an incorrect string is at most

$$\Pr\big[Y \neq \text{majmod}_p(X) \oplus \text{parity}(X)\big] \leq \left(\frac{1}{p} + \delta\right)\left(\frac{p}{2}\left(1 - \frac{2}{\pi}\right) + \frac{1}{2}\right) \tag{5.36}$$

$$= \frac{1}{2} - \frac{1}{\pi} + \frac{\delta p}{2}\left(1 - \frac{2}{\pi}\right) + \frac{1}{2}\left(\frac{1}{p} + \delta\right) \tag{5.37}$$

$$= \frac{1}{2} - \left(\frac{1}{\pi} - \frac{1}{2p}\right) + O(p\delta) \tag{5.38}$$

All that's left is to upper bound δ, the total variation distance between $|X| \mod p$ and U_p. For this we use the following Fact from [Vio12].

Fact 1 (special case of Fact 3.2 in [Vio12])**.** *Let $(x_1, x_2, \ldots, x_t) \in \{0,1\}^n$ be sampled uniformly. Then the total variation distance between $\sum_{i=1}^t x_i \mod p$ and U_p, the uniform distribution over $\{0, 1, \ldots, p-1\}$ is at most $\sqrt{p}e^{-t/p^2}$*

Using this fact, we get the upper bound $\delta \leq p^{1/2}e^{-n/p^2}$. The probability the measured string is incorrect is then

$$\Pr\big[Y \neq \text{majmod}_p(X) \oplus \text{parity}(X)\big] \leq \frac{1}{2} - \frac{1}{\pi} + \frac{1}{2p} + O(p^{3/2}e^{-n/p^2}). \tag{5.39}$$

□

The final step of this proof is translating the probability of (X, Y) being incorrect into total variation distance between the measurement of the quantum circuit (X, Y) and the ideal distribution $D = (X, \text{majmod}_p(X) \oplus \text{parity}(X))$.

Claim 11. $\Delta\left((X, Y), D\right) = \Pr\big[Y \neq \textit{majmod}_p(X) \oplus \textit{parity}(X)\big]$

Proof. For the sake of clarity, we denote the event $Y = \text{majmod}_p(X) \oplus \text{parity}(X)$ as "Y correct", and its negation as "Y incorrect". Let $V \subseteq \{0,1\}^{n-1} \times \{0,1\}$ be the subset of all correct strings and $V^{\mathsf{c}} := \{0,1\}^{n-1} \times \{0,1\} \setminus V$ the set of all incorrect strings. We calculate the total variation distance as follows

$$\Delta((X,Y), D) = \frac{1}{2} \sum_{\substack{x \in \{0,1\}^{n-1} \\ y \in \{0,1\}}} \Big| \Pr[(D = (x,y))] - \Pr[(X,Y) = (x,y)] \Big| \tag{5.40}$$

$$= \frac{1}{2} \sum_{(x,y) \in V} \left| \frac{1}{2^{n-1}} - \Pr[(X,Y) = (x,y)] \right| + \frac{1}{2} \sum_{(x,y) \in V^{\mathsf{c}}} \Pr[(X,Y) = (x,y)] \tag{5.41}$$

$$= \frac{1}{2} \sum_{x \in \{0,1\}^{n-1}} \Pr[X = x]\left(1 - \Pr[Y \text{ correct} | X = x]\right) \tag{5.42}$$

$$+ \frac{1}{2} \sum_{x \in \{0,1\}^{n-1}} \Pr[Y \text{ incorrect} \wedge X = x] \tag{5.43}$$

$$= \sum_{x \in \{0,1\}^{n-1}} \Pr[Y \text{ incorrect} \wedge X = x] = \Pr[Y \text{ incorrect}] \tag{5.44}$$

□

It follows directly from Claim 10 and Claim 11 that the total variation distance between the measured output of the quantum circuit and $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$ is at most $\frac{1}{2} - \frac{1}{\pi} + \frac{1}{2p} + O(p^{3/2}e^{-n/p^2})$. Therefore, setting n to be sufficiently large, this probability is at most $\frac{1}{2} - \frac{1}{\pi} + \frac{1}{2p} < 0.19 + \frac{1}{4} = 0.44$ since $p \geq 2$.

□

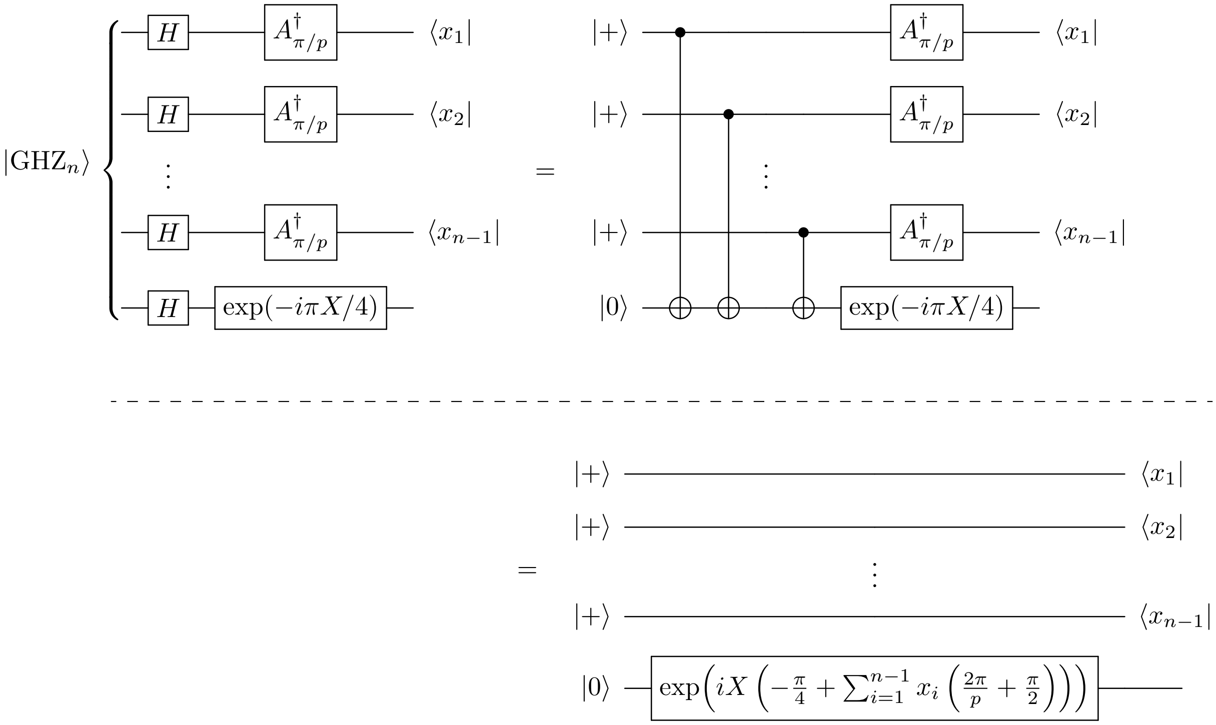


Figure 5.5: Diagrammatic analysis of the circuit presented in the proof of Theorem 12. The first line follows from Equation (5.5), while the second follows from Lemma 6.

5.1.2 Removing non-unitary operations

We now construct a fully quantum circuit that takes a GHZ state as input and produces a state which, when measured in the computational basis, samples approximately from the distribution $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$. Our starting point is the non-unitary circuit constructed in Section 5.1.1. First, we modify this circuit by replacing the non-unitary A_θ gates with a different set of non-unitary gates, and show then sampling statistics are unchanged. Then we show these new non-unitary gates are close to unitary gates, and hence the circuit can be made fully unitary with minimal change to the output statistics.

