Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 130 (2024)

Improved Circuit Lower Bounds and Quantum-Classical Separations

Sabee Grewal Vinayak M. Kumar[*]

Abstract

Kumar (CCC, 2023) used a novel switching lemma to prove exponential-size lower bounds for a circuit class GC^0 that not only contains AC^0 but can—with a *single* gate—compute functions that require exponential-size TC^0 circuits. Their main result was that switching-lemma lower bounds for AC^0 lift to GC^0 *with no loss in parameters*, even though GC^0 requires exponential-size TC^0 circuits. Informally, GC^0 is AC^0 with unbounded-fan-in gates that behave arbitrarily inside a sufficiently small Hamming ball but must be constant outside it. While seemingly exotic, GC^0 captures natural circuit classes, such as circuits of biased linear threshold gates.

We show that polynomial-method lower bounds for $\mathsf{AC}^0[p]$ lift to $\mathsf{GC}^0[p]$ *with no loss in parameters*, complementing Kumar's result for GC^0 and the switching lemma. As an application, we prove Majority requires depth-d $\mathsf{GC}^0[p]$ circuits of size $2^{\Omega(n^{1/2(d-1)})}$, matching the state-of-the-art lower bounds for $\mathsf{AC}^0[p]$ proven by Razborov and Smolensky. We also apply the algorithmic method to show that E^{NP} requires exponential-size GCC^0 circuits (the union of $\mathsf{GC}^0[m]$ for all m), extending the result of Williams (JACM, 2014).

It is striking that the combinatorial switching lemma, the algebraic polynomial method, and the algorithmic method all generalize to GC^0-related classes, with the first two methods doing so without any loss in parameters. Notably, our results give the least restricted classes of non-monotone circuits for which we have exponential-size lower bounds for explicit functions.

By strengthening classical lower bounds from prior work, we also establish the strongest known *unconditional* separations between quantum and classical circuits. We prove:

- $\mathsf{BQLOGTIME} \not\subseteq \mathsf{GC}^0$. This implies an oracle relative to which BQP is not contained in the class of languages decidable by uniform families of size-$2^{n^{O(1)}}$ GC^0 circuits, generalizing Raz and Tal's relativized separation of BQP from the polynomial hierarchy (STOC, 2019).
- There is a search problem that QNC^0 circuits can solve but is average-case hard for exponential-size GC^0 circuits.
- For any prime p, there is a search problem that $\mathsf{QNC}^0/\mathsf{qpoly}$ circuits can solve but is average-case hard for exponential-size $\mathsf{GC}^0[p]$ circuits.
- For any prime p, there is an interactive problem that QNC^0 circuits can solve but exponential-size $\mathsf{GC}^0[p]$ circuits cannot.

[*] {sabee, vmkumar}@cs.utexas.edu. Department of Computer Science, The University of Texas at Austin.

ISSN 1433-8092

Contents

1 Introduction

Proving superpolynomial circuit lower bounds for an explicit function is a longstanding challenge in computer science, and it remains one of our only viable approaches to resolving the $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ question [Aar16]. These lower bounds also find applications throughout complexity theory, for example, in structural complexity [FSS84, Hås86, Aar10, HRST17, RT22], proving unconditional quantum advantage [BGK18, WKST19, GS20], and pseudorandomness [NW94, IW97].

Motivated in part by the relativization barrier of Baker, Gill, and Solovay [BGS75], considerable effort was put forth in the mid-1970s to early 1980s to prove circuit lower bounds for explicit functions. After a burst of progress [Sch74, Pau75, Sto76, Sch80, Blu81, Blu83], the best lower bound for an explicit function was $3n - o(n)$. The current state of the art is $3.1n - o(n)$, and the (seemingly) marginal improvement in the leading constant was highly nontrivial to obtain [DK11, FGHK16, GK16, LY22].

A "bottom-up" approach to circuit lower bounds has also been explored, where the goal is to prove lower bounds for highly restricted circuits, then slightly relax those restrictions and repeat. This approach has led to two techniques: switching lemmas (or more broadly, the method of random restrictions) [Ajt83, FSS84, Yao85, Hås86] and the polynomial method [Raz87, Smo87]. The former technique has been used to show lower bounds against AC^0, constant-depth circuits of AND, OR, and NOT gates with unbounded fan-in. The latter technique has been used to prove lower bounds against $\mathsf{AC}^0[p]$, constant-depth circuits that include unbounded fan-in MOD_p gates, where p is prime, in addition to AND, OR, and NOT gates.[1]

Alas, this bottom-up approach stalled in the late 1980s. Furthermore, the natural proofs barrier of Razborov and Rudich [RR97] showed that the random restriction and polynomial methods fail to prove superpolynomial-size lower bounds against TC^0, constant-depth, polynomial-size circuits of AND, OR, NOT and MAJ gates with unbounded fan-in—a circuit class far weaker than polynomial-depth, polynomial-size circuits.[2] Additionally, Aaronson and Wigderson [AW09] identified a third barrier, the algebrization barrier, another hurdle any new lower bound technique must overcome.

The gold standard in circuit complexity is new lower bound techniques that get around the known barriers [BGS75, RR97, AW09]. One shining example is Williams' algorithmic method, which was used to prove ACC^0 lower bounds [Wil14].[3] However, new techniques are few and far between. In this work, we focus instead on the capabilities of our existing techniques, hoping to shed light on where they break down and what new techniques may look like. This continues a line of work initiated by Kumar [Kum23], and, at a high level, we want to answer the question: *What is the largest circuit class for which our existing techniques are useful for proving circuit lower bounds?*

Several reasons motivate this question. An immediate consequence is obtaining lower bounds for more powerful circuits. This was the primary motivation of Yao [Yao89], who observed that many circuit lower bounds extend to circuits augmented with "local computation": bounded fan-in gates that can compute arbitrary functions. Additionally, when a lower bound technique remains useful for a more powerful circuit class, it can also be viewed as a limitation of the technique—a view taken by Chen, Hirahara, Oliveira, Pich, Rajgopal, and Santhanam [CHO+22] who also studied circuits augmented with local computation. For example, if an AC^0 lower bound technique lifts to a larger circuit class C, it suggests that the lower bound technique is exploiting a property that isn't unique to AC^0 but rather the larger class C. By characterizing this class and understanding this property more precisely, one can hope to refine the lower bound technique to prove stronger lower bounds

[1] MOD_p outputs 0 iff the sum of the input bits is congruent to 0 (mod p).

[2] MAJ outputs 1 iff at least half of the input bits are 1.

[3] ACC^0 is the union of $\mathsf{AC}^0[m]$ for all m.

against AC^0. Indeed, the fact that the lower bound technique is useful against C indicates that the technique is too unrefined to address AC^0 specifically. Looking ahead, a conceptual contribution of this work is to propose a broader notion of locality—specifically, arbitrary computation within small Hamming balls—as the property exploited by both the switching lemma and the polynomial method. This generalizes the more conventional notion of locality—arbitrary computation over a few input bits—as used in [Yao89] and [CHO+22].

Kumar [Kum23] used a switching lemma to prove exponential-size lower bounds for GC^0, a circuit class that not only contains AC^0 but can—with a *single* gate—compute functions that require exponential-size TC^0 circuits. Informally, GC^0 is the class of constant-depth unbounded-fan-in circuits whose gates can behave arbitrarily on inputs with sufficiently small Hamming weight, but must otherwise be constant. GC^0 captures natural circuit classes, such as circuits of biased linear threshold gates. Conceptually, Kumar showed that switching lemmas are just as effective at proving lower bounds against circuits that are exponentially more powerful than AC^0.

The main contribution of this work is to show analogous results for the polynomial and algorithmic methods, which can prove lower bounds on circuit classes that provably do not satisfy a switching lemma. We prove exponential-size circuit lower bounds for $\mathsf{GC}^0[p]$ (GC^0 with MOD_p gates) and GCC^0 (the union of $\mathsf{GC}^0[m]$ over all m). These circuit classes are much stronger than $\mathsf{AC}^0[p]$ and ACC^0, respectively. In particular, our results give the least restricted classes of non-monotone circuits for which we have exponential-size circuit lower bounds for explicit functions (see Remark 1.6 for further detail).

Our results can also be interpreted as a barrier result, like relativization, naturalization, and algebrization [BGS75, RR97, AW09]. Specifically, if one can implement a function f with size-s $\mathsf{GC}^0[p]$ circuits, then this implies that one cannot prove better than size-s lower bounds against $\mathsf{AC}^0[p]$ for f using the Razborov-Smolensky polynomial method or the switching lemma.[4] This could shed light on why we still don't have tight lower bounds for MAJ against $\mathsf{AC}^0[p]$.

Our lower bounds have several applications, including proving *unconditional separations* between constant-depth quantum circuits and the classical circuit classes studied in this work. Our quantum-classical circuit separations subsume all prior work, which we discuss further in Section 1.2.2. To dive deeper into our results, we must introduce the $\mathsf{G}(k)$ gate.

1.1 The Main Character: The $\mathsf{G}(k)$ Gate

Kumar [Kum23] introduced the $\mathsf{G}(k)$ gate, an unbounded fan-in gate with the following behavior. If the Hamming weight of the input is at most k, then the $\mathsf{G}(k)$ gate can compute any function f of the circuit designer's choosing. Otherwise, the $\mathsf{G}(k)$ gate outputs a constant $c \in \{0,1\}$ of the designer's choosing.

The value of k controls the power of the $\mathsf{G}(k)$ gate. When k is a constant, it is easy to see that a single $\mathsf{G}(k)$ gate can be computed by a depth-two polynomial-size AC^0 circuit. When $k = n$, a single $\mathsf{G}(k)$ gate can compute any function. Much of the landscape between $k = O(1)$ and $k = n$ is not yet understood, which we discuss further in Open Problems.

