Interactive quantum advantage with noisy, shallow Clifford circuits

Daniel Grier[*][†] Nathan Ju[*][‡] Luke Schaeffer[*][§]

Abstract

Recent work by Bravyi et al. constructs a relation problem that a noisy constant-depth quantum circuit (QNC^0) can solve with near certainty (probability $1 - o(1)$), but that any bounded fan-in constant-depth classical circuit (NC^0) fails with some constant probability. We show that this robustness to noise can be achieved in the other low-depth quantum/classical circuit separations in this area. In particular, we show a general strategy for adding noise tolerance to the interactive protocols of Grier and Schaeffer. As a consequence, we obtain an unconditional separation between noisy QNC^0 circuits and $\mathsf{AC}^0[p]$ circuits for all primes $p \geq 2$, and a conditional separation between noisy QNC^0 circuits and log-space classical machines under a plausible complexity-theoretic conjecture.

A key component of this reduction is showing average-case hardness for the classical simulation tasks—that is, showing that a classical simulation of the quantum interactive task is still powerful even if it is allowed to err on some constant fraction of inputs. We show that is possible even for quantum tasks which are $\oplus\mathsf{L}$-hard to simulate. To do this, we borrow techniques from randomized encodings used in cryptography.

1 Introduction

A major goal in quantum complexity theory is identifying problems which are efficiently solvable by quantum computers and not efficiently solvable by classical computers. If willing to believe certain conjectures, one can be convinced of this separation by the discovery of quantum algorithms that solve classically hard problems. For example, the belief that classical computers cannot efficiently factor integers contrasts with Shor's algorithm for factoring integers on a quantum computer [Sho97]. However, a demonstration of Shor's algorithm on instances that are not efficiently solvable by classical computers would require quantum resources far out of reach of near-term capabilities. This has spurred developments

[*] Institute for Quantum Computing, University of Waterloo, Canada

[†] Cheriton School of Computer Science, University of Waterloo, Canada

[‡] Department of Computer Science, University of Illinois, Urbana-Champaign, IL

[§] Department of Combinatorics and Optimization, University of Waterloo, Canada

in devising sampling problems that separate efficient, near-term quantum computers and classical computers like IQP circuit sampling [BJS10], BosonSampling [AA11], and random circuit sampling [Boi+18]. However, convincing evidence that *noisy* quantum computers outperform classical computers in these tasks suffers from the necessity of assuming some non-standard complexity-theoretic conjectures that are often native to each proposal.

Surprisingly, if you restrict to the setting of constant-depth circuits, a noisy, unconditional separation *is* possible. At first, these unconditional separations were known only in the noiseless setting. That line of work was initiated by pioneering work of Bravyi, Gosset, and König [BGK18] who showed a strict separation between constant-depth quantum circuits (QNC^0) and constant-depth classical circuits with bounded fan-in gates (NC^0). The separation is based on the relation problem[1] associated with measuring the outputs of a shallow Clifford circuit, which they called the Hidden Linear Function (HLF) problem due to certain algebraic properties of the output. Furthermore, they show that NC^0 cannot even solve this problem on average, a result which was later strengthened in several ways [CSV18; Le 19; Ben+19]. Nevertheless, these works still assumed that the quantum circuit solving the task was noise free.

Follow-up work of Bravyi, Gosset, König, and Tomamichel [Bra+20] showed that it was possible to encode the qubits of the quantum circuit in such a way that it preserved the separation while affected by noise. Interestingly, this was accomplished not by explicitly carrying out the quantum error correction procedure, but by simply measuring the syndrome qubits of the code and requiring the classical circuit to do the same. In fact, their procedure provided a more-or-less general recipe for taking a constant-depth quantum/classical circuit separation and turning it into a separation in which the quantum circuit was also allowed noise.

This raises an obvious question: how many circuit separations can we upgrade in this way? NC^0 circuits are fairly weak—they cannot even compute the logical AND of all input bits—and so we would like to show that even larger classes of classical devices cannot solve a problem that a noisy shallow quantum circuit can.

As a warm-up, we first consider the separation of Bene Watts, Kothari, Schaeffer, and Tal [Ben+19], which shows that constant-depth classical circuits with *unbounded* fan-in gates (AC^0) cannot solve the HLF problem on average.[2] Combining this result with the general error-correction recipe for relation problems, we arrive at the following result:

Theorem 1.1. *There is a relation task solved by a noisy*[3] QNC^0 *circuit with probability* $1 - o(1)$ *on all inputs. On the other hand, any* AC^0 *circuit can solve the problem on at most a* $\exp(-n^\alpha)$ *fraction of inputs for some constant* $\alpha > 0$.

The result of Bene Watts et al. is the strongest known low-depth separation of its kind, but stronger separations are known for tasks which admit some amount of interactivity. Consider the shallow Clifford measurement problem discussed above, where the measurements are

[1] Generally speaking, a relation problem is defined by a relation $R \subseteq \Sigma^* \times \Sigma^*$. Given an input x, the task is to find some y such that $(x, y) \in R$.

[2] The authors of that paper refer to their task as the "Relaxed Parity Halving Problem," but it is still essentially the problem of measuring the outputs of a constant-depth Clifford circuit.

[3] We employ the same local stochastic noise model used in [Bra+20]. We refer the reader to Section 2.2 for those details.

made in two rounds. In the first round, the quantum device is given the bases in which to measure some of the qubits and returns their measurement outcomes; and in the second round, the quantum device is given bases in which to measure the remaining qubits and returns their measurement outcomes. Grier and Schaeffer [GS20] show that any classical device which can solve such problems must be relatively powerful. More specifically, if the initial Clifford state is a constant-width grid state, then the classical device can be used to solve problems in NC^1, and if the starting state is a poly-width grid state, then the classical device can be used to solve problems in $\oplus\mathsf{L}$.[4] Because $\mathsf{AC}^0[p] \subsetneq \mathsf{NC}^1$ unconditionally, the above interactive task can be solved by a QNC^0 circuit but not an $\mathsf{AC}^0[p]$ circuit, i.e., an AC^0 circuit with unbounded MOD_p gates[5] for some prime p.

One of the contributions of this work is massaging the noisy circuit separation recipe for relations problems into a recipe for *interactive* problems as well. Starting from an interactive protocol which exhibits a separation with a noise-free quantum circuit, there are three key steps to upgrade the separation to the noisy setting:

1. *Augment the interactive protocol with the surface code encoding of Bravyi et al. [Bra+20].* This is straightforward, but it's worth noting that it changes the problem definition—not just because there are more physical qubits due to the encoding, but because we cannot prepare the initial state exactly or decode the syndrome in constant depth. As for the relational case, the burden of these steps is offloaded into the problem definition.
2. *Show classical average-case hardness.* That is, show that even when the classical circuit simulating the interactive protocol is allowed to err on some constant fraction of its inputs, it can still be leveraged to solve a hard problem (e.g., a problem in NC^1 or $\oplus\mathsf{L}$). This step is the most involved, and new ideas will be required to upgrade existing interactive separations in this way.
3. *Connect to separations of classical complexity classes.* In some cases, this will lead to an unconditional separation between noisy shallow quantum circuits and shallow classical circuits, and in some cases this will lead to a conditional separation. We note that these separations will not be identical to those obtained in Ref [GS20] due to the fact that we use quasipolynomial-size circuits to decode the syndrome qubits of the surface code.

Fortunately, it was shown in Ref [GS20] that Step 2 holds[6] for the NC^1-hardness result. We immediately obtain the following separation:

Theorem 1.2. *There is a two-round interactive task solved by a noisy QNC^0 circuit with probability $1 - o(1)$ on all inputs. Any $\mathsf{AC}^0[p]$ circuit (for primes $p \geq 2$) fails the task with some constant probability.*

Unfortunately, Step 2 is left as an open question in Ref [GS20] for the $\oplus\mathsf{L}$-hardness result. The second major contribution of this paper is to show that we can, in fact, obtain average-case hardness for this setting:

[4] See Section 2.1 for definitions of all the relevant complexity classes needed for the paper.

[5] The MOD_p gates outputs 1 iff the sum of the inputs bits is 0 mod p.

[6] Although it is not strictly required, we prove a slightly stronger average-case hardness in Lemma 4.6.

Theorem 1.3. *There is a two-round interactive task solved by a* QNC^0 *circuit with certainty. There exists a constant* $\delta > 0$ *such that any sufficiently powerful classical device which solves the task with probability at least* $1 - \delta$ *can also solve problems in* $\oplus\mathsf{L}$. *That is,*

$$\oplus\mathsf{L} \subseteq (\mathsf{BPAC}^0)^{\mathcal{R}}$$

where $\mathcal{R}$ *is an oracle for the classical solution.*

The proof of this theorem borrows an idea from cryptography called *randomized encodings*. In particular, we will employ the construction of Applebaum, Ishai, and Kushilevitz [AIK06] which randomizes instances of the following problem—given a layered DAG, determine the parity of the number of paths from vertex s to vertex t. In fact, we will use that this problem reduces to the $\oplus\mathsf{L}$-hardness result in [GS20]. Importantly, we show that when we compose the randomized encoding with the rest of the reduction, the distribution over inputs in the promise will be fairly uniform. This leads to a general way to boost the randomization in worst-to-average-case reductions using the framework in [GS20].

Using the recipe for interactive circuit separations, we obtain the following consequence:

Theorem 1.4. *There is a two-round interactive task solved by a noisy* QNC^0 *circuit with probability* $1 - o(1)$ *on all inputs. Assuming* $\oplus\mathsf{L} \notin (\mathsf{qBPAC}^0)^{\mathsf{L}}$, *any log-space machine fails the task with some constant probability.*

Let us briefly unpack the $\oplus\mathsf{L} \notin (\mathsf{qBPAC}^0)^{\mathsf{L}}$ assumption. First, consider the plausible assumption that $\oplus\mathsf{L} \notin \mathsf{L}$. An L machine is deterministic, while a $\oplus\mathsf{L}$ machine is nondeterministic and accepts if the parity of accepting paths is zero. On the other hand, it is well-known that the parity function is not in qBPAC^0 (i.e., random AC^0 circuits of quasipolynomial size). Therefore, one might also expect that $(\mathsf{qBPAC}^0)^{\mathsf{L}}$ is insufficiently powerful to compute $\oplus\mathsf{L}$ functions.

We do not attempt to give an exhaustive list of separations obtainable from Theorem 1.3. Much like the results of [GS20], there is an inherent tradeoff to the separation. We can weaken the assumption at the expense of weakening the separation.

Finally, we explore the regime between the NC^1-hardness result (i.e., an interactive task on constant-width grids) and the $\oplus\mathsf{L}$-hardness result (i.e., an interactive task on poly-width grids). To this end, we consider the interactive task on general width-w grids, and connect them to the problem of solving width-w permutation branching programs. We prove the analogue of Theorem 1.3 in this setting, which once again leads to conditional separations between noisy shallow quantum circuits and complexity classes solved by width-w permutation branching programs.

1.1 Open Problems

Our work still leaves several unresolved questions. We show average-case hardness results for classical devices solving quantum interactive tasks—if the classical machine only errs on some small fraction of inputs, it can be leveraged to solve hard problems. We ask what happens in the error regime below this threshold. Can the success probabilities be exponentially reduced by parallel repetition of the same problems? Direct product theorems are often useful in proving these types of error amplifications, but it is unclear how they apply in this setting. Parallel repetition could also *improve* the success probability of the noisy quantum circuit if we only require that some fraction of the instances are solved correctly. This could boost the error probability from inverse quasipolynomial to inverse exponential.

One could also approach this problem from the other direction by showing that there are circuits that can tolerate even higher amounts of error. More generally, we ask what is the optimal amount of allowable error for each problem. We show error thresholds of 29/30, 105/106, and 420/421 for classical devices solving NC^1 problems, width-w permutation branching programs, and $\oplus\mathsf{L}$ problems, respectively. Surely, these bounds are not tight. How far can they be improved?

Finally, we leave open questions in the noiseless setting. We know that we cannot hope for anything greater than $\oplus\mathsf{L}$-hardness using our current paradigm of Clifford circuits. Is there a way of leaving this paradigm without sacrificing the provable guarantees? We also note that without interaction, much less is known. Is it possible to show that some relation problem is solved by a QNC^0 circuit that cannot be solved in a classical complexity class larger than AC^0? For instance, we suspect that $\mathsf{AC}^0[3]$ circuits are unable to solve the shallow Clifford measurement tasks considered in the introduction.

2 Preliminaries

This section discusses much of the background needed for this paper. Readers familiar with previous results in this area, particularly [Bra+20] and [GS20], can comfortably skip this section, except for Section 2.4 where we formally define the main task we will consider for the rest of the paper.

Section 2.1 briefly touches on the low-depth circuit classes relevant to this paper. Section 2.2 explains the local stochastic noise model [Bra+20], i.e., the type of error we allow in the quantum circuit. Section 2.3 describes the surface code (also following [Bra+20]) and its behavior under local stochastic noise. Section 2.4 defines a shallow Clifford circuit measurement task that unifies both the relation and interactive problem statements of [BGK18] and [GS20]. Nevertheless, to make this distinction clear, we discuss the differences in relation and interactive tasks in Section 2.5. Finally, in Section 2.6 we discuss the connection between our problem and measurement-based quantum computation.