Introducing multi-qubit non-unitary operations

We start by defining the m-qubit non-unitary operation $A_{\theta,m}$ whose action on the m qubit basis state $|x\rangle = |x_1 x_2 ... x_m\rangle$ is given by:

$$A_{\theta,m} |x_1 x_2 ... x_m\rangle = \exp(i\theta x_m) |x_1\rangle \otimes \exp(i\theta x_1) |x_2\rangle \otimes ... \otimes \exp(i\theta x_{m-1}) |x_m\rangle . \tag{5.45}$$

Intuitively, we can think of the $A_{\theta,m}$ operation as consisting of m distinct A_θ operations, just with the qubits they act on "shifted" away from the qubits controlling the gate by 1 modulo m.

Now we observe that, in certain situations, an $A_{\theta,m}$ operation can replace a tensor product of m different A_θ operations.

Lemma 7. *For any m-qubit computational basis state $|x\rangle = |x_1 x_2 ... x_m\rangle$ and arbitrary one qubit state $|\psi\rangle$, the following equivalence holds:*

$$\begin{aligned} &\langle x|_{1...m} \left(A^\dagger_{\theta,m}\right)_{1...m} \left(\prod_{i=1}^{m} CNOT_{i,m+1}\right) |+\rangle^{\otimes m} \otimes |\psi\rangle \\ &\qquad = \langle x|_{1...m} \left(\prod_{i=1}^{m} \left(A^\dagger_\theta\right)_i CNOT_{i,m+1}\right) |+\rangle^{\otimes m} \otimes |\psi\rangle \end{aligned} \tag{5.46}$$

Proof. The proof is similar to the proof of Lemma 6. In what follows we identify indices mod m so, in particular, we have $x_0 = x_m$. Then we see:

$$\langle x|_{1...m} \left(A^\dagger_{\theta,m}\right)_{1...m} \left(\prod_{j=1}^{m} \mathrm{CNOT}_{j,m+1}\right) |+\rangle^{\otimes m} \otimes |\psi\rangle$$

$$= \langle x|_{1...m} \left(\prod_{j=1}^{m} \exp(i\theta X_j x_{j-1}) \mathrm{CNOT}_{j,m+1}\right) |+\rangle^{\otimes m} \otimes |\psi\rangle$$

$$= \langle x|_{1...m} \left(\prod_{j=1}^{m} \mathrm{CNOT}_{j,m+1} \exp(i\theta X_j X_{m+1} x_{j-1})\right) |+\rangle^{\otimes m} \otimes |\psi\rangle \tag{5.47}$$

$$= \langle x|_{1...m} \left(\prod_{j=1}^{m} \mathrm{CNOT}_{j,m+1}\right) |+\rangle^{\otimes m} \otimes \exp\left(i\theta X \sum_{j=1}^{m} x_{j-1}\right) |\psi\rangle \tag{5.48}$$

$$= \langle x|_{1...m} \left(\prod_{j=1}^{m} \mathrm{CNOT}_{j,m+1}\right) |+\rangle^{\otimes m} \otimes \exp\left(i\theta X \sum_{j=1}^{m} x_j\right) |\psi\rangle \tag{5.49}$$

$$= \langle x|_{1...m} \left(\prod_{j=1}^{m} \exp(i\theta X_j x_j) \mathrm{CNOT}_{j,m+1}\right) |+\rangle^{\otimes m} \otimes |\psi\rangle \tag{5.50}$$

$$= \langle x|_{1...m} \left(\prod_{j=1}^{m} \left(A^\dagger_\theta\right)_j \mathrm{CNOT}_{j,m+1}\right) |+\rangle^{\otimes m} \otimes |\psi\rangle . \tag{5.51}$$

Here the first line follows from definition of $A_{\theta,m}$, the second line follows from commuting an $\exp(i\theta X)$ gate past a CNOT gate as in the proof of Lemma 6, the third line follows because $|+\rangle$ is a 1 eigenstate of the X operator and the fourth line follows from a simple relabeling of indices. The fifth line follows from applying the same argument as in the second and third lines, just in the reverse direction, and the sixth line follows by definition of A_θ. Figure 5.6 gives a diagrammatic version of this proof. □

A straightforward consequence of Lemma 7 and the arguments of Section 5.1.1 is that constant depth quantum circuits augmented with $A_{\theta,m}$ gates and acting on a GHZ state can also approximately sample from the distribution $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$.

Corollary 3. *Let m and D be integers, and $n = Dm + 1$. Then the state*

$$\left(\left(A^\dagger_{\pi/p,m}\right)^{\otimes D} \otimes \exp(-i\pi X/4)\right) H^{\otimes n} |GHZ_n\rangle\,, \tag{5.52}$$

when measured in the computational basis, approximately samples from the distribution $(X, majmod_p(X) \oplus parity(X))$.

Proof. By Lemma 7 and Equation (5.5) we have

$$\begin{aligned}
&\left(\left(A^\dagger_{\pi/p,m}\right)^{\otimes D} \otimes \exp(-i\pi X/4)\right) H^{\otimes n} |GHZ_n\rangle \\
&\qquad = \left(\left(A^\dagger_{\pi/p,m}\right)^{\otimes D} \otimes \exp(-i\pi X/4)\right) \left(\prod_{i=1}^{n-1} \mathrm{CNOT}_{i,n}\right) |+\rangle^{\otimes n-1} \otimes |0\rangle & (5.53)\\
&\qquad = \left(\left(A^\dagger_{\pi/p}\right)^{\otimes n-1} \otimes \exp(-i\pi X/4)\right) \left(\prod_{i=1}^{n-1} \mathrm{CNOT}_{i,n}\right) |+\rangle^{\otimes n-1} \otimes |0\rangle & (5.54)\\
&\qquad = \left(\left(A^\dagger_{\pi/p}\right)^{\otimes n-1} \otimes \exp(-i\pi X/4)\right) H^{\otimes n} |GHZ_n\rangle & (5.55)
\end{aligned}$$

In the proof of Theorem 12 we show this state, when measured in the computational basis, approximately samples from the distribution $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$. □

Replacing multi-qubit non-unitary operations with unitary operations

In this section, we construct a fully unitary circuit which takes a GHZ state as input and produces an output which, when measured in the computation basis, samples approximately from the distribution $(X, \mathrm{majmod}_p(X) + \mathrm{parity}(X))$. We do this by proving that we can replace the non-unitary operations $A_{m,\theta}$ introduced in the previous section with unitary operations while causing minimal change to a circuit using these elements.

To make these statements formal, we first recall some definitions and useful standard facts about matrix norms.