We also emphasize that the circuit designer can use the $\mathsf{G}(k)$ gate however they like. On the tamer side, the $\mathsf{G}(k)$ gate can, e.g., evaluate parity on k bits or majority on $2k$ bits, and, on the wilder side, it can, e.g., evaluate uncomputable functions like the halting function or the busy beaver function (with the caveat that it must output a constant if the Hamming weight of the input is above k).

[4]For if one could prove a size-s lower bound against $\mathsf{AC}^0[p]$, our results imply that they lift to a size-s lower bound against $\mathsf{GC}^0[p]$, a contradiction.

While seemingly exotic, $\mathsf{G}(k)$ gates capture natural gates as special cases. For example, $\mathsf{G}(k)$ gates naturally generalize AND and OR gates to biased linear threshold gates. Let $\theta, w_1, \ldots, w_n \in \mathbb{R}$, with the w_i's sorted such that $|w_1| \le |w_2| \le \cdots \le |w_n|$. Let $f(x) = \operatorname{sgn}(\sum_{i=1}^n w_i x_i - \theta)$. If $\sum_{i>k}|w_i| - \sum_{i\le k}|w_i| < |\theta|$, then f can be computed by a $\mathsf{G}(k)$ gate. Kumar [Kum23] showed that circuits comprised of biased linear threshold gates interpolate between AC^0 and TC^0 as the parameter k varies. We note that there is a connection between linear threshold functions and neural networks that dates back to the 1940s [MP43], and there is a precise connection between feed-forward neural networks and TC^0 circuits [Mur71, MSS91] (see also [AGS21, Section 2.5.1]). Circuits with $\mathsf{G}(k)$ gates capture a subset of neural networks whose activation functions are *biased* linear threshold functions.

Let $\mathsf{GC}^0(k)$ denote constant-depth, unbounded fan-in circuits of AND, OR, NOT, and $\mathsf{G}(k)$ gates.[5] Kumar [Kum23] used a novel switching lemma to prove that constant-depth-d $\mathsf{GC}^0(0.1n^{1/d})$ circuits require size $2^{\Omega(n^{1/d})}$ to compute the parity of n bits, which *nearly match* the $2^{\Omega(n^{1/(d-1)})}$ size lower bound for depth-d AC^0 obtained via the AC^0 switching lemma.[6] Observe that, with $k = 0.1n^{1/d}$, a *single* $\mathsf{G}(k)$ gate can compute parity on $0.1n$ bits, which is known to require *exponential-size* AC^0 circuits. (In fact, we show in Theorem 5.39 that, in this regime of k, a single $\mathsf{G}(k)$ gate can compute functions requiring exponential size TC^0 circuits!) To summarize, Kumar proved that the exponential-size lower bounds for depth-d AC^0 established via the switching lemma lift with *no loss in parameters* to depth-d $\mathsf{GC}^0(0.1n^{1/d})$ circuits, which not only contain AC^0 but can compute functions with a single gate that require exponential-size TC^0 circuits.

1.2 Our Results

Following the bottom-up approach to circuit lower bounds, it is natural to ask if we can still show strong lower bounds if we add MOD_2 gates to GC^0 because MOD_2 was the hard function used to establish exponential-size lower bounds for GC^0. Formally, let $\mathsf{GC}^0(k)[m]$ be $\mathsf{GC}^0(k)$ circuits with the additional ability to apply MOD_m gates. Define $\mathsf{GCC}^0(k) \coloneqq \bigcup_{m\in\mathbb{N}} \mathsf{GC}^0(k)[m]$. Note that these circuit classes are $\mathsf{AC}^0[m]$ and ACC^0, respectively, with the additional ability to apply $\mathsf{G}(k)$ gates. In what follows, p and q are arbitrary prime numbers.

$\mathsf{GC}^0(k)[p]$ and $\mathsf{GCC}^0(k)$ not only contain $\mathsf{AC}^0[p]$ and ACC^0, respectively, but are exponentially more powerful. As previously mentioned, we have that for large enough k, a single $\mathsf{G}(k)$ gate can compute functions that require exponential-size $\mathsf{AC}^0[p]$ and ACC^0 circuits, and, by our Theorem 5.38, can compute functions outside of $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$.

Unfortunately, $\mathsf{AC}^0[p]$ and ACC^0 provably cannot have a switching lemma (MOD_p does not simplify under restriction), so the work of [Kum23] does not imply whether lower bounds for these circuit classes can lift (in any sense) to $\mathsf{GC}^0[p]$ or GCC^0. The polynomial method of Razborov and Smolensky [Raz87, Smo87] was developed to prove lower bounds against $\mathsf{AC}^0[p]$, and the algorithmic method of Williams [Wil14] was developed to prove lower bounds for ACC^0. Can lower bounds achieved by these stronger techniques lift to their GC^0 counterpart?

We show that the Razborov-Smolensky polynomial method lifts *without loss in parameters*. As a consequence, we establish $2^{\Omega(n^{1/2(d-1)})}$-size lower bounds for depth-d $\mathsf{GC}^0(.1n^{1/2d})[p]$, matching the best-known lower bounds for $\mathsf{AC}^0[p]$. We also show that lower bounds against ACC^0 established via the algorithmic method lift to $\mathsf{GCC}^0(n^\varepsilon)$, although with some loss in parameters. Hence, all

[5] The AND, OR, and NOT gates are superfluous, but we want to emphasize that $\mathsf{GC}^0(k)$ is a natural generalization of AC^0. To see this, observe that NOT has fan-in 1, so for all $k \in \mathbb{N}$, $\mathsf{G}(k)$ gates can compute the NOT gate. Next, observe that, for all $k \in \mathbb{N}$, $\mathsf{G}(k)$ can compute OR, i.e., the function that is constantly 1 on all inputs of Hamming weight ≥ 1. Finally, with NOT and OR one can compute AND via De Morgan's law.

[6] Kumar also showed that constant-depth-d size-$2^{O(n^{1/d})}$ $\mathsf{GC}^0(0.1n^{1/d})$ circuits suffice to compute parity.

known lower bound techniques for unbounded fan-in AND/OR gates generalize to unbounded fan-in $\mathsf{G}(k)$ gates. This is extremely surprising to us because the combinatorial switching lemma, algebraic polynomial method, and algorithmic method are all incomparable techniques of different flavors, yet they all generalize to $\mathsf{G}(k)$ gates, with the first two techniques doing so losslessly. We discuss the details of these lower bounds and a few of their applications.

1.2.1 Our Classical Results

Polynomial Method Our first result uses the polynomial method to prove exponential-size lower bounds for $\mathsf{GC}^0(k)[p]$.

Theorem 1.1 ($\mathsf{GC}^0(k)[p]$ lower bound, Restatement of Corollary 4.3)**.** *Let p and q be distinct prime numbers, and let $k = O(n^{1/2d})$. Any constant-depth-d $\mathsf{GC}^0(k)[p]$ circuit that computes either MAJ or MOD_q on n input bits must have size $2^{\Omega\left(n^{1/2(d-1)}\right)}$.*

Notice that our lower bound exactly matches the best-known lower bound of $2^{\Omega(n^{1/2(d-1)})}$ for depth-d $\mathsf{AC}^0[p]$. In contrast, the naïve approach of simulating $\mathsf{GC}^0(k)$ in AC^0 and then using AC^0 lower bounds would incur a significant loss in parameters. In more detail, one can convert a depth-d size-$2^{\Theta(n^{1/(4d-2)}/k)}$ $\mathsf{GC}^0(k)[p]$ circuit into a depth-$2d$ size-$2^{O(n^{\frac{1}{2(2d-1)}})}$ $\mathsf{AC}^0[p]$ circuit because each $\mathsf{G}(k)$ gate of fan-in n can be expanded as a CNF of size $n^{O(k)}$. Then, by applying the known $\mathsf{AC}^0[p]$ lower bound, one would get a $2^{\Theta(n^{1/(4d-2)}/k)}$-size lower bound for $\mathsf{GC}^0(k)[p]$. However, this is far weaker than our Theorem 1.1. Even for constant k, the naïve approach is not as large as $2^{\Omega(n^{1/2d})}$, and, when $k \geq n^{1/(4d-2)}$, the naïve approach no longer gives nontrivial lower bounds. On the other hand, Theorem 1.1 gives us size $2^{\Omega(n^{1/2(d-1)})}$ lower bounds for k up to $O(n^{1/2d})$. We emphasize that the size $n^{O(k)}$ blow up from converting $\mathsf{G}(k)$ gates to CNFs/DNFs is necessary by a Shannon counting argument.

Our lower bound follows the same roadmap as the Razborov-Smolensky lower bound for $\mathsf{AC}^0[p]$ [Raz87, Smo87], which has two parts: (1) show that $\mathsf{AC}^0[p]$ can be approximated by a low-degree $\mathbb{F}_p$ polynomial, and (2) any $\mathbb{F}_p$ polynomial computing MAJ or MOD_q for a prime $q \neq p$ must have high degree.

We re-use the second part of this roadmap without modification. Hence, the challenge is to show that $\mathsf{GC}^0(k)[p]$ circuits can be approximated by low-degree $\mathbb{F}_p$ polynomials. More precisely, because the $\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}$, and MOD_p gates are handled by the original lower bound, our contribution is to show that $\mathsf{G}(k)$ gates can be approximated by low-degree $\mathbb{F}_p$ polynomials. At the core of this is the following lemma that says any $\mathsf{G}(k)$ gate can be computed by a probabilistic polynomial of extremely low degree (Definition 3.3).

Lemma 1.2 (Restatement of Lemma 3.6)**.** *For any prime p and $\mathsf{G}(k)$ gate G of fan-in n, there is an ε-probabilistic $\mathbb{F}_p$-polynomial of degree $O(k + \log(1/\varepsilon))$ computing G.*

We also show that Lemma 1.2 is optimal.