2.1 Low-depth circuit classes

Circuits for solving certain problems are generally defined as families of circuits, one for each input size, where the corresponding circuits are

- NC^i: $\log^i$-depth bounded fan-in AND/OR/NOT circuits.
- AC^i: NC^i circuits with unbounded fan-in gates.
- $\mathsf{AC}^\mathsf{i}[p]$: AC^i circuits with MOD_p gates.
- TC^i: AC^i circuits with majority gates.
- $\mathsf{BP}\mathcal{C}$: $\mathcal{C}$ circuits that have access to random bits and two-sided bounded error.

- $\mathsf{q}\mathcal{C}$: $\mathcal{C}$ circuits of $\exp\big(\log^{O(1)} n\big)$ size.

In addition, we have the following inclusions that are proven strict ($\subsetneq$) and believed to be strict ($\subset$): $\mathsf{NC}^0 \subsetneq \mathsf{AC}^0 \subsetneq \mathsf{AC}^0[p] \subsetneq \mathsf{TC}^0 \subset \mathsf{NC}^1 \subset \mathsf{L} \subset \oplus\mathsf{L}$. We note that the inclusion $\mathsf{AC}^0[p] \subseteq \mathsf{TC}^0$ is only known to be strict when p is prime.

2.2 Local Stochastic Noise Model

While a noise-free quantum computation can reliably execute a sequence of operations, a noisy quantum computation may have sources of errors that corrupt several key parts of the computation including state initialization, gate execution, and measurement. To capture these sources of error, we consider the *local stochastic quantum noise* model [FGL18; Bra+20]. Under this model, random errors occur at each timestep of the execution of a quantum circuit. For example, a gate error occurs when random noise enters the computation prior to the execution of the gate. Similarly, an erroneous measurement outcome is modeled by random noise affecting the state of the system right before measurement.

The types of random noise that we consider are random Pauli errors on each qubit. For a Pauli error $E \in \{I, X, Y, Z\}^{\otimes n}$, we borrow the convention of $\mathrm{Supp}(E) \subseteq [n]$ to denote the subset of indexed qubits for which E acts by a X, Y, or Z. In other words, $\mathrm{Supp}(E)$ is the subset of qubits on which E acts non-trivially. Local stochastic noise is parameterized by the noise rate p:

Definition 2.1. *Let $p \in [0,1]$. A random n-qubit Pauli error E is "p-local stochastic" if*

$$\Pr[F \subseteq \mathrm{Supp}(E)] \leq p^{|F|} \quad \forall\, F \subseteq [n] \tag{1}$$

Notice that this allows distant qubits to have correlated errors, but the probability that k qubits are corrupted simultaneously decreases exponentially in k. When we say that a layer of local stochastic noise E is sampled with noise rate p, we use the notation $E \sim \mathcal{N}(p)$. There is a property of local stochastic noise that will be useful for our analysis.

Fact 1. *Suppose $E \sim \mathcal{N}(p)$ and E' is another Pauli error such that Supp$(E') \subseteq$ Supp(E) with certainty. Then $E' \sim \mathcal{N}(p)$.*

2.3 The 2-D surface code

The 2D surface code is a CSS-type error correcting code that encodes one logical qubit into m physical qubits on a 2D lattice. For a detailed discussion of the construction we employ throughout this paper, we refer the reader to Section IV of [Bra+20]. We will abstract away the physical surface code and henceforth denote the encoded version of a state with a line over the state vector, e.g., the logical $|0\rangle$ state is encoded as the $|\overline{0}\rangle$ state. We follow the same convention when speaking of encoded circuits and measurement observables. If $\mathcal{Y}$ is the physical measurement outcome over multiple codeblocks, we denote $\mathcal{Y} = \mathcal{Y}^1 ... \mathcal{Y}^n$ with each $\mathcal{Y}^i$ the m outcomes of the i'th codeblock. The space of binary physical measurement outcomes on one codeblock forms a linear subspace called the codespace, and we refer to it by $\mathcal{L}$.

A standard quantum computation begins with qubits prepared in a basis state, e.g. multiple copies of $|0\rangle$. However, the surface code must begin with an *encoded* basis state, $|\overline{0}\rangle$, so we require a constant-depth procedure to produce such a state. We can do this, albeit at the cost of extra ancilla qubits and a Pauli recovery operator dependent on the measurement outcomes of the ancillae. We also opt for the encoded Bell state, $|\overline{\Phi}\rangle$, as the starting state.

Lemma 2.1. *(Basis state preparation, Theorem 23 in [Bra+20]) There is a constant-depth Clifford circuit on $2m + m_{anc}$ qubits that measures m_{anc} qubits with measurement outcome $s \in \{0,1\}^{m_{anc}}$ and leaves the remaining $2m$ qubits in the state $\mathsf{Rec}(s)\left|\overline{\Phi}\right\rangle$ for some Pauli operator Rec completely determined by s.*

Following basis-state preparation, we would like to perform a constant-depth Clifford circuit on the surface code.

Lemma 2.2. *(Constant-depth Cliffords, Lemma 20 in [Bra+20]) The encoded $\overline{H}$, $\overline{S}$, $\overline{CNOT}$ gates have constant depth implementations on the surface code.*

In particular, an unencoded Clifford circuit can be transformed to an encoded Clifford circuit on the surface code with only constant overhead. If we add local stochastic noise to a Clifford circuit, we can propagate the errors to the end of the circuit.

Lemma 2.3. *(Propagating noise, Theorems 17 and 23 in [Bra+20]) Suppose we have a quantum circuit with noise rate p that creates multiple $\mathsf{Rec}(s)\left|\overline{\Phi}\right\rangle$ states using Lemma 2.1. Then suppose that it performs a depth-D Clifford circuit on these states. The state of the system is equivalent to a noiseless computation with only one layer of local stochastic noise, E, following the circuit such that $E \sim \mathcal{N}(O(p^{2^{-O(D)}}))$.*

Suppose that we have finished performing an encoded Clifford circuit on a surface code as Lemma 2.3 describes. We could express the final layer of local stochastic noise as $X(v)Z(w)$ where $X(v)$ is a Pauli X only on qubits with their corresponding bit in v set to 1, and similarly for $Z(w)$. If we measured one surface code, we would have a length-m bitstring $\mathcal{Y}$ that encodes the measurement outcome of a logical qubit. The Dec function decodes the m outcomes to a single bit, and it is able to tolerate noise.

Lemma 2.4. *(Lemma 21 in [Bra+20]) Suppose that there is only one layer of local stochastic noise $X(v) \sim \mathcal{N}(r)$ with $r \leq 0.01$ which occurs right before the measurement of any codeblock. Then*

$$\Pr_{X(v)\sim\mathcal{N}(r)}[\mathsf{Dec}(x \oplus v) = \mathsf{Dec}(x)] \geq 1 - \exp\left(-\Omega(m^{1/2})\right) \tag{2}$$

for any $x \in \mathcal{L}$.

This lemma says that when the layer of local stochastic noise is below a constant threshold of 0.01, then Dec will successfully decode the surface code measurement outcome for any x in the codespace.

2.4 Generic Graph State Measurement Problem

All of the problems we consider fit into the following framework.

Problem 1 (k-Round Graph State Measurement problem)**.** *Let $k \geq 1$ be an integer. Let $\{G_n = (V_n, E_n)\}_{n\geq 1}$ be a uniform family of graphs, where G_n has $|V_n| = O(\mathsf{poly}(n))$ vertices. Furthermore, for each graph G_n, suppose the vertices are colored with k colors, i.e., there exists $\chi_n\colon V_n \to [k]$ for all $n \geq 1$. Each choice of k, $\{G_n\}$, and $\{\chi_n\}$ defines a problem within this framework.*

The problem is to prepare the graph state $|G_n\rangle$, where

$$|G_n\rangle \coloneqq \prod_{(i,j)\in E_n} \mathrm{CZ}(i,j)\, |+\rangle^n ,$$

and then measure the vertices in k rounds of interaction. In the ith round, measurement bases (either X or Y) are provided for all vertices of color i (i.e., $\chi^{-1}(i)$), and the device is expected to output corresponding measurement outcomes (either $+1$ or -1) for each such vertex. At the end of k rounds, the device succeeds if measuring $|G_n\rangle$ in the input measurement bases could generate (with non-zero probability) the output measurement outcomes.

Within this paper, we will consider only non-interactive relation problems ($k = 1$, following [BGK18; Bra+20], etc.), and two-round interactive problems ($k = 2$, following [GS20]). We will also avoid precisely defining the family of graphs since they can be found in the literature, and because the proofs only depend on the following properties:

- The family of graphs is efficiently constructible and uniform (i.e., all graphs are constructed by the same machine). In other words, we will assume any basic processing of the graph (e.g., enumerating the vertices, determining adjacency, etc.) is not a bottleneck in our complexity reductions.
- The maximum degree of the vertices is $O(1)$ to allow constant depth (QNC^0) construction of $|G_n\rangle$.
- The state $|G_n\rangle$ may be used as a resource for MBQC (measurement-based quantum computation) for some family of quantum circuits that *is* precisely defined.

Theorem 2.1. *Any k-Round Graph State Measurement problem on graphs $\{G_n\}_{n\geq 1}$ with $O(1)$ maximum degree is solved by a family of $O(1)$-depth, classically-controlled Clifford circuits. Furthermore, the circuits are uniform (i.e., generated by a fixed Turing machine, say, given the input n) if the family of graphs is uniform.*

The theorem above follows almost by definition of the problem itself. Figure 1 depicts this quantum circuit.

2.5 Relation and Interactive problems

As discussed above, we will only consider problems with one or two rounds of interactivity. A one-round protocol corresponds to what would ordinarily be called a *relation problem*, i.e., given an input $x \in \{0,1\}^n$, produce an output $y \in \{0,1\}^m$ (where m is polynomially related

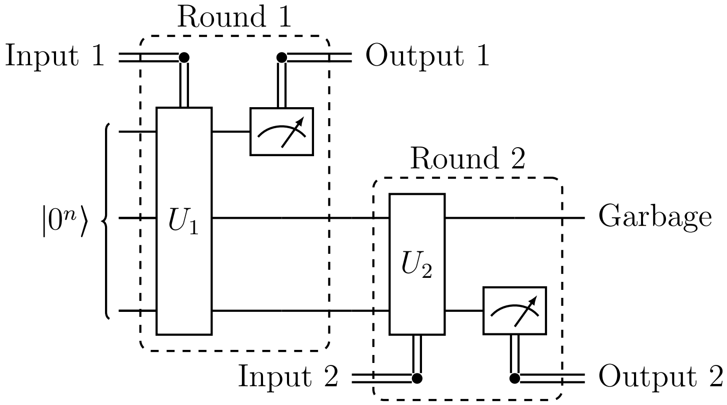


Figure 1: General form of a quantum circuit for a 2-round interactive protocol.

to n) such that (x, y) belongs to some relation $R \subseteq \{0,1\}^n \times \{0,1\}^m$. In this paper, for all $x \in \{0,1\}^n$ in the promise there exists some $y \in \{0,1\}^m$ satisfying the relation, but we note that it may not be unique (for us, often *half* the strings satisfy the relation). We use the notation $R(x, y) = 1$ to indicate success (i.e., (x, y) satisfies the relation problem), otherwise $R(x, y) = 0$.

The two-round tasks are back-and-forth interactions between a question-asker (challenger) and a computational device. The challenger provides the first input x_1, and the device replies with its first output y_1. The challenger provides a second input, x_2, possibly dependent on y_1, and the device is expected to reply with some output y_2. The sequence of back-and-forth messages (x_1, y_1, x_2, y_2) is known as the *transcript*. The problem itself is defined by a set of transcripts T, and the device succeeds if the final transcript belongs to T.

For the purposes of complexity theoretic reductions, we treat the device as an *oracle with state*, in a very natural way. We feed the first round input, it returns the first round output, then the next input it treats as the second round input (i.e., continuing the previous interaction, making use of its internal state), and answers with the second round output. After the last round of input/output, feeding another input to the oracle will cause it to restart the whole protocol again from the beginning.

2.5.1 Rewind Oracle

Ordinarily, oracles are treated as black boxes in reductions. However, the results of Grier and Schaeffer [GS20] depend on a pivotal difference between a *classical* implementation of an interactive oracle and a *quantum* implementation. Crucially, when a classical device solves an interactive problem, the internal state of the device may be duplicated at any time during the protocol, and later restored to that point. That is, in addition to restarting the interactive protocol afresh, we have the option to *rewind* it to any previous point in the computation, and perhaps continue with different inputs. It turns out that for many classical complexity classes, if the oracle belongs to that class, then so does the oracle with rewinding.

Fact 2. *Suppose $\mathcal{O}$ is an oracle for an interactive problem, and let $\mathcal{R}$ be the* rewind oracle *obtained by giving rewind capability to $\mathcal{O}$. If $\mathcal{O}$ is implemented by a device in some classical* *complexity class $\mathcal{C}$, then $\mathcal{R}$ is also in $\mathcal{C}$ as long as $\mathcal{C}$ is one of* $\mathsf{AC}^0, \mathsf{AC}^0[p], \mathsf{TC}^0, \mathsf{NC}^1, \mathsf{L}$.

