Exponential Separation between Shallow Quantum Circuits and Unbounded Fan-In Shallow Classical Circuits

Adam Bene Watts*
Massachusetts Institute of Technology
Cambridge, MA, United States
abenewat@mit.edu

Robin Kothari
Microsoft Research
Quantum Architectures and Computation group (QuArC)
Redmond, WA, United States
robin.kothari@microsoft.com

Luke Schaeffer†
Massachusetts Institute of Technology
Cambridge, MA, United States
lrs@mit.edu

Avishay Tal‡
Stanford University
Stanford, CA, United States
avishay.tal@gmail.com

ABSTRACT

Recently, Bravyi, Gosset, and König (Science, 2018) exhibited a search problem called the 2D Hidden Linear Function (2D HLF) problem that can be solved exactly by a constant-depth quantum circuit using bounded fan-in gates (or QNC^0 circuits), but cannot be solved by any constant-depth classical circuit using bounded fan-in AND, OR, and NOT gates (or NC^0 circuits). In other words, they exhibited a search problem in QNC^0 that is not in NC^0.

We strengthen their result by proving that the 2D HLF problem is not contained in AC^0, the class of classical, polynomial-size, constant-depth circuits over the gate set of *unbounded* fan-in AND and OR gates, and NOT gates. We also supplement this worst-case lower bound with an average-case result: There exists a simple distribution under which any AC^0 circuit (even of nearly exponential size) has exponentially small correlation with the 2D HLF problem. Our results are shown by constructing a new problem in QNC^0, which we call the Parity Halving Problem, which is easier to work with. We prove our AC^0 lower bounds for this problem, and then show that it reduces to the 2D HLF problem.

CCS CONCEPTS

• **Theory of computation → Quantum complexity theory**.

KEYWORDS

Quantum circuits, Low-depth circuits, Switching lemma, Non-local games

*ABW was supported by NSF grant CCF-1729369.

†Part of this work was done while the author was an intern in the QuArC group at Microsoft Research.

‡This work was done in part while the author was visiting the Simons Institute for the Theory of Computing. Partially supported by a Motwani Postdoctoral Fellowship and by NSF grant CCF-1763311.

STOC '19, June 23–26, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6705-9/19/06…$15.00
https://doi.org/10.1145/3313276.3316404

ACM Reference Format:
Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. 2019. Exponential Separation between Shallow Quantum Circuits and Unbounded Fan-In Shallow Classical Circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on the Theory of Computing (STOC '19), June 23–26, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3313276.3316404

1 INTRODUCTION

One of the basic goals of quantum computing research is to identify problems that quantum computers can solve more efficiently than classical computers. We now know several such problems, such as the integer factorization problem, which we believe can be solved exponentially faster on a quantum computer [23]. However, running this algorithm requires a large general-purpose quantum computer, which we do not yet have. Hence it is interesting to find examples of quantum speedup using weaker models of quantum computation, such as models with limited space or time, limited gate sets, or limited geometry of interactions.

Shallow Quantum Circuits. One such model of quantum computation that has been studied for over 20 years is the class of shallow or constant-depth quantum circuits [8, 9, 11, 15, 18, 19, 24, 26]. Such circuits may be viewed as parallel quantum computers with a constant running time bound. Several variations on this theme have been studied (see [2] for a survey of older results), and in recent years there has been a resurgence of interest [3, 5, 7, 16, 25] in constant-depth quantum circuits, for at least two reasons.

First, shallow quantum circuits are well motivated from a practical perspective, as we might actually be able to implement such circuits on near-term quantum computers! In the current era of Noisy Intermediate-Scale Quantum (NISQ) computers, due to high error rates of quantum gates, we are limited to running quantum algorithms for a short amount of time before errors accumulate and noise overwhelms the signal. Hence we seek interesting problems that can still be implemented by limited quantum hardware.

Second, constant-depth circuits (either classical or quantum) are very interesting to theoretical computer scientists, as it is possible to prove unconditional impossibility results. For example, while we strongly believe that the factoring problem mentioned above requires exponential time on a classical computer, we cannot prove

this. On the other hand, many of the early successes of complexity theory involved exhibiting explicit functions that could not be computed by constant-depth classical circuits [1, 10, 13, 29]. Indeed, constant-depth circuits remain the frontier of circuit lower bounds and an active area of research in classical complexity theory today [20, 28].

This motivates the search for problems that can be solved by constant-depth quantum circuits, while being hard for constant-depth (or even more powerful) classical circuits.

Prior Work. While there has been prior work on establishing the power of shallow quantum circuits assuming complexity theoretic conjectures [3, 26], this work is not directly related to our work as we prove unconditional lower bounds.

In this realm the most relevant result is the recent exciting result of Bravyi, Gosset, and König [5], who defined a search or relational problem[1] called the 2D Hidden Linear Function (2D HLF) problem. (We define this problem in Section 4.) The 2D HLF problem can be solved by a constant-depth quantum circuit that uses bounded fan-in quantum gates. Indeed, the quantum circuit solving 2D HLF can be implemented on a 2-dimensional grid of qubits with spatially local quantum gates.

Furthermore, Bravyi, Gosset, and König [5] show that the 2D HLF problem cannot be solved by any constant-depth classical circuit using unbounded fan-out and bounded fan-in gates. Their lower bound even holds when the classical circuit is allowed to sample from an arbitrary probability distribution on polynomially many bits that does not depend on the input. (In complexity theory, this resource is called "randomized advice.") More formally, the class of classical circuits of polynomial-size, constant-depth, unbounded fan-out, and bounded fan-in gates is called NC^0.[2] An NC^0 circuit with the additional ability to sample from any probability distribution on polynomially many bits that is independent of the input, but that can depend on the input size, is called an $\mathsf{NC}^0/\mathrm{rpoly}$ circuit. The class of polynomial-size, constant-depth quantum circuits with bounded fan-in gates is called QNC^0. Note that because quantum gates have the same number of inputs and outputs, QNC^0 circuits also have bounded fan-out, unlike classical NC^0 circuits, which have unbounded fan-out.

With this notation, we can now summarize the Bravyi et al. result as follows [5].

Theorem (Bravyi, Gosset, and König). *The* 2D HLF *problem can be solved exactly by a* QNC^0 *circuit on a 2D grid, but no* $\mathsf{NC}^0/\mathrm{rpoly}$ *circuit can solve the problem with probability greater than* $7/8$ *on every input.*

The fact that the separating problem in [5] is a search problem and not a function (or decision) problem is unavoidable, since any function in QNC^0 has output bits that only depend on a constant number of input bits, due to the bounded fan-in gates, and hence such a function would also be in NC^0.

This result was also recently improved by Coudron, Stark, and Vidick [7], and (independently) Le Gall [16], who extended the lower bound to an average-case lower bound. As opposed to saying that no NC^0 circuit can solve the problem on *all* inputs, an average-case hardness result says that no NC^0 circuit can solve the problem even on some fraction of the inputs.[3] These results show that no NC^0 circuit can solve the problem with input size n on an $\exp(-n^\alpha)$ fraction of the inputs for some $\alpha > 0$.

Main Result. In this work, we strengthen these results and prove a strong average-case lower bound for the 2D HLF problem against the class AC^0. AC^0 is a natural and well-studied class that generalizes NC^0 by allowing the circuit to use unbounded fan-in AND and OR gates. Note that $\mathsf{NC}^0 \subsetneq \mathsf{AC}^0$ because AC^0 can compute functions that depend on all bits, such as the logical OR of all its inputs, whereas NC^0 cannot. Our main result is the following.

Theorem 1. *The* 2D HLF *problem on n bits cannot be solved by an* AC^0 *circuit of depth d and size at most* $\exp(n^{1/2d})$. *Furthermore, there exists an (efficiently sampleable) input distribution on which any* AC^0 *circuit (or* $\mathsf{AC}^0/\mathrm{rpoly}$ *circuit) of depth d and size at most* $\exp(n^{1/2d})$ *only solves the* 2D HLF *problem with probability at most* $\exp(-n^\alpha)$ *for some* $\alpha > 0$.

Thus our result proves a separation against a larger complexity class and implies the worst-case lower bound of Bravyi, Gosset, and König [5]. It also implies the average-case lower bounds of Coudron, Stark, and Vidick [7] and Le Gall [16].

1.1 High-Level Overview of the Main Result

We now describe the problems we study en route to proving Theorem 1 and give a high-level overview of the proof.

Theorem 1 is proved via a sequence of increasingly stronger results. We first introduce a problem we call the Parity Halving Problem (PHP). PHP is not in QNC^0, but it can be solved exactly by a $\mathsf{QNC}^0/\mathrm{qpoly}$ circuit, which is a QNC^0 circuit with quantum advice. Similar to randomized advice, a circuit class with quantum advice is allowed to start with any polynomial-size quantum state that is independent of the input, but can depend on the input length. For the Parity Halving Problem (and other problems introduced later), the quantum advice state is a very simple state called the cat state, which we denote by $|😸_n\rangle := \frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle)$. We denote the subclass of $\mathsf{QNC}^0/\mathrm{qpoly}$ where the advice state is the cat state $\mathsf{QNC}^0/😸$.

Here's a bird's eye view of our proof: Our first result establishes that PHP is in $\mathsf{QNC}^0/😸$, but any nearly exponential-size AC^0 circuit only solves the problem with probability exponentially close to $1/2$. Next we define a new problem called the Relaxed Parity Halving Problem on a grid (Grid-RPHP), which is indeed in QNC^0, but any nearly exponential-size AC^0 circuit only solves the problem with probability exponentially close to $1/2$. We then define parallel versions of these two problems, which we call Parallel-PHP and Parallel Grid-RPHP. We show that Parallel-PHP $\in \mathsf{QNC}^0/\mathrm{qpoly}$

[1] A search or relational problem can have many valid outputs for a given input, unlike a function problem that has exactly one valid output. A decision problem is a function problem with a 1-bit output.