Lemma 1.3 (Restatement of Lemma 3.7)**.** *There exists a $\mathsf{G}(k)$ gate that requires probabilistic degree $\Omega(k + \log(1/\varepsilon))$.*

Our lower bounds lift losslessly from $\mathsf{AC}^0[p]$ to $\mathsf{GC}^0(k)[p]$ because the degree in Lemma 1.2 is sufficiently small—anything less than optimal would not suffice! If the degree was, say, $O(k\log(1/\varepsilon))$ instead, our lower bounds in Theorem 1.1 would be far weaker than the state-of-the-art lower bounds for $\mathsf{AC}^0[p]$.[7]

[7] The specific size lower bound one would obtain is $\exp(\Omega(n^{1/2d}/k))$, which does improve over the naïve $\exp(\Omega(n^{1/4d}/k))$ bound discussed earlier, but still incurs the $1/k$ loss in the exponent.

We use Lemma 1.2 to prove that $\mathsf{GC}^0(k)[p]$ can be approximated by proper $\mathbb{F}_p$ polynomials (i.e., polynomials that have Boolean outputs when the inputs are Boolean, see Definition 3.1).

Theorem 1.4 (Restatement of Theorem 3.8)**.** *Let p be a prime. For any constant-depth-d size-s $\mathsf{GC}^0(k)[p]$ circuit, there exists an proper polynomial $q(x) \in \mathbb{F}_p[x_1, \dots, x_n]$ with*

$$\deg(q) \le O\left(k^d + \log^{d-1} s\right)$$

such that

$$\Pr_{x \sim U_n}[q(x) \neq C(x)] \le 0.1.$$

Combining Theorem 1.4 with the well-known fact that any $\mathbb{F}_p$ polynomial approximating MAJ or MOD_q must have degree $\Omega(\sqrt{n})$ yields our Theorem 1.1.

We improve upon our worst-case lower bound by giving an average-case lower bound for $\mathsf{GC}^0(k)[p]/\mathsf{rpoly}$. Recall that $/\mathsf{rpoly}$ denotes that the circuit gets random advice, i.e., the circuit draws one sample from a probability distribution of choice on polynomially many bits where the distribution is independent of the input (but does depend on the input size).

Theorem 1.5 (Average-case lower bound for $\mathsf{GC}^0(k)[p]$, Restatement of Theorem 4.4)**.** *Let p and q be distinct prime numbers, let $k = O(n^{1/2d})$, and let $F \in \{\mathsf{MOD}_q, \mathsf{MAJ}\}$. There exists an input distribution on which any $\mathsf{GC}^0(k)[p]/\mathsf{rpoly}$ circuit of depth d and size at most $\exp\left(O\left(n^{1/2.01d}\right)\right)$ only computes F with probability $\frac{1}{2} + \frac{1}{n^{\Omega(1)}}$.*

Remark 1.6. We remark on how our contribution fits into the goal of proving superpolynomial-size lower bounds for TC^0, a problem that has been open for over forty years. Following notation from Kumar [Kum23], let $\mathsf{TC}^0(k)[p]$ denote the class of circuits with AND, OR, NOT, MOD_p, and the biased linear threshold gates discussed in Section 1.1. Since $\mathsf{TC}^0(k)[p]$ is a subclass of $\mathsf{GC}^0(k)[p]$, our Theorem 1.5 implies average-case hardness against exponential-size $\mathsf{TC}^0(\Omega(n^{1/2d}))[p]$ circuits. Additionally, $\mathsf{TC}^0(k)[p]$ lies between $\mathsf{AC}^0[p]$ and TC^0. Therefore, Theorems 1.1 and 1.5 represent the closest circuit class to TC^0 for which we have exponential-size lower bounds.

Algorithmic Method In a celebrated result, Williams [Wil14] used the algorithmic method to prove that there are languages in E^{NP} and NEXP that fail to have exponential-size ACC^0 circuits. Recall that E^{NP} is the class of languages that can be decided by a Turing machine in time $2^{O(n)}$ with access to an NP oracle.

For a circuit class C, C-CircuitSAT is the problem where we're given as input a description of a circuit $C \in \mathsf{C}$, and we must decide whether there exists an input $x \in \{0,1\}^n$ such that $C(x) = 1$. At a high level, the algorithmic method shows that fast algorithms for C-CircuitSAT imply circuit lower bounds for C.

Williams showed a $2^{n^{1/f_{\mathsf{ACC}}(d)}}$-size ACC^0 lower bound against E^{NP}, for some function $f_{\mathsf{ACC}}(d) \in 2^{O(d)}$. In this work, we prove that there are languages in E^{NP} that fail to have exponential-size $\mathsf{GCC}^0(k)$ circuits. A similar result can be shown for NEXP, but we focus on E^{NP} because we get a stronger size-depth tradeoff.

Theorem 1.7 ($\mathsf{E}^{\mathsf{NP}} \not\subseteq \mathsf{GCC}^0(k)$, Restatement of Theorem 4.13)**.** *Let $f_{\mathsf{ACC}} \in 2^{O(d)}$ be the function defined above. For every constant d, there is a constant $C > 1$ such that for all $k \le O(n^{1/Cf_{\mathsf{ACC}}(d)}/\log n)$, there exists a language in E^{NP} that fails to have $\mathsf{GCC}^0(k)$ circuits of depth d and size $\exp\left(\Omega(n^{1/Cf_{\mathsf{ACC}}(d)}/k)\right)$.*

We again note that if we expanded out our depth-d size-s $\mathsf{GCC}^0(k)$ circuit as a depth-$2d$ size-$s^{O(k)}$ ACC^0 circuit (by converting $\mathsf{G}(k)$ gates to CNFs), and applied Williams' bound, we would yield only a size $\exp(\Omega(n^{1/Cf_{\mathsf{ACC}}(2d)}/k))$ lower bound. Hence our lemma lifts lower bounds from ACC^0 to GCC^0 without incurring the loss caused by the doubling in depth. This is quite substantial as the size lower bound has a triply exponential dependence on d. Unfortunately, our bound suffers a $1/k$ dependence in the exponent, so the possibility of losslessly lifting ACC^0 bounds remains open.

Our GCC^0 lower bound follows from a fast algorithm for $\mathsf{GCC}^0(k)$-CircuitSAT, which we obtain by generalizing Williams' fast algorithm for ACC^0-CircuitSAT.

Theorem 1.8 ($\mathsf{GCC}^0(k)$-CircuitSAT algorithm, Restatement of Theorem 4.12)**.** *For every $d > 1$ and $\varepsilon = \varepsilon(d) := .99/f_{\mathsf{ACC}}(d/2)$, the satisfiability of depth-d $\mathsf{GCC}^0(k)$ circuits with n inputs and $2^{n^\varepsilon/k}$ size can be determined in time $2^{n-\Omega(n^\delta/k)}$ for some $\delta > \varepsilon$.*

Let SYM^+ be the class of depth-two circuits where a layer of AND gates feed into a gate computing some symmetric function. The ACC^0-CircuitSAT algorithm works by giving an algorithm to transform an ACC^0 circuit into a SYM^+ circuit and then running a SYM^+-CircuitSAT algorithm. Our $\mathsf{GCC}^0(k)$-CircuitSAT algorithm works similarly.

The key ingredient to transform a $\mathsf{GCC}^0(k)$ circuit to a SYM^+ circuit is a randomness-efficient probabilistic circuit (Definition 2.2) that computes $\mathsf{G}(k)$ gates. Our probabilistic polynomial construction in Lemma 1.2 uses $\mathrm{poly}(n)$ random bits to construct a probabilistic polynomial of degree $O(k)$ for any $\mathsf{G}(k)$ gate. This construction uses too many random bits, and attempting to use it in the algorithmic method would yield a GCC^0-CircuitSAT algorithm that is too slow. Furthermore, the randomness is used in a complicated manner, making it unclear how to convert it into a probabilistic circuit.

Instead, we construct a probabilistic circuit of degree $O(k^2 \log^2 n)$ for any $\mathsf{G}(k)$ gate that uses $O(k^2 \log^2 n)$ random bits, generalizing a work of Allender and Hertrampf [AG94]. So, at the cost of a $\mathrm{poly}(k, \log n)$ factor in the degree, we exponentially improve the randomness efficiency.

Theorem 1.9 (Restatement of Theorem 3.12)**.** *Let q be a prime. Any $\mathsf{G}(k)$ gate on n bits can be computed by a depth-2 probabilistic circuit using $O(k^2 \log^2 n \log(1/\varepsilon))$ random bits, and consists of a MOD_q of fan-in $2^{O(k^3 \log^2 n \log(1/\varepsilon))}$ at the top, and AND gates of fan-in $O(k^3 \log^2 n \log(1/\varepsilon))$ at the bottom layer. Furthermore, the circuit can be constructed in $2^{O(k^3 \log^2 n \log(1/\varepsilon))}$ time.*

To summarize, Theorem 1.9 is the key ingredient to transform $\mathsf{GCC}^0(k)$ circuits into SYM^+ circuits. The details of this transformation are given in Theorem 4.10. Then the fast algorithm in Theorem 1.8 follows from running a SYM^+-CircuitSAT algorithm on the transformed circuit. Finally, our lower bound (Theorem 1.7) is implied by the fast algorithm via the algorithmic method.

PAC Learning $\mathsf{GC}^0(k)[p]$ Using the framework of Carmosino, Impagliazzo, Kabanets, and Kolokolova [CIKK16], we give a quasipolynomial time learning algorithm for $\mathsf{GC}^0(k)[p]$ in the PAC model over the uniform distribution with membership queries (Definition 4.14).

Corollary 1.10 (Learning $\mathsf{GC}^0(k)[p]$ in quasipolynomial time, Restatement of Corollary 4.18)**.** *For every prime p and $k = O(n^{1/2d})$, there is a randomized algorithm that, using membership queries, learns a given n-variate Boolean function $f \in \mathsf{GC}^0(k)[p]$ of size s_f to within error ε over the uniform distribution, in time* $\mathrm{quasipoly}(n, s_f, 1/\varepsilon)$.