In the context of 2-round graph state measurement, the oracle's answers represent measurement outcomes for a round of qubits. With a rewind oracle, we can measure the second round qubits and then rewind back to the end of the first round. We are then free to choose a different basis (i.e., change the second round input) and measure again, as many times as we choose. This is a powerful ability which an actual quantum device does not have since, in general, it is impossible to make two non-commuting measurements on a state, even if the measurements are known in advance. We formally state all of our hardness reductions in terms of the rewind oracle, meaning that the problem is only (NC^1, $\oplus\mathsf{L}$, etc.)-hard for classical devices, since a quantum solution does not imply a rewind oracle.

Last, although rewind oracles appear to give the power to measure the second round qubits in arbitrary bases, the outcomes are not truly *samples* from the distribution of measurement outcomes. Because of our definition of interactive protocols (and the k-round graph state measurement problem), we can only claim that oracle returns a *possible* outcome, not necessarily with the same probability, or perhaps not even random at all!

In fact, we must assume the oracle produces adversarial outputs, designed to thwart our hardness reductions at every turn. For example, suppose we wish to distinguish between two non-orthogonal states. For any measurement, there is some outcome consistent with both states, and in the worst case the oracle will always give us that outcome. [GS20] combats this in two ways: First, they use demonstrations of non-contextuality (specifically, the magic square and magic pentagram games) to force the oracle to reveal some information about the state. Second, they use self-randomization subroutines to conceal the original query within a uniformly random input query, so that the oracle cannot tell which pair of states we want to distinguish, and thus cannot reliably choose an outcome common to both. See [GS20] for a full description of this process; we re-use many of their ideas, but make changes in our proofs of Theorem 4.2 and Theorem 4.5 to increase the input query randomization. We discuss the motivation behind this change in the corresponding sections.

2.6 Measurement-based Quantum Computation

Many of the results in [GS20] are based on the hardness of computing the final state (or even distinguishing between two possible final states) in some depth-$\Omega(n)$ Clifford circuit. However, we need the simulation to be performed by a constant-depth quantum circuit, so we use the following theorem from measurement-based quantum computation (MBQC) [RB01; RBB03]:

Theorem 2.2. *Fix the* layout *of a circuit of one- and two-qubit gates, i.e., a circuit diagram where all the gates are placeholders to be replaced with a concrete one- or two-qubit unitary later. Let us map each placeholder gate to a* gadget *(details omitted, see [RB01; RBB03] or [GS20]), in the form of a constant-size graph, and connect these gadgets together (as they are connected in the circuit) to form a graph G.*

For any concrete unitary U, there exists a set of measurement bases for the qubits of the corresponding gadget, such that measuring in those bases performs PU for some Pauli operation P depending on the measurement outcomes. That is, if we plug in unitaries to get a circuit $U_k \cdots U_1$ and measure the gadget qubits in the appropriate bases, then the remaining *qubits are in state $P_k U_k \cdots P_1 U_1 |+^n\rangle$ for Pauli operations $P_1, \ldots, P_k$. Moreover, if the unitary operations are Clifford then X- and Y-basis measurements suffice.*

In other words, we can simulate any Clifford circuit with a particular layout by measuring a constant-degree graph, and thus with a constant-depth circuit. The only catch is that there are Pauli errors throughout the circuit. Fortunately, the Clifford group normalizes the Pauli group by definition, so each Pauli operation can be "pushed" through the circuit at the expense of some computation. Unfortunately, pushing all the error to the end of the circuit is effectively as hard as simulating the circuit outright, and therefore not practical in our hardness reductions.

Finally, we note that MBQC works as expected with interactive protocols. In our interactive protocols, for instance, we let nearly all of the gates be in the first round, and only a handful, comprising a constant-depth circuit at the end, into the second round. Then for all the gates in a round, we measure the qubits of their gadgets appropriately, computing the state $|\psi\rangle := P_2 U_2 P_1 U_1 |+\rangle^n$ where U_1, U_2 are the Clifford unitaries represented by the two parts of the circuits, and P_1, P_2 are Pauli operations. Since U_1 is a deep Clifford circuit, we are unable to compute P_1 in our reductions, but we *can* compute P_2 since it is easy to push Pauli operations through a constant-depth circuit such as U_2. In the second round we also measure $|\psi\rangle$ in, say, the X-basis (this is part of the second round), which means we can tell which outcomes P_2 flipped, and translate the actual measurement outcomes to outcomes for the state $U_2 P_1 U_1 |+\rangle^n$.

Thus, we can think about our interactive protocols like this: the first round input specifies Clifford gates for the placeholders in some layout. The first round output determines P_1, but not in an easily decodable way, not unlike a cryptographic commitment. The second round input specifies more gates, but since they have constant depth, they are used only to change the measurement basis. Finally, the second round output tells us, with some constant-depth classical processing, the result of measuring $P_1 U_1 |+\rangle^n$ in the measurement basis specified in the second round input. With a rewind oracle, this gives us the power to measure $P_1 U_1 |+\rangle^n$ in different bases with the *same* unknown P_1 each time.

3 The noisy extension and AC^0 separation

In this section, we review the noisy relaxation of the 1-Round Graph State Measurement Problem of [Bra+20] called the *noisy extension*. We will revisit the main results of that paper, and show how their separation between noisy QNC^0 and NC^0 can be extended to a separation between noisy QNC^0 and AC^0 using the results of Bene Watts et al. [Ben+19]. The results of this section are not independent from our results for interactive Clifford simulation tasks since the noisy extension will play a critical role there, as well.

3.1 The noisy extension

Suppose that we have a relation problem defined by $R \subseteq \{0,1\}^* \times \{0,1\}^*$ that is solved with certainty by a classically-controlled Clifford circuit C_x that begins with multiple copies of $|\Phi\rangle$ and measures all qubits as output. To make this relation noise-tolerant, we convert R to its *noisy-extended* version. The noisy-extended version is defined using a 2D surface code with desirable properties. Its effect on a relation problem is the following: For some input x, the set of y such that $(x,y) \in R$ is enlarged to the set of $\mathcal{Y}$ that decode to y.

However, recall that the procedure for basis state preparation on the surface code incurs an additional Pauli operator $\mathsf{Rec}(s)$. Fortunately, the effect of this Pauli operator on the overall quantum computation can be propagated through the circuit. Consider the classically-controlled Clifford circuit $\overline{C_x}$. Since the Clifford group normalizes the Pauli group, we can define $f(s,x)$ and $h(s,x)$ by

$$X(f)Z(h) \sim \overline{C_x}\mathsf{Rec}(s)\overline{C_x}^{\dagger} \tag{3}$$

where $f = f^1 \dots f^n$ and each f^i is m bits describing the Pauli X operator on the i'th codeblock of the circuit. We are now ready to define the noisy-extended relation.

Definition 3.1. *The* noisy-extended relation R' *associated with relation* R *is defined as*

$$R'(x,(\mathcal{Y},s)) = \begin{cases} 1 & \textit{if } R(x,y) = 1 \textit{ for } y_i = \mathsf{Dec}(\mathcal{Y}^i \oplus f^i(s,x)) \quad \forall\, i \in [n] \\ 0 & \textit{otherwise} \end{cases} \tag{4}$$

It should be the case that if a *noiseless* quantum circuit can solve a certain relation problem with certainty, then another noiseless quantum circuit can solve the related noisy-extended relation problem with certainty.

Suppose that a quantum circuit has prepared the encoded basis state by measuring syndrome outputs $s^1 \dots s^n$, resulting in the state $(\mathsf{Rec}(s^1) \otimes \dots \otimes \mathsf{Rec}(s^n)) \left|\overline{\Phi}^n\right\rangle = \mathsf{Rec}(s) \left|\overline{\Phi}^n\right\rangle$ through the procedure of Lemma 2.1. Then it performs a classically-controlled Clifford circuit $\overline{C_x}$ in constant-depth using Lemma 2.2 and measures the output $\mathcal{Y}$ with the property

$$|\langle \mathcal{Y}| \overline{C_x} \mathsf{Rec}(s) \left|\overline{\Phi}^n\right\rangle|^2 > 0 \tag{5}$$

By propagating the Pauli $\mathsf{Rec}(s)$ over Clifford circuits using Equation (3), we have that $|\langle \mathcal{Y}| X(f)Z(h)\overline{C_x} \left|\overline{\Phi}^n\right\rangle|^2 > 0$. Since Pauli Z operators have no effect on Z-basis measurement, we get

$$|\langle \mathcal{Y} \oplus f(s,x)| \overline{C_x} \left|\overline{\Phi}^n\right\rangle|^2 > 0 \tag{6}$$

Because $\overline{C_x}$ is the encoded version of C_x, then the bit string y defined by

$$y_i = \mathsf{Dec}(\mathcal{Y}^i \oplus f^i(s,x)) \; \forall i \in [2n] \tag{7}$$

satisfies the original relation R.

Of course, the whole point of defining the noisy extension of a relation problem is to show that the outputs of noisy quantum circuits can still satisfy it:

Theorem 3.1. *(Noise-tolerance of noisy extension, Theorem 17 in [Bra+20]) Suppose a constant-depth Clifford circuit solves a relation problem. Then another constant depth-D Clifford circuit solves the noisy extended relation problem with probability at least* $1 - \exp(-\Omega(m^{1/2}))$ *when the noise rate is below* $\exp(-\exp(O(D)))$.

3.1.1 AC^0 decoding circuit

The noisy extension allows a noisy constant-depth quantum circuit to solve the modified relation problem with high probability. However, in order to demonstrate a separation between the quantum and classical circuit's capabilities, we need to argue that the noisy extended relation problem remains hard for classical circuits. Our goal is to show that any qAC^0-capable device can actually decode measurements on the physical qubits into measurements on the logical qubits. Therefore, any such device which can solve the noisy extended relation problem can also solve the original relation problem. This allows us to port previous noiseless separations to the noisy setting.

Recall that the encoded quantum circuit will output $\mathcal{Y}$ and s such that the bitstring y defined by Equation (7) is the output of an unencoded circuit. We show that decoding can be carried out by an $\exp(m, m_{anc})$-size AC^0 circuit.[7] The construction is shown in Figure 2.

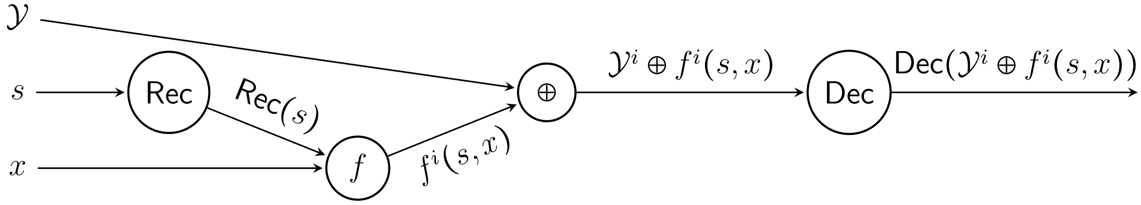

$\mathcal{Y}$
s
x
Rec
Rec(s)
f
$f^i(s,x)$
$\oplus$
$\mathcal{Y}^i \oplus f^i(s,x)$
Dec
$\mathsf{Dec}(\mathcal{Y}^i \oplus f^i(s,x))$

Figure 2: Decoding gadget

Lemma 3.1. *Given $\mathcal{Y}$, s, and x, $\mathsf{Dec}(\mathcal{Y}^i \oplus f^i(s,x))$ can be computed by an AC^0 circuit of size $\exp(m_{anc}, m)$ for each i.*

Proof. Note that Rec and Dec can be computed by $\exp(m, m_{anc})$ size circuits by using a truth-table circuit for these functions. We would like to conjugate the Pauli $\mathsf{Rec}(s) = \mathsf{Rec}(s^1) \otimes ... \otimes \mathsf{Rec}(s^n)$ with the classically-controlled Clifford circuit, $\overline{C_x}$, to obtain $f^i(s,x)$ in Equation (3). Consider any depth-one Clifford circuit, C, on mn qubits composed of one- and two-qubit gates. Conjugating the Pauli operator $\mathsf{Rec}(s)$ by C is locally computable by a polynomial size AC^0 circuit because C only uses one- and two-qubit Clifford gates. Repeating this process for each layer of the constant-depth circuit $\overline{C_x}$, XOR'ing each bit with $\mathcal{Y}^i$, and plugging the output into Dec gives the lemma. □

In summary, we have computed $\mathsf{Dec}(\mathcal{Y}^i \oplus f^i(s,x))$ for each i with a $\exp(m, m_{anc})$ size AC^0 circuit. Because $m, m_{anc} = \mathsf{polylog}(n)$ in all of our constructions, we have shown that the decoding can be done in qAC^0. This quasipolynomial blowup will not affect the conclusion of this section since neither poly-size nor quasipoly-size AC^0 circuits can solve parity. However, this blowup will be important to our conclusions about interactive problems (see, for instance, Theorem 1.4 in the Introduction).

[7] In fact, this size is nearly optimal for AC^0 circuits because computing the parity of n bits reduces (using only fan-out) to Dec on $O(n^2)$ bits.

3.2 Noise-tolerant AC^0 separation

We begin by reviewing the relevant problem that is solvable by *noiseless* QNC^0 circuits but is hard for AC^0 circuits to solve. By applying the noisy extension introduced in the previous section, we can prove that a separation persists as the quantum circuit is subject to noise.