[2] In this paper, we will employ a common abuse of notation and use class names like NC^0 and AC^0 to generally talk about a type of circuit, as opposed to decision problems solved by such circuits. Hence, for example, we speak of "decision problems in AC^0" and "search problems in AC^0" although formally AC^0 would be the class of decision problems solved by such circuits, and FAC^0 would be the class of search problems solved by such circuits.

[3] Note that [5, Appendix C.3] already shows mild average-case hardness for this problem.

and Parallel Grid-RPHP $\in \mathrm{QNC}^0$, but any nearly exponential-size AC^0 circuit only solves these problems with exponentially small probability.[4] Finally we show that Parallel Grid-RPHP can be reduced to 2D HLF, and hence our lower bound applies to 2D HLF as well. We now describe these problems and our proof techniques in more detail.

Parity Halving Problem. In the Parity Halving Problem on n bits, which we denote by PHP_n, we are given an input string $x \in \{0,1\}^n$ promised to have even parity: i.e., the Hamming weight of x, denoted $|x|$, satisfies $|x| \equiv 0 \pmod 2$. The goal is to output a string $y \in \{0,1\}^n$ that satisfies

$$|y| \equiv |x|/2 \pmod 2. \tag{1}$$

In other words, the output string's Hamming weight (mod 2) is half of that of the input string. Note that $|x|/2$ is well defined above because $|x|$ is promised to be even. An alternate way of expressing this condition is that $|y| \equiv 0 \pmod 2$ if $|x| \equiv 0 \pmod 4$ and $|y| \equiv 1 \pmod 2$ if $|x| \equiv 2 \pmod 4$.

We show in Section 2 that PHP can be solved with certainty on every input by a simple depth-2 $\mathrm{QNC}^0/$🐱 circuit. A quantum circuit solving PHP on 3 bits is shown in Figure 1. The circuit has one layer of controlled phase gates followed by Hadamard gates on the output qubits, followed by measurement.

Although the problem is easy for constant-depth quantum circuits, we show that even an exponential-size $\mathrm{AC}^0/\mathrm{rpoly}$ circuit cannot solve the problem on the uniform distribution (over valid inputs) with probability considerably better than $1/2$, which is trivially achieved by the circuit that outputs the all-zeros string on all inputs.

Theorem 2 (PHP). *The Parity Halving Problem* (PHP_n) *can be solved exactly by a* $\mathrm{QNC}^0/$🐱 *circuit. But on the uniform distribution over all valid inputs (even parity strings), any* $\mathrm{AC}^0/\mathrm{rpoly}$ *circuit of depth* d *and size at most* $\exp\left(n^{\frac{1}{2d}}\right)$ *only solves the problem with probability* $\frac{1}{2} + \exp(-n^{\alpha})$ *for some* $\alpha > 0$.

Note that the parameters of the AC^0 lower bound in this theorem are essentially optimal, since the parity function on n bits can be computed by a depth-d AC^0 circuit of size $\exp\left(n^{\frac{1}{d-1}}\right)$ [13, Theorem 2.2]. Once we can compute the parity of the input bits, it is easy to solve PHP.

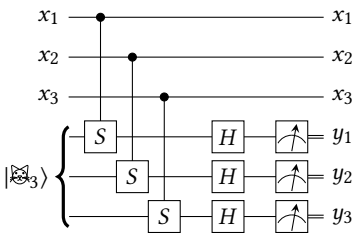

Figure 1: Quantum circuit for the Parity Halving Problem on 3 bits, PHP_3**.**

[4] This proof is only found in the full version of the paper.

Since the quantum circuit for PHP is simple, it is clear that the difficult part of Theorem 2 is the AC^0 lower bound. One reason for this difficulty is that if we allowed the output string y in PHP to be of quadratic size, then there is a simple depth-1 NC^0 circuit that solves this problem! The circuit simply takes the AND of every pair of input bits and outputs this string of size $\binom{n}{2}$. A simple calculation shows that the Hamming weight of this string will be $\binom{|x|}{2}$, which satisfies the conditions of the problem. Hence to prove the AC^0 lower bound, the technique used has to be sensitive to the output size of the problem. However, traditional AC^0 lower bound techniques were developed for decision problems, and do not explicitly take the output size of the problem into account. Hence we modify some known techniques and establish the this lower bound in three steps.

First, we use Håstad's switching lemmas [13, 14], or more precisely a recent refinement of it due to Rossman [22]. However, directly using the result of Rossman off the shelf gives us a weaker result than Theorem 2; we are only able to establish the theorem with a quasi-polynomially small correlation instead of the exponentially small correlation in Theorem 2. To obtain the result we want, we refine Rossman's result to work better for multi-output functions (Lemma 12). This result is quite technical, but the conceptual ideas already appear in the works of Håstad [14] and Rossman [22]. Applying this switching lemma reduces the problem of proving an average-case AC^0 lower bound to that of showing an average-case NC^0 lower bound for a modified version of the Parity Halving Problem. This modified version of the problem is similar to PHP, except it has n inputs and slightly more (say, $n^{1.01}$) outputs.

Our second step is to use a combinatorial argument to reduce this question to showing an average-case lower bound against NC^0 circuits with locality 1 (i.e., where each output only depends on a single input) for a further modified version of PHP.

The third and final step is to show that NC^0 circuits with locality 1 cannot solve this modifed PHP on a random input. We prove this by generalizing known lower bounds in the literature on quantum non-local games. Specifically we generalize lower bounds present in the work of Mermin [17], and Brassard, Broadbent, and Tapp [4].

This proof is presented in Section 2. We first prove the lower bound against NC^0 circuits of locality 1 in Section 2.2, then show the lower bound against general NC^0 circuits in Section 2.3, and finally prove the switching lemma and conclude the proof of Theorem 2 in Section 2.4.

Now Theorem 2 is weaker than what we want (Theorem 1) in two ways. Aside from the fact that the lower bound is for a problem different from the 2D HLF problem, the problem in Theorem 2 is in $\mathrm{QNC}^0/$🐱 and not QNC^0, and the correlation lower bound is close to $1/2$ instead of being exponentially small. We now tackle the first problem and get rid of the cat state.

Relaxed Parity Halving Problem. Since the cat state cannot be constructed in QNC^0 (proved in Theorem 14), we have to modify the Parity Halving Problem to get by without a cat state.

Although we cannot create the cat state in QNC^0, we can construct a state we call a "poor man's cat state," which is the state

$$\frac{1}{\sqrt{2}}\left(|z\rangle + |\bar{z}\rangle\right), \tag{2}$$

where $z \in \{0,1\}^n$ is a bit string and $\bar{z}$ denotes its complement. When $z = 0^n$, this is indeed the cat state, but in general this is some entangled state that can be converted to the cat state by applying the X gate to some subset of the qubits.

Interestingly, we can create a poor man's cat state in QNC^0 for a uniformly random z. Here is one simple construction. First arrange the n qubits on a line and set them all to be in the $|+\rangle$ state. Then in a separate set of $n-1$ qubits, compute the pairwise parities of adjacent qubits. In other words, we store the parity of qubit 1 and 2, 2 and 3, 3 and 4, and so on until qubit $n-1$ and n. And then we measure these $n-1$ qubits, and denote the measurement outcomes $d_1, \ldots, d_{n-1}$, which we will call the "difference string." It is easy to verify that if all the $d_i = 0$, then the resulting state is indeed the cat state. In general, for any $d \in \{0,1\}^{n-1}$, the resulting state is a poor man's cat state, and z can be determined from the string d up to the symmetry between z and $\bar{z}$. Since z and $\bar{z}$ are symmetric in the definition of the poor man's cat state, let us choose the convention that $z_1 = 0$. Now we can determine the remaining z_i from d using the fact that $d_1 = z_1 \oplus z_2 = z_2$, $d_2 = z_2 \oplus z_3$, and so on until $d_{n-1} = z_{n-1} \oplus z_n$. Note that because of this construction, some z_i depend on many bits of d_i. For example, z_n is the parity of all the bits in d.

This construction of the poor man's cat state easily generalizes to graphs other than the 1D line. We could place the n qubits on a balanced binary tree and measure the parity of all adjacent qubits, and hence get one d_i for every edge in the tree. If we call the root node $z_1 = 0$, then the value of any z_i will be the parity of all d_i on edges between vertex i and the root. In this case each z_i depends on at most $\log n$ bits of d_i. Similarly, we can choose a 2D grid instead of a balanced binary tree, and set the top left qubit to be $z_1 = 0$. Then each z_i will depend on at most $2\sqrt{n}$ bits of d. This grid construction is described more formally in Section 3.1.

Now there's an obvious strategy to try: Simply use a poor man's cat in our quantum circuit for PHP instead of using an actual cat state, and redefine the problem to match the output of this quantum circuit! So we simply run the circuit in Figure 1 on a poor man's cat state $\frac{1}{\sqrt{2}}(|z\rangle + |\bar{z}\rangle)$ and see what the quantum circuit outputs. Unfortunately the output depends on z, but the poor man's cat state has been destroyed by the circuit and we do not have a copy of z around. But we do still have the string d from which it is possible to recover z, although this may not be computationally easy since a single bit of z may depend on a large number of bits of d. More subtly, a single bit of d may be involved in specifying many bits of z, which is also a complication for circuits without fan-out. Instead of trying to recover z, we can just modify the problem to include d as an output. The problem will now have two outputs, one original output y, and a second output string d, which is the difference string of the z in the poor man's cat state. This is the Relaxed Parity Halving Problem, which is more formally defined in Section 3.2.

More precisely, the Relaxed Parity Halving Problem, or RPHP, depends on the choice of the underlying graph, and is well defined for any graph. We choose the 2D grid to get a problem that reduces to 2D HLF.[5] We call this problem Grid-RPHP.

We show in Section 3.2 that Grid-RPHP can be solved by the 2D QNC^0 circuit we described, but even a nearly exponentially large AC^0 circuit cannot solve the problem with probability significantly larger than $1/2$ on the uniform distribution over valid inputs.