Using circuit lower bounds to obtain learning algorithms dates back to the seminal work of Linial, Mansour, and Nisan [LMN93] where they gave a quasipolynomial time learning algorithm for AC^0 in the PAC model over the uniform distribution (hereafter, the "LMN algorithm"). One can interpret the LMN algorithm as exploiting the existence of a natural property that is useful against AC^0 (in the sense of Razborov and Rudich [RR97], see Definition 4.15).

Carmosino, Impagliazzo, Kabanets, and Kolokolova [CIKK16] prove that for any circuit class C containing AC^0, a natural property that is useful against C implies a quasipolynomial time learning algorithm in the PAC model over the uniform distribution with membership queries. It is not hard to show that our $\mathsf{GC}^0(k)[p]$ lower bound is natural.

Theorem 1.11 (Informal version of Theorem 4.16)**.** *For every prime p and $k = O(n^{1/2d})$, there is a natural property of n-variate Boolean functions that is useful against $\mathsf{GC}^0(k)[p]$ circuits of depth d and of size up to $\exp\left(\Omega(n^{1/2d})\right)$.*

A New Multi-Output Multi-Switching Lemma For $\mathsf{GC}^0(k)$ In Section 1.2.2, we explain new separations between quantum and classical circuits. One of our separations uses a new multi-switching lemma for $\mathsf{GC}^0(k)$ tailored to handle circuits with multiple outputs. The details of the switching lemma are somewhat involved, and we refer the interested reader to Section 5.2.1 for details. The general switching lemma is stated in Theorem 5.20. We heavily rely on Kumar's multi-switching lemma [Kum23], which we use in a black-box manner.

This new switching lemma lets us show that $\mathsf{GC}^0(k)$ circuits have exponentially small correlation with a search problem that can be solved by a constant-depth quantum circuit. If we just used Kumar's switching lemma for $\mathsf{GC}^0(k)$ [Kum23], we could obtain a similar separation but the correlation would be larger.

$\mathsf{G}(k)$ Is Incomparable to $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$ For a different quantum-classical separation, we need to prove that $\mathsf{G}(k)$ gates are incomparable to certain classical circuit classes. We prove that, for large enough k, a single $\mathsf{G}(k)$ gate can compute functions that are not in NC^i or even $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$, which we think is interesting in its own right.

Theorem 1.12 (Restatement of Theorem 5.38)**.** *There is a function $f : \{0,1\}^n \to \{0,1\}$ computable by a single $\mathsf{G}(k)$ gate that is not computable in NC^i for any constant i and $k = \omega(\log^{i-1}(n))$. When $k \in \log^{\omega(1)}(n)$, then there are functions $f : \{0,1\}^n \to \{0,1\}$ that are computable by a single $\mathsf{G}(k)$ gate that cannot be computed in $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$.*

Notably, all of our lower bounds for $\mathsf{GC}^0(k)$, $\mathsf{GC}^0(k)[p]$, and $\mathsf{GCC}^0(k)$ hold in a regime of k where they are incomparable to $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$.

We can similarly show that a single $\mathsf{G}(k)$ gate requires exponential-size TC^0 circuits. We find this interesting in its own right, because proving lower bounds for TC^0 is currently beyond our techniques.

Theorem 1.13 (Informal version of Theorem 5.39)**.** *There is a function $f : \{0,1\}^n \to \{0,1\}$ computable by a single $\mathsf{G}(k)$ gate that requires exponential-size TC^0 circuits for $k = \Omega(n^{\varepsilon})$ for constant $\varepsilon > 0$.*

1.2.2 Exponential Separations Between Quantum and Classical Circuits

A central goal in quantum complexity theory is to identify problems that are tractable for quantum computers but intractable for classical ones. One way to formalize this goal is to show that BQP (Bounded-Error Quantum Polynomial Time) strictly contains P (Polynomial Time). Alas, even showing that P is strictly contained in PSPACE is currently beyond the reach of complexity theory.

One can separate BQP from P *conditionally*, for example, under the assumption that there is no polynomial-time algorithm for factoring integers [Sho97, Reg24]. There is also a long line of research that separates quantum and (randomized) classical computation in the black-box model [BV97, Sim97, AA15].

Yet another option (and the one that is most relevant to this work) is to look at restricted models of computation. In a groundbreaking work, Bravyi, Gosset, and König [BGK18] exhibited a search problem that is solvable by QNC^0 (constant-depth bounded-fan-in quantum circuits), but is hard for NC^0 (constant-depth bounded-fan-in classical circuits). This is an *unconditional separation* between a quantum and classical model of computation.

Since then, there has been a lot of progress [WKST19, LG19, CSV19, GS20, BGKT20, GJS21, GKMdO24]. We briefly summarize the state of the art prior to this work: there is a decision problem that separates $\mathsf{BQLOGTIME}$ (Definition 5.1) and quasipolynomial-size AC^0 [RT22]; a search problem that separates QNC^0 and exponential-size AC^0 [WKST19]; and a search problem that separates $\mathsf{QNC}^0/\mathsf{qpoly}$ and polynomial-size $\mathsf{AC}^0[p]$ for any prime p [WKST19, GKMdO24]. Recall that $\mathsf{QNC}^0/\mathsf{qpoly}$ is the class of constant-depth bounded-fan-in quantum circuits that start with a quantum advice state, i.e., an input-independent quantum state of choice. Grier and Schaeffer [GS20] also obtain a separation between QNC^0 and exponential-size $\mathsf{AC}^0[p]$ for an interactive problem. Finally, Bravyi, Gosset, König, and Temamichel [BGKT20] and Grier, Ju, and Schaeffer [GJS21] showed that these separations hold even when the quantum circuits are subject to certain types of noise.[8]

Building on this line of work, we can subsume all previously known separations between quantum and classical circuits. We re-use the problems used to obtain the above quantum-classical separations; our contribution is to strengthen all of the lower bounds to hold for $\mathsf{GC}^0(k)$ or $\mathsf{GC}^0(k)[p]$. All of our separations are exponential, meaning that the problems can be solved with polynomial-size quantum circuits but require exponential-size classical circuits. Previously the best separation between $\mathsf{QNC}^0/\mathsf{qpoly}$ and polynomial-size $\mathsf{AC}^0[p]$ circuits. In the remainder of this subsection, we state our separations in more detail.

$\mathsf{BQLOGTIME}$ vs. GC^0 In Section 5.1, we exhibit a promise problem that separates $\mathsf{BQLOGTIME}$ from $\mathsf{GC}^0(k)$.

Theorem 1.14 (Restatement of Corollary 5.7)**.** *There is a promise problem in* $\mathsf{BQLOGTIME}$ *(Definition 5.1) that is not solvable by constant-depth* $\mathsf{GC}^0(k)$ *for* $k = \frac{O(n^{1/4d})}{(\log n)^{\omega(1)}}$ *and size* $\mathrm{quasipoly}(n)$.

By well-known reductions, this implies an oracle relative to which BQP is not contained in the class of languages decided by uniform GC^0 circuit families.

Corollary 1.15 (Restatement of Corollary 5.8)**.** *There is an oracle relative to which* BQP *is not contained in the class of languages decidable by uniform families of circuits* $\{C_n\}$*, where for all* $n \in \mathbb{N}$*,* C_n *is a size-*$2^{n^{O(1)}}$ *depth-*d $\mathsf{GC}^0(k)$ *circuit with* $k \in \frac{2^{n/4d}}{n^{\omega(1)}}$.

Raz and Tal [RT22] showed that $\mathsf{BQLOGTIME} \not\subseteq \mathsf{AC}^0$, which implied an oracle relative to which BQP is not contained in the class of languages decided by uniform families of size-$2^{n^{O(1)}}$ constant-depth AC^0 circuits. It is well-known that this class is precisely the polynomial hierarchy PH. Hence, because $\mathsf{GC}^0(k)$ contains AC^0 (and can even compute functions that require exponential-size AC^0 circuits), Corollary 1.15 is a generalization of the relativized separation of BQP and PH.

[8] Watts and Parham [WP24] also studied unconditional separations for input-independent sampling problems. In this work, we focus on computational problems that have inputs and outputs.

One reason Raz and Tal [RT22] is such a striking result is that it shows even the enormous power of PH fails to simulate quantum computation in a relativizing way. This is made more precise in the beautiful follow-up work of Aaronson, Ingram, and Kretschmer [AIK22] who show (among many other results) that there is an oracle relative to which $\mathsf{P} = \mathsf{NP}$ but $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$. In words, they show that even in a world where NP is easy, BQP can still be extremely powerful. Our oracle separation result complements these results (and relies on Raz and Tal).

We give one concrete implication of Corollary 5.8. Namely, we show that there is an oracle relative to which BQP is outside of hierarchies of counting classes, where the counting classes can count whether there are a small number of accepting witnesses. This is perhaps surprising because $\mathsf{BQP} \subseteq \mathsf{PP}$ relative to all oracles [ADH97]. Hence, we show that it is actually necessary to count a larger number of witnesses to simulate BQP in a relativizing way. The counting classes are defined in Definitions 5.9 and 5.10, and the oracle separation is given in Corollary 5.11.

QNC^0 vs. GC^0 In Section 5.2, we exhibit a search problem that separates QNC^0 from $\mathsf{GC}^0(k)$. Our separation is based on the 2D Hidden Linear Function (2D HLF) problem (Definition 5.12) introduced by Bravyi, Gosset, and König [BGK18].

Theorem 1.16 (Restatement of Theorem 5.15)**.** *The* 2D HLF *problem (Definition 5.12) on n bits cannot be solved by a constant-depth-d size-*$\exp(O(n^{1/4d}))$ $\mathsf{GC}^0(k)$ *circuit with $k = O(n^{1/4d})$. Furthermore, for the same value of k, there exists an (efficiently samplable) input distribution on which any $\mathsf{GC}^0_d(k)$ circuit (or $\mathsf{GC}^0_d(k)/\mathsf{rpoly}$ circuit) of size at most $\exp(n^{1/4d})$ only solves the* 2D HLF *problem with probability at most $\exp(-n^c)$ for some $c > 0$.*

Theorem 1.16 generalizes the separation between QNC^0 and AC^0 obtained by Watts, Kothari, Schaeffer, and Tal [WKST19]. The proof requires a new multi-output multi-switching lemma for $\mathsf{GC}^0(k)$, which we prove in Section 5.2.1.