It was shown in [Ben+19] that there is a problem that is solved with certainty by a QNC^0 circuit but is hard for AC^0 circuits to solve. The problem, a promise version of 1-Round Graph State Measurement Problem that they call the Relaxed Parity Having Problem (RPHP), is a relation problem with inputs uniformly chosen from a set $P_n \subseteq \{0,1\}^n$ for all n that is solved with certainty by a constant-depth Clifford circuit. We will actually use *Parallel* RPHP (under the name RPHP, for brevity), a version with polynomially many copies of vanilla RPHP, because the probability of a classical circuit succeeding is then exponentially small.

The same problem is hard for AC^0 circuits with an error rate exponentially close to 1.

Lemma 3.2 (Theorem 26 in [Ben+19])**.** *Any AC^0 circuit of size s and depth d cannot solve RPHP with probability exceeding*

$$\exp\left(\frac{-n^{1/2-o(1)}}{O(\log s)^{2d}}\right) \tag{8}$$

over a uniformly random input $x \in P_n$.

Define *Noisy* RPHP to be the noisy extended relation problem associated with RPHP (see Definition 3.1) with $m, m_{anc} = \Theta(\mathsf{polylog}(n))$. By construction, a noisy quantum circuit can solve Noisy RPHP with high probability. Plugging this relation problem into Theorem 3.1, we get:

Proposition 1. *Let D be a constant and the local stochastic noise rate p be bounded by $p < p_{th} = \exp(-\exp(O(D)))$. For all $x \in P_n$, a depth D Clifford circuit with noise rate p can solve Noisy RPHP with probability exceeding*

$$1 - \exp(-\Omega(\mathsf{polylog}(n))) \tag{9}$$

We conclude with the separation between AC^0 and noisy QNC^0 using Noisy RPHP:

Theorem 3.2. *Any AC^0 circuit of size s and depth d cannot solve Noisy RPHP with probability exceeding*

$$\exp\left(\frac{-n^{1/2-o(1)}}{O(\log(s + \exp(\mathsf{polylog}(n))))^{2d+O(1)}}\right) \tag{10}$$

over the uniformly random $x \in P_n$.

Proof. We decode the output of Noisy RPHP to one of RPHP using Lemma 3.1. This incurs an extra size overhead of the AC^0 circuit by $\exp(m, m_{anc}) = \exp(\mathsf{polylog}(n))$. By applying this size expansion to Lemma 3.2, we arrive at the desired bound. □

4 Interactive hardness

To start, let us extend the 2-Round Graph State Measurement Problem (Problem 1) to its noisy variant by encoding the qubits in the surface code in the natural way:

Problem 2 (Noisy k-Round Graph State Measurement)**.** *Consider the k-Round Graph State Measurement Problem where each logical qubit is encoded in $\Theta(\mathsf{polylog}(n))$ physical qubits according to the surface code (see Section 2.3). That is, we start with some encoded graph state*

$$\left|\overline{G_n(s)}\right\rangle := \prod_{(i,j)\in E_n} \overline{\mathrm{CZ}}(i,j)\mathsf{Rec}(s)\left|\overline{+}\right\rangle^n ,$$

where s denotes the syndrome qubits associated with preparing the n logical $|\overline{+}\rangle$ states. The choice of s is left to the device. In the ith round of interaction, logical measurement bases (either X or Y) are provided for all vertices of color i, and the device is expected to output the corresponding measurement outcomes for every physical qubit representing each such vertex. At the end of the k rounds, the device also outputs s.

The device succeeds if measuring $\left|\overline{G_n(s)}\right\rangle$ in the input measurement bases could generate (with non-zero probability) the output measurement outcomes. This acceptance criterion is equivalent to requiring that the measurements of the encoded graph state satisfies the noisy-extended relation (see Definition 3.1).

Showing that there exists a noisy, constant-depth circuit which solves the 2-round problem for constant-degree graphs is relatively straightforward given the properties of the code.

Theorem 4.1. *There exists a noisy, constant depth-D Clifford circuit can pass the Noisy 2-Round Graph State Clifford Measurement Problem on constant-degree graphs with probability*

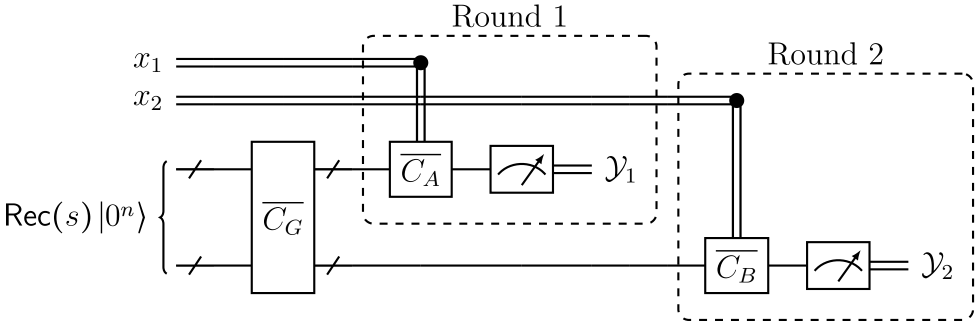
$$\geq 1 - \exp(-\Omega(\mathsf{polylog}(n)))$$

when the noise rate p is bounded by

$$p < p_{th} = \exp(-\exp(O(D)))$$

Proof. We will construct a classically-controlled Clifford circuit $\mathcal{C}$ to satisfy the protocol. First, the circuit prepares $\mathsf{Rec}(s)\,|\overline{0^n}\rangle$, where $\mathsf{Rec}(s) := \mathsf{Rec}(s^1)\otimes\ldots\otimes\mathsf{Rec}(s^n)$ using Lemma 2.1.[8] Next, the circuit constructs the graph state using a constant-depth circuit $\overline{\mathcal{C}_G}$ using Lemma 2.2. Finally, in each round of the protocol, $\mathcal{C}$ performs a basis change Clifford operation ($\overline{\mathcal{C}_A}$ and $\overline{\mathcal{C}_B}$, respectively) and measures in the Z basis. This can be seen in the following figure: Let us analyze the effect of noise in this process. By propagating noise via Lemma 2.3 and combining noise via Fact 1, the noisy computations are equivalent to single layers of noise before first and second round measurements with noise rate $p^{\exp(-O(D))}$ where D is the depth of $\mathcal{C}$. The measurement success rate given by Lemma 2.4 implies that if the single layer of local stochastic noise before measurement has noise rate below 0.01, then a codeblock outputs a correct measurement outcome with probability exceeding $1 - \exp(-\Omega(m^{1/2}))$. We only require that $p < \exp(-\exp(O(D)))$ to achieve the 0.01 noise rate bound before measurements. By choosing $m \geq \Omega(\mathsf{polylog}(n))$ and a union bound, we can conclude that all n codeblocks are correct with probability at least $1 - \exp(-\Omega(\mathsf{polylog}(n)))$. □

[8] We note that the Bell state $|\Phi\rangle$ and zero state $|0\rangle$ are equal up to a constant-depth Clifford, so these protocols are essentially identical.



4.1 ⊕L-hardness

The main theorem of this section is an average-case ⊕L-hardness result for the 2-Round Graph State Measurement Problem on a $\mathsf{poly}(n) \times \mathsf{poly}(n)$ grid state. We will do this by showing that the worst-case ⊕L-hard problem of [GS20] has a random self-reducibility property, and thus average case hardness.

First, let us review this ⊕L-hard problem, which concerns the parity of paths in a directed acyclic graph. Define M to be the set of upper-triangular adjacency matrices of monotone graphs—i.e., a directed graph on vertices $V = \{1, \ldots, n\}$ with no edges from j to i for $j \geq i$.

Lemma 4.1. *([Dam90]) Let* MGap *be the problem of deciding the parity of paths from* 1 *to* n *in* $A \in M$*. Then*

$$\oplus\mathsf{L} \subseteq (\mathsf{NC}^0)^{\mathsf{MGap}}$$

For a detailed description of how an instance of MGap is reduced to an instance of the 2-Round Graph State Measurement Problem, we refer the reader to [GS20]. Our goal in this section will be to show that randomizing an instance of the MGap problem suffices for an average-case hardness result for the graph state measurement problem. It will be useful to describe this randomized algorithm as a composition of two algorithms, $F \circ D$, which takes $A \in M$ as input and outputs a sequence of operations[9] which serve as input to the graph state measurement problem through measurement-based computation gadgets. We will describe F and D in greater detail in the next section.

To obtain our average-case hardness result, we constrain the first round inputs of the graph state measurement problem to be from the set $F \circ D(M)$. We constrain the second round inputs to be from a small set of Pauli measurements I_2, defined formally in Appendix B.

Theorem 4.2. *Let $\mathcal{R}$ be the rewind oracle for 2-Round Graph State Measurement Problem on a $\mathsf{poly}(n) \times \mathsf{poly}(n)$ grid promised that input is from the image of $F \circ D(M)$ in the first round and I_2 in the second round. If $\mathcal{R}$ fails on $\epsilon < \frac{1}{421}$ fraction of inputs, then*

$$\oplus\mathsf{L} \subseteq (\mathsf{BPAC}^0)^{\mathcal{R}}$$

Appealing to Lemma 3.1, we obtain the following corollary for the Noisy 2-Round Graph State Measurement Problem:

[9] Formally, each one of these operations consists of a CNOT gate followed by a circuit of CZ and Phase gates.

Corollary 4.1. *If the rewind oracle $\mathcal{R}'$ fails for $\epsilon < \frac{420}{421}$ fraction of inputs to the Noisy 2-Round Graph State Measurement Problem (with the same promise in Theorem 4.2), then*

$$\oplus\mathsf{L} \subseteq (\mathsf{qBPAC}^0)^{\mathcal{R}'}.$$

The corollary above suffices to prove the conditional separation mentioned in the Introduction (Theorem 1.4): assuming $\oplus\mathsf{L} \notin (\mathsf{qBPAC}^0)^{\mathsf{L}}$, any L machine fails the Noisy 2-Round Graph State Measurement Problem with some constant probability. Suppose that some L machine fails with probability $o(1)$. Then by Corollary 4.1, we have $\oplus\mathsf{L} \subseteq (\mathsf{qBPAC}^0)^{\mathsf{L}}$, contradicting the assumption.

4.1.1 Input half-randomization and $F \circ D$ description

The two halves of our randomization procedure, F and D, are derived from different sources. The F procedure is based on the randomization in [GS20], which mostly randomizes the *second* round of the interactive protocol. The result is that an adversarial oracle cannot prevent us from learning the state $|\psi\rangle$ at the end of the first round. We keep F so we do not have to re-invent the wheel for the second round. However, F does very little to randomize the first round of the protocol, and thus it does a poor job concealing the instance of the $\oplus\mathsf{L}$-hard problem we are trying to solve (i.e., $\mathsf{LDAGParity}$). It is conceivable that the oracle has some small probability of error, but coincidentally concentrates that error on the instance (or small number of instances which F randomizes to the same distribution) that we wish to solve.

The other procedure, D, is adapted from work of Applebaum, Ishai and Kushilevitz [AIK06]. They introduce the notion of a *randomized encoding*, where it is possible to transform x to a probability distribution that hides everything about x except some function $f(x)$. For example, and especially relevant to us, they show how a binary matrix of a certain form can be mapped to a uniformly random matrix of the same form, and same determinant modulo 2. In this case, the randomized encoding hides everything about the input (which, recall, is essentially the problem instance) except for one bit, a property we call *half randomizing*. It turns out the parity of the determinant is even $\oplus\mathsf{L}$-complete, but since our quantum gates are inherently reversible (i.e., non-zero determinant), we must find a more nuanced way to randomize a $\oplus\mathsf{L}$-complete problem to adequately randomize the first round.

Let us describe this approach in more detail. We will define two additional problems: $\mathsf{LDAGParity}$ (which is a layered version of MGap) and $\mathsf{CNOTMult}^*$ (which serves as a more natural input for MBQC). We note that D maps an instance of MGap to $\mathsf{LDAGParity}$ and F maps an instance of $\mathsf{LDAGParity}$ to $\mathsf{CNOTMult}^*$.

Define $\mathsf{LDAGParity}$ to be the problem of deciding whether a layered DAG has an even number of paths from a fixed start vertex to a fixed end vertex, and let $\mathsf{CNOTMult}^*$ be the problem of deciding whether CNOT gates $g_1, ..., g_n \in \mathrm{CNOT}_m$ multiply to the identity, promised that they multiply to either the identity or a fixed 3-cycle. It was shown [GS20] that $\mathsf{LDAGParity}$ is NC^0-reducible to $\mathsf{CNOTMult}^*$, using a randomized algorithm F.

Lemma 4.2. *There is randomized NC^0 circuit, F, that takes as input an LDAG adjacency matrix A of dimension $n \times n$ and outputs a sequence of $g_1, ..., g_{\mathsf{poly}(n)} \in \mathrm{CNOT}_m$ such that*

1. *$A \in \mathsf{LDAGParity}$ iff $(g_1, ..., g_{\mathsf{poly}(n)}) \in \mathsf{CNOTMult}^*$.*

2. *For any distinct LDAG adjacency matrices A and A', $F(A)$ and $F(A')$ are uniform distributions over disjoint sets of the same cardinality*

We discuss F in more detail in Appendix B.1 and prove the second point of the lemma in Lemma B.4. Note that F is a *randomized* algorithm, outputting a random sequence of $g_1, \ldots, g_{\mathsf{poly}(n)} \in \mathrm{CNOT}_m$. This randomization offers a weak form of random self-reduction. We want to boost the randomization done by F to promote this procedure to a good random self-reduction. The new layer of randomization, D, does this by *pre*-randomizing the input to F; it occurs *before* the reduction on an $\mathsf{LDAGParity}$ instance occurs. As a result, the image of $F \circ D$ becomes the set of first round inputs to the rewind oracle. We will often use r_f and r_d as notation for the random bits of F and D.