Theorem 3 (Grid-RPHP). *Grid-RPHP$_n$ can be solved exactly by a QNC^0 circuit on a 2D grid. But on the uniform distribution over all valid inputs (even parity strings), any AC^0 circuit (or AC^0/rpoly circuit) of depth d and size at most $\exp(n^{1/2d})$ can solve the problem with probability at most $\frac{1}{2} + \exp(-n^{\alpha})$ for some $\alpha > 0$.*

Note that just like Theorem 2, the lower bound here is essentially optimal, since the parity function itself can be computed by a depth-d AC^0 circuit of size $\exp\left(n^{\frac{1}{d-1}}\right)$ [13, Theorem 2.2].

Our separation essentially works for any graph with sublinear diameter, such as the grid or the balanced binary tree, but not the 1D line. In fact, when the underlying graph is the 1D line, RPHP becomes easy to solve, even for NC^0 circuits.[6]

We prove Theorem 3 by showing a reduction from the Parity Halving Problem with input size n and output size $O(n^{3/2})$ to Grid-RPHP. This version of PHP is indeed hard for AC^0 circuits and this result follows from the work done in Section 2. This reduction and theorem are proved formally in Section 3.2.

Now Theorem 3 is still weaker than what we want (Theorem 1). The correlation lower bound is still close to $1/2$ and not exponentially small. We fix this issue using a simple idea.

Parallel Grid-RPHP. Let Parallel Grid-RPHP be the problem where we are given many instances of Grid-RPHP in parallel and are required to solve all of them correctly. For this problem the quantum circuit is obvious: Simply use the quantum circuit for Grid-RPHP for each instance of the problem. Clearly if the quantum circuit solves each instance correctly, it solves all of them correctly. But since a classical circuit only solves an instance with some probability close to $1/2$, we expect that solving many copies of the problem gets much harder.

Theorem 4 (Parallel Grid-RPHP). *Parallel Grid-RPHP$_n$ can be solved exactly by a QNC^0 circuit on a 2D grid. But on the uniform distribution over all valid inputs (even parity strings for each instance of Grid-RPHP), any AC^0 circuit (or AC^0/rpoly circuit) of depth d and size at most $\exp(n^{1/2d})$ can solve the problem with probability at most $\exp(-n^{\alpha})$ for some $\alpha > 0$.*

This proof of Theorem 4 is left to the full version of this paper. Note Theorem 4 looks very similar to Theorem 1, except that the hardness is shown for Parallel Grid-RPHP and not the 2D HLF problem.

Reduction to the Hidden Linear Function problem. The final step of our program is carried out in Section 4. First we show in Theorem 19 that the Relaxed Parity Halving Problem (for any graph G) can be reduced to the Hidden Linear Function problem (not necessarily the 2D HLF). In particular, our proof shows that Grid-RPHP reduces to the 2D HLF problem, which we prove in Corollary 21. This reduction along with Theorem 4 implies Theorem 1.

[5] Picking the balanced binary tree would give better parameters, but qualitatively similar results. We choose the 2D grid to obtain .

[6] One can output $y = 0^n$ and $d_i = x_i$ for all $i \in \{1, \ldots, n-1\}$ to solve the Relaxed Parity Halving Problem on the 1D line.

1.2 Additional Results

We also consider the question of showing a separation between QNC^0 and $\mathrm{AC}^0[2]$, where $\mathrm{AC}^0[2]$ is AC^0 with unbounded fan-in XOR gates. We implement the first two steps of the strategy above, where we come up with a problem in $\mathrm{QNC}^0/\boxtimes$ that cannot be solved by an $\mathrm{AC}^0[2]$ circuit, even on a $o(1)$ fraction of the inputs. But we do not know how to remove the reliance on the cat state in this setting. These results are left for the full version of this paper.

1.3 Discussion and Open Problems

Our main results show that there is a search problem (either the 2D HLF problem or the Parallel Grid-RPHP) in QNC^0 that is not in AC^0, and that there is a search problem (Parallel PBP) in $\mathrm{QNC}^0/\boxtimes$ that is not in $\mathrm{AC}^0[2]$. One open problem is to generalize both separations and show that there is a search problem in QNC^0 that is not in $\mathrm{AC}^0[2]$, or more generally $\mathrm{AC}^0[p]$ for any prime p. This is essentially the frontier of circuit lower bounds, and it will be difficult to go further without radically new techniques.

One could try to achieve a quantum advantage using even weaker classes than QNC^0 or classes incomparable to QNC^0. The recent result of Raz and Tal [21] exhibits a decision problem in BQLOGTIME (bounded-error quantum logarithmic time) that is not in AC^0. Note that as classes of search problems, BQLOGTIME and QNC^0 are incomparable, since both can solve search problems the other cannot.

It is also interesting to note that all the separations using QNC^0 that make the circuit geometrically local use 2 dimensions. Is there a search problem that can be implemented by a 1D QNC^0 circuit that is not in NC^0 or even AC^0. On the flip side, perhaps 1D QNC^0 is not very powerful and can be classically simulated. On that note, can we upper bound the power of QNC^0 in terms of some classical complexity class that does not also upper bound the power of BQP?

2 PARITY HALVING PROBLEM

Recall the Parity Halving Problem from the introduction. We now define a more general version of the problem with n input bits and m output bits.

Problem 1 (Parity Halving Problem, $\mathrm{PHP}_{n,m}$)**.** Given an input $x \in \{0,1\}^n$ of even parity, output a string $y \in \{0,1\}^m$ such that

$$|y| \equiv \frac{1}{2}|x| \pmod 2. \tag{3}$$

Alternately, y must have even parity if $|x| \equiv 0 \pmod 4$ and odd parity if $|x| \equiv 2 \pmod 4$. We also define PHP_n to be $\mathrm{PHP}_{n,n}$.

The main result of this section is to show this problem is in $\mathrm{QNC}^0/\boxtimes$, but not $\mathrm{AC}^0/\mathrm{rpoly}$. We now restate this result (Theorem 2) more formally:

Theorem 2 (formal). *The Parity Halving Problem* (PHP_n) *can be solved exactly by a depth-2, linear-size quantum circuit starting with the* $|\boxtimes_n\rangle$ *state. But on the uniform distribution over all valid inputs (even parity strings), any* $\mathrm{AC}^0/\mathrm{rpoly}$ *circuit of depth* d *and size* $s \leq \exp\left(n^{\frac{1}{2d}}\right)$ *only solves the problem with probability* $\frac{1}{2} + \exp(-n^{1-o(1)}/O(\log s)^{2(d-1)})$.

We prove this theorem in several parts. First we prove the quantum upper bound in Section 2.1 (Theorem 5). The lower bound on AC^0 circuits via a sequence of incrementally stronger lower bounds, culminating in the claimed lower bound. We start in Section 2.2 by showing a lower bound (Theorem 6) for a very simple class of circuits, NC^0 circuits of locality 1, i.e., NC^0 circuits where every output is an arbitrary function of exactly one input bit. We then extend the lower bound to arbitrary NC^0 circuits in Section 2.3 (Theorem 8), and to AC^0 circuits in Section 2.4 culminating in the AC^0 lower bound for $\mathrm{PHP}_{n,m}$ in Theorem 13, from which the lower bound in Theorem 2 follows straightforwardly by setting $m = n$.

2.1 Quantum Upper Bound

Before we get into the details of the proof, let us motivate the problem. Observe that the problem naturally defines an interesting n-player cooperative non-local game, which we call the Parity Halving Game. In this game, there are n players, and each player gets one of the n input bits and outputs a single bit, with no communication with the other players. The input and output conditions are the same as in PHP_n: The input is promised to be of even Hamming weight, and the players win the game if their output's parity satisfies the condition in Problem 1.

Because the players are not allowed to communicate, the strategies permitted in the non-local game are far more restricted than an AC^0 circuit or even an NC^0 circuit for PHP_n since each output bit is only allowed to depend on one input bit. We will call this model NC^0 with locality 1.

Now that we have defined a game, we can study the probability of success for classical players versus the probability of success for quantum players who share entanglement before the game begins. In fact, when $n = 3$, the Parity Halving Game coincides with the well-known Greenberger-Horne-Zeilinger (GHZ) game [12]. It is known that quantum players sharing entanglement, and specifically the state $|\boxtimes_3\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$, can always win the GHZ game with certainty, but classical players can win the GHZ game with probability at most $3/4$.

This n-player generalization of the GHZ game is very natural and quantum players can win the Parity Halving Game exactly using a $|\boxtimes_n\rangle$ state. This game has been studied before, and we are aware of two other works that analyze this game: the first by Mermin [17], and the second by Brassard, Broadbent, and Tapp [4]. Both papers exhibit the quantum strategy that wins perfectly and argue that classical strategies fail (as we do in the next section).

The strategy for winning the 3-player GHZ game generalizes to yield a perfect strategy for winning the n-player game as well, which yields a depth-2 linear-size quantum circuit for PHP_n. We now describe the quantum strategy and the corresponding constant-depth quantum circuit.

Theorem 5 (Quantum circuit for PHP_n). *The Parity Halving Problem* (PHP_n) *can be solved exactly by a depth-2, linear-size quantum circuit starting with the* $|\boxtimes_n\rangle$ *state.*

Proof. We describe this circuit in the language of the n-player Parity Halving Game described above. The circuit is depicted in Figure 1 (on page 3). Let the input to the i^{th} player in the Parity Halving Game be called x_i, and their output be called y_i. In our protocol, the players will share an n-qubit cat state $|\boxtimes_n\rangle = \frac{1}{\sqrt{2}}(|0^n\rangle + |1^n\rangle)$, and each player receives one qubit of the cat state at the beginning.