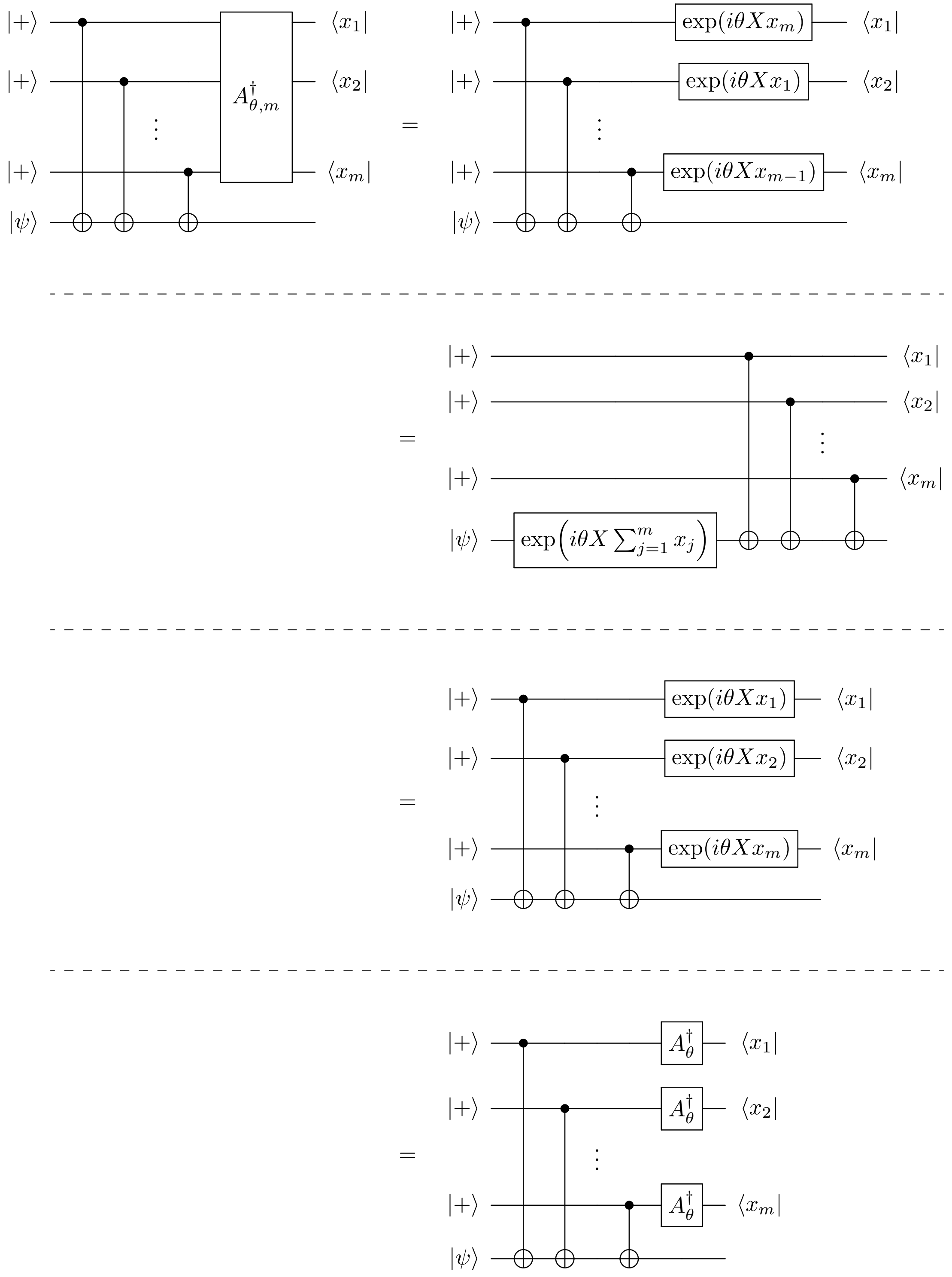


Figure 5.6: Diagrammatic proof of Lemma 7. $|\psi\rangle$ is an arbitrary single qubit state. The equivalence between lines is explained in the proof of the lemma.

Definition 22. *The Frobenius norm of a matrix M, denoted $\|M\|_F$, is defined by*

$$\|M\|_F = \sqrt{\mathrm{tr}[M^*M]} \tag{5.56}$$

Definition 23. *The infinity (or operator) norm of a matrix M, denoted $\|M\|_\infty$, is defined by*

$$\|M\|_\infty = \max_{|\psi\rangle : \||\psi\rangle\|=1} \|M\,|\psi\rangle\|, \tag{5.57}$$

where $\||\psi\rangle\|$ denotes the regular Euclidean norm of any vector $|\psi\rangle$.

The proofs of the following Facts can be found in Appendix A.

Fact 2. *For any matrix M, the Frobenius norm upper bounds the operator norm*

$$\|M\|_\infty \leq \|M\|_F. \tag{5.58}$$

Fact 3. *Given matrices $A_1, A_2, ...A_s$ and $B_1, B_2, ..., B_s$ with*

$$\|A_i - B_i\|_\infty \leq \epsilon, \tag{5.59}$$

$$\|A_i\| \leq 1 \tag{5.60}$$

for all $i \in [s]$, and

$$s\epsilon < 1, \tag{5.61}$$

we also have

$$\left\| \bigotimes_{i\in[s]} A_i - \bigotimes_{i\in[s]} B_i \right\|_\infty \leq 2s\epsilon. \tag{5.62}$$

Fact 4. *Given two states $|\rho\rangle$ and $|\sigma\rangle$, let $p(x)$ and $q(x)$ denote the resulting classical distributions when $|\rho\rangle$ and $|\sigma\rangle$ are measured in some basis $\{|x\rangle\}$. Then we have*

$$\sum_x |p(x) - q(x)| \leq 4\||\rho\rangle - |\sigma\rangle\| \tag{5.63}$$

Next, we recall the definition of the matrix $A_{m,\theta}$ in terms of its action on computational basis states.

$$A_{m,\theta}\,|x_1x_2...x_m\rangle := \exp(i\theta X x_m)\,|x_1\rangle \otimes \exp(i\theta X x_1)\,|x_2\rangle \otimes ... \otimes \exp(i\theta X x_{m-1})\,|x_m\rangle\,. \tag{5.64}$$

The matrix $A_{m,\theta}$ would be a unitary matrix if and only if it mapped computational basis states to some set of orthonormal basis states.[2] The following lemma shows that this condition is close to being satisfied. In what follows, for any bitstring $x = x_1x_2...x_m \in \{0,1\}^m$ we let $\overline{x}$ denote the bitwise complement of x. We also interpret all subscripts in the remainder of this section mod m so, in particular, $x_0 = x_m$.

[2] More generally it is unitary if and only if it maps any set of orthonormal basis states to some other orthonormal basis.

Lemma 8. *For any $\theta \in \mathbb{R}, m \in \mathbb{Z}^+$ and $x = x_1 x_2 ... x_m \in \{0,1\}^m$ the matrix $A_{\theta,m}$ satisfies the following properties:*

1. $\langle x | A^\dagger_{\theta,m} A_{\theta,m} | x \rangle = 1$.
2. $\langle \overline{x} | A^\dagger_{\theta,m} A_{\theta,m} | x \rangle = -i^{m+2|x|} \sin^m(\theta)$.
3. $\langle y | A^\dagger_{\theta,m} A_{\theta,m} | x \rangle = 0$ *for any* $y \in \{0,1\}^m \backslash \{\overline{x}, x\}$.

Proof. The proof of Items 1 and 2 are purely computational. For any $x = x_1 x_2 ... x_m \in \{0,1\}^m$ we have

$$\langle x | \, A^\dagger_{m,\theta} A_{m,\theta} \, | x \rangle = \prod_{j \in [m]} \langle x_j | \exp(-i\theta x_{j-1}) \exp(i\theta x_{j-1}) \, | x_j \rangle \tag{5.65}$$

$$= \prod_{j \in [m]} \langle x_j | x_j \rangle = 1, \tag{5.66}$$

proving Item 1. A similar calculation gives

$$\langle \overline{x} | A^\dagger_{m,\theta} A_{m,\theta} | x \rangle = \prod_{j \in [m]} \langle \overline{x}_j | \exp(-i\theta X \overline{x}_j) \exp(i\theta X x_j) | x_j \rangle \tag{5.67}$$

$$= \prod_{j \in [m]} \langle \overline{x}_j | \exp\big(i^{1+2\overline{x}_j} \theta X\big) | x_j \rangle \tag{5.68}$$

$$= \prod_{j \in [m]} \langle \overline{x}_j | \cos(\theta) + i^{1+2\overline{x}_j} \sin(\theta) X | x_j \rangle \tag{5.69}$$

$$= \prod_{j \in [m]} i^{1+2\overline{x}_j} \sin(\theta) \tag{5.70}$$

$$= i^{m+2|\overline{x}|} \sin^m(\theta) \tag{5.71}$$

$$= -i^{m+2|x|} \sin^m(\theta), \tag{5.72}$$

where we used that $X \, |\overline{x}_j\rangle = |x_j\rangle$ by definition of the compliment on the fourth line and that $|\overline{x}| + |x| = m$ for any x in the final line. This proves Item 2.