To show that combining the random self-reductions of F and D produces highly random instances to the first-round input, we will need that $F \circ D$ has a property called *half-randomizing.*

Definition 4.1. *Let $\Pi = (\Pi_{yes}, \Pi_{no})$ be a promise problem. A randomized algorithm $\tilde{f}: \Pi \to \mathcal{C}$ is* half-randomizing *if there are two disjoint, equal-sized subsets of $\mathcal{C}$, C_{yes} and C_{no}, such that for $i \in \{yes, no\}$*

$$\forall x \in \Pi_i,\ \tilde{f}(x) \equiv U_{C_i}$$

where U_{C_i} is the uniform distribution over C_i.

There is an immediate consequence of Definition 4.1:

Proposition 2. *If δ fraction of $C_{yes} \cup C_{no}$ have some property, p, then for any $x \in \Pi_{yes} \cup \Pi_{no}$,*

$$Pr[\tilde{f}(x) \text{ has property } p] \leq 2\delta$$

To give some intuition for why Proposition 2 may be useful, consider an oracle that fails for ϵ fraction of inputs. By randomizing its input using $\tilde{f}$, an algorithm that uses this oracle will only have a 2ϵ probability of querying the oracle with a failing input. This is precisely the idea that we use to achieve average-case hardness.[10]

Randomized algorithm D is very similar to the randomized encoding techniques introduced by [AIK06], except we introduce an extra step that an NC^0 circuit can perform. It takes as input a monotone adjacency matrix from M and converts it to a LDAG adjacency matrix. It will be useful to define $D = D_{adj} \circ D_{re}$ where D_{re} is the randomized encoding map from [AIK06] (to be defined) and D_{adj} is the modification that creates the adjacency matrix of a LDAG. We describe some useful properties related to D_{re}.

Fact 3. *(Fact 4.13, [AIK06]) Let A be an $n \times n$ monotone adjacency matrix, and let L be the $(n-1) \times (n-1)$ top-right submatrix of $A - I$. Then $\det(L) \mod 2$ is the parity of the number of paths from 1 to n in A.*

So the determinant of L encodes the answer to a $\oplus\mathsf{L}$-hard problem. Note that L is upper-triangular *except* for -1 on its second diagonal. Furthermore, we can sample uniformly from a certain representation of all matrices that have the same determinant as L and have -1 on their second diagonal. We call the next sampling procedure D_{re}.

[10] Of course, the trouble is in making sure that the input to the oracle preserves information about the problem instance, which half-randomizing algorithms do.

Proposition 3. *(D_{re}, Lemma 4.17, [AIK06]) Given A, a randomized NC^0 circuit can sample uniformly 0/1-values*

$$\{(K^{(1)}_{i,j}, ..., K^{(k)}_{i,j})\}_{1 \le i \le j \le n-1}$$

(where k is polynomially related to n) such that the $(n-1) \times (n-1)$ matrix $K^{\oplus}$ defined by $K^{\oplus}_{i,j} = \oplus_\ell K^{(\ell)}_{i,j}$, -1 on its second diagonal, and 0 below it has

$$\det(K^{\oplus}) \equiv \det(L) \mod 2 \tag{11}$$

Following D_{re}, we let D_{adj} convert the $\{(K^{(1)}_{i,j}, ..., K^{(k)}_{i,j})\}_{1 \le i \le j \le n-1}$ matrix representation to the adjacency matrix of a LDAG. Before doing this, we define the $n \times n$ 0/1 matrix B: Let the 0/1 entries above the main diagonal of B be the same 0/1 entries on or above the main diagonal of $K^{\oplus}$ and let every other entry in B be 0.

Proposition 4. *B is a uniformly random monotone adjacency matrix such that the parity of paths from 1 to n is the same as in A.*

Proof. Note that B is a valid monotone adjacency matrix, and $K^{\oplus}$ is the $(n-1) \times (n-1)$ top-right submatrix of $B - I$. The entries on or above the main diagonal of $K^{\oplus}$ are uniformly random subject to the determinant constraint in Equation (11), so the entries above the main diagonal of $B - I$, and hence B, are uniformly random subject to the constraint. However, the constraint is equivalent to A and B having the same parity of paths from 1 to n due to Fact 3. □

We are now ready to define D_{adj}, which takes $\{(K^{(1)}_{i,j}, ..., K^{(k)}_{i,j})\}_{1 \le i \le j \le n-1}$ and deterministically converts it to a LDAG $\mathcal{L}$ that preserves the parity of the number of paths from source to sink as B. $\mathcal{L}$ has n layers labeled N_i for each $i \in [n]$ and $(n-1)$ layers labeled J_i for each $i \in [n-1]$, and the layers are ordered from left to right in the following way: $N_1 J_1 N_2 J_2 ... J_{n-1} N_n$. Each N_i has vertices $\{1, ..., n\}$ and each J_i has vertices $\{1, ..., n\} \cup \{K^{(\ell)}_{i,j}\}_{j,\ell}$. The only edges between layers fall under two classes:

1. (N_i to J_i) For every $q \in [n]$, vertex q in N_i always connects to vertex q in J_i. Vertex i in N_i connects to vertex $K^{(\ell)}_{i,j}$ in J_i iff the value $K^{(\ell)}_{i,j}$ is 1.
2. (J_i to N_{i+1}) For every $q \in [n]$, vertex q in J_i always connects to vertex q in N_{i+1}. Vertex $K^{(\ell)}_{i,j}$ in J_i always connects to vertex j in N_{i+1}.

The D_{adj} procedure can easily be computed in NC^0 from the $\{(K^{(1)}_{i,j}, ..., K^{(k)}_{i,j})\}_{1 \le i \le j \le n-1}$ matrix representation to $\mathcal{L}$ using the description above.

Lemma 4.3. *Given $A \in M$, $\mathcal{L} = D(A)$ is a uniformly random output of the image of $D(M)$ such that the parity of number of paths from vertex 1 in N_1 to vertex n in N_n is equal to the parity of the number of paths from 1 to n in A. Furthermore, D is half-randomizing.*

Proof. For the first part of the claim, it suffices to show that for a given $K = \{K^{(\ell)}_{i,j}\}_{i,j,\ell}$ the parity of number of paths from vertex 1 in N_1 to vertex n in N_n in $\mathcal{L} = D_{adj}(K)$ is equal to the parity of the number of paths from 1 to n in the B associated with K via and due to Proposition 4.

We prove inductively that the parity of paths from vertex i in N_i to vertex n in N_n is the same as i to n in B. Note that the edge i to i between N_i and J_i does not contribute to the parity of paths from $i \in N_i$ to $n \in N_n$. The parity of paths from $i \in N_i$ to $j \in N_{i+1}$ is equal to $K^{\oplus}_{i,j}$, and there is only one path from $j \in N_{i+1}$ to $j \in N_j$. Hence, the paths from $i \in N_i$ to $j \in N_j$ only change the parity if $K^{\oplus}_{i,j} = 1$ and the parity of paths from j to n in B is odd (by assumption). Therefore, the parity of paths from vertex i in N_i to vertex n in N_n is equal to those from i to n in B.

For the second part of the claim, let Π_{yes} be odd (1 to n) path-parity adjacency matrices and Π_{no} be even adjacency matrices. D_{re} is half-randomizing due to Proposition 4, and D_{adj} is injective, so $D = D_{adj} \circ D_{re}$ is half-randomizing. □

This concludes the definition of D. Now it is straightforward to prove the core half-randomizing property of $F \circ D$.

Lemma 4.4. *$F \circ D$ is half-randomizing*

Proof. Recall that for any distinct $A, A' \in M$, $F(A)$ and $F(A')$ are uniform distributions over disjoint sets of the same cardinality by Lemma 4.2. Combined with the fact that D is half-randomizing, we get that $F \circ D$ is half-randomizing. □

In other words, the half-randomizing procedure $F \circ D$ maps odd parity adjacency matrices to gates that multiply to the 3-cycle, and even parity adjacency matrices to gates that multiply to the identity.

Note that F alone is not half-randomizing and does not suffice for a worst-to-average-case reduction. As before, for any distinct adjacency matrices A and A', $F(A)$ and $F(A')$ are completely disjoint, so all the error in the rewind oracle can, for example, be concentrated in $F(A)$ (an exponentially small fraction of all possible input) making oracle access to $F(A)$ useless. As a result, proving average-case hardness using a worst-to-average-case reduction does not work in this case. However, if this type of concentration of error does not occur, we get something better.

Proposition 5. *Let A be an LDAG adjacency matrix of dimension $n \times n$ and $\mathcal{R}$ be the rewind oracle 2-Round Graph State Measurement with the promises following Theorem 4.2. Then*

$$\Pr_{r_f}[\mathcal{R} \textit{ fails on any input in } F_{r_f}(A) \times I_2] < \frac{1}{21}$$

then a $(\mathsf{BPAC}^0)^{\mathcal{R}}$ can decide the the parity of the number of paths from 1 *to* n *in* A.

We also defer the proof of Proposition 5 to the appendix in Section B.1. Our goal is to show that we can often satisfy the hypothesis in Proposition 5 and leverage its circuit to solve a $\oplus\mathsf{L}$-hard problem.

4.1.2 Proof of average-case $\oplus\mathsf{L}$-hardness

We are now ready to prove the main theorem of this section. Given the adjacency matrix of a monotone graph A, it is $\oplus\mathsf{L}$-hard to decide if the parity of paths from 1 to n is even or odd due to Lemma 4.1. By assumption, ϵ fraction of input from $F \circ D(M)$ in the first round combined with a set of Pauli operator from I_2 in the second round will lead to faulty output by the rewind oracle. Consequently, at most 2ϵ fraction of inputs from $F \circ D(A) \times I_2$ will lead to faulty output because $F \circ D$ is half-randomizing by Lemma 4.2 and the error bound in Proposition 2. Furthermore,

$$\mathop{\mathbb{E}}_{r_d}[\Pr_{r_f}[\mathcal{R} \text{ fails on any input in } F_{r_f} \circ D_{r_d}(A) \times I_2]] \leq 10\epsilon$$

by a union bound with $|I_2| = 5$. By Markov's inequality,

$$\Pr_{r_d}[\Pr_{r_f}[\mathcal{R} \text{ fails on any input in } F_{r_f} \circ D_{r_d}(A) \times I_2] \geq \frac{1}{21}] \leq 210\epsilon. \tag{12}$$

Deciding the parity of the number of paths from the start to end vertex of $D_{r_d}(A)$ also decides the parity of the number of paths from 1 to n in A by Lemma 4.3. We can reduce the error in the BPAC^0 circuit C from Proposition 5, so that it succeeds with probability $1 - \alpha$ for any constant $\alpha > 0$. Using this error-reduced circuit, we get the following

$$\Pr_{r_d}[C \text{ fails to decide parity of } D_{r_d}(A)] \leq 210\epsilon + \alpha(1 - 210\epsilon) < \frac{1}{2}$$

where the first inequality comes from Proposition 5 and Equation (12) and the final inequality comes from choosing small enough α and $\epsilon < \frac{1}{421}$. To finish, we repeat the process by sampling $O(\lg n)$ elements from $D(A)$ and applying circuit C on the sampled instance. By taking a majority vote of outputs over all instances, we recover the parity of the paths from 1 to n in A with high probability. This proves the claim in Theorem 4.2.

4.2 NC^1-hardness

In this section, we consider an interactive task that a noisy quantum circuit can solve (with high probability) but classically solving the task is NC^1-hard. Roughly speaking, this is how we will show that a $\mathsf{AC}^0[p] \subsetneq \mathsf{NC}^1$ circuit cannot solve such a problem.

Without noise, the problem for this section is CliffSim[2] from [GS20]. That is, an instantiation of the 2-round Graph State Measurement problem on a $2 \times O(n)$ grid (the brick-like pattern of Figure 3 was used in [GS20] to reduce the number of qubits per Clifford operation, but the result is qualitatively the same with a grid), so that the first round is suitable for an MBQC simulation of a sequence of 2-qubit Clifford gates, and the second round is used to apply one more Clifford gate to permit arbitrary Clifford measurements of a 2-qubit state.