Each player starts by applying a phase gate, $S = \left(\begin{smallmatrix} 1 & 0 \\ 0 & i \end{smallmatrix}\right)$, to their qubit of the cat state if their input bit is 1. If their input bit is 0, they do nothing. In other words, the player applies a control-S gate with x_i as the source and their qubit of the cat state as the target. After this step, the cat state has been transformed to

$$\frac{1}{\sqrt{2}}\left(|0^n\rangle + i^{|x|}\,|1^n\rangle\right). \tag{4}$$

But since x has even parity, this state is either $|\boxtimes_n\rangle$ or the "minus cat state" $\frac{1}{\sqrt{2}}(|0^n\rangle - |1^n\rangle)$. We will denote this state by $Z\,|\boxtimes_n\rangle$ since this is the state one obtains by applying the $Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ gate to any one qubit of the cat state. When $|x| \equiv 0 \pmod 4$, this state will be $|\boxtimes_n\rangle$ and when $|x| \equiv 2 \pmod 4$, this will be $Z\,|\boxtimes_n\rangle$. Note that $|\boxtimes_n\rangle$ and $Z\,|\boxtimes_n\rangle$ are orthogonal states.

Finally, each player applies the Hadamard gate $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ to their qubit of the cat state, measures the qubit, and outputs that as y_i. The operator $H^{\otimes n}$ maps the cat state $|\boxtimes_n\rangle$ to a uniform superposition over even parity strings, and maps $Z|\boxtimes_n\rangle$ to a uniform superposition over odd parity strings. This follows from the following equations:

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle, \qquad \text{and} \tag{5}$$

$$H^{\otimes n}|1^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{|x|}|x\rangle. \tag{6}$$

Thus, when the players measure their qubits, they will get either a random even parity string when $|x| \equiv 0 \pmod 4$ or a random odd parity string when $|x| \equiv 2 \pmod 4$, as desired. □

Note that the idea of inducing a relative phase proportional to the Hamming weight of a string is studied more generally and called "rotation by Hamming weight" in [15].

2.2 Lower Bound for NC^0 Circuits of Locality 1

We now discuss the success probability of classical strategies for the Parity Halving Game. This was already studied by Mermin [17], and Brassard, Broadbent, and Tapp [4]. Both papers argue that classical strategies only succeed with probability exponentially close to 1/2 on the uniform distribution over even-parity inputs.

We reprove these lower bounds on the Parity Halving Game and also prove lower bounds for a restricted version of the game. In the restricted version of the game we only consider inputs consistent with some restriction of the input bits, i.e., where the values of some input bits have been fixed and are known to all the players, and we only consider all even-parity inputs consistent with this fixing of input bits. We need this generalization later on in the proof since some input bits will be fixed by a random restriction in the AC^0 lower bound argument.

Theorem 6 (Classical lower bound for the Parity Halving Game). *On the uniform distribution over even-parity strings, the success probability of any classical strategy for the Parity Halving Game with n players is at most* $\frac{1}{2} + 2^{-\lceil n/2\rceil}$.

Now consider the restricted *Parity Halving Game with n players, where d of the input bits have fixed values known to all players. On the uniform distribution over even-parity strings consistent with the fixed input bits, the success probability of any classical strategy is at most* $\frac{1}{2} + 2^{-\lceil (n-d)/2\rceil}$.

Proof. We start with the lower bound for the unrestricted Parity Halving Game. Since we consider classical strategies against a fixed input distribution, we can without loss of generality only consider deterministic strategies. This is because a randomized strategy is simply a probability distribution over deterministic strategies, and we can pick the strategy that does the best against the chosen input distribution. (This is the easy direction of Yao's minimax principle.)

Since each player only has one input bit x_i, and one output bit y_i, there are only four deterministic strategies: output $y_i = 0$, $y_i = 1$, $y_i = x_i$, or $y_i = x_i \oplus 1$. In any case, each y_i is a degree-1 polynomial (over $\mathbb{F}_2$) in x_i. It follows that the parity of the outputs, $\bigoplus_{i=1}^n y_i$, can be expressed as multivariate linear polynomial in $x_1, \ldots, x_n$, say $a + b \cdot x$ for some $a \in \mathbb{F}_2$ and $b \in \mathbb{F}_2^n$. We want to upper bound the success probability of any such strategy.

Now consider the function $f(x) = \Re(i^{|x|})$. We have

$$f(x) = \begin{cases} 1 & \text{if } |x| \equiv 0 \pmod 4 \\ -1 & \text{if } |x| \equiv 2 \pmod 4 \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

The function $f(x)$ matches the parity of the output bits (as ± 1) of the PHP_n function on an input x. More precisely, $f(x)$ gives the correct parity (as ± 1) when x satisfies the promise of PHP_n, and evaluates to 0 for inputs outside the promise.

It follows that the product $(-1)^{a+b\cdot x} f(x)$ is 1 if the strategy corresponding to $a + b \cdot x$ is correct, -1 if it is incorrect, and 0 on inputs that are outside the promise. We define the correlation χ of a classical strategy as the absolute value of the fraction of valid inputs on which it is correct minus the fraction of valid inputs on which it is incorrect. We can compute this quantity as follows:

$$\chi = \left| \mathop{\mathbb{E}}_{x\in\mathbb{F}_2^n : \sum_i x_i = 0} \left[(-1)^{a+b\cdot x} f(x)\right] \right| \tag{8}$$

$$= \left| \frac{1}{2^{n-1}} \sum_{x\in\mathbb{F}_2^n} (-1)^{a+b\cdot x} \Re(i^{|x|}) \right| \tag{9}$$

$$\le \frac{1}{2^{n-1}} \left| \Re\left(\sum_{x\in\mathbb{F}_2^n} (-1)^{b_1x_1+\cdots+b_nx_n} \cdot i^{x_1+\cdots+x_n} \right) \right| \tag{10}$$

$$= \frac{1}{2^{n-1}} \left| \Re\left(\sum_{x_1\in\mathbb{F}_2} (-1)^{b_1x_1} i^{x_1} \cdots \sum_{x_n\in\mathbb{F}_2} (-1)^{b_nx_n} i^{x_n} \right) \right| \tag{11}$$

$$= \frac{1}{2^{n-1}} \left| \Re\left((1 + i^{1+2b_1}) \cdots (1 + i^{1+2b_n}) \right) \right|. \tag{12}$$

That is, we want to know the real part of a product of n terms, each of which is $1 \pm i$. Since $1 \pm i$ is $\sqrt{2}$ times a primitive eighth root of unity, the product is $2^{n/2}$ times an eighth root of unity. After factoring out the $\sqrt{2}$ from each term, we have to determine the possible values of the product of n numbers of the form $\frac{1}{\sqrt{2}}(1 \pm i)$. When n is even, their product must lie in the set $\{\pm 1, \pm i\}$, and when n is odd it must lie in the set $\frac{\pm 1 \pm i}{\sqrt{2}}$. In both cases, we see that the real part of the product is either 0 or $\pm 2^{\lfloor n/2\rfloor}$, so the correlation is

$\chi = 0$ or $\chi = 2^{-\lceil n/2 \rceil + 1}$. Since the success probability is $(1+\chi)/2$, this proves the first part of the theorem.

Now let us move on the to restricted version of the game and fix some of the inputs. If some individual bit x_j is restricted, then the term $\sum_{x_j \in \mathbb{F}_2} (-1)^{b_j x_j} i^{x_j}$ in the analysis above becomes either 1 or $(-1)^{b_j} i$. This term is a fourth root of unity, so it does not contribute to the magnitude of the product, since the fourth roots of unity have magnitude 1. Furthermore, it does not change the set of potential phases, since both the sets above are invariant under multiplication by a fourth root of unity. Since the constraint also halves the number of possible inputs, the effect on the correlation is the same as just removing that bit. In other words, χ is at most $2^{-\lceil (n-d)/2 \rceil + 1}$. It follows that the success probability of a classical strategy is

$$\frac{1+\chi}{2} = \frac{1}{2} + 2^{-\lceil (n-d)/2 \rceil}. \tag{13}$$

□

It is interesting to note that for the unrestricted game, Brassard, Broadbent, and Tapp [4] show that there are strategies matching this upper bound.

2.3 From NC^0 Circuits of Locality 1 to General NC^0 Circuits

We can view NC^0 circuits as a more powerful model of computation than the game considered in the previous section. Now each player is allowed to look at the input bits of a constant number of other players before deciding what to output. For example, the players could band together into constant-sized groups and look at all the other bits in the group to make a slightly more informed choice. However, intuitively it seems that the players cannot do much better than before. We will show this formally by proving that NC^0 circuits cannot solve PHP_n.

First, we define some terms. Fix a circuit C and define the *interaction graph* of the circuit C to be a bipartite graph on the input bits and output bits where there is an edge from an input bit x_i to an output bit y_j if there is a path from x_i to y_j in the circuit C (i.e., if x_i can affect y_j in C). The neighborhood of a vertex in this graph is sometimes called its *light cone*. That is, the light cone of an output bit, $LC(y_i)$, is the set of input bits which can affect it, and the light cone of an input bit, $LC(x_i)$ is the set of output bits which it can affect. For example, if all gates have fan-in 2, then the light cone of any output bit in a circuit of depth d is of size at most 2^d. In general, we say that a circuit C has locality ℓ if the light cone of any output bit is of size at most ℓ.

Note that while the fan-in of gates sets an upper bound on the light cone of an output bit, the fan-out sets an upper bound for the light cone of input bits. In all the classical circuit classes we study in this paper, fan-out is unbounded, hence even in a constant-depth circuit one input bit can affect all output bits.

Proposition 7. *Let C be a circuit with n inputs, m outputs, and locality ℓ. There exists a subset of inputs bits S of size $\Omega\left(\min\left\{n, \frac{n^2}{\ell^2 m}\right\}\right)$ such that each output bit depends on at most one bit from S.*

Proof. Since each output bit has a light cone of size at most ℓ, the interaction graph has at most ℓm edges. This implies that, on average, an input bit has a light cone of size $\frac{\ell m}{n}$. Our goal is to find a set of input bits S such that their light cones are pairwise disjoint, since then the light cone of any output contains at most one element of S.