To prove Item 3 note that for any m bit strings x and y with $x \notin \{\overline{y}, y\}$ there exists a $k \in [m]$ with $x_{k-1} = y_{k-1}$ and $x_k \neq y_k$. Fixing k to be that value we find:

$$\langle y | A^\dagger_{m,\theta} A_{m,\theta} | x \rangle = \prod_{j=1}^{m} \langle x_j | \exp(-i\theta X y_{j-1}) \exp(i\theta X x_{j-1}) | y_j \rangle \tag{5.73}$$

$$= \langle y_k | \exp(i\theta X (x_k - y_k)) | x_k \rangle \times \prod_{j \in [m] \backslash \{k\}} \langle y_j | \exp(i\theta X (x_{j-1} - y_{j-1})) | x_j \rangle \tag{5.74}$$

$$= \langle y_k | x_k \rangle \times \prod_{j \in [m] \backslash \{k\}} \langle y_j | \exp(i\theta X (x_{j-1} - y_{j-1})) | x_j \rangle \tag{5.75}$$

$$= 0 \tag{5.76}$$

since $y_k \neq x_k$ by definition. This completes the proof of Item 3. □

We show that, as a consequence of Lemma 8, there exists an m qubit unitary matrix which is close (in Frobenius norm) to the non-unitary matrix $A_{\theta,m}$. We construct this unitary by applying Gram-Schmidt orthonormalization applied to the state's output by $A_{m,\theta}$ acting on computational basis states.

Lemma 9. *For any m, there exists unitary matrices $U_{m,\theta}$ satisfying*

$$\left\|A_{m,\theta} - U_{m,\theta}\right\|_F \in O\left(\theta^{-m}\right) \tag{5.77}$$

as $\theta \to 0$.

Proof. We will define $U_{m,\theta}$ by its action on computational basis states. First, fix B^m to be any set containing half the bit strings of length m with the property that for any $x \in \{0,1\}^m$ either $x \in B^m$ or $\overline{x} \in B^m$. (That is, B^m contains one representative element from the equivalence classes of the set $\{0,1\}^m$ induced by the equivalence relation $x \sim y$ if $x = y$ or $\overline{x} = y$). Then define:

$$U_{m,\theta}\left|x\right\rangle := \begin{cases} A_{m,\theta}\left|x\right\rangle & \text{if } x \in B^m \\ C^{-1}\left(A_{m,\theta}\left|x\right\rangle + i^{m+2|x|}\sin^m(\theta) A_{m,\theta}\left|\overline{x}\right\rangle\right) & \text{otherwise.} \end{cases} \tag{5.78}$$

with $C := \sqrt{1 - \sin^{2m}(\theta)}$ a normalizing constant. Observe that, by Item 2 of Lemma 8, for $x \notin B^m$ we can also write

$$U_{m,\theta}\left|x\right\rangle = C^{-1}\left(A_{m,\theta}\left|x\right\rangle - \langle\overline{x}|A^\dagger_{m,\theta}A_{m,\theta}|x\rangle\, A_{m,\theta}\left|\overline{x}\right\rangle\right) \tag{5.79}$$

and

$$C = \left(1 - \left|\,\langle\overline{x}|A^\dagger_{m,\theta}A_{m,\theta}|x\rangle\right|^2\right)^{1/2}. \tag{5.80}$$

We now prove that $U_{m,\theta}$ is unitary. To do this, we prove $U_{m,\theta}$ maps computational basis states to an orthonormal basis. First note that Item 1 of Lemma 8 gives that for any $x \in B^m$:

$$\langle x|U^\dagger_{m,\theta}U_{m,\theta}|x\rangle = \langle x|A^\dagger_{m,\theta}A_{m,\theta}|x\rangle = 1 \tag{5.81}$$

while a similar calculation gives for any $x \notin B^m$:

$$\langle x|U^\dagger_{m,\theta}U_{m,\theta}|x\rangle = C^{-2}\left(\langle x|\,A^\dagger_{m,\theta} - \langle\overline{x}|A^\dagger_{m,\theta}A_{m,\theta}|x\rangle^\dagger\,\langle\overline{x}|\,A^\dagger_{m,\theta}\right)\left(A_{m,\theta}\left|x\right\rangle - \langle\overline{x}|A^\dagger_{m,\theta}A_{m,\theta}|x\rangle\, A_{m,\theta}\left|\overline{x}\right\rangle\right) \tag{5.82}$$

$$= C^{-2}\left(1 - \left|\,\langle\overline{x}|A^\dagger_{m,\theta}A_{m,\theta}|x\rangle\right|^2\right) = 1. \tag{5.83}$$

Where we used Equations (5.79) and (5.80) on the first and second lines, respectively. Then we see the states $\{U_{m,\theta}\left|x\right\rangle\}$ for $x \in \{0,1\}^m$ acting on computational basis states are correctly normalized.

It remains to show that these states are orthogonal. First, we note that Item 3 of Lemma 8 gives that for any $x, y \in \{0,1\}^m$ with $y \notin \{x, \overline{x}\}$ we have

$$\langle y | A^\dagger_{\theta,m} A_{\theta,m} | x \rangle = \langle \overline{y} | A^\dagger_{\theta,m} A_{\theta,m} | x \rangle = \langle y | A^\dagger_{\theta,m} A_{\theta,m} | \overline{x} \rangle = \langle \overline{y} | A^\dagger_{\theta,m} A_{\theta,m} | \overline{x} \rangle = 0 \tag{5.84}$$

and then a quick proof by cases shows that $\langle y | U^\dagger_{\theta,m} U_{\theta,m} | x \rangle = 0$ for any $x \in \{0,1\}^m$ and $y \notin \{x, \overline{x}\}$. Finally, we consider the inner product $\langle \overline{x} | U^\dagger_{\theta,m} U_{\theta,m} | x \rangle$. By definition of B^m, exactly one of x or $\overline{x}$ is in B^m. Assume for the moment that $x \notin B^m$. Then using Equation (5.79) we have

$$\begin{aligned} \langle \overline{x} | A^\dagger_{\theta,m} A_{\theta,m} | x \rangle &= C^{-1} \left(\langle \overline{x} | \, A^\dagger_{m,\theta} \right) \left(A_{m,\theta} \, | x \rangle - \langle \overline{x} | A^\dagger_{m,\theta} A_{m,\theta} | x \rangle \, A_{m,\theta} \, | \overline{x} \rangle \right) && (5.85) \\ &= C^{-1} \left(\langle \overline{x} | A^\dagger_{m,\theta} A_{m,\theta} | x \rangle - \langle \overline{x} | A^\dagger_{m,\theta} A_{m,\theta} | x \rangle \, \langle \overline{x} | A^\dagger_{m,\theta} A_{m,\theta} | \overline{x} \rangle \right) && (5.86) \\ &= C^{-1} \left(\langle \overline{x} | A^\dagger_{m,\theta} A_{m,\theta} | x \rangle - \langle \overline{x} | A^\dagger_{m,\theta} A_{m,\theta} | x \rangle \right) = 0 && (5.87) \end{aligned}$$

as desired. We conclude $U_{m,\theta}$ is unitary.

Finally, to show $U_{m,\theta}$ is close to $A_{m,\theta}$ we compute

$$\begin{aligned} \| A_{m,\theta} - U_{m,\theta} \|_F^2 &= \sum_{x \in \{0,1\}^m} | (A_{m,\theta} - U_{m,\theta}) \, | x \rangle |^2 && (5.88) \\ &= \sum_{x \in B^m} \left| \left(1 - C^{-1}\right) A_{m,\theta} \, | x \rangle - i^{m+2|x|} C^{-1} \sin^m(\theta) A_{m,\theta} \, | \overline{x} \rangle \right|^2 && (5.89) \\ &\leq \sum_{x \in B^m} \left(1 - C^{-1}\right)^2 + C^{-2} \sin^{2m}(\theta) && (5.90) \\ &\leq 2^{m/2} \left(\frac{\sin^{4m}(\theta)}{2} + \frac{\sin^{2m}(\theta)}{1 - \sin^{2m}(\theta)} \right) \in O\left(\theta^{2m}\right) && (5.91) \end{aligned}$$

where the final big O approximation holds for any fixed m as $\theta \to 0$. Taking a square root then completes the proof. □

Finally, we are in a position to describe the fully unitary $(X, \mathrm{majmod}_p(X) + \mathrm{parity}(X))$ sampling circuit.