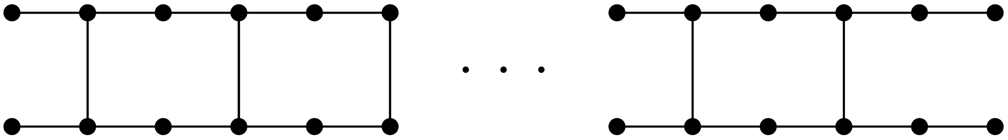


Figure 3: Cluster state $|\mathcal{H}_n\rangle$

Let us quickly recap how [GS20] connects this problem to NC^1-hard computation. First, a result of Barrington and Thérien shows that computing products in non-solvable groups is

NC^1-hard [BT88], even to distinguish between the product being the identity or some other predetermined group element. The 2-qubit Cliffords modulo 2-qubit Pauli operations, which we denote $\{\mathcal{C}_2/\mathcal{P}_2\}$, turn out to be a non-solvable group (specifically, isomorphic to S_6). For this special case, Barrington and Thérien's result translates to the following:

Theorem 4.3. *Let $C_1, \ldots, C_n \in \mathcal{C}_2/\mathcal{P}_2$ be given. Promised that the product $C_1 \cdots C_n$ is either $I \otimes I$ or $H \otimes H$ modulo Pauli operations, deciding whether $|\psi\rangle := C_1 \cdots C_n |{+}{+}\rangle$ is an X-basis state (i.e., $|{+}{+}\rangle, |{+}{-}\rangle, |{-}{+}\rangle, |{-}{-}\rangle$) or a Z-basis state ($|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$) is NC^1-hard.*

Our problem for this section is explicitly designed to prepare $|\psi\rangle$ (up to an unknown Pauli $P \in \mathcal{P}_2$) and measure it in an arbitrary Clifford basis. Our goal is to show that a rewind oracle for the problem solves an NC^1-hard problem. All that remains is to show how repeated measurements of $P|\psi\rangle$ in arbitrary Clifford bases can be used to determine something about the state, and ultimately leveraged to distinguish X-basis states from Z-basis states.

We borrow a tool of [GS20] to extract some information about the first round state from the rewind oracle $\mathcal{R}$. We simply quote the lemma here without proof, but it uses a demonstration of non-contextuality, the magic square game, to force the oracle to reveal something about the state, and then it uses a plethora of randomization tricks so the oracle does not control what it reveals.

Lemma 4.5. *(Theorem 24 in [GS20]) Let $\mathcal{R}$ be a rewind oracle solving the $2 \otimes O(n)$ interactive problem, possibly with some error (i.e., it may fail the task some fraction of the time). Suppose we are given gates $C_1, \ldots, C_n \in \mathcal{C}_2$. There is a randomized $(\mathsf{AC}^0)^{\mathcal{R}}$ circuit which*

- *processes $C_1, \ldots, C_n$ to a related but uniformly random sequence of operations,*
- *calls $\mathcal{R}$ with the above uniformly random sequence in the first round, and makes 6 uniformly random (but not independent) measurements in the second round (i.e., rewinds 5 times), then*
- *post-processes the outputs from $\mathcal{R}$ in conjunction with the randomization.*

The result is an algorithm which outputs either a uniformly random stabilizer Pauli (excluding II) or uniformly random non-stabilizer Pauli of $C_1 \cdots C_n |{+}{+}\rangle$. Moreover, the circuit outputs a stabilizer only if at least one call to $\mathcal{R}$ was in error.

We apply this lemma repeatedly to learn many non-stabilizers of $C_1 \cdots C_n |{+}{+}\rangle$, until we accumulate enough non-stabilizers to determine the state $C_1 \cdots C_n |{+}{+}\rangle$ up to Pauli operations. Thus, a rewind oracle for the 2-Round Graph State Measurement Problem on a $2 \times \mathsf{poly}(n)$ grid, with randomized AC^0 reduction circuits, can solve the NC^1-hard problem in Theorem 4.3. We do not state the theorem (or proof) since it will generalize it to the noisy version of the problem in Lemma 4.6, with proof in Appendix A.

4.2.1 Noisy NC^1 separation

Now let us add an error correcting code with $\mathsf{polylog}(n)$ physical qubits per logical qubit, so that a noisy constant-depth quantum circuit can solve the problem with high probability. That is, we extend the original 2-round interactive problem on a $2 \otimes \mathsf{poly}(n)$ grid to its noisy-extended version using Problem 2.

By applying Theorem 4.1, we immediately get a constant depth quantum circuit that solves the problem.

Corollary 4.2. *There is a quantum circuit of constant depth D that passes the Noisy 2-Round Graph State Measurement Problem with probability at least* $1 - \exp(-\Omega(\mathsf{polylog}(n)))$ *for all possible inputs when the noise rate p is bounded by* $p < p_{th} = \exp(-\exp(O(D)))$.

We now prove classical hardness for the same task. We first show that a $(\mathsf{BPAC}^0)^{\mathcal{R}}$ circuit can decide a NC^1-hard problem even if the rewind oracle $\mathcal{R}$ fails on $\frac{1}{30}$ possible inputs. This is a slight improvement over the $\frac{2}{75}$ present in [GS20]. Then we reduce the from the noisy problem to the noiseless problem to complete the reduction.

Lemma 4.6. *Let $\mathcal{R}$ be the rewind oracle for the noiseless interactive problem in Section 4.2. Suppose for a uniformly random input (first & second round), $\mathcal{R}$ is incorrect with probability $\epsilon < \frac{1}{30}$. Then*

$$\mathsf{NC}^1 \subseteq (\mathsf{BPAC}^0)^{\mathcal{R}} \tag{13}$$

We defer the proof to Appendix A. Clearly, replacing a rewind oracle for the noiseless problem with one for the noisy problem (having the same error rate) makes no difference, except that we require quasipoly-size AC^0 for decoding. By decoding its output using Lemma 3.1, we conclude that a $(\mathsf{qBPAC}^0)^{\mathcal{R}}$ solves an NC^1-hard problem.

Theorem 4.4. *Let $\mathcal{R}$ be the rewind oracle for the noisy interactive problem in Section 4.2. Suppose for a uniformly random input (first & second round), $\mathcal{R}$ is incorrect with probability $\epsilon < \frac{1}{30}$. Then*

$$\mathsf{NC}^1 \subseteq (\mathsf{qBPAC}^0)^{\mathcal{R}} \tag{14}$$

Quasi-polynomial size bounded-probability AC^0 circuits (qBPAC^0) may seem like an unnatural class, but we can relate this result to more familiar classes.

Corollary 4.3. *For any prime p, any $\mathsf{AC}^0[p]$ circuit of depth d and size* $\exp(n^{1/2(d+4)})$ *cannot pass the noise-tolerant interactive measurement problem with probability at least $\frac{29}{30}$ over a uniform input.*

Proof. Let $\mathsf{AC}^0[p](d,s)$ be the set of $\mathsf{AC}^0[p]$ circuits of depth d and size s as stated. Suppose for contradiction that such a circuit could pass the interactive protocol with probability $\frac{29}{30}$ over the uniform input distribution. Then there must exist an $\mathsf{AC}^0[p](d,s)$ circuit for the rewind oracle that succeeds with probability at least $\frac{29}{30}$, so $\mathsf{NC}^1 \subseteq \mathsf{BPAC}^0[p](d,s)$. A result by Ajtai and Ben-Or is that BPAC^0 is contained in non-uniform AC^0 with four more layers in depth and polynomial overhead in size [AB84]. Their proof also holds when considering any class of AC^0 circuits of at least polynomial size equipped with MOD_p gates, so $\mathsf{BPAC}^0[p](d,s) \subseteq \mathsf{AC}^0[p](d+4,s)$. $\mathsf{NC}^1 \subseteq \mathsf{AC}^0[p](d+4,s)$ contradicts the exponential lower bounds by Razborov and Smolensky [Raz87; Smo87], so we conclude that no $\mathsf{AC}^0[p](d,s)$ circuit can pass the interactive measurement problem with average probability $\frac{29}{30}$. □

4.3 w-PBP-hardness

The main theorem of this section is an average-case w-PBP-hardness of classically simulating w-width 2D grids of constant-depth quantum circuits. More specifically, we want to relate Problem 1 on a w-width grid to a w-PBP-hard problem.

4.3.1 Statement of main theorem and reduction

Going further, the main theorem of this section does not deal with $g_1, ..., g_n \in \text{CNOT}_m$ like in the $\oplus\mathsf{L}$ case. Instead, we have $g_1, ..., g_n \in \langle \text{SWAP} \rangle$, and we randomize input to the rewind oracle by giving it input from $H'_m := \langle \text{SWAP}, \text{CZ}, \text{S} \rangle_m$ in the first round. Let us state the main theorem.

Theorem 4.5. *Let $w \leq \mathsf{poly}(n)$ and $\mathcal{R}$ be the rewind oracle for the 2-round Graph State Measurement Problem on a $(w+2) \times \mathsf{poly}(n)$ grid promised that input is from H'_m in the first round and I_2 in the second round. If $\mathcal{R}$ fails with probability $\epsilon < \frac{1}{106}$, then*

$$w\text{-}\mathsf{PBP} \subseteq (\mathsf{BPTC^0})^{\mathcal{R}}$$

From well-known results in complexity theory, we have the following corollaries:

1. $(w = 5)$ $\mathsf{NC^1} \subseteq (\mathsf{BPAC^0})^{\mathcal{R}}$. Note that the error bound ϵ is weaker here than X.
2. $(w = \mathsf{poly}(n))$ $\mathsf{L} \subseteq (\mathsf{BPTC^0})^{\mathcal{R}}$

The reduction in Theorem 4.5 utilizes $\mathcal{R}$ to simulate a product of permutations by performing transpositions of qubits via SWAP gates, enabled by measurement-based computing. However, it will be useful to reduce the problem of determining a product of permutations to the problem of deciding whether a permutation is the 3-cycle or identity to leverage some of the tools we used in the $\oplus\mathsf{L}$ result. Toward that end, we have the following reduction:

Lemma 4.7. *Let $w \leq \mathsf{poly}(n)$ and $\mathcal{O}$ be the oracle that decides whether a sequence of permutations on $w+2$ elements multiplies to C_3 or I. Then*

$$w\text{-}\mathsf{PBP} \subseteq (\mathsf{NC^0})^{\mathcal{O}}$$

Proof. Let $\pi = \pi_1 ... \pi_n$ be the product of the permutations on w elements induced by the w-PBP. A YES instance implies $\pi = \alpha$ for some fixed non-identity permutation α. Otherwise, a NO instance implies $\pi = I$. Because α is non-identity, then $\alpha(x) = y$ for some $x \neq y \in [w]$. So deciding whether $\beta = (1\ x)\pi(1\ y)$ fixes the first element also decides whether π is α or I.

If we define the permutation $D = \beta(1\ (w+1))\beta^{-1}$, then $D = (1\ (w+1))$ if β fixes the first element. Otherwise, $D = (z\ (w+1))$ for some $z \in \{2, ..., w\}$. If we define the permutation $E = D(1\ (w+2))D(1\ w+2)$, then $E = (1\ (w+2)\ (w+1))$ if $D = (1\ (w+1))$. Otherwise, $E = I$ if $D = (z\ (w+1))$. Therefore, distinguishing E as C_3 or I determines whether β fixes the first element, and hence, decides w-PBP. Note that the only parts necessary in this reduction by a circuit are fan-out and nonuniformity, which are provided by $\mathsf{NC^0}$. □

Equipped with Lemma 4.7, we can reuse some of the same tools from our proof of Theorem 4.2 to learn information about the permutation π. Recall in that proof, we introduced the pre-randomizer, D, to increase the randomization in the random self-reduction because F could not strongly randomize its input. The random self-reduction D relied crucially on the fact that we were concerned with a parity calculation originating from a $\oplus\mathsf{L}$-hard problem. In this case, we are concerned with learning the permutation induced by a permutation branching program, so applying D in this case is less straightforward. Instead, we aim to directly increase the randomization in F. We leave the full proof to Appendix C.

5 Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). DG is also supported in part by IBM Research. We also thank David Gosset for useful discussions.

References

[AB84] Miklos Ajtai and Michael Ben-Or. "A Theorem on Probabilistic Constant Depth Computations". In: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*. STOC '84. Association for Computing Machinery, 1984, pp. 471–474.

[Raz87] A. A. Razborov. "Lower bounds on the size of bounded depth circuits over a complete basis with logical addition". In: *Mathematical Notes of the Academy of Sciences of the USSR* 41.4 (1987), pp. 333–338.

[Smo87] Roman Smolensky. "Algebraic methods in the theory of lower bounds for Boolean circuit complexity". In: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*. STOC '87. Association for Computing Machinery, 1987, pp. 77–82.

[BT88] David A. Mix Barrington and Denis Thérien. "Finite Monoids and the Fine Structure of NC^1". In: *J. ACM* 35.4 (Oct. 1988), pp. 941–952.

[Dam90] Carsten Damm. "Problems complete for $\oplus\mathsf{L}$". In: *International Meeting of Young Computer Scientists* (1990), pp. 130–137.

[MV91] Yossi Matias and Uzi Vishkin. "Converting High Probability into Nearly-Constant Time—with Applications to Parallel Hashing". In: *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*. STOC '91. New Orleans, Louisiana, USA: Association for Computing Machinery, 1991, pp. 307–316.

[Sho97] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM J. Comput.* 26.5 (Oct. 1997), pp. 1484–1509.

[RB01] Robert Raussendorf and Hans J. Briegel. "A One-Way Quantum Computer". In: *Phys. Rev. Lett.* 86 (22 May 2001), pp. 5188–5191.

[RBB03] Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. "Measurement-based quantum computation on cluster states". In: *Phys. Rev. A* 68 (2 Aug. 2003), p. 022312.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. "Cryptography in NC^0". In: *SIAM J. Comput.* 36.4 (2006), pp. 845–888.

[BJS10] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. “Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2010, pp. 459–472.

[AA11] Scott Aaronson and Alex Arkhipov. “The Computational Complexity of Linear Optics”. In: *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*. STOC ’11. Association for Computing Machinery, 2011, pp. 333–342. ISBN: 9781450306911.

[Boi+18] S. Boixo et al. “Characterizing quantum supremacy in near-term devices”. In: *Nature Physics* 14.6 (2018), pp. 595–600.