Consider the intersection graph between input variables. That is, we consider the graph on $x_1, \ldots, x_n$, where x_i is connected to x_j if their light cones intersect. A variable x_i that had degree d in the original graph has degree at most $d\ell$ in the intersection graph, since each output vertex has locality ℓ. Hence the average degree in the intersection graph, denoted by D, is at most $\frac{\ell^2 m}{n}$. By Turán's theorem, in any graph on n vertices with average degree at most D there exists an independent set of size at least $n/(1+D)$. Thus, we get a set $S \subseteq \{x_1, \ldots, x_n\}$ of size $\Omega\left(\min\left\{n, \frac{n^2}{\ell^2 m}\right\}\right)$ such that the light cones of every pair of input bits in S do not intersect. □

We are now ready to prove a lower bound on NC^0 circuits of locality ℓ solving $\mathrm{PHP}_{n,m}$.

Theorem 8 (PHP is hard for NC^0). *Let C be an NC^0 circuit with n inputs, m outputs, and locality ℓ. Then C solves $\mathrm{PHP}_{n,m}$ on a random even-parity input with probability at most $\frac{1}{2} + 2^{-\Omega\left(\min\left\{n, \frac{n^2}{\ell^2 m}\right\}\right)}$.*

Proof. Let the circuit C solve $\mathrm{PHP}_{n,m}$ on a random even-parity input with probability p. By the previous theorem, there is a set of input bits with disjoint light cones, S, and $|S| = \Omega\left(\min\left\{n, \frac{n^2}{\ell^2 m}\right\}\right)$. For the remainder of this proof fix any such S.

Now consider choosing an arbitrary assignment for the bits outside S and running the circuit C on the distribution of random even-parity strings consistent with this arbitrary assignment. The probability of success of circuit C may depend on the arbitrary assignment chosen, but since the success probability for a random choice is p, there exists one assignment for which the success probability is at least p. Let us fix this assignment of bits outside S. Now we have an assignment for bits outside S such that C is correct with probability at least p on a random even-parity input consistent with this assignment.

We will now argue that the circuit gives a strategy for the restricted Parity Halving Game on n players with $n - |S|$ restricted bits with probability of success at least p. To do so, we assign a player for every input bit. Only the players assigned to bits in S will have unrestricted inputs. Since the light cones of bits in S do not intersect, a player with input bit in S can compute the values of all outputs in its light cone (since all the bits outside S are fixed and known to everyone). This player can now output the parity of all these output bits. Some output bits may not appear in any input light cone; we add the parity of these bits to an arbitrary player's output. Now the the parity of the players' outputs is the same as the parity of the circuit's output. This gives a classical strategy for the restricted Parity Halving Game with n players and $n - |S|$ restricted bits with success probability at least p. Finally, from Theorem 6 we get that $p \leq \frac{1}{2} + 2^{-\Omega\left(\min\left\{n, \frac{n^2}{\ell^2 m}\right\}\right)}$. □

Note that this theorem is essentially tight. It says that to achieve a high probability of success, we need $n^2 = \Theta(\ell^2 m)$. We can indeed achieve success probability 1 at both extremes: when $m = \Theta(n^2)$ and $\ell = 2$, or when $m = 1$ and $\ell = n$. For the first setting of parameters, as noted in the introduction, there is a simple depth-1 NC^0 circuit

of locality 2 that solves the problem when $m = \binom{n}{2}$. The second parameter regime is even simpler, since any Boolean function can be computed by an NC^0 circuit of locality $\ell = n$.

2.4 From NC^0 Circuits to AC^0 Circuits

In this section we finally extend our lower bound to AC^0 circuits as stated in Theorem 2.

To do this, we use a technical tool known as a switching lemma [1, 10, 13, 29]. Informally, a switching lemma says that with high probability randomly restricting a large fraction of the input bits to an AC^0 circuit produces a circuit with small locality.

Average-case reductions from NC^0 to AC^0 have previously appeared in the literature (cf. [27]), based on the original switching lemma [13]. In this paper, we will use multi-switching lemmas, which handle multiple output circuits much better, and were recently proved by Håstad [14] and Rossman [22]. Using the multi-switching lemmas instead of Håstad's original switching lemma [13] allows us to improve the parameters dramatically.[7]

2.4.1 Preliminaries. We start with some definitions. In the following, we consider restrictions and random restrictions. A restriction $\rho \in \{0,1,*\}^n$ defines a partial assignment to the inputs of a Boolean string of length n. For $i = 1, \ldots, n$, when $\rho_i \in \{0,1\}$ we say that the restriction fixes the value of the i-th coordinate, and when $\rho_i = *$ we say that the restriction keeps the i-th coordinate alive.

A p-random restriction is a restriction sampled according to the following process: for each $i = 1, \ldots, n$ independently, sample $\rho_i = *$ with probability p, $\rho_i = 0$ with probability $(1-p)/2$ and $\rho_i = 1$ with probability $(1-p)/2$. We denote by $\mathbf{R}_p$ the distribution of p-random restrictions.

For a Boolean function $f : \{0,1\}^n \to \{0,1\}^m$ we denote by $f|_\rho : \{0,1\}^n \to \{0,1\}^m$ the restricted function defined by

$$f|_\rho(x) = f(y) \qquad \text{where} \qquad y_i = \begin{cases} x_i & \rho_i = * \text{ and} \\ \rho_i & \text{otherwise.} \end{cases} \tag{14}$$

Next, we give the standard definition of a decision tree. For an excellent survey on this topic, please see [6].

Definition 9 (Decision Tree). *A* decision tree *is a rooted ordered binary tree T, where each internal node of T is labeled with a variable x_i and each leaf is labeled with a value 0 or 1. Given an input $x \in \{0,1\}^n$, the tree is evaluated as follows. Start at the root. If this is a leaf then stop. Otherwise, query the variable x_i that labels the root. If $x_i = 0$, then recursively evaluate the left subtree, if $x_i = 1$ then recursively evaluate the right subtree. The output of the tree is the value (0 or 1) of the leaf that is reached eventually. Note that an input x deterministically determines the leaf reached at the end, and thus the output. We say a decision tree* computes *f if its output equals $f(x)$, for all $x \in \{0,1\}^n$. The complexity of such a tree is its depth, i.e., the number of queries made on the worst-case input. We denote by $\mathrm{DT}(t)$ the class of functions computed by decision trees of depth at most t.*

[7] Based on the original switching lemma, we can show that PHP is hard to compute by AC^0 circuits on more than $1/2 + 1/n^{\Omega(\log n)}$ of the inputs. On the other hand, based on the multi-switching lemmas, we will show that PHP is, in fact, hard to compute on more than $1/2 + \exp(-n^{1-o(1)})$ of the inputs.

Note that the decision tree complexity of a function f is also called the deterministic query complexity of f.

Definition 10 ($\mathcal{F}$-Decision Tree). *Suppose $\mathcal{F}$ is a class of functions mapping $\{0,1\}^n$ to $\{0,1\}^m$. An $\mathcal{F}$-partial decision tree is a standard decision tree, except that the leaves are marked with functions in $\mathcal{F}$ (instead of constants). Given an input $x \in \{0,1\}^n$, the $\mathcal{F}$-Decision Tree is evaluated as follows. Starting from the tree's root, we go along the path defined by the input x until we reach a leaf. Then, we evaluate the function $f_v \in \mathcal{F}$ that labels the leaf v on the input x, and output its value, $f_v(x)$. We denote by $\mathrm{DT}(t) \circ \mathcal{F}$ the class of functions computed by $\mathcal{F}$-decision trees of depth at most t.*

Note that $\mathcal{F}$-decision trees compute functions from $\{0,1\}^n \to \{0,1\}^m$ where n and m are the input and output lengths for the functions in $\mathcal{F}$, respectively.

Definition 11 (Tuples of functions classes). *Suppose $\mathcal{F}$ is a class of functions mapping $\{0,1\}^n$ to $\{0,1\}$. We denote by $\mathcal{F}^m$ the class of functions $F : \{0,1\}^n \to \{0,1\}^m$ of the form $F(x) = (f_1(x), \ldots, f_m(x))$, where each $f_i \in \mathcal{F}$. That is, $\mathcal{F}^m$ is the class of m-tuples of functions in $\mathcal{F}$.*

2.4.2 The Multi-Switching Lemma. The main lemma that we are going to use is a slight adaption of Rossman's lemma [22], which combines both switching lemmas of Håstad [13, 14]. The lemma claims that a multi-output AC^0 circuit mapping $\{0,1\}^n \to \{0,1\}^m$ would reduce under a random restriction, with high probability, to a function in the class $\mathrm{DT}(2t) \circ \mathrm{DT}(q)^m$ (for some parameters t and q).

Let us pause for a second to spell out what is the class $\mathrm{DT}(2t) \circ \mathrm{DT}(q)^m$. This is the class of depth-$2t$ decision trees, whose leaves are labeled by m-tuples of depth-q decision trees, one per output bit. In other words, these are functions mapping $\{0,1\}^n$ to $\{0,1\}^m$ that can be evaluated by adaptively querying at most $2t$ coordinates globally, after which each of the m output bits can be evaluated by making at most q additional adaptive queries. Note that while the first $2t$ queries are global, the last q queries could differ from one output bit to another. We would typically set the parameters so that t is much larger than q (for example, $t = n^{1-o(1)}$ and $q = o(\log n)$).

Lemma 12. *Let $f : \{0,1\}^n \to \{0,1\}^m$ be an AC^0 circuit of size s, depth d. Let $q \in \mathbb{N}$ be a parameter, and set $p = 1/(m^{1/q} \cdot O(\log s)^{d-1})$. Then*

$$\forall t : \Pr_{\rho \sim \mathbf{R}_p} [f|_\rho \notin \mathrm{DT}(2t) \circ \mathrm{DT}(q)^m] \le s \cdot 2^{-t}. \tag{15}$$

This is an adaptation of Rossman's lemma [22]. See the full paper for a proof.