Theorem 13. *For n sufficiently large and $p = n^c$ for some constant $c \in (0,1]$ there is a constant depth circuit consisting of one and two qubit unitary gates and $U_{m',\theta'}$ gates with $m' = \lceil c^{-1} + 1 \rceil$ and $\theta' = \pi/p$ which takes an n qubit GHZ state as input and produces an output which, when measured in the computational basis, produces an n bit output with total variation distance $\frac{1}{2} - \Omega(1)$ from the distribution $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$.*

Proof. For convenience, we first describe the circuit when $n = Dm' + 1$ for some constant D. This circuit consists of a Hadamard gate applied to each qubit of the GHZ state, followed by $U^\dagger_{m',\theta'}$ gates applied to all qubits except the final qubit and an $\exp(-i\pi X/4)$ rotation applied to the final qubit. Figure 5.7 illustrates this circuit. Note the quantum state produced by this circuit pre-measurement is

$$\left(\left(U^{\dagger}_{\theta',m'}\right)^{\otimes D} \otimes \exp(-i\pi X/4)\right) H^{\otimes n} \left|\psi\right\rangle . \tag{5.92}$$

To prove this circuit samples from the correct distribution first note that Lemma 9 and Fact 2 give that

$$\left\|U_{\pi/p,m} - A_{\pi/p,m}\right\|_{\infty} \in O(\theta'^{m}) = O(n^{-mc}) \leq O(n^{-(1+c)}) \tag{5.93}$$

Them, Fact 3 gives that

$$\left\|\left(\left(U^{\dagger}_{\theta',m'}\right)^{\otimes D} \otimes \exp(-i\pi X/4)\right) H^{\otimes n} - \left(\left(A^{\dagger}_{\pi/p,m}\right)^{\otimes D} \otimes \exp(-i\pi X/4)\right) H^{\otimes n}\right\|_{\infty} \in O(Dn^{-(1+c)}) \tag{5.94}$$

$$\leq O(n^{-c}). \tag{5.95}$$

Combining this observation with Fact 4 and the definition of the operator norm $\|\|_{\infty}$ gives that the classical distributions resulting from computation basis measurements of the states

$$\left(\left(U^{\dagger}_{\theta',m'}\right)^{\otimes D} \otimes \exp(-i\pi X/4)\right) H^{\otimes n} \left|\psi\right\rangle . \tag{5.96}$$

and

$$\left(\left(A^{\dagger}_{\pi/p,m}\right)^{\otimes D} \otimes \exp(-i\pi X/4)\right) H^{\otimes n} \left|\psi\right\rangle \tag{5.97}$$

are $O(n^{-c})$ in total variation distance away from each other. Then Corollary 3 completes the proof. □

5.2 Classical Hardness of sampling $(X, \textbf{majmod}_p(X) \oplus \textbf{parity}(X))$

In this section we prove the classical hardness of sampling from $(X, \text{majmod}_p(X) \oplus \text{parity}(X))$ for each prime number p, where X is sampled from the uniform distribution over $\{0,1\}^n$. Recall that the total variation distance distributions D_1, D_2 over $\{0,1\}^m$ is

$$\Delta(D_1, D_2) := \max_{T \subseteq \{0,1\}^m} \left| \Pr[D_1 \in T] - \Pr[D_2 \in T] \right| \tag{5.98}$$

By the definition of Δ, each set $T \subseteq \{0,1\}^m$, witnesses a lower bound on $\Delta(D_1, D_2)$ of $\left|\Pr[D_1 \in T] - \Pr[D_2 \in T]\right|$. To prove a lower bound on $\Delta(D_1, D_2)$, we construct a particular

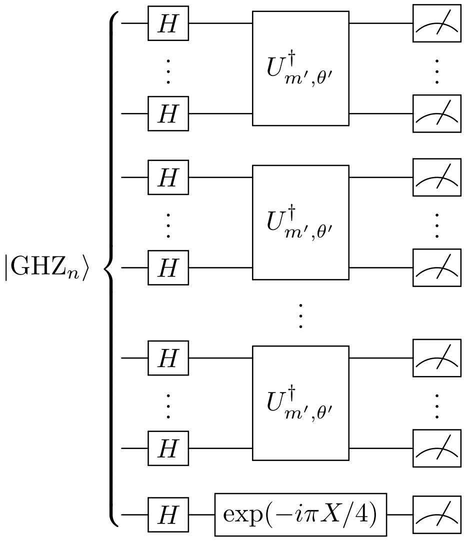


Figure 5.7: Constant depth fully unitary circuit producing approximate samples from the distribution $(\mathrm{majmod}_p(X) + \mathrm{parity}(X), X)$. Here $p = n^c$ for some constant $c \in (0, 1]$, $\theta' = \pi/p$, $m' = \lceil c^{-1} + 1\rceil$ and $n = Dm' + 1$ for some large integer D.

$T \in \{0,1\}^m$ and refer to it as our *statistical test*, and we say that D_i "passes" the statistical test with probability $\Pr[D_i \in T]$.

We are interested in the total variation distance between the true distribution $D = (X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$, and the output distribution of some local function $f : \{0,1\}^\ell \to \{0,1\}^{n+1}$ that takes a uniformly random ℓ-bit string U as input. That is, we aim to lower-bound $\Delta(f(U), D)$.

Theorem 14. *For all $\delta < 1$ there exists an $\epsilon > 0$ such that for all sufficiently large n and prime number $p = \Theta(n^\alpha)$ for $\alpha \in (\delta/3, 1/3)$: Let $f : \{0,1\}^\ell \to \{0,1\}^{n+1}$ be an $\epsilon\log(n)$-local function, with $\ell \leq n + n^\delta$. Then $\Delta(f(U), (X, majmod_p(X) \oplus parity(X))) \geq 1/2 - O(1/\log n)$*

Proof. This proof follows closely to the analogous proof for $(X, \mathrm{majmod}_p(X))$ in [Vio12], with similar notation. Let d be the locality of f, $d = \epsilon\log(n)$. We start by permuting the outputs, as shown in [Vio12]. Note that $\circ$ denotes concatenation.

Lemma 10 ([Vio12])**.** *There exists a partition of the input $u \in \{0,1\}^\ell$ into $u = (x, y)$, and permutation of the output bits such that*

$$f(x, y) = g_1(x_1, y) \circ g_2(x_1, y) \circ \cdots \circ g_s(x_s, y) \circ h(y).$$

With $g_i : \{0,1\} \times \{0,1\}^{\ell-s} \to \{0,1\}^{|B_i|}$, $|B_i| \leq O(d)$ and $s \geq \Omega(n/d^2)$.

We will refer to each $g_i(x_i, y)$ as the ith *block* of the output, for $i \in [s]$. Note that if we fix y, each block is independent, and block $i \in [s]$ only depends on x_i. We say that g_i is *y-fixed* for some $y \in \{0,1\}^{\ell-s}$ if $g_i(0, y) = g_i(1, y)$.

Without loss of generality, and for simplicity of notation, let's assume that the last output bit does not get permuted, so $f(x,y)_{n+1}$ still corresponds to $\mathrm{majmod}_p \oplus$ parity of the first n outputs, and that it only depends on y. Next we define our statistical test.

Statistical Test: Let $N_0 := 3n^{3\alpha}, N_F := 2n^{3\alpha}$, we define our statistical test as $T := T_0 \cup T_F \cup T_S$, with

$$
\begin{aligned}
T_0 &:= \{z \in \{0,1\}^{n+1} : g_i(x_i, y) = 0^{|B_i|}\text{for } \le N_0 \text{ blocks } i \in [s]\} \\
T_F &:= \{z \in \{0,1\}^{n+1} : \exists (x,y) : f(x,y) = z \text{ and } \ge N_F \text{ blocks } g_i(x_i,y) \text{ are } y\text{-fixed}\} \\
T_S &:= \{(z', b) \in \{0,1\}^n \times \{0,1\} : b \neq \mathrm{majmod}_p(z') \oplus \mathrm{parity}(z')\} \qquad (\text{“incorrect strings”})
\end{aligned}
$$

We will show that $f(U)$ passes the statistical test ($f(U) \in T$) with probability at least $1/2 - O(1/\log n)$ and $(X, \mathrm{majmod}_p(X) \oplus \mathrm{parity}(X))$ passes with probability at most $1/n$. To this end, the following Fact and Corollary will be useful.