[BGK18] Sergey Bravyi, David Gosset, and Robert Koenig. “Quantum advantage with shallow circuits”. In: *Science* 362 (6412 Oct. 2018), pp. 308–311.

[CSV18] Matthew Coudron, Jalex Stark, and Thomas Vidick. “Trading locality for time: certifiable randomness from low-depth circuits”. In: *arXiv preprint arXiv:1810.04233* (2018).

[FGL18] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. “Constant overhead quantum fault-tolerance with quantum expander codes”. In: IEEE. 59th Annual Symposium on Foundations of Computer Science (FOCS), 2018, pp. 743–754.

[Ben+19] Adam Bene Watts et al. “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*. STOC ’19. Association for Computing Machinery, 2019.

[Le 19] François Le Gall. “Average-case quantum advantage with shallow circuits”. In: *34th Computational Complexity Conference (CCC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2019.

[Bra+20] Sergey Bravyi et al. “Quantum advantage with noisy shallow circuits”. In: *Nature Physics* (2020), pp. 595–600.

[GS20] Daniel Grier and Luke Schaeffer. “Interactive shallow Clifford circuits: quantum advantage against NC^1 and beyond”. In: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*. STOC ’20. Association for Computing Machinery, 2020.

A Improved NC^1-hardness error bound

Here we improve the error threshold at which CliffSim[2], the problem on an $2 \times O(n)$ grid from Section 4.2, becomes hard. Previously, the threshold was $\epsilon < \frac{2}{75}$ [GS20], and we improve it to $\epsilon < \frac{1}{30}$.

Lemma A.1. *Let $\mathcal{R}$ be the rewind oracle for the interactive problem in Section 4.2. Suppose for a uniformly random input (first & second round), $\mathcal{R}$ is incorrect with probability $\epsilon < \frac{1}{30}$. Then*

$$\mathsf{NC}^1 \subseteq (\mathsf{BPAC}^0)^{\mathcal{R}} \tag{15}$$

Proof. Let $C_1, \ldots, C_n$ be two-qubit Cliffords. By Lemma 4.5, we can use $\mathcal{R}$ to output one of fifteen non-trivial Paulis:[11] either a uniformly random non-stabilizer or a uniformly random stabilizer (excluding II) for the state $|\psi\rangle := C_1 \cdots C_n |{+}{+}\rangle$.

Since Lemma 4.5 makes 6 uniformly random calls to $\mathcal{R}$, and each one fails with probability at most ϵ, there is a failure with probability at most 6ϵ. Only when there is a failure can the algorithm output a stabilizer Pauli, and there are three nontrivial stabilizers of $|\psi\rangle$, so the probability of returning any particular stabilizer $S \in \mathcal{P}_2 \backslash \{II\}$ is

$$\Pr[\text{output } S] < 6\epsilon \cdot \frac{1}{3}. \tag{16}$$

On the other hand, at least $1 - 6\epsilon$ fraction of the time the algorithm returns one of twelve nonstabilizer Paulis. Thus, any particular nonstabilizer $N \in \mathcal{P}_2$ is observe with probability

$$\Pr[\text{output } N] \geq (1 - 6\epsilon) \cdot \frac{1}{12}. \tag{17}$$

As long as $\epsilon < \frac{1}{30}$, we have that Equation (16) and Equation (17) are bounded above by $\frac{1}{15}$ and below by $\frac{1}{15}$, respectively. In fact, the gap between the bounds is a constant, $\sigma := (1 - 30\epsilon)/12$, so with $O(1/\sigma^2)$ samples we can empirically estimate the probability for each Pauli well enough to distinguish stabilizers from nonstabilizers (with constant probability). It follows $(\mathsf{BPAC}^0)^R$ circuit can learn the stabilizer group of $|\psi\rangle$, and thus solve an NC^1 hard problem. □

B Error analysis

Here we provide an error analysis for a protocol originating from [GS20]. We review the protocol in the necessary detail for proving the error analysis relevant to Theorem 4.2 and Theorem 4.5 and refer the reader to [GS20] for proofs of correctness.

We define three groups and a set of 3-qubit Paulis. Let the group G_m consist of single CNOT gates followed by circuits generated by CZ and S gates on m qubits; i.e. $G_m = \mathrm{CNOT}_m \langle \mathrm{CZ}, \mathrm{S} \rangle_m$; where the binary operation is composition. Let $H_m = \langle \mathrm{CZ}, \mathrm{S} \rangle_m$, where we notice that H_m is a normal subgroup of G_m. Finally, let $H_3^{\oplus}$ be the subgroup of H_m that consists of even numbers of CZ and S gates on the first three qubits. In summary, we have defined the following relationship: $H_3^{\oplus} \leqslant H_m \trianglelefteq G_m$.

Then let

$$I_2 = \{\{XXX, XYY, YXY, YYX\}, \{IYI, XII, XYY, IIY\}, \{IIY, YXY, YII, IXI\}, \{XXX, XII, IIX, IXI\}, \{IYI, IIX, YII, YXY\}\}$$

be a collection of sets of 3-qubit Paulis, and define $\star$ be the union over these sets. Let us define the operation $A \cdot B$ as ABA^{-1}. It can be checked that $|H_3^{\oplus} \cdot \star| = 24$ where $\cdot$ is performed pairwise between the elements of the two sets. Next, we review the protocol from [GS20]. It receives as input $f \in H_3^{\oplus}$ and $g_1, \ldots, g_n \in \mathrm{CNOT}_m$ that is promised to multiply to the 3-cycle or identity, and it outputs a 3-qubit Pauli that reveals some information about whether the product is the 3-cycle or identity. The protocol, R_f, is defined to be the following:

[11] We remind the reader that we consider Paulis only up to sign, and exclude II, which is why there are fifteen rather than 32 or 64

Algorithm 1: Randomized algorithm, R_f, carried out by $(\mathsf{BPAC}^0)^{\mathcal{R}}$ circuit

```
   Input: $f \in H_3^{\oplus}$, $g_1, ..., g_n \in \mathrm{CNOT}_m$ promised that $\pi = g_1 ... g_n \in \{C_3, I\}$
   Output: A 3-qubit Pauli in ☆
1  Sample $f' \leftarrow H_3^{\oplus}$;
2  Sample $h_1, ..., h_{2n-1} \leftarrow H_m$;
   /* The following is Kilian randomization                                  */
3  $(g'_1, ..., g'_{2n}) \leftarrow (f' g_1 h_1, h_1^{-1} g_2 h_2, ..., h_{n-1}^{-1} g_n h_n, h_n^{-1} f g_n h_{n+1}, ..., h_{2n-2}^{-1} g_2 h_{2n-1}, h_{2n-1}^{-1} g_1)$;
4  Input $(g'_1, ..., g'_{2n})$ to $\mathcal{R}$ in first round;
5  for Pauli line $(P_1, P_2, P_3, P_4)$ in $I_2$ do
6      Input $(P_1, P_2, P_3, P_4)$ to $\mathcal{R}$ in the second round;
7      Record measurement outcome of $P_i$ for $i \in [4]$ from $\mathcal{R}$;
8      Rewind $\mathcal{R}$ to beginning of second round;
   /* Each 3-qubit Pauli $P_i$ is measured exactly twice in the loop         */
9  $P \leftarrow$ any 3-qubit Pauli for which $\mathcal{R}$ returns inconsistent results;
10 return $f'^{-1} \cdot P$
```

The first round input $(g'_1, ..., g'_{2n})$ is uniformly random with the constraints: (1) $g'_i H_m = g_i H_m$ and $g'_{n+i} H_m = g_{n-i+1} H_m$ for $i \in [n]$ and (2) $g'_1 ... g'_{2n} \in H_3^{\oplus}$. Let us denote by $I_1(g_1, ..., g_n)$ the support of this distribution.

If the oracle makes no errors, then the output of R_f will be a sample from the distribution of 3-qubit Paulis $(\pi f \pi^{-1}) \cdot \mathcal{D}_{\mathcal{R}}$ where $\mathcal{D}_{\mathcal{R}}$ is a distribution over the 20 nonstabilizers of $|+^3\rangle$ in $S := H_3^{\oplus} \cdot$ ☆ . Let us call this errorless distribution $R_{f,0}$. Otherwise, suppose that the oracle fails on $(g'_1, ..., g'_{2n})$ with probability ϵ for $(g'_1, ..., g'_{2n})$ sampled from $I_1(g_1, ..., g_n)$, where we say the oracle fails for $(g'_1, ..., g'_{2n}) \in I_1(g_1, ..., g_n)$ if it fails for any input in $(g'_1, ..., g'_{2n}) \times I_2$. Then with probability ϵ, R_f samples from any fixed distribution over S, and with probability $1 - \epsilon$, R_f samples from $R_{f,0}$. Let us call this faulty distribution $R_{f,\epsilon}$. Notice that in both cases, $\mathcal{D}_{\mathcal{R}}$ is fixed regardless of the value of f. This leads us to choose f in a useful way for distinguishing the 3-cycle or identity.

Lemma B.1. *(Theorem 33, [GS20]) For every Pauli $P \in \mathcal{D}_{\mathcal{R}}$, a NC^0 circuit can determine an $f \in H_3^{\oplus}$ such that $f \cdot P$ has no weight in $R_{f,0}$ if $\pi = C_3$.*

Lemma B.1 becomes useful only after we have a Pauli P on which to apply it. After we have identified such a P, the problem of determining whether π is the 3-cycle or identity becomes tractable for BPAC^0 circuits, and this roughly becomes the idea for learning π.

Lemma B.2. *Suppose that a BPAC^0 circuit can sample from a distribution $\gamma(g_1, ..., g_n)$ over first round inputs to $\mathcal{R}$ of the form in Line 3 of Algorithm 1 and*

$$\delta := Pr[\mathcal{R} \text{ fails for any } (g'_1, ..., g'_{2n}) \times I_2 \mid (g'_1, ..., g'_{2n}) \leftarrow \gamma(g_1, ..., g_n)] < \frac{1}{21}$$

Then a $(\mathsf{BPAC}^0)^{\mathcal{R}}$ circuit can determine whether $\pi = g_1 ... g_n$ is the 3-cycle or identity.

Note that the following protocol is similar to that in Theorem 33 of [GS20], except we provide an error analysis that is crucial to the main average-case hardness results.

Proof. The $(\mathsf{BPAC}^0)^{\mathcal{R}}$ circuit proceeds in two phases. In the first phase, the circuit learns information about the distribution $\mathcal{D}_{\mathcal{R}}$. Then in the second phase, the circuit uses this information to deduce the value of π.

In further detail, the first phase of the circuit samples $O(\lg n)$ 3-qubit Paulis from $R_{I,\epsilon}$, where the oracle $\mathcal{R}$ fails with probability $\delta < \frac{1}{21}$ for first round input. With probability $1-\delta$, a sample comes from $R_{f,0} = \mathcal{D}_{\mathcal{R}}$. There is a 3-qubit Pauli $P \in S$ that attains the maximal weight $\mathcal{D}_{\mathcal{R}}(P) \geq \frac{1}{20}$. The Pauli P also has the maximal weight in $R_{I,\delta}$ because $\frac{1}{20}(1-\delta) \geq \delta$. By sampling $O(\lg n)$ Paulis from $R_{I,\delta}$, the circuit recovers P with high probability. Furthermore, by Lemma B.1, the circuit determines a $f \in H_3^{\oplus}$ such that $f \cdot P$ has no weight in $R_{f,0}$

The second phase of the circuit samples $O(\lg n)$ Paulis from $R_{f,\delta}$. With probability $1-\delta$, a sample will be from the distribution $R_{f,0} = (\pi f \pi^{-1}) \cdot \mathcal{D}_{\mathcal{R}}$. If $\pi = I$, then $R_{f,0} = f \cdot \mathcal{D}_{\mathcal{R}}$, so $f \cdot P$ has equal weight in $R_{f,0}$ as P has weight in $\mathcal{D}_{\mathcal{R}}$. If $\pi = C_3$, then $f \cdot P$ has no weight in $R_{f,0}$ by Lemma B.1. Due to the previous facts, we have the bound

$$R_{f,\delta:\pi=I}(f \cdot P) \geq (1-\delta)\mathcal{D}_{\mathcal{R}}(P) \geq \frac{1}{21} > \delta \geq R_{f,\delta:\pi=C_3}(f \cdot P)$$

so $R_{f,\delta:\pi=I}(f \cdot P)$ and $R_{f,\delta:\pi=C_3}(f \cdot P)$ are at least a constant difference apart. Thus, the circuit can distinguish $\pi \in \{C_3, I\}$ using the $O(\lg n)$ Pauli samples from $R_{f,\delta}$ with high probability by checking whether $f \cdot P$ appears in significantly more than a $1/21$ ratio of all samples. □

B.1 Applications to average-case ⊕L-hardness

We prove the second point of Lemma 4.2 regarding the algorithm F, and apply Lemma B.2 to F to prove the error-tolerance lemma in Proposition 5. We begin by describing F, which is partly composed of the following algorithm.