We would use the lemma as follows. First, we apply a p-random restriction that reduces the AC^0 circuit to a $\mathrm{DT}(2t) \circ \mathrm{DT}(q)^m$ function with high probability. Then, we further query at most $2t$ coordinates, and fix their values, by following a path in the common partial decision tree. After which, the restricted function would be an m-tuple of depth-q decision trees. Then, using the simple fact that a depth-q decision tree is a function with locality at most 2^q, we reduced an AC^0 circuit to an NC^0 circuit with locality at most 2^q with high probability.

On the choice of parameters. We have the freedom to choose q and t when applying Lemma 12 in Theorem 13. First, we discuss

the choice of q. We would like the lemma to yield on one hand an NC^0 circuit with small locality, and on the other hand to keep many input variables alive. To get small locality, q should be small, say $q = o(\log n)$. To keep many variables alive, pn should be large, and since $p = 1/O(m^{1/q}(\log s)^{d-1})$, we would like q to be large, say $q = \omega(1)$. Balancing these two requirements leads to the choice $q = \Theta(\sqrt{\log n})$.

Once q is set, we would like to make t as large as possible, as it controls the failure probability in Lemma 12, but at the same time we want the number of alive variables after the two-step restriction process above to remain high. Since this number is roughly $pn - t$ we would choose t to be a small constant fraction of pn (which is $n^{1-o(1)}$). With these choices, we would be left with at least $\Omega(pn)$ variables alive and locality at most 2^q with extremely high probability.

2.4.3 AC^0 Lower Bound.

Theorem 13. *Let $n \le m \le n^2$. Any $\mathrm{AC}^0/\mathrm{rpoly}$ circuit F of depth d and size $s \le \exp(n^{1/2d})$ solves $\mathrm{PHP}_{n,m}$ on the uniform distribution over valid inputs (even parity strings) with probability at most $\frac{1}{2} + \exp(-n^2/(m^{1+o(1)} \cdot O(\log s)^{2(d-1)}))$.*

We omit the proof from this version of the paper. Briefly, we apply Lemma 12 as described above, producing a circuit with bounded fan-in gates and limited locality. We finish the proof with Theorem 8.

3 RELAXED PARITY HALVING PROBLEM

In this section we deal with the issue that the QNC^0 circuit for PHP (Problem 1) needs a cat state, but QNC^0 cannot create a cat state.

In Section 3.1, we first prove that a QNC^0 circuit cannot create a cat state. But, as we show, QNC^0 circuits can construct what we call a "poor man's cat state." This is a state of the form $\frac{1}{\sqrt{2}}(|z\rangle + |\bar{z}\rangle)$ for some uncontrolled $z \in \{0,1\}^n$ alongside classical "side information" about z that allows us to determine it.

In Section 3.2 we show the poor man's cat state lets us solve a relaxed version of the Parity Halving Problem, which is nevertheless hard for AC^0 circuits.

3.1 A Poor Man's Cat State

First, we note that a QNC^0 circuit cannot construct a cat state in constant depth. See the full version for a proof.

Theorem 14 (Cat states cannot be created in QNC^0). *Let C be a depth-d QNC^0 circuit over the gates set of all 2-qubit gates that maps $|0^{n+m}\rangle$ to $|\text{⋈}_n\rangle \otimes |0^m\rangle$. Then $d \ge (\log n)/2$.*

Now although QNC^0 circuits cannot create the cat state, we show that QNC^0 circuits are able to construct states of the form $\frac{1}{\sqrt{2}}(|z\rangle + |\bar{z}\rangle)$, where z is some string in $\{0,1\}^n$ and $\bar{z}$ the complement of z. Note that this state is exactly the cat state when $z = 0^n$ or $z = 1^n$. The circuits that create this state also output an auxiliary classical string d such that z can be determined from d, up to the symmetry between z and $\bar{z}$. There is actually a family of QNC^0 circuits which construct these states, which we now describe.

Theorem 15 (Poor man's cat state construction). *For any connected graph $G = (V, E)$ with maximum degree Δ, there is a depth $\Delta + 2$ QNC^0 circuit which outputs a $|V|$ qubit state $\frac{1}{\sqrt{2}}(|z\rangle + |\bar{z}\rangle)$, along with a bit string $d \in \{0,1\}^E$. Indexing the bits of z by vertices of V and the bits of d by edges of E, z and d satisfy the property that*

$$z_u + z_v \equiv \sum_{e \in P(u,v)} d_e \pmod 2 \tag{16}$$

for are any two vertices $u, v \in V$, and any path $P(u,v)$ from u to v. Note that this condition also implies that the sum of d_e along any cycle in the graph is $0 \pmod 2$.

Proof. We first describe the QNC^0 circuit. Begin with $|V| + |E|$ qubits in the state $|0\rangle$, and identify each of the qubits with either an edge or a vertex of the graph. Apply the Hadamard transform for each vertex qubit. Now the state is $|+\rangle^{|V|} \otimes |0\rangle^{|E|}$. Then, for every edge $e = (u, v)$ in the graph, XOR the qubits indexed by u and v onto the edge qubit indexed by e (i.e., let the edge qubit store the parity of the two vertex qubits). Explicitly, this can be done by implementing CNOT gates from qubits u, v onto qubit e. (As discussed below, this can be done in $\Delta + 1$ parallel local steps.) Finally, measure all edge qubits in the standard basis.

To complete this proof we need to establish two claims: First, that the circuit leaves the unmeasured vertex qubits in the state $\frac{1}{\sqrt{2}}(|z\rangle + |\bar{z}\rangle)$, while the measured edge qubits give the classical bitstring d. Second, that the circuit can be implemented in depth $\Delta + 2$.

We begin with the first claim. Imagine that we first only measure the $n - 1$ edges of some spanning tree T. Before measurement, the vertex qubits were in a uniform superposition over all possible 2^n states. Each measurement on an edge qubit had two equally probable outcomes, and observing the result of this measurement reduced the number of states in the superposition by half. More precisely, measuring the qubit for edge $e = (u, v)$ yields a bit $d_e \in \{0,1\}$, which gives a linear equation on the state: $z_u \oplus z_v = d_e$. Thus, after all edges in the spanning tree are measured, the vertex qubits must be left in some two state superposition. Furthermore, after the spanning tree is measured any two vertex qubits u and v must differ by the parity of the observed measurements on edge qubits along the path from u to v. This shows the vertex qubits must be in the state $\frac{1}{\sqrt{2}}(|z\rangle + |\bar{z}\rangle)$, with the measurements on the edge qubits so far consistent with the requirements of Theorem 15. Now for any edge $e = (v, w)$ not in the spanning tree, the XOR measurements on the associated edge qubit is fixed to be equal to the XOR of edge qubit measurements along the path in T from v to w. This shows this measurement must also be consistent with the requirements of Theorem 15 and cannot affect the state of the vertex qubits. The establishes the first claim.

The second claim is more straightforward. The first layer of our QNC^0 circuit consists of Hadamard gates applied to all vertex qubits. At the end there is a computational basis measurement of all edge qubits, but final measurements traditionally do not count towards the depth. It remains to show that we can implement all the desired CNOT gates in depth $\Delta + 1$. To show this we introduce a new graph G' with $|V| + |E|$ vertices, that is obtained from G by replacing each edge $e = (a, b)$ in E with a vertex v_e connected to its two end-points, a and b. Note the edges of G' are in one to one correspondence with the CNOT gates we want to implement in our circuit. By our assumptions on G, G' has degree at most Δ, and so Vizing's theorem tells us the edges of G' can be colored

using at most $\Delta + 1$ colors. Since the edges in each color class are non-overlapping we can apply all the CNOT gates in one color class simultaneously, and thus apply all the CNOT gates in depth $\Delta + 1$. □

In the remainder of this paper, we primarily apply Theorem 15 when G is a spanning tree of a 2D grid, with diameter $2\sqrt{n}$ as depicted in Figure 2, which also describes the associated QNC^0 circuit. This QNC^0 circuit has the nice feature that it is spatially local,[8] while any bit of z is specified by relatively few, $O(\sqrt{n})$, bits of d. This graph has constant degree $\Delta = 3$.

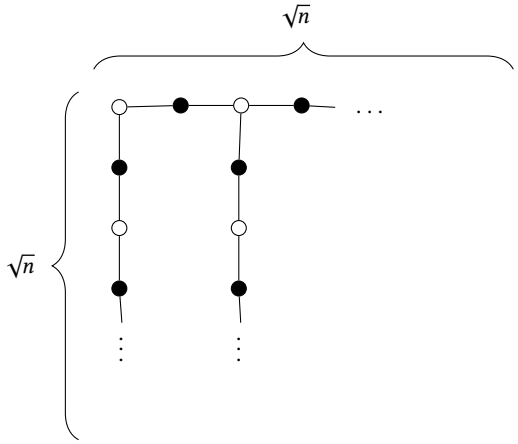

Figure 2: Grid Implementation of a Poor Man's Cat State. Black vertices are "edge" qubits, and are used to measure the parity of their neighbours. White vertices are "vertex" qubits. They are initialized in the $|+\rangle$ state, and make up the poor man's cat state after the edge qubits are measured.

It is worth mentioning that if we relax the requirement that our implementation be spatially local, we can improve on the number of bits of d required to specify any bit of z. In particular, applying Theorem 15 to a balanced binary tree gives an output string d with at most $\log(n)$ bits of d required to specify any bit of z. This version of the problem would lead to slightly better parameters in Theorem 3, but then our final problem would not longer be solved by a 2D quantum circuit and would no longer reduce to the 2D HLF problem. Hence the construction illustrated in Figure 2 will be sufficient for our purposes.

3.2 The Relaxed Parity Halving Problem

Having constructed a poor man's cat state, a natural idea would be to try and use this state instead of the cat state to solve the Parity Halving Problem. For example, we can feed the poor man's cat state into the quantum circuit (instead of a true cat state) and hope for the best. Unsurprisingly, this does not solve the Parity Halving Problem.

However, we can make lemonade from the lemons we have been handed. We can define a new problem from this failed attempt, which has QNC^0 circuits by construction. We call this problem the *Relaxed Parity Halving Problem*, although we will see that the precise definition of the problem depends on how the poor man's cat state is constructed.