Fact 5 (Fact 3.2 in [Vio12]). *Let $a_1, a_2, \dots a_t$ be integers modulo p, and let $(x_1, x_2, \dots, x_t) \in \{0,1\}^n$ be sampled uniformly. Then the total variation distance between $\sum_{i=1}^t a_i x_i \mod p$ and U_p, the uniform distribution over $\{0, 1, \dots, p-1\}$ is at most $\sqrt{p}e^{-t/p^2}$*

Corollary 4. *For $t = \Theta(p^3) = \Theta(n^{3\alpha})$, $a_0, a_1, \dots a_t$ integers modulo p, and $A \subseteq \{0, 1, \dots, p-1\}$*

$$
\frac{|A|}{p} - O(1/n) \le \Pr_{x \in \{0,1\}^t}\left[a_0 + \sum_{i=1}^t a_i x_i \in A\right] \le \frac{|A|}{p} + O(1/n)
$$

Proof. It is sufficient to prove that $\Delta(U_p, a_0 + \sum_{i=1}^t a_i x_i) \le O(1/n)$, by the definition of total variation distance.

$$
\Delta(U_p, a_0 + \sum_{i=1}^t a_i x_i) \le \sqrt{p}e^{-t/p^2} = \sqrt{p}e^{-\Theta(p)} = \Theta(n^{\alpha/2})e^{-\Theta(n^{\alpha/2})} \le O(1/n).
$$

□

Claim 12. $\Pr[f(U) \in T] \ge 1/2 - O(1/\log n)$

Proof of Claim 12. We will show that for each y, $\Pr_x[f(x,y) \in T] \ge 1/2 - 1/n$. Suppose we fix y arbitrarily.

If y fixes at least N_F, blocks $g_i(x_i, y)$, then $\Pr_x[f(x,y) \in T_F] = 1$. Moreover, if there are $\le N_0$ blocks g_i such that $g_i(x_i, y) = 0^{|B_i|}$ for some $x_i \in \{0,1\}$, then for each x, there will also be $\le N_0$ blocks with $g_i(x_i, y) = 0^{|B_i|}$, so $\Pr_x[f(x,y) \in T_0] = 1$.

Therefore, we assume that there are $< N_F$ blocks g_i that are y-fixed, and $> N_0$ blocks with $g_i(x_i, y) = 0^{|B_i|}$ for some $x \in \{0,1\}^s$. Thus, there are more than $N_0 - N_F = n^{3\alpha}$ blocks g_i such that for some $x_i \in \{0,1\}$, $g_i(x_i, y) = 0^{|B_i|}$ and $g_i(1 - x_i, y) \neq 0^{|B_i|}$. Let $J \subseteq [s]$ denote this subset of blocks, with $|J| \ge n^{3\alpha}$.

We arbitrarily fix the x_i for $i \in [s] \setminus J$. Now, the total hamming weight of the first n bits of $f(x,y)$ (denoted as $|f(x,y)_{1:n}|$) only depends on the x_i for $i \in J$. We will show over uniformly random choice of $\{x_i\}_{i\in J}$, that $|f(x,y)_{1:n}|$ is close to the uniform distribution of integers mod p.

Let S_i denote the hamming weight of the ith block for each $i \in [s]$. Note that for each $i \in J$, $S_i = 0$ with probability 1/2, and S_i is some positive integer modulo p, with probability 1/2, since $|B_i| \leq O(d) = O(\epsilon \log n) < p$. Moreover, for each $i \in [s] \setminus J$, S_i is fixed. Therefore,

$$|f(x,y)_{1:n}| = a + \sum_{j\in J} |g_i(x_i, y)| = a + \sum_{i\in J} S_i$$

for some positive integer a that does not depend on $\{x_i\}_{i\in J}$.

Since the last bit $b := f(x,y)_{n+1}$ is fixed, the correctness of the output is determined by the majmod_p and parity of $f(x,y)_{1:n}$. We will slightly abuse our notation and also define majmod_p and parity as functions over integers, so

$$\mathrm{majmod}_p(j) = \begin{cases} 0 & \text{if } j \in \{0,1,\ldots,(p-1)/2\} \mod p \\ 1 & \text{if } j \in \{(p+1)/2,\ldots,p-1\} \mod p \end{cases}, \quad \mathrm{parity}(j) = j \mod 2, \quad \text{for } j \in \mathbb{Z}.$$

Thus, we have that $f(x,y) \in T_S \iff \mathrm{majmod}_p(a + \sum_{i\in J} S_i) \oplus \mathrm{parity}(a + \sum_{i\in J} S_i) \neq b$. Note that we can write $a + \sum_{i\in J} S_i = a + \sum_{i\leq |J|} a_i r_i$ for some uniformly random $r \in \{0,1\}^{|J|}$, and for each a_i a fixed positive integer mod p. Therefore,

$$\Pr_{x_J}[f(x,y) \in T_S] = \Pr_{r\in\{0,1\}^{|J|}}[\mathrm{majmod}_p(a + \sum_{i=1}^{|J|} a_i r_i) \oplus \mathrm{parity}(a + \sum_{i=1}^{|J|} a_i r_i) \neq b] \tag{5.99}$$

Let's consider the case that at least 1/2 of the a_i for $i \in J$ are even. Then we fix all x_i such that a_i is odd, and we let $J^e = \{i \in J : a_i \text{ even}\}$. Note that now the parity is fixed to $c := \mathrm{parity}(a + \sum_{i\in J\setminus J^e} a_i x_i)$. Let $a'_i = a_{J^e_i}$ for each $i \in \{1, 2, \ldots, |J^e|\}$.

$$\begin{aligned} \Pr_{x_{J^e}}[f(x,y) \in T_S] &= \Pr_{r\in\{0,1\}^{|J^e|}}[\mathrm{majmod}_p(a' + \sum_{i\leq |J^e|} a'_i r_i) \oplus c \neq b] \\ &= \Pr\left[a' + \sum_{i\leq |J^e|} a'_i r_i \in M_{c\oplus b}\right] \end{aligned}$$

Where $M_0 = \{0,1,\ldots,(p-1)/2\}$ and $M_1 = \{(p+1)/2,\ldots,p-2,p-1\}$. Since $|M_0| = (p+1)/2$, $|M_1| = (p-1)/2$, and using Corollary 4 with $t = |J^e| = \Theta(n^{3\alpha})$, it follows that

$$\Pr_{x_{J^e}}[f(x,y) \in T_S] \geq (p-1)/2p - O(1/n) = 1/2 - O(1/\log(n))$$

All that's left is to consider the case where more than half of the a_i for $i \in J$ are odd. In this case we will fix x_i for each $i \in J$ with a_i even, setting $a' := a + \sum_{i\in J^e} a_i x_i$. We denote the set of such "odd" elements of J as $J^o = \{i \in J : a_i \text{ odd}\}$, and we let $t = |J^o| = \Theta(n^{3\alpha})$ and set $a'_i = a_{J^o_i}$ for each $i \in [t]$. Note that since each a'_i is odd, we have $\text{parity}(a' + \sum_{i \le t} a'_i r_i) = a' \oplus (\text{parity}(r, \dots, r_t))$.