Using the frameworks developed by Bravyi et al. [BGKT20] and Grier et al. [GJS21], we show in Theorem 5.23 that this separation holds even when the quantum circuits are subjected to certain types of noise.

$\mathsf{QNC}^0/\mathsf{qpoly}$ vs. $\mathsf{GC}^0[p]$ In Sections 5.3 and 5.4, we exhibit a family of search problems that separates $\mathsf{QNC}^0/\mathsf{qpoly}$ from $\mathsf{GC}^0(k)[p]$ for all primes p. The family of search problems is a generalization of the Parity Bending problem introduced by Watts, Kothari, Schaeffer, and Tal [WKST19] and was also studied in a recent work of Grilo, Kashefi, Markham, and Oliveira [GKMdO24].

Theorem 1.17 (Restatement of Theorem 1.17)**.** *For any prime p, there is a search problem that is solvable by $\mathsf{QNC}^0/\mathsf{qpoly}$ with probability $1 - o(1)$, but any $\mathsf{GC}^0(k)[p]/\mathsf{rpoly}$ circuit of depth d and size at most $\exp\left(O(n^{1/2.01d})\right)$ with $k = O(n^{1/2d})$ cannot solve the search problem with probability exceeding $n^{-\Omega(1)}$.*

Previously the best separations were between polynomial-size QNC^0 and *polynomial-size* $\mathsf{AC}^0[p]$ obtained in the works of Watts et al. [WKST19] and Grilo et al. [GKMdO24]. Our Theorem 1.17 is a separation between polynomial-size QNC^0 and *exponential-size* $\mathsf{GC}^0(k)[p]$.

Interactive QNC^0 vs. $\mathsf{GC}^0(k)[p]$ Grier and Schaeffer [GS20] studied quantum-classical separations that can be obtained in certain interactive models. Among some conditional results, they obtain an unconditional separation between QNC^0 and $\mathsf{AC}^0[p]$ for all primes p. We generalize their separation to $\mathsf{GC}^0(k)[p]$.

Theorem 1.18 (Restatement of Theorem 5.41)**.** *Let $k = O(n^{1/2d})$. There is an interactive task that QNC^0 circuits can solve that depth-d, size-s $\mathsf{GC}^0(k)[p]$ circuits cannot for $s \leq \exp\left(O(n^{1/2.01d})\right)$.*

Concurrent Work An independent and concurrent work of Hsieh, Mendes, Oliveira, and Subramanian [HMdOS24] overlaps with our work in one way. They give an exponential separation between $\mathsf{GC}^0(k)$ and QNC^0, which is essentially the same as our separation (Theorem 5.15).[9] Like us, they also prove a new muli-output multi-switching lemma for $\mathsf{GC}^0(k)$ (Theorem 5.20) to obtain their separation. The similarity in our arguments comes from the fact that we both use the exponential separation between AC^0 and QNC^0 of Watts, Kothari, Schaeffer, and Tal [WKST19] as a starting point. Watts et al. proved a similar multi-output multi-switching lemma for AC^0, building on the prior work of Rossman [Ros17].

Hsieh et al. also show that their separation holds if the quantum circuits are subjected to a certain noise model, which we also do in Theorem 5.23. This noise-robustness result follows from applying the framework introduced by Bravyi, Gosset König, and Temamichel [BGKT20] and further developed by Grier, Ju, and Schaeffer [GJS21]. Hsieh et al. also study extending this framework to prime-dimensional qudits.

1.3 Open Problems

There are several important directions for future work. We begin by discussing some of the broader directions of the study of $\mathsf{G}(k)$-related circuit classes and then conclude with some more specific open problems.

Combined with the work of Kumar [Kum23], we now know that AC^0 size lower bounds from the combinatorial technique of switching lemmas, as well as $\mathsf{AC}^0[p]$ lower bounds using the algebraic technique of probabilistic polynomials, both lift *almost losslessly* to GC^0 and $\mathsf{GC}^0(k)[p]$, respectively. It is extremely surprising to us that both techniques, while extremely different in flavor, generalize so cleanly to $\mathsf{G}(k)$ gates. This observation raises many questions about how $\mathsf{G}(k)$ gates can help us understand the limitations of our lower bound techniques.

- Do $\mathsf{G}(k)$ gates exactly capture the switching lemma technique as well as the probabilistic polynomial technique? This would let us know whether there is an even more general class of gates that capture the power of these techniques.
- Can we use $\mathsf{G}(k)$ gates (or its generalizations derived from the last item) to show barrier results for current lower bounds we have? For example, implementing explicit functions in $\mathsf{GC}^0(k)$ or $\mathsf{GC}^0(k)[p]$ would demonstrate a limitation on the size lower bounds achievable for AC^0 or $\mathsf{AC}^0[p]$ via switching lemmas or the polynomial method.
- Can lower bounds for E^{NP} using Williams' algorithmic method be lifted losslessly from ACC^0 to GCC^0 as well?

There are also general questions about how GC^0 and their counterparts fit in the landscape of circuit classes.

- How do $\mathsf{GC}^0(k)$, $\mathsf{GC}^0(k)[p]$, and $\mathsf{GCC}^0(k)$ compare to more classical circuit classes like NC^1 and TC^0? We know that when $k = n$, $\mathsf{GC}^0(k)$ can compute any function, and when $k = 1$, $\mathsf{GC}^0(k) = \mathsf{AC}^0$. What is the smallest k such that $\mathsf{TC}^0 \subset \mathsf{GC}^0(k)$? We know this is true when $k \geq n/2$, but is it true for smaller k? Similar questions can be raised for $\mathsf{GC}^0(k)[p]$.

[9] Hsieh et al. denote $\mathsf{GC}^0(k)$ by $\mathsf{bPTF}^0[k]$.

- Can we get stronger quantum-classical separations? Specifically, can we obtain separations between QNC^0 and $\mathsf{GC}^0(k)[p]$ without giving the quantum circuit an advice state?

- [Kum23] gave a natural subclass of $\mathsf{G}(k)$ consisting of biased linear threshold gates. Are there other natural gates contained in $\mathsf{G}(k)$?

2 Preliminaries

We presume the reader is familiar with common concepts in the theory of computation (circuit complexity and quantum computing, in particular). All prerequisite knowledge can be found in standard textbooks such as [Gol08, AB09, NC02].

We obey the following notation and conventions throughout. For a positive integer n, $[n] \coloneqq \{1, \ldots, n\}$. For us, the natural numbers do not include 0, i.e., $\mathbb{N} \coloneqq \{1, 2, 3, \ldots\}$. Define $\binom{n}{\leq k} \coloneqq \sum_{i=0}^{k} \binom{n}{i}$. For $S \subseteq [n]$ and $x \in \{0,1\}^n$, define $x^S \coloneqq \prod_{i \in S} x_i$. Let quasipoly$(n)$ denote all functions that have an upper bound of the form $2^{O(\log^c n)}$ for some constant c.

We denote the Hamming weight of a string $x \in \{0,1\}^n$ as $|x| = \sum_i x_i$. More generally, for $x \in \mathbb{F}_q^n$ (for some prime q), $|x| = \sum_i x_i \pmod q$. The *Hamming ball of radius* k is the set $\{x \in \{0,1\}^n : |x| \leq k\}$.

For a distribution $\mathcal{D}$ over support S, $x \sim \mathcal{D}$ denotes sampling an $x \in S$ according to the distribution $\mathcal{D}$. For a set S, we denote drawing a sample $s \in S$ uniformly at random by $s \sim S$. U_n denotes the uniform distribution over length-n bit strings. For a distribution $\mathcal{D}$ and a function f, $\mathbf{E}[f(\mathcal{D})] \coloneqq \mathbf{E}_{x \sim \mathcal{D}}[f(x)]$. For two discrete distributions p and q supported on S, the total variation distance (also called the statistical distance) is defined as $\frac{1}{2}\sum_{s \in S} |p(s) - q(s)|$.

We also use Fermat's little theorem.

Theorem 2.1 (Fermat's little theorem)**.** *For any integer $a \not\equiv 0 \pmod p$ for a prime p, $a^{p-1} \equiv 1 \pmod p$.*

All circuit classes studied in this work are constant depth, and d always denotes a constant. Circuits are comprised of layers of gates. When we refer to the "top" of a classical circuit, we are referring to the last layer of the circuit. In particular, for a Boolean-valued circuit, the top is a single gate. The "bottom" of a circuit refers to the first layer of gates.

For an integer m, the MOD_m gate is the unbounded fan-in Boolean gate that outputs 0 iff the sum of the input bits is congruent to $0 \pmod m$. The MAJ gate computes the majority function, i.e., the unbounded fan-in Boolean gate that outputs 1 iff the majority of the input bits are 1. The THR^k gate is the unbounded fan-in Boolean gate that outputs 1 iff the Hamming weight of the input is $> k$.

Recall the following well-studied circuit classes:

- NC^i: $O(\log^i n)$-depth circuits of bounded fan-in AND, OR, and NOT gates.

- AC^i: $O(\log^i n)$-depth circuits of unbounded fan-in AND, OR, and NOT gates.

- $\mathsf{AC}^i[p]$: $O(\log^i n)$-depth circuits of unbounded fan-in AND, OR, NOT, and MOD_p gates.

- ACC^i: The union of $\mathsf{AC}^i[m]$ for all m.

- TC^i: $O(\log^i n)$-depth circuits of unbounded fan-in $\mathsf{AND}, \mathsf{OR}, \mathsf{NOT}$, and MAJ gates.

- QNC^i: $O(\log^i n)$-depth quantum circuits of bounded fan-in quantum gates.