Theorem 16 (PHP circuit applied to a poor man's cat state). *The quantum circuit for* PHP_n *applied to the state* $\frac{|z\rangle+|\bar{z}\rangle}{\sqrt{2}}$*, where* $z \in \{0,1\}^n$ *(instead of the cat state), and an input* $x \in \{0,1\}^n$ *of even parity, yields an output string* $y \in \{0,1\}^n$ *such that*

$$|y| \equiv \frac{1}{2}|x| + \langle z, x\rangle \pmod 2, \tag{17}$$

where $\langle z, x\rangle := \sum_{i\in[n]} z_i \cdot x_i$*. Note that this is the same condition as for* PHP_n *(Problem 1) except for the addition of the* $\langle z, x\rangle$ *term.*

Proof. Let us apply the quantum circuit solving PHP (as depicted in Figure 1) to the poor man's cat state. We first apply a phase gate (S gate) to qubit i of the poor man's cat state if $x_i = 1$. This yields the state

$$\frac{i^{\langle z,x\rangle}|z\rangle + i^{\langle \bar{z},x\rangle}|\bar{z}\rangle}{\sqrt{2}} = i^{\langle z,x\rangle}\frac{|z\rangle + i^{|x|-2\langle z,x\rangle}|\bar{z}\rangle}{\sqrt{2}} \tag{18}$$

$$= i^{\langle z,x\rangle}\frac{|z\rangle + (-1)^{|x|/2-\langle z,x\rangle}|\bar{z}\rangle}{\sqrt{2}}. \tag{19}$$

Up to a global phase, which can be ignored, the state is $\frac{1}{\sqrt{2}}\cdot(|z\rangle + (-1)^{|x|/2-\langle z,x\rangle}\cdot|\bar{z}\rangle)$. The next stage of the algorithm applies Hadamard gates to all input qubits. Thus we have

$$H^{\otimes n}\left(\frac{|z\rangle + (-1)^{|x|/2-\langle z,x\rangle}|\bar{z}\rangle}{\sqrt{2}}\right) \tag{20}$$

$$= \frac{1}{\sqrt{2^{n+1}}}\sum_{y\in\{0,1\}^n}\left((-1)^{\langle y,z\rangle} + (-1)^{\langle y,\bar{z}\rangle+|x|/2-\langle z,x\rangle}\right)|y\rangle\,. \tag{21}$$

On measuring this state in the computational basis, we get only those y whose coefficient is nonzero. Hence we get a uniform distribution over all strings y satisfying $\langle y, z\rangle \equiv \langle y, \bar{z}\rangle + |x|/2 - \langle z, x\rangle \pmod 2$ or equivalently,

$$|y| \equiv \frac{1}{2}|x| + \langle z, x\rangle \pmod 2. \tag{22}$$

□

We now define a new problem based on this observation.

Problem 2 (Relaxed Parity Halving Problem for graph G)**.** Fix a connected graph $G = (V, E)$. Given an input $x \in \{0,1\}^V$ promised to have even parity, the *Relaxed Parity Halving Problem* or RPHP outputs $y \in \{0,1\}^V$ and $d \in \{0,1\}^E$, such that there exists a $z \in \{0,1\}^V$ with the property

$$\forall (u,v) \in E,\ z_u \oplus z_v = d_{(u,v)}, \quad \text{and} \tag{23}$$

$$|y| \equiv \frac{1}{2}|x| + \langle z, x\rangle \pmod 2. \tag{24}$$

Note that in the definition above, a string z satifying the first constraint exists if and only if the parity of d along every cycle is 0. If there are no cycles, then there always exists a z. When z does exist, it is unique up to complement, which does not change the second condition in the problem statement because $\langle z, x\rangle \equiv \langle \bar{z}, x\rangle$

[8] Here spatially local means here that circuit may be implemented in hardware with the qubits placed on a 2D grid and CNOT gates allowed only between neighbouring qubits.

(mod 2). To see this, recall that x has even parity, and hence $0 \equiv \langle 1, x\rangle \equiv \langle z, x\rangle + \langle \bar{z}, x\rangle \pmod 2$.

To fully specify the problem, we need a family of graphs with $|V| = n$ for infinitely many n. For this paper, we are primarily interested in the 2D grid (so that the quantum circuit is specially local) and some reasonable (say, diameter $O(\sqrt{n})$) spanning tree of this graph. This gives us the Grid Relaxed Parity Halving Problem below. We use a spanning tree rather than the grid graph itself because deleting edges from G only makes the problem easier (since we can drop the corresponding bits of the output string d), which makes our lower bounds stronger. Additionally, choosing a spanning tree ensures that a string z satisfying the constraints of the problem always exists. It turns out the upper bounds (i.e., QNC^0 circuit we construct) can be easily modified to solve the problem on the entire 2D grid without deleting edges.

Problem 3 (Grid Relaxed Parity Halving Problem)**.** Consider the 2D grid of size $\sqrt{n}\times\sqrt{n}$ and fix an n vertex spanning tree $G = (V, E)$ of low diameter. For concreteness, fix the spanning tree which takes the first row of edges and all columns (as depicted in Figure 2). Then the *Grid Relaxed Parity Halving Problem* is the RPHP associated with G. That is, given $x \in \{0,1\}^V$, output $y \in \{0,1\}^V$ and $d \in \{0,1\}^E$ such that

$$\forall (u, v) \in E,\ z_u \oplus z_v = d_{(u,v)}, \quad \text{and} \tag{25}$$

$$|y| \equiv \frac{1}{2}|x| + \langle z, x\rangle \pmod 2. \tag{26}$$

We can use other graphs instead of the grid to define different variants of this problem. For instance, a balanced binary tree has lower diameter, which leads to slightly better parameters, but we use the grid to achieve a spatially local quantum circuit.

A path graph (sometimes called the line graph) is even simpler than the grid or tree. Unfortunately, the Relaxed Parity Halving Problem corresponding to the path graph can be solved by an NC^0 circuit, which makes it unsuitable for proving a separation against NC^0.

3.3 Quantum Circuit and AC^0 Lower Bound

In this section we establish Theorem 3, which states that Grid-RPHP can be solved in QNC^0, but it is average-case hard for AC^0 circuits. We start by establishing the quantum upper bound.

Theorem 17 (Grid-RPHP is in QNC^0). *There exists a depth-5 spatially local* QNC^0 *circuit that exactly solves* Grid-RPHP.

Proof. Let $G = (V, E)$ be a $O(\sqrt{n})$-diameter spanning tree of the $\sqrt{n}\times\sqrt{n}$ grid graph with $|V| = n$ and $|E| = n - 1$. This graph has degree $\Delta = 3$. As shown in Theorem 15, there is a spatially local QNC^0 circuit of depth $\Delta + 2$ to construct a random poor man's cat state $\frac{|z\rangle+|\bar{z}\rangle}{\sqrt{2}}$ (for some $z \in \{0,1\}^V$) and the associated string $d \in \{0,1\}^E$ for graph G such that

$$d_{(u,v)} = z_u \oplus z_v \tag{27}$$

for all $(u, v) \in E$. We run the QNC^0 circuit for the Parity Halving Problem as per Theorem 16, and get an output $y \in \{0,1\}^V$ such that

$$|y| \equiv \frac{1}{2}|x| + \langle z, x\rangle \pmod 2. \tag{28}$$

We have defined Grid-RPHP so that it is not necessary to compute z, just a vector d consistent z, which we have from the construction of the poor man's cat state. Hence, we return y and d satisfying the condition, and we are done. □

This result is not surprising, since the relaxed parity halving problem is a relaxation of the parity halving problem explicitly constructed with the goal of having a QNC^0 circuit.

The nontrivial direction of the argument is to show that AC^0 cannot solve Grid-RPHP. We accomplish this by exhibiting an NC^0 reduction from Grid-RPHP to an instance of $\mathrm{PHP}_{n,m}$ with $m = \Theta(n^{3/2})$, which is still hard for AC^0. Note that although our reduction is an NC^0 reduction, it cannot be carried out with a QNC^0 circuit, so we are not showing that PHP is in QNC^0. While this might seem mysterious at first, the reason is that NC^0 has one ability that we have not given QNC^0: unbounded fan-out. Our reduction uses the fact that NC^0 can make unlimited copies of the output of a gate, whereas QNC^0 cannot do so.

Theorem 18 (Grid-RPHP is not in AC^0). *There is an* AC^0 *reduction from* $\mathrm{PHP}_{n,O(n^{3/2})}$ *to* Grid-RPHP$_n$. *In particular, an* AC^0 *circuit of size* $s \le \exp(n^{1/2(d+1)})$ *and depth* d *cannot solve* Grid-RPHP *with probability better than*

$$\frac{1}{2} + \exp(-n^{1/2-o(1)}/O(\log s)^{2d}) \tag{29}$$

on a random input with even-parity.

Proof. Suppose we want to solve an instance of $\mathrm{PHP}_{n,O(n^{3/2})}$, and we can solve Grid-RPHP$_n$. Let $T = (V, E)$ be a $O(\sqrt{n})$-diameter spanning tree of an n vertex grid graph.

Take the input $x \in \{0,1\}^n$ from the PHP instance as input for a Grid-RPHP$_n$ instance, mapping the n bits arbitrarily to vertices of the grid. Solving this Grid-RPHP instance gives us vectors $y \in \{0,1\}^V$ and $d \in \{0,1\}^E$ such that

$$|y| \equiv |x|/2 + \langle z, x\rangle \pmod 2 \tag{30}$$

where $z \in \{0,1\}^V$ satisfies the parity constraints in d. In particular, if we fix $z_1 = 0$ then each z_i is the parity of all d_j along a path from z_1 to z_i in the graph. Let $D_i \subseteq |E|$ denote the edges in the path from z_1 to z_i. Then we can write

$$\langle z, x\rangle = \sum_i z_i x_i = \sum_i \sum_{j\in D_i} d_j x_i. \tag{31}$$

Since the diameter of the grid graph is $O(\sqrt{n})$, we may assume each D_i has size at most $O(\sqrt{n})$. Thus, we have expressed $\langle z, x\rangle$ as a sum of $O(n^{3/2})$ terms of the form $d_j x_i$. We concatenate these to y, and return that as the output to our $\mathrm{PHP}_{n,O(n^{3/2})}$ instance.