$$\begin{aligned}
\Pr_{x_{J^o}}[f(x,y) \in T_S] = \Pr_{r \in \{0,1\}^t}&\left[\text{majmod}_p\left(a' + \sum_{i \le t} a'_i r_i\right) \oplus \text{parity}(r) \neq b \oplus a'\right] \\
= &\frac{1}{2}\Pr_r\left[\text{majmod}_p\left(a' + \sum_{i \le t} a'_i r_i\right) \neq b \oplus a' \middle| \text{parity}(r) = 0\right] \\
&+ \frac{1}{2}\Pr_r\left[\text{majmod}_p\left(a' + \sum_{i \le t} a'_i r_i\right) = b \oplus a' \middle| \text{parity}(r) = 1\right]
\end{aligned}$$

Sampling a uniformly random t bit string $z_1 z_2 \dots z_t$ with even hamming weight is equivalent to sampling a random $t-1$ bit and setting the last bit to $z_t = \text{parity}(z_1, \dots, z_{t-1})$. So the equation above is equal to

$$\begin{aligned}
= &\frac{1}{2}\Pr_{r_1, \dots r_{t-1}}\left[\text{majmod}_p\left(a' + \sum_{i \le t-1} a'_i r_i + a'_t \cdot \text{parity}(r_1, \dots, r_{t-1})\right) \neq b \oplus a'\right] \\
&+ \frac{1}{2}\Pr_{r_1, \dots r_{t-1}}\left[\text{majmod}_p\left(a' + \sum_{i \le t-1} a'_i r_i + a'_t \cdot \text{parity}(1, r_1, \dots, r_{t-1})\right) = b \oplus a'\right].
\end{aligned}$$

For any positive integers z_1, z_2, s, t such that $s < t$ and $t - s - z_2 \ge 0$, if Z_2 is a positive random variable such that $Z_2 \le z_2$, then $\Pr[z_1 + Z_2 \in [s,t]] \ge \Pr[z_1 \in [s, t - z_2]]$. Therefore, with all addition done modulo p, we lower bound the above expression as

$$\begin{aligned}
\ge &\frac{1}{2}\Pr\left[a' + \sum_{i \le t-1} a'_i r_i \in [0, p/2 - a_t)\right] + \frac{1}{2}\Pr\left[a' + \sum_{i \le t-1} a'_i r_i \in (p/2, p - 1 - a_t]\right] \\
\ge &\frac{1}{2p}((p+1)/2 - a_t + (p-1)/2 - a_t) - O(1/n) \\
= &\frac{1}{2} - \frac{a_t}{2p} - O(1/n) \\
= &\frac{1}{2} - \frac{O(\log n)}{2n^\alpha} - O(1/n) \ge \frac{1}{2} - O(1/\log n).
\end{aligned}$$

Where we used Corollary 4, and that $a_t \le |B_t| \le O(\epsilon \log n)$ and $p = \Theta(n^\alpha)$.

In conclusion, we've showed that after arbitrarily fixing y, $\Pr_x[f(x,y) \in T] \ge \frac{1}{2} - O(1/\log n)$. Therefore, $\Pr_{x,y}[f(x,y) \in T] \ge \frac{1}{2} - O(1/\log n)$, as desired.

□

Claim 13. $\Pr\big[(X, \textit{majmod}_p(X) \oplus \textit{parity}(X)) \in T\big] \le O(1/n)$

Proof. This proof follows that of Claim 3.3 in [Vio12]. Let $D := (X, \text{majmod}_p(X) \oplus \text{parity}(X))$. By the union bound $\Pr[D \in T] \le \Pr[D \in T_0] + \Pr[D \in T_F] + \Pr[D \in T_S]$. Clearly

$\Pr[D \in T_S] = 0$, since T_S is the set of invalid strings. Therefore, it is sufficient for us to show that $\Pr[D \in T_F], \Pr[D \in T_0] \leq \frac{1}{2n}$.

$\Pr[D \in T_F] = |T_F|/2^n$, so it is sufficient to upper bound $|T_F|$. Recall that $z \in T_F$ if $z = f(x, y)$ for some x, y such that at least N_F blocks are y-fixed. Thus, each $z \in T_F$ is characterized by y, and the bits of x that do not belong to fixed blocks. That is, we need at most $\ell - N_F$ bits to characterize z. Since $\ell \leq n = n^\delta$ and $N_F = 2n^{3\alpha}$,

$$\begin{aligned} |T_F| &\leq 2^{n+n^\delta-2n^{3\alpha}} \\ &\leq 2^{n-n^{3\alpha}} \end{aligned}$$

since $\delta < 3\alpha$. So

$$\Pr[D \in T_F] \leq 2^{-n^{3\alpha}} \leq \frac{1}{2n}.$$

All that's left is to bound $\Pr[D \in T_0]$, the probability that at most $N_0 = 3n^{3\alpha}$ blocks i are all zero, $D_{B_i} = 0^{|B_i|}$. Since the first n bits of D are independently random, the probability that the block D_{B_i} is all zero is independent of other blocks D_{B_j} for $i \neq j \in [s]$. The probability that block $i \in [s]$ is all zero is

$$\Pr\left[D_{B_i} = 0^{|B_i|}\right] = (1/2)^{|B_i|} \geq (1/2)^{O(d)} = (1/2)^{O(\epsilon \log n)} = \left(\frac{1}{n}\right)^{O(\epsilon)}.$$

Now, the probability that at most $N_0 = 3n^{3\alpha}$ are fixed is

$$\begin{aligned} \Pr[D \in T_0] &= \Pr\left[\bigcup_{\substack{T \subseteq [s]: \\ |T| = N_0}} \{D_{B_i} \neq 0^{|B_i|} \text{ for each } i \in [s] \setminus T\}\right] \\ &\leq \binom{s}{N_0}\left(1 - \frac{1}{n^{O(\epsilon)}}\right)^{s-N_0} \\ &\leq \binom{s}{N_0} e^{-\frac{s-N_0}{n^{O(\epsilon)}}} \end{aligned}$$

Since $s \geq \Omega(n/d^2) = \Omega(\frac{n}{\epsilon^2 \log^2 n})$, $s \leq n$ and $N_0 = 3n^{3\alpha}$,

$$\begin{aligned} &\leq \binom{n}{3n^{3\alpha}} e^{-n^{-O(\epsilon)}(\frac{n}{\epsilon^2 \log^2 n} - 3n^{3\alpha})} \\ &\leq \left(\frac{n}{3n^{3\alpha}}\right)^{3n^{3\alpha}} e^{-n^{1-O(\epsilon)}/\log^2 n} e^{3n^{3\alpha}} \\ &\leq n^{3n^{3\alpha}} e^{-n^{1-O(\epsilon)}/\log^2 n} \\ &\leq \frac{1}{2n} \end{aligned}$$

For sufficiently large n and small ϵ. In conclusion, $\Pr[D \in T] \leq \frac{1}{n}$, as desired. □

□

Bibliography

[Aar04] S Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. submitted. *arXiv preprint quant-ph/0412187*, 3, 2004.

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach.* Cambridge University Press, 2009.

[BFLL22] Adam Bouland, Bill Fefferman, Zeph Landau, and Yunchao Liu. Noise and the frontier of quantum supremacy. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1308–1317. IEEE, 2022.

[BGK18] Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018.

[BGKT20] Sergey Bravyi, David Gosset, Robert Koenig, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, 2020.

[BMS16] Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Physical review letters*, 117(8):080501, 2016.

[CSV21] Matthew Coudron, Jalex Stark, and Thomas Vidick. Trading locality for time: certifiable randomness from low-depth circuits. *Communications in mathematical physics*, 382(1):49–86, 2021.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[FGHZ05] Stephen Fenner, Frederic Green, Steven Homer, and Yong Zhang. Bounds on the power of constant-depth quantum circuits. In *International Symposium on Fundamentals of Computation Theory*, pages 44–55. Springer, 2005.

[FH16] Edward Farhi and Aram W Harrow. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv preprint arXiv:1602.07674*, 2016.

[Gal18] François Le Gall. Average-case quantum advantage with shallow circuits. *arXiv preprint arXiv:1810.12792*, 2018.

[Has19] Matthew B. Hastings. Classical and quantum bounded depth approximation algorithms. *quantum Information and Computation*, 19(13 & 14):1116–1140, Nov 2019.