$\mathsf{NC} := \bigcup_i \mathsf{NC}^i$, and AC and TC are defined analogously. It is known that $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$. The size of a circuit is the number of gates in the circuit besides NOT gates. We always specify the circuit size when relevant; however, if the size is not explicitly mentioned, it should be assumed to be polynomial.

A search problem (also called a relation problem or relational problem) is a computational problem with many valid outputs, as opposed to a function problem which only has one valid output for each input. A two-round interactive problem is a computational problem where in the first round you are given an input and produce an output and in the second round, you produce another input and output. The correctness of an interactive algorithm is based on the entire transcript of the interaction, and a computational device solving an interactive problem gets to keep state from the first round during the second round.

In a common abuse of notation we use e.g. AC^0 or $\mathsf{GC}^0(k)[p]$ to interchangeably talk about a type circuit or a class of (decision/relation/interactive) problems, where the context clarifies what we are referring to.

We also will use probabilistic circuits.

Definition 2.2. A *probabilistic circuit* that computes a function $f : \{0,1\}^n \to \{0,1\}$ is a circuit C that takes input $x \in \{0,1\}^n$ and uniformly random bits r, and satisfies the property that for all $x \in \{0,1\}^n$,

$$\Pr_r[C(x,r) \neq f(x)] \leq \varepsilon.$$

3 Approximating $\mathsf{G}(k)$ Gates by Low-Degree Polynomials

We show that any $\mathsf{G}(k)$ gate can be approximated by proper low-degree polynomials. To discuss our results in more detail, we must introduce some terminology.

Definition 3.1 (Proper polynomial)**.** Let q be a prime number. A polynomial $p(x) \in \mathbb{F}_q[x_1, \ldots, x_n]$ is proper when $p(x) \in \{0,1\}$ for all inputs $x \in \{0,1\}^n$.

Definition 3.2 (ε-approximating polynomial)**.** An ε-approximate polynomial for a function $f : \{0,1\}^n \to \{0,1\}$ is a proper polynomial p such that

$$\Pr_{x \sim U_n}[f(x) \neq p(x)] \leq \varepsilon.$$

Definition 3.3 (ε-probabilistic polynomial)**.** An ε-probablistic polynomial of degree d for a function $f : \{0,1\}^n \to \{0,1\}$ is a distribution $\mathcal{P}$ over proper polynomials of degree $\leq d$ such that for every $x \in \{0,1\}^n$,

$$\Pr_{p \sim \mathcal{P}}[p(x) \neq f(x)] \leq \varepsilon.$$

In Section 3.1, we show that $\mathsf{G}(k)$ gates can approximated by low-degree polynomials. As a consequence, we show that any $\mathsf{GC}^0(k)[q]$ circuit can be approximated by low-degree polynomials, generalizing the Razborov-Smolensky polynomial method [Raz87, Smo87] for $\mathsf{AC}^0[q]$ to $\mathsf{GC}^0(k)[q]$. This allows us to prove circuit lower bounds for $\mathsf{GC}^0(k)[q]$; we discuss this application and others in Sections 4 and 5.

In Section 3.2, we construct probabilistic polynomials for $\mathsf{G}(k)$ gates that use very few bits of randomness. The randomness-efficiency of our construction will be essential to invoke the algorithms-to-lower-bounds technique of Williams [Wil14], which we do in Section 4.2.

3.1 Approximating $\mathsf{GC}^0[p]$ by Low-Degree Polynomials

We show that size-s $\mathsf{GC}^0(k)[q]$ circuits can be well-approximated by $\mathbb{F}_q$-polynomials of degree $\mathrm{poly}(k, \log s)$. To do so, we need a standard lemma stating that one can interpolate a truth table on a radius k Hamming ball by a degree-k polynomial. We give a proof for convenience.

Lemma 3.4. *For any $f : \{0,1\}^n \to \mathbb{F}_q$ and prime q, there exists a unique $\mathbb{F}_q$-polynomial p with $\deg(p) \leq k$ such that for all $x \in \{0,1\}^n$ with $|x| \leq k$, $f(x) = p(x)$. Furthermore, this polymomial can be constructed in $n^{O(k)}$ time.*

Proof. Consider the $\mathbb{F}_q$-linear system of equations given by $\sum_{|S|\leq k} c_S a^S = f(a)$ for each $a \in \{0,1\}^n$ such that $|a| \leq k$. These equations are linearly independent, and since the number of equations equals the number of variables, there is a unique set of coefficients $\{c_S\}$ that satisfies this system. Therefore, the polynomial with these coefficients, $p(x) \coloneqq \sum_{|S|\leq k} c_S x^S$, is the desired polynomial. Furthermore, these coefficients can be retrieved in $n^{O(k)}$ time via Gaussian elimination on the $\binom{n}{k}$ linear equations. $\square$

Next, we prove a technical lemma that says there are low-degree probabilistic polynomials for $\mathsf{G}(k)$ gates. Our construction uses probabilistic polynomials for THR^k, where THR^k is the unbounded fan-in gate that outputs 1 iff the Hamming weight of the input is $> k$.

Lemma 3.5 ([STV21, Theorem 3])**.** *For any prime q, there is an ε-probabilistic $\mathbb{F}_q$ polynomial of degree $O(\sqrt{k\log(1/\varepsilon)} + \log(1/\varepsilon))$ that computes THR^k.*

Lemma 3.6. *For any $\mathsf{G}(k)$ gate G of fan-in n and constant prime q, there is an ε-probabilistic $\mathbb{F}_q$-polynomial of degree $O(k + \log(1/\varepsilon))$ computing G.*[10]

Proof. Because $G \in \mathsf{G}(k)$, we can express its behavior as

$$G(x) = \begin{cases} c & |x| > k \\ f(x) & |x| \leq k \end{cases}$$

for an arbitrary $f : \{0,1\}^n \to \{0,1\}$ and $c \in \{0,1\}$. By Lemma 3.4, there exists a (deterministic) degree-k polynomial $p(x)$ that matches $f(x) - c$ when $|x| \leq k$. Furthermore, by Lemma 3.5, there exists a probabilistic polynomial $Q(x)$ of degree $O(\sqrt{k\log(1/\varepsilon)} + \log(1/\varepsilon))$ that computes THR^k with error ε.

Consider the probabilistic polynomial

$$P(x) \coloneqq (p(x)(1 - Q(x)) + c)^{q-1}.$$

Notice that $\deg(P) = O(k + \sqrt{k\log(1/\varepsilon)} + \log(1/\varepsilon)) = O(k + \log(1/\varepsilon))$, and that the support of P is over proper polynomials by Fermat's Little Theorem (Theorem 2.1).

When $|x| \leq k$, observe that $\mathbf{Pr}[Q(x) = 0] \geq 1 - \varepsilon$. Hence, with probability at least $1 - \varepsilon$, we have

$$P(x) = (p(x) + c)^{q-1} = f(x)^{q-1} = f(x) = G(x),$$

where we use the fact that $p(x) = f(x) - c$ when $|x| \leq k$ and the third equality follows from Fermat's Little Theorem (Theorem 2.1).

When $|x| > k$, $\mathbf{Pr}[Q(x) = 1] \geq 1 - \varepsilon$. Therefore, with probability at least $1 - \varepsilon$, we have

$$P(x) = c^{q-1} = c = G(x).$$

Thus in either case, it follows that P computes G with error $\leq \varepsilon$. $\square$

[10] By *constant prime*, we mean that q does not grow with n. In particular, the $O(\cdot)$ expressions may hide factors depending on q.

We can also show that the degree of the probabilistic polynomial in Lemma 3.6 is optimal.

Lemma 3.7. *There exists a $\mathsf{G}(k)$ gate that requires probabilistic degree $\Omega(k + \log(1/\varepsilon))$.*

Proof. To show a probabilistic degree lower bound of d against a $\mathsf{G}(k)$ gate G, it suffices by Yao's minimax principle to construct a hard distribution $\mathcal{D}$ supported over $\{0,1\}^n$ such that for any degree-d polynomial p, $\mathbf{Pr}_{x\sim\mathcal{D}}[p(x) \neq G(x)] > \varepsilon$. We will show a lower bound of $\max(k/2, \log(1/\varepsilon)) = \Omega(k + \log(1/\varepsilon))$ by showing there exists a gate $\mathsf{G}(k)$ gate G and hard distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ such that any polynomial ε-approximating G under $\mathcal{D}_1$ requires degree $\geq k/2$, and any polynomial ε-approximating G under $\mathcal{D}_2$ requires degree $\lfloor \log(1/\varepsilon) \rfloor$. We will use the probabilistic method and pick $G \in \mathsf{G}(k)$ uniformly at random.

We will set $\mathcal{D}_1$ to be uniform over all strings x with $|x| \leq k$. For a fixed polynomial p, we see by a Chernoff bound that

$$\mathop{\mathbf{Pr}}_{G}\left[\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}_1}[p(x) \neq G(x)] < \varepsilon\right] \leq e^{-\frac{1}{4}\binom{n}{\leq k}}.$$

Union bounding over all $q^{\binom{n}{\leq k/2}}$ degree-$(k/2)$ $\mathbb{F}_q$-polynomials tells us that G cannot be computed by any degree-$k/2$ polynomial with error ε with probability

$$\geq 1 - q^{\binom{n}{\leq k/2}} \cdot e^{-\frac{1}{4}\binom{n}{\leq k}} \geq 1 - e^{-\Omega\left(\binom{n}{\leq k}\right)}.$$

Now let $\mathcal{D}_2$ be the sample $1^k 0^{n-k-\lfloor \log(1/\varepsilon)\rfloor} y$, where $y \sim U_{\lfloor \log(1/\varepsilon)\rfloor}$. Notice that with probability $1/2$, $G'(y) := G(1^k 0^{n-k-\lfloor \log(1/\varepsilon)\rfloor} y)$ is either an AND or OR up to negation (and with the other $1/2$ probability it is constant). Furthermore, if there exists even one y such that

$$p(1^k 0^{n-k-\lfloor \log(1/\varepsilon)\rfloor} y) \neq G(1^k 0^{n-k-\lfloor \log(1/\varepsilon)\rfloor} y),$$

then $\mathbf{Pr}_{x\sim\mathcal{D}_2}[p(x) \neq G(x)] \geq \frac{1}{2^{\lfloor \log(1/\varepsilon)\rfloor}} > \varepsilon$. Therefore, any polynomial p ε-approximating G under $\mathcal{D}_2$ must have the restricted polynomial $p'(y) := p(1^k 0^{n-k-\lfloor \log(1/\varepsilon)\rfloor} y)$ exactly compute G'. Conditioning on G being an AND/OR up to negation, we note that the AND/OR over m variables has $\mathbb{F}_q$-degree m, and so $\deg(p) \geq \deg(p') = \deg(G') = \lfloor \log(1/\varepsilon) \rfloor$.

Consequently by a union bound, a randomly picked G will require degree $k/2$ to approximate under $\mathcal{D}_1$ and degree $\lfloor \log(1/\varepsilon) \rfloor)$ to approximate under $\mathcal{D}_2$ with probability $\geq \frac{1}{2} - e^{-\Omega\left(\binom{n}{\leq k}\right)} > 0$. Hence, our desired G exists, and the lower bound holds. $\square$

We are now ready to show the main theorem. Namely, that proper low-degree $\mathbb{F}_q$ polynomials can approximate any $\mathsf{GC}^0(k)[q]$ circuit.

Theorem 3.8. *Let q be a constant prime. For any $\mathsf{GC}^0_d(k)[q]$ circuit C of size s, there exists a proper polynomial $p(x) \in \mathbb{F}_q[x_1, \ldots, x_n]$ with $\deg(p) \leq O\left((k + \log(1/\varepsilon)(k + \log(s/\varepsilon))^{d-1}\right)$ such that*

$$\mathop{\mathbf{Pr}}_{x\sim U_n}[p(x) \neq C(x)] \leq \varepsilon.$$

Proof. We will construct a probabilistic low-degree polynomial for each gate in the circuit. By composing these polynomials according to the structure of the circuit, we will obtain a probabilistic low-degree polynomial for the entire circuit. This final probabilistic polynomial is the low-degree polynomial approximating the circuit.

For each gate $G \in C$ with fan-in n_G, we will associate a probabilistic low-degree polynomial P_G that approximates it. If $G = \mathsf{NOT}$, then $n_G = 1$ and we set $P_G(x) = x + 1$. If $G = \mathsf{MOD}_q$, then we set $P_G(x) = \sum x_i$. If $G \in \mathsf{G}(k)$ and G is not the top gate, we will set P_G to be the probabilistic polynomial with degree $O(k + \log(2s/\varepsilon))$ that computes G with error probability at most ε/s, as given by Lemma 3.6. Otherwise if G is the top gate, we will set P_G to be the probabilistic polynomial with degree $O(k+\log(2/\varepsilon))$ that computes G with error probability at most $\varepsilon/2$. Note that for all gates G below the top gate in the circuit and all inputs x, $\mathbf{Pr}[G(x) \neq P_G(x)] \leq \varepsilon/2s$, and $\deg(P_G) \leq O(k + \log(s/\varepsilon))$, whereas the top gate G satisfies $\mathbf{Pr}[G(x) \neq P_G(x)] \leq \varepsilon/2$ with $\deg(P_G) \leq O(k + \log(2/\varepsilon))$.

Now, if we replace each gate G with the probabilistic polynomial P_G and compose the polynomials together, we get a probabilistic polynomial P with $\deg(P) \leq O((k+\log(2/\varepsilon))(k+\log(2s/\varepsilon))^{d-1})$. Fix an input x to the circuit. Let $x_G \in \{0,1\}^{n_G}$ be the bits of x read by gate G. If $P_G(x_G) = G(x_G)$ for all gates G in C, then $P(x) = C(x)$. Therefore, by a union bound and accounting for the larger error on the top gate, we have that

$$\mathop{\mathbf{Pr}}_{p\sim P}[p(x) \neq C(x)] \leq \sum_G \mathbf{Pr}[P_G(x_G) = G(x_G)] \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2s} \cdot s = \varepsilon.$$

Since x was arbitrary, the above holds for *all* x, which means

$$\varepsilon \geq \mathop{\mathbf{E}}_x[\mathop{\mathbf{Pr}}_{p\sim P}[p(x) \neq C(x)]] = \mathop{\mathbf{E}}_{p\sim P}[\mathop{\mathbf{Pr}}_x[p(x) \neq C(x)]].$$

Hence, by an averaging argument, there exists a polynomial p in the support of P that agrees with $C(x)$ on all but an ε fraction of inputs. ☐

3.2 Probabilistic Circuits for $\mathsf{G}(k)$ Gates With Very Few Random Bits

We prove that $\mathsf{G}(k)$ gates can be approximated by a randomness-efficient depth-2 probabilistic circuit (Definition 2.2) comprised of AND gates of small fan-in in the bottom layer and a MOD_q gate for any prime q in the top layer, generalizing a prior work of Allender and Hertrampf [AH94]. This result will be crucial for invoking the lower bound technique of Williams [Wil14] as we do in Section 4.2.

Depth-2 probabilistic circuits with AND gates at the bottom and a MOD_q gate at the top are an instance of probabilistic $\mathbb{F}_q$-polynomials. In particular, if the AND gates all have fan-in at most d, then these depth-2 circuits are probabilistic polynomials of degree d. Therefore, one can view the main result of this subsection (Theorem 3.12) as a version of Lemma 3.6 that uses very few bits of randomness. To compare, our Lemma 3.6 uses $\text{poly}(n)$ random bits to construct a probabilistic $\mathbb{F}_q$-polynomial of degree $O(k)$ for any $\mathsf{G}(k)$ gate. In this section, we use $O(k^2 \log^2 n)$ random bits to construct a probabilistic $\mathbb{F}_q$-polynomial of degree $O(k^3 \log^2 n)$ for any $\mathsf{G}(k)$ gate. So, at the cost of a $\text{poly}(k, \log n)$ factor in the degree, we can obtain an exponential savings in the number of random bits used in our construction.

Our construction uses the following theorem of Valiant and Vazirani.

Theorem 3.9 ([VV85])**.** *Let $n \in \mathbb{N}$ and let $S \subseteq \{0,1\}^n$ be a nonempty set. Suppose $w_1, w_2, \ldots, w_n$ are randomly chosen from $\{0,1\}^n$. Let $S_0 = S$ and let*

$$S_i = \{v \in S : \langle v, w_1\rangle = \langle v, w_2\rangle = \ldots = \langle v, w_i\rangle = 0\}, \text{ for each } i \in [n]$$

(where the dot product of two vectors v, w of length n is $\langle v, w\rangle = \sum_{j=1}^n v_j w_j \pmod 2$). Let $P_n(S)$ be the probability that $|S_i| = 1$ for some $i \in \{0, \ldots, n\}$. Then $P_n(S) \geq \frac{3}{4}$.

We start by constructing a depth 5 circuit and then reducing it to depth 2.

Theorem 3.10. *Let q be a constant prime number. Any $\mathsf{G}(k)$ gate on n bits can be computed by a uniform family of probabilistic circuits of size $n^{O(k)}\log(1/\varepsilon)$, with $O(k^2\log^2 n\log(1/\varepsilon))$ random bits and error ε. Furthermore, the circuit has the following structure from top to bottom.*

- *The first layer (the top output gate) is an AND of fan-in $O(k\log n\log(1/\varepsilon))$.*
- *The second layer consists of MOD_p gates with fan-in $n^{O(k)}$.*
- *The third layer consists of AND gates of fan-in k.*
- *The fourth layer consists of MOD_p gates of size $n^{O(k)}$.*
- *The fifth layer consists of AND gates of fan-in $O(k\log n)$.*

Furthermore, this circuit can be constructed in $n^{O(k)}$ time.

Proof. Let G be an arbitrary $\mathsf{G}(k)$ gate. Assume that $G(x)=0$ for $|x|>k$. Otherwise, we can construct a circuit C computing $\neg G$, and then negate it by using a MOD_q gate connected to C and $q-1$ 1's. We begin by describing our circuit construction (with some commentary to help digest the circuit's behavior). A rigorous analysis of the construction will then follow.

Construction. It will be helpful to think of the circuit as the AND of two subcircuits, C_1 and C_2. On inputs x with $|x|\le k$, C_1 will compute G exactly while C_2 will output 1. On the remaining inputs with $|x|>k$, C_1 will have arbitrary behavior while C_2 will output 0 with high probability over the probabilistic bits.

Our circuit C_1 is the low-degree polynomial constructed in Lemma 3.4. This degree-k polynomial can be constructed in $n^{O(k)}$ time, represented as a depth-2 circuit with fan-in-k AND gates at the bottom, one MOD_q gate at the top, and requires no random bits.

Next, we describe the circuit C_2 layer-by-layer, from the inputs to the output gate. Define $m\coloneqq\lfloor\log\binom{n}{k+1}\rfloor+1$. In the first layer, we will have $n+m^2$ bits as input: the input x along with m^2 random bits. Identify the random bits as m vectors $w_1\dots w_m\in\{0,1\}^m$. Arbitrarily associate each $S\in\binom{[n]}{k+1}$ with a distinct bit string in $\{0,1\}^m$, and denote the length-m bit string associated with S by $(S_1,S_2,\dots,S_m)$. We can then define $\langle S,w_i\rangle\coloneqq\sum S_iw_i \pmod 2$.