Lemma B.3. *(Lemma 44, [GS20]) There is a function E computed by an NC^0 circuit that takes as input an LDAG adjacency matrix A of dimension $n \times n$ and outputs a sequence of $g_1, \ldots, g_{\mathsf{poly}(n)} \in \mathrm{CNOT}_m$ such that*

1. *$A \in \mathsf{LDAGParity}$ iff $(g_1, \ldots, g_{\mathsf{poly}(n)}) \in \mathsf{CNOTMult}^*$*
2. *E is injective*

Let us define $F := \gamma \circ E$, where γ is the randomization performed in line 3 of Algorithm 1. As stated earlier, for any $g_1, \ldots, g_n \in \mathrm{CNOT}_m$, the output $(g'_1, \ldots, g'_{2n}) \leftarrow \gamma(g_1, \ldots, g_n)$ is uniformly random with the constraints: (1) $g'_i H_m = g_i H_m$ and $g'_{n+i} H_m = g_{n-i+1} H_m$ for $i \in [n]$ and (2) $g'_1 \ldots g'_{2n} \in H_3^{\oplus}$.

Lemma B.4. *For any distinct LDAG adjacency matrices A and A', $F(A)$ and $F(A')$ are uniform distributions over disjoint sets of the same cardinality.*

Proof. By the injectivity of E, we already have $E(A) \neq E(A')$, so constraint (1) above implies that the distributions $F(A)$ and $F(A')$ are disjoint. This is because for any distinct $g_i, g_j \in \mathrm{CNOT}_m$, we have that $g_i H_m \neq g_j H_m$. Furthermore, the supports of $F(A)$ and $F(A')$ have equal size because (1) and (2) are the only constraints. □

This proves the second point of Lemma 4.2. By the injectivity of E, we also immediately arrive at Proposition 5 by simply applying Lemma B.2.

C w-PBP-hardness

We prove the main w-PBP average-case hardness result, Theorem 4.5, in this section. Determining whether $g_1, ..., g_n \in \text{SWAP}_m$ multiplies to the 3-cycle or identity suffices to solve a w-PBP-hard problem, see Lemma 4.7. A crucial step in the remaining section involves the protocol and analysis of Appendix B, so we recommend reading that section before this section.

The main idea for the proof of w-PBP hardness 2-Round Graph State Measurement Problem on a w-Width grid is increasing the randomization power in the algorithm F, line 3 in Algorithm 1. Let us more precisely illustrate the reason that we need to strengthen F's randomization and provide a roadmap for how to do this.

By using F in the same way as Algorithm 1, the input to the oracle, $g_1, ..., g_n$, is uniformly randomized to $g'_1, ..., g'_{2n}$ where one of the constraints on $g'_1, ..., g'_{2n}$ is $g'_i H_m = g_i H_m$ and $g'_{n+i} H_m = g_{n-i+1} H_m$ for all $i \in [n]$. However, for any distinct sets of SWAPs, $(g_1^{(1)}, ..., g_n^{(1)}) \neq (g_1^{(2)}, ..., g_n^{(2)})$ as protocol input, the constraint implies that the supports of the input distributions to the oracle, $(g_1'^{(1)}, ..., g_{2n}'^{(1)})$ and $(g_1'^{(2)}, ..., g_{2n}'^{(2)})$, are disjoint. This is a problem because even if an exponentially small fraction of input to the oracle fails, a weaker assumption than a constant $\epsilon = \frac{1}{106}$ fails, then it could be the case that every input in the distribution of $(g_1'^{(1)}, ..., g_{2n}'^{(1)})$ fails for the oracle, so the procedure cannot reliably determine whether $(g_1^{(1)}, ..., g_n^{(1)})$ amounts to I or C_3.

The lesson is that we require a stronger form of randomization than present in F that successfully "hides" the protocol's SWAPs, $(g_1, ..., g_n)$, in the oracle input distribution. We achieve "hiding" the SWAPs by adding random SWAPs to H_m. However, sampling random swaps requires the ability to sample random permutations, and low-depth circuits appear unable to do this [MV91]. Instead, we use an *approximate* permutation sampling procedure.

Proposition 6. *Let $g_1, ..., g_n \in \langle \text{SWAP} \rangle_m$ be a sequence of permutations promised to be the 3-cycle or identity. Then the output of Kilian randomization, Line 3 in Algorithm 1, where $h_1, ..., h_{2n-1}$ are sampled uniformly from $H_m = \langle \text{SWAP}, \text{CZ}, S \rangle_m$, is uniform over $I_1(g_1, ..., g_n)$.*

Recall that $I_1(g_1, ..., g_n)$ is all $\hat{g_1}, ..., \hat{g_{2n}} \in H_m$ such that $\hat{g_1} ... \hat{g_{2n}} \in H_3^{\oplus}$. Notice that the constraint on the output of Kilian randomization is *independent* of $(g_1, ..., g_n)$, implying that we have successfully hidden the input to the oracle. Fortunately, we can *almost* achieve this in AC^0 by using known *approximate* permutation sampling techniques [MV91]. By using the approximate sampling of $\langle \text{SWAP} \rangle_m$, we essentially fix the second problem, up to some small error. Then we can "hide" the sets of SWAPs in our input distribution to the oracle, similarly to Proposition 6, which fixes the first problem. Toward this solution, let us consider the approximate uniform permutation sampling that we can achieve.

To quantify the distance between two distributions, we define the variation distance:

Definition C.1. *(Variation distance) Suppose two random variables, $\mathcal{X}$ and $\mathcal{Y}$, take on values over the same set. Then the variation distance between $\mathcal{X}$ and $\mathcal{Y}$ is defined to be*

$$\frac{1}{2} |\mathcal{X} - \mathcal{Y}|_1$$

where $|\mathcal{X} - \mathcal{Y}|_1$ is the 1-norm of entry-wise difference in distributions of $\mathcal{X}$ and $\mathcal{Y}$.

This allows us to quantify the following result:

Lemma C.1. *([MV91]) A randomized,* $\mathsf{poly}(n)$*-size* AC^0 *circuit can sample from a distribution that is* $\delta = 2^{-\mathsf{poly}(n)}$ *close in variation distance to the uniform distribution over permutations of* $[n]$*.*

Using this idea, we can approximately sample uniformly random elements from $H_m = \langle \mathrm{SWAP}, \mathrm{CZ}, \mathrm{S} \rangle_m$.

Proposition 7. *A randomized,* $\mathsf{poly}(m)$*-size* AC^0 *circuit can sample from a distribution,* J*, that is* $\delta = 2^{-\mathsf{poly}(m)}$ *close in variation distance to the uniform distribution over* H_m*.*

Proof. The circuit samples an element $s \in \langle \mathrm{SWAP} \rangle_m$ using Lemma C.1, and it composes it with a uniformly random element $d \in \langle \mathrm{CZ}, \mathrm{S} \rangle_m$. The $s \in \langle \mathrm{SWAP} \rangle_m$ is $2^{-\mathsf{poly}(m)}$ close in variation distance to uniform, and d is uniform, so the distribution over sd, J, must be $2^{-\mathsf{poly}(m)}$ close to uniform. □

When randomizing the input $(g_1, ..., g_n)$, the circuit performs Kilian randomization by sampling elements of H_m from Proposition 7 instead of the uniform distribution over H_m. We want to argue that the oracle input distribution is essentially uniform, up to some small error.

Lemma C.2. *Let* $g_1, ..., g_n \in \langle \mathrm{SWAP} \rangle_m$ *be a sequence of permutations promised to be the 3-cycle or identity. Then the output of Kilian randomization, Line 3 in Algorithm 1, where* $h_1, ..., h_{2n-1}$ *are sampled by Proposition 7, is* $2^{-\mathsf{poly}(n)}$ *close in variation distance to the uniform distribution over* $I_1(g_1, ..., g_n)$*.*

Proof. The uniform distribution over $\hat{g_1}, ..., \hat{g_{2n}} \in H_m$ such that $\hat{g_1} ... \hat{g_{2n}} \in H_3^{\oplus}$ is produced by fixing any $f \in H_3^{\oplus}$, sampling $f' \leftarrow U_{H_3^{\oplus}}$ and $h_1, ... h_{2n-1} \leftarrow U_{H_m}$ to produce

$$(\hat{g_1}, ..., \hat{g_{2n}}) = (f' g_1 h_1, h_1^{-1} g_2 h_2, ..., h_{n-1}^{-1} g_n h_n, h_n^{-1} f g_n h_{n+1}, h_{n+1}^{-1} g_{n-1} h_{n+2}, ..., h_{2n-1}^{-1} g_1) \tag{18}$$

It suffices to show that the variation distance between the distribution over $(\hat{g_1}, ..., \hat{g_{2n}})$ in Equation (18) and $\mathcal{D}_{\mathcal{I}}$ is at most $2^{-\mathsf{poly}(n)}$. The distribution $\mathcal{D}_{\mathcal{I}}$ is created by sampling $f' \leftarrow U_{H_3^{\oplus}}$ and $j_1, ... j_{2n-1} \leftarrow J$ from Proposition 7, and multiplying to get

$$(g'_1, ..., g'_{2n}) = (f' g_1 j_1, j_1^{-1} g_2 j_2, ..., j_{n-1}^{-1} g_n j_n, j_n^{-1} f g_n j_{n+1}, j_{n+1}^{-1} g_{n-1} j_{n+2}, ..., j_{2n-1}^{-1} g_1) \tag{19}$$

To bound the distance between Equation (18) and Equation (19), it suffices to bound the distance between

$$(g_1 h_1, h_1^{-1} g_2 h_2, ..., h_{n-1}^{-1} g_n h_n, h_n^{-1} g_n h_{n+1}, h_{n+1}^{-1} g_{n-1} h_{n+2}, ..., h_{2n-2}^{-1} g_2 h_{2n-1}) \tag{20}$$

and

$$(g_1 j_1, j_1^{-1} g_2 j_2, ..., j_{n-1}^{-1} g_n j_n, j_n^{-1} g_n j_{n+1}, j_{n+1}^{-1} g_{n-1} j_{n+2}, ..., j_{2n-2}^{-1} g_2 j_{2n-1}) \tag{21}$$

because f' is uniform, f permutes both distributions in the same way, and the final entries in both distributions, $h_{2n-1}^{-1} g_1$ and $j_{2n-1}^{-1} g_1$, are fixed after conditioning on the values of the other entries. In Equation (20), an entry $h_i g_j h_{i+1}$ has the property that $h_i g_j$ is fixed after conditioning on the entries to the left, and a similar property holds for Equation (21). Therefore, it also suffices to bound the distance between

$$(h_1, ..., h_{2n-1}) \tag{22}$$

and

$$(j_1, ..., j_{2n-1}) \tag{23}$$

All h_i are independent from one another (similarly for j_i), and the distance between h_i and j_i is δ. In this case, variation distance is subadditive, so the distance between Equation (22) and Equation (23) is at most $(2n-1)\delta$. Noticing that $\delta \leq 2^{-\mathsf{poly}(n)}$, we get the desired bound. □

C.1 Proof of average-case w-PBP-hardness

We are now ready to prove the main theorem of this section. The problem of deciding whether $\pi = g_1, ..., g_n \in \langle \mathrm{SWAP} \rangle_m$ multiplies to the 3-cycle or identity is w-PBP-hard due to Lemma 4.7.

If $\mathcal{R}$ fails for $\frac{1}{106}$ of all input, a constant smaller than $\frac{1}{105}$, then it fails for $\leq \frac{5}{106} < \frac{1}{21}$ of first round input because $|I_2| = 5$. Because first round input is sampled from $\mathcal{D}_{\mathcal{I}}$ which is $2^{-\mathsf{poly}(n)}$-close to $U_{I_1(g_1,...,g_n)}$ by Lemma C.2, then the circuit samples first round input such that

$$\Pr[\mathcal{R} \text{ fails for any } (g'_1, ..., g'_{2n}) \times I_2 \mid (g'_1, ..., g'_{2n}) \leftarrow \mathcal{D}_{\mathcal{I}}] < \frac{5}{106} + 2^{-\mathsf{poly}(n)} < \frac{1}{21}$$

Combining this with Lemma B.2, the circuit determines π with high probability.

It remains to check that the input to $\mathcal{R}$ can be made classically-controlled (CC); ideally, each gate that $\mathcal{R}$ simulates depends on $O(1)$ input bits to facilitate reduction to the 2-Round Graph State Measurement problem. The input given to the oracle in I_2 can easily be converted to such a form in NC^0. To convert an element of I_1 to a CC form, we first notice that any $g \in G_m$ can be written as an element of $\langle \mathrm{SWAP} \rangle_m \langle \mathrm{CZ}, \mathrm{S} \rangle_m$. The gates CZ and S are diagonal, so they commute with one another, and $\langle \mathrm{CZ}, \mathrm{S} \rangle_m$ can be made CC in a straightforward way from this fact.

Now we turn to converting a permutation $\sigma = (\sigma(1)...\sigma(m)) \in \langle \mathrm{SWAP} \rangle_m$ to CC form. Any series of local transpositions that sorts $(\sigma(1)...\sigma(m))$ also creates the permutation σ, so it suffices to sort $(\sigma(1)...\sigma(m))$ to create a CC circuit of SWAPs. Using threshold gates, a TC^0 circuit can determine the value of

$$|\{j < i \,:\, \sigma(j) > \sigma(i)\}|$$

for all i in parallel. In fact, this is the only comparison needed to create a CC circuit of SWAPs that performs insertion sort on $(\sigma(1)...\sigma(m))$. Therefore, the TC^0 circuit converts $\sigma \in \langle \mathrm{SWAP} \rangle_m$ to a CC circuit. This proves the claim in Theorem 4.5.