[HHT97] Yenjo Han, Lane A Hemaspaandra, and Thomas Thierauf. Threshold computation and cryptographic security. *SIAM Journal on Computing*, 26(1):59–78, 1997.

[JSB10] R Jozsa, D Shepherd, and M Bremner. Classical simulation of commuting qauntum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, rspa20100301*, 2010.

[MT20] Tomoyuki Morimae and Suguru Tamaki. Additive-error fine-grained quantum supremacy. *Quantum*, 4:329, 2020.

[TD02] Barbara M Terhal and David P DiVincenzo. Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games. *arXiv preprint quant-ph/0205133*, 2002.

[Tod89] Seinosuke Toda. On the computational power of pp and (+) p. In *30th Annual Symposium on Foundations of Computer Science*, pages 514–519. IEEE Computer Society, 1989.

[Vio12] Emanuele Viola. The complexity of distributions. *SIAM Journal on Computing*, 41(1):191–218, 2012.

[Viz65] Vadim G Vizing. Critical graphs with given chromatic class (in russian). *Metody Discret. Analiz.*, 5:9–17, 1965.

[WKST19] Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 515–526, 2019.

APPENDICES

Appendix A

Ommitted Proofs

Lemma 11. *If the function $f : \{0,1\}^m \to \{0,1\}^n$ is ℓ-local, it can be implemented by a classical circuit with fan-in 2, depth $O(\ell)$ and size $(n\ell 2^\ell)$.*

Proof. Consider the function $f_i : \{0,1\}^\ell \to \{0,1\}$ for each output bit $i \in [n]$. It is sufficient to prove that each of these can be implemented with fan-in 2, depth $O(\ell)$ and size $O(\ell 2^\ell)$.

For each $i \in [n]$, let $S_i \subseteq \{0,1\}^\ell$ be the set of all 1-strings for f_i.

$$S_i := \{z \in \{0,1\}^\ell : f_i(z) = 1\} \tag{A.1}$$

Now evaluating $f_i(z)$ for some $z \in \{0,1\}^\ell$ is equivalent to determining if z is in S_i. Consider some $i \in [n]$. For each $s \in S_i$, let $c_s : \{0,1\}^\ell \to \{0,1\}$ be a function which checks if $s = z$.

$$c_s(z) := \begin{cases} 1 & \text{if } s = z \\ 0 & \text{if } s \neq z \end{cases} \qquad \text{for each } s \in S_i, z \in \{0,1\}^\ell \tag{A.2}$$

We can implement $c_s(z)$ by checking in parallel that each bit matches, then taking an AND.

$$c_s(z) = \bigwedge_{i \in [\ell]} (z_i = s_i) \tag{A.3}$$

This AND of ℓ variables can be implemented with fan-in 2, depth $\log \ell$, and size $O(\ell)$. Now, to implement $f_i(z)$, we check if any of the $c_s(z)$ are 1

$$f_i(z) = \bigvee_{s \in S_i} c_s(z) \tag{A.4}$$

Since $|S_i| \leq 2^\ell$, we can implement all of the c_s in parallel with fan-in 2, depth $\log \ell$ and total size $O(\ell 2^\ell)$. Then, the OR of each of these $O(2^\ell)$ bits can be implemented with fan-in 2 and depth ℓ, and size $O(\ell)$. Therefore, f_i can be implemented with fan-in 2, depth $O(\ell)$, and size $O(\ell 2^\ell)$ for each $i \in [n]$. □

Fact 2. *For any matrix M, the Frobenius norm upper bounds the operator norm*

$$\|M\|_\infty \leq \|M\|_F. \tag{58}$$

Proof. For an arbitrary matrix M, let $\lambda_1, ..., \lambda_d$ denote the eigenvalues of M^*M, with $\lambda_1 \geq \lambda_2 \geq ...\lambda_d$. Note all λ_i are positive. Then we have

$$\|M\|_\infty^2 = \lambda_1 \leq \sum_{i=1}^{d} \lambda_i = \|M\|_F^2 \tag{5}$$

as desired. □

Fact 3. *Given matrices $A_1, A_2, ...A_s$ and $B_1, B_2, ..., B_s$ with*

$$\begin{aligned} \|A_i - B_i\|_\infty &\leq \epsilon, \qquad (59)\\ \|A_i\| &\leq 1 \qquad (60) \end{aligned}$$

for all $i \in [s]$, and

$$s\epsilon < 1, \tag{61}$$

we also have

$$\left\| \bigotimes_{i\in[s]} A_i - \bigotimes_{i\in[s]} B_i \right\|_\infty \leq 2s\epsilon. \tag{62}$$

Proof. First note that $\|M\|_\infty$ is equal to the largest singular value of the matrix M, from which it follows that

$$\|M \otimes N\|_\infty = \|M\|_\infty \|N\|_\infty \tag{6}$$

for any matrices M and N. Then an inductive argument gives

$$\begin{aligned} \left\| \bigotimes_{i=1}^{s} A_i - \bigotimes_{i=1}^{s} A_i \right\|_\infty &= \left\| \bigotimes_{i=1^s} A_i - B_1 \bigotimes_{i=2}^{s} A_i + B_1 \bigotimes_{i=2}^{s} A_i - \bigotimes_{i=1}^{s} B_i \right\|_\infty \qquad (7)\\ &\leq \left\| (A_1 - B_1) \bigotimes_{i=2}^{s} A_i \right\| + \left\| B_1 \otimes \left(\bigotimes_{i=2}^{s} A_i - \bigotimes_{i=2}^{s} B_i \right) \right\| \qquad (8)\\ &\leq \epsilon + (1+\epsilon) \left\| \bigotimes_{i=2}^{s} A_i - \bigotimes_{i=2}^{s} B_i \right\| \qquad (9)\\ &= \epsilon + (1+\epsilon)(2\epsilon(s-1)) \leq 2s\epsilon \qquad (10) \end{aligned}$$

as desired. □

Fact 4. *Given two states $|\rho\rangle$ and $|\sigma\rangle$, let $p(x)$ and $q(x)$ denote the resulting classical distributions when $|\rho\rangle$ and $|\sigma\rangle$ are measured in some basis $\{|x\rangle\}$. Then we have*

$$\sum_x |p(x) - q(x)| \leq 4\| |\rho\rangle - |\sigma\rangle \| \tag{63}$$

Proof. First, we note that for any two states $|\rho\rangle$ and $|\sigma\rangle$ and PSD matrix $M \leq I$ we have

$$\begin{aligned} 2\| |\rho\rangle - |\sigma\rangle \| &\geq 2\| M(|\rho\rangle - |\sigma\rangle)\| && (11) \\ &\geq 2\left(\|M |\rho\rangle\| - \|M |\sigma\rangle\|\right) && (12) \\ &\geq \left(\|M |\rho\rangle\| - \|M |\sigma\rangle\|\right)\left(\|M |\rho\rangle\| + \|M |\sigma\rangle\|\right) && (13) \\ &= \|M |\rho\rangle\|^2 - \|M |\sigma\rangle\|^2 && (14) \end{aligned}$$

Then defining probability distributions $p(x)$ and $q(x)$ and the basis $\{|x\rangle\}$ as above, let

$$P_x := \{x : p(x) \geq q(x)\} \tag{15}$$

and

$$M_x = \sum_{x \in P_x} |x\rangle\langle x| \,. \tag{16}$$

Then note

$$\begin{aligned} \|M_x |\rho\rangle\|^2 - \|M_x |\sigma\rangle\|^2 &= \sum_{x \in P_x} |\langle x|\rho\rangle|^2 - |\langle x|\sigma\rangle|^2 && (17) \\ &= \sum_{x \in P_x} (p(x) - q(x)) && (18) \\ &= \frac{1}{2} \sum_x |p(x) - q(x)| && (19) \end{aligned}$$

with the final inequality holding because both $p(x)$ and $q(x)$ must sum to one. Combining the two inequalities above proves the result. □