In the second layer, we will compute $\langle S,w_i\rangle\coloneqq\sum_{j=1}^m x_iw_{i,j} \pmod 2$ for each $S\in\binom{[n]}{k+1}$ and $i\in[m]$. Each $\langle S,w_i\rangle$ can be computed with a single $\oplus$ gate with fan-in $\le m$ by adding a wire from w_{ij} to the gate iff $S_j=1$. To turn $\oplus$ into MOD_q and AND gates, each $\oplus$ gate can be expanded into a DNF of size $\binom{n}{k}^{O(1)}=n^{O(k)}$. Because at most one of the bottom-layer AND clauses can be satisfied simultaneously, we can replace the top OR gate with a MOD_q gate. This conversion is done for each $\oplus$ gate, so, in total, we have $\binom{n}{k}\cdot m$ depth-2 subcircuits of size $n^{O(k)}$, where each subcircuit has a layer of fan-in-m AND gates in the bottom layer and a single MOD_q gate at the top. Denote the MOD_q gate computing $\langle S,w_i\rangle$ by $A_{S,i}$.

In the third layer, for all $S\in\binom{[n]}{k+1}$ and $0\le\ell\le m$ we will compute the predicates

$$B_{S,\ell}\coloneqq\mathbb{1}\left\{(x^S=1)\wedge(\forall i\le\ell,\langle S,w_i\rangle=0)\right\}.$$

These predicates are easily computed using the $A_{S,i}$'s. In particular, to compute $B_{S,k}$, take the AND of x_i for $i\in S$, as well as the $A_{S,i}$ for all $i\le k$. This uses a single AND gate of fan-in $O(m)$. Notice that if $|x|\le k$, $B_{S,\ell}$ is false for all S,ℓ.

In the fourth layer, for $0\le i\le m$, we will compute the predicates

$$D_\ell\coloneqq\mathbb{1}\left\{\left|\left\{S\in\binom{[n]}{k}:x^S=1\text{ and }\forall i\le\ell,\langle S,w_i\rangle=0\right\}\right|\not\equiv 1\pmod q\right\}.$$

In words, D_ℓ is 1 iff the number of sets $S \in \binom{[n]}{k}$ such that $x^S = 1$ and $\forall i \leq \ell, \langle S, w_i \rangle = 0$ is *not* one more than a multiple of q. This is accomplished by taking the MOD_q of $B_{S,\ell}$ for all S, along with $q-1$ 1's. Notice if $|x| > k$, then the set of all $S \in \binom{[n]}{k+1}$ such that $x^S = 1$ is nonempty. Hence by Theorem 3.9, with probability $\geq 1/4$ there will exist some ℓ such that there is exactly one S with $x^S = 1$ and $\forall i \leq \ell, \langle S, w_i \rangle = 0$. In this case, we will have that $D_\ell = 0$.

In the fifth layer, we simply take the AND of all the D_ℓ, which will have fan-in m. By the analysis above, we know this AND gate will output 0 with probability $\geq 1/4$ when $|x| > k$.

We also note that by algorithmically constructing C_2 exactly in the manner we described, we can produce C_2 in $n^{O(k)}$ time.

Analysis. Consider an input x to C.

If $|x| \leq k$, then we know by our construction of C_1 and Lemma 3.4 that $C_1(x) = G(x)$, and from our construction of C_2 that $B_{S,\ell}$ is 0 for all S and ℓ. It is clear that if all $B_{S,\ell}$'s 0, then all the D_ℓ's must be 1. Therefore, $C_2(x) = 1$. Hence, in this case we have,

$$C(x) = C_1(x) \wedge C_2(x) = G(x) \wedge 1 = G(x).$$

Now if $|x| > k$, $C_1(x)$ may be arbitrary, but, as argued above, $C_2(x) = 0$ with probability $\geq 1/4$. We can amplify the error probability of C by replacing C_2 with C_2', which is an AND of $O(\log(1/\varepsilon))$ copies of C_2. It is easy to see that the behavior of C is preserved when $|x| \leq k$. Now when $|x| > k$,

$$\mathbf{Pr}[C(x) = 0] = \mathbf{Pr}[C_1(x) \wedge C_2'(x) = 0] \geq \mathbf{Pr}[C_2'(x) = 0] \geq 1 - (3/4)^{O(\log(1/\varepsilon))} \geq 1 - \varepsilon.$$

We have shown that our circuit C has the desired behavior: computing G with error ε. C_1 has size and construction runtime $n^{O(k)}$ and uses no random bits, and C_2 has size and construction runtime $n^{O(k)}$ and uses m^2 random bits. Hence C will have size and construction runtime $O(n^{O(k)} \log(1/\varepsilon))$ and use $O(k^2 \log^2 n \log(1/\varepsilon))$ random bits.

One can also verify easily that the construction has the desired structure (upon collapsing the cluster of AND gates at the top of the circuit, and trivially extending circuit C_1 past layer 2 using fan-in one gates). □

To shorten this construction to depth-2, we use the following depth-reduction lemma of Allender and Hertrampf [AH94].

Lemma 3.11 ([AH94, Lemma 3]). *Let q be prime. Then every depth-4 circuit consisting of*

- *one* MOD_p *gate with fan-in s_1 on the top level,*
- AND *gates with fan-in t on the second level,*
- MOD_p *gates with fan-in s_2 on the third level, and*
- AND *gates with fan-in r on the last level*

can be converted into a depth-2 circuit that is a MOD_p *of $s_1 \cdot s_2^{t\cdot(p-1)}$* AND *gates, each with fan-in $r \cdot t \cdot (p-1)$. Furthermore, this conversion can be done in $O(s_1 s_2^{t(p-1)} + rt)$ time.*

By applying this lemma twice to our depth-5 probabilistic circuit, we get the following depth-2 probabilistic circuit approximating a $\mathsf{G}(k)$ gate.

Theorem 3.12. *Let q be a constant prime. Any $\mathsf{G}(k)$ gate on n bits can be computed by a depth-2 probabilistic circuit using $O(k^2 \log^2 n \log(1/\varepsilon))$ random bits, and consists of a* MOD_q *of fan-in $2^{O(k^3 \log^2 n \log(1/\varepsilon))}$ at the top, and* AND *gates of fan-in $O(k^3 \log^2 n \log(1/\varepsilon))$ at the bottom layer. Furthermore, the circuit can be constructed in $2^{O(k^3 \log^2 n \log(1/\varepsilon))}$ time.*

Proof. We take the construction of Theorem 3.10 and apply Lemma 3.11 to all the depth-4 subcircuits. This yields a circuit with an AND of fan-in $O(k \log n \log(1/\varepsilon))$ at the top, followed by MOD_p gates of fan-in $n^{O(k)} \cdot n^{O(k \cdot k(q-1))} = 2^{O(k^2 \log n)}$ in the next layer, followed by a final layer of AND gates of fan-in $O(k^2 \log n)$.

We now apply Lemma 3.11 again on this resulting circuit, where we add a dummy fan-in 1 AND gate at the top. This gives a depth-2 circuit whose top gate is a MOD_q of fan-in $2^{O(k^3 \log^2 n \log(1/\varepsilon))}$, and whose bottom layer are AND gates of fan-in $O(k^3 \log^2 n \log(1/\varepsilon))$ as desired. □

4 Applications to Classical Complexity

Theorems 3.8 and 3.12 generalize the seminal works of Razborov [Raz87], Smolensky [Smo87], and Allender-Hertrampf [AH94], which have found use throughout theoretical computer science for nearly four decades. We expect most (if not all) of these applications to hold equally well for $\mathsf{GC}^0(k)[p]$ and GCC^0, given our results in the previous section. To illustrate this, we have selected three applications to present here.

In Section 4.1, we prove average-case lower bounds against $\mathsf{GC}^0(k)[p]$. In particular, we prove that exponential-size circuits are necessary for a $\mathsf{GC}^0(k)[p]$ circuit to compute MAJ or MOD_q for any prime $q \neq p$. This was the original application of the theorems of Razborov and Smolensky.

In Section 4.2, we prove that E^{NP} does not have non-uniform GCC^0 circuits of exponential size. This generalizes the celebrated result of Williams [Wil14].

Finally, in Section 4.3, we apply a framework of Carmosino, Impagliazzo, Kabanets, and Kolokolova [CIKK16] to give a quasipolynomial time learning algorithm for $\mathsf{GC}^0(k)[p]$ in the PAC model over the uniform distribution with membership queries.

4.1 Average-Case Lower Bounds for $\mathsf{GC}^0[q]$

We prove that exponential-size $\mathsf{GC}^0(k)[q]$ circuits are necessary to compute MAJ and MOD_r for any prime $r \neq q$. Our lower bounds generalize the lower bounds of Razborov [Raz87] and Smolensky [Smo87] and follow the same structure. The lower bound argument has two main pieces: (1) $\mathsf{GC}^0(k)[q]$ circuits can be approximated by low-degree polynomials and (2) MAJ and MOD_r gates require large degree to be approximated by a polynomial. The former result was shown in Theorem 3.8, and the latter is a result of Razborov and Smolensky.

Proposition 4.1 ([Raz87, Smo87])**.** *Let q and r be distinct prime numbers, and let $F \in \{\mathsf{MAJ}, \mathsf{MOD}_r\}$. For all degree-$t$ polynomials $p(x) \in \mathbb{F}_q[x_1, \dots, x_n]$,*

$$\Pr_{x \in \{0,1\}^n} [p(x) = F(x)] \leq \frac{1}{2} + O\left(\frac{t}{\sqrt{n}}\right).$$

We can prove correlation bounds against $\mathsf{GC}^0(k)[q]$ by combining Theorem 3.8 and Proposition 4.1.

Theorem 4.2 (Correlation bounds against $\mathsf{GC}^0(k)[q]$)**.** *Let $F \in \{\mathsf{MAJ}, \mathsf{MOD}_r\}$. For any depth-$d$ size-s $\mathsf{GC}^0(k)[q]$ circuit C, we have*

$$\Pr_{x \in \{0,1\}^n} [C(x) = F(x)] \leq \frac{1}{2} + O\left(\frac{(k + \log n)(k + \log(ns))^{d-1}}{\sqrt{n}}\right) + \frac{1}{n}.$$

Proof. By Theorem