Note that an AC^0 circuit of size s and depth d solving Grid-RPHP gives a circuit of size $\mathrm{poly}(s)$ and depth $d+1$ solving PHP. Applying Theorem 13 gives the required bound assuming $s \le \exp(n^{1/2(d+1)})$. □

4 RELATION TO HIDDEN LINEAR FUNCTION PROBLEMS

Finally, to establish our main result (Theorem 1), we have to show that Parallel Grid-RPHP reduces to the 2D HLF problem. Since we

have already established the required hardness for Parallel Grid-RPHP in Theorem 4, we will then be done.

We start by recalling the general Hidden Linear Function problem (HLF) defined by Bravyi, Gosset, and König [5].

Problem 4 (Hidden Linear Function problem)**.** We are given as input a symmetric matrix $A \in \{0,1\}^{n\times n}$ and vector $b \in \{0,1,2,3\}^n$. From these, define a quadratic form $q\colon \mathbb{F}_2^n \to \mathbb{Z}_4$ as $q(x) := x^T Ax + b^T x \pmod 4$. Define $\mathcal{L}_q$ as follows:

$$\mathcal{L}_q := \{x \in \mathbb{F}_2^n : q(x\oplus y) \equiv q(x)+q(y) \pmod 4 \ \ \forall y \in \mathbb{F}_2^n\}. \quad (32)$$

Bravyi et al. [5] show that $\mathcal{L}_q$ is a linear subspace of $\mathbb{F}_2^n$, $q(x)$ is in $\{0,2\}$ for all $x \in \mathcal{L}_q$, and that q is linear on $\mathcal{L}_q$. Since q is linear on $\mathcal{L}_q$, there exists a $z \in \{0,1\}^n$ such that $q(x) \equiv 2z^T x \pmod 4$ for all $x \in \mathcal{L}_q$. The goal is to output any string z satisfying this condition.

Theorem 19. *There is an* NC^0 *reduction from* RPHP *on any graph* G *to the* HLF *problem.*

The proof is calculation heavy, and may be found in the full version of the paper. The intuition is that since the quantum circuit for RPHP (where graph G is used to construct the poor man's cat state) is exactly the same as the quantum circuit for HLF on a related graph G' (where there is a new vertex for each edge, splitting the edge in two), the two problems should be related.

The main problem studied in Bravyi, Gosset and König [5] is actually a version of HLF on an $N \times N$ grid called the 2D Hidden Linear Function problem (2D HLF), where the matrix A is supported only on the grid in the sense that $A_{ij} = 0$ if there is no edge from vertex i to vertex j. Fortunately, RPHP on the grid translates to HLF on a grid in the reduction in Theorem 19, so we have the following corollary.

Corollary 20. *There is an* NC^0 *reduction from* Grid-RPHP *to* 2D HLF.

Furthermore, we can embed multiple disjoint grids into a larger grid, and therefore solve multiple instances of Grid-RPHP by solving a single instance of 2D HLF.

Corollary 21. *Parallel* Grid-RPHP *reduces to* 2D HLF.

The proofs of these corollaries may be found in the full version.

ACKNOWLEDGEMENTS

We would like to thank Nicolas Delfosse, Aram Harrow, Anna Gál, Anand Natarajan, Benjamin Rossman, and Emanuele Viola for very helpful discussions.

REFERENCES

[1] Miklos Ajtai. 1983. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic* 24 (1983), 1–48. https://doi.org/10.1016/0168-0072(83)90038-6

[2] Debajyoti Bera, Frederic Green, and Steven Homer. 2007. Small Depth Quantum Circuits. *ACM SIGACT News* 38, 2 (June 2007), 35–50. https://doi.org/10.1145/1272729.1272739

[3] Juan Bermejo-Vega, Dominik Hangleiter, Martin Schwarz, Robert Raussendorf, and Jens Eisert. 2018. Architectures for Quantum Simulation Showing a Quantum Speedup. *Physical Review X* 8 (Apr 2018), 021010. Issue 2. https://doi.org/10.1103/PhysRevX.8.021010

[4] Gilles Brassard, Anne Broadbent, and Alain Tapp. 2005. Recasting Mermin's Multi-player Game into the Framework of Pseudo-telepathy. *Quantum Information & Computation* 5, 7 (Nov. 2005), 538–550. http://dl.acm.org/citation.cfm?id=2011656.2011658

[5] Sergey Bravyi, David Gosset, and Robert König. 2018. Quantum advantage with shallow circuits. *Science* 362, 6412 (2018), 308–311. https://doi.org/10.1126/science.aar3106

[6] Harry Buhrman and Ronald de Wolf. 2002. Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.* 288, 1 (2002), 21–43. https://doi.org/10.1016/S0304-3975(01)00144-X

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

[8] M. Fang, S. Fenner, F. Green, S. Homer, and Y. Zhang. 2006. Quantum Lower Bounds for Fanout. *Quantum Information & Computation* 6, 1 (Jan. 2006), 46–57. http://dl.acm.org/citation.cfm?id=2011679.2011682

[9] Stephen Fenner, Frederic Green, Steven Homer, and Yong Zhang. 2005. Bounds on the Power of Constant-Depth Quantum Circuits. In *Fundamentals of Computation Theory*. 44–55. https://doi.org/10.1007/11537311_5

[10] Merrick Furst, James B. Saxe, and Michael Sipser. 1984. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory* 17, 1 (Dec 1984), 13–27. https://doi.org/10.1007/BF01744431

[11] Frederic Green, Steven Homer, Cristopher Moore, and Christopher Pollett. 2002. Counting, Fanout and the Complexity of Quantum ACC. *Quantum Information & Computation* 2, 1 (Dec. 2002), 35–65. http://dl.acm.org/citation.cfm?id=2011417.2011420

[12] Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger. 1989. Going beyond Bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe*. Springer, 69–72. https://doi.org/10.1007/978-94-017-0849-4_10

[13] Johan Håstad. 1986. *Computational limitations of small-depth circuits*. Ph.D. Dissertation. Massachusetts Institute of Technology. https://www.nada.kth.se/~johanh/thesis.pdf

[14] Johan Håstad. 2014. On the Correlation of Parity and Small-Depth Circuits. *SIAM J. Comput.* 43, 5 (2014), 1699–1708. https://doi.org/10.1137/120897432

[15] Peter Høyer and Robert Špalek. 2005. Quantum Fan-out is Powerful. *Theory of Computing* 1 (2005), 23. https://doi.org/10.4086/toc.2005.v001a005

[16] François Le Gall. 2018. Average-Case Quantum Advantage with Shallow Circuits. *arXiv preprint arXiv:1810.12792* (2018). arXiv:1810.12792

[17] N. David Mermin. 1990. Extreme quantum entanglement in a superposition of macroscopically distinct states. *Physical Review Letters* 65 (Oct 1990), 1838–1840. Issue 15. https://doi.org/10.1103/PhysRevLett.65.1838

[18] Cristopher Moore. 1999. Quantum Circuits: Fanout, Parity, and Counting. *arXiv:quant-ph/9903046* (March 1999). arXiv:quant-ph/9903046

[19] Cristopher Moore and Martin Nilsson. 2002. Parallel Quantum Computation and Quantum Codes. *SIAM J. Comput.* 31, 3 (March 2002), 799–815. https://doi.org/10.1137/S0097539799355053 arXiv:quant-ph/9808027

[20] Cody Murray and Ryan Williams. 2018. Circuit Lower Bounds for Nondeterministic Quasi-polytime: An Easy Witness Lemma for NP and NQP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018)*. 890–901. https://doi.org/10.1145/3188745.3188910

[21] Ran Raz and Avishay Tal. 2018. Oracle Separation of BQP and PH. In *Electronic Colloquium on Computational Complexity (ECCC)*, Vol. 18-107. https://eccc.weizmann.ac.il/report/2018/107/

[22] Benjamin Rossman. 2017. An entropy proof of the switching lemma and tight bounds on the decision-tree size of AC^0. (2017). http://www.math.toronto.edu/rossman/logsize.pdf Manuscript.

[23] Peter W. Shor. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* 26, 5 (1997), 1484–1509. https://doi.org/10.1137/S0036144598347011

[24] Yasuhiro Takahashi and Seiichiro Tani. 2016. Collapse of the Hierarchy of Constant-Depth Exact Quantum Circuits. *Computational Complexity* 25, 4 (01 Dec 2016), 849–881. https://doi.org/10.1007/s00037-016-0140-0

[25] Yasuhiro Takahashi and Seiichiro Tani. 2018. Power of Uninitialized Qubits in Shallow Quantum Circuits. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018) (Leibniz International Proceedings in Informatics (LIPIcs))*, Vol. 96. 57:1–57:13. https://doi.org/10.4230/LIPIcs.STACS.2018.57

[26] Barbara M. Terhal and David P. DiVincenzo. 2004. Adaptive Quantum Computation, Constant Depth Quantum Circuits and Arthur-Merlin Games. *Quantum Information & Computation* 4, 2 (March 2004), 134–145. http://dl.acm.org/citation.cfm?id=2011577.2011582

[27] Emanuele Viola. 2014. Extractors for Circuit Sources. *SIAM J. Comput.* 43, 2 (2014), 655–672. https://doi.org/10.1137/11085983X

[28] Ryan Williams. 2014. Nonuniform ACC Circuit Lower Bounds. *J. ACM* 61, 1, Article 2 (Jan. 2014), 32 pages. https://doi.org/10.1145/2559903

[29] Andrew C-C. Yao. 1985. Separating the Polynomial-time Hierarchy by Oracles. In *Proc. 26th Annual Symposium on Foundations of Computer Science*. 1–10. http://dl.acm.org/citation.cfm?id=4479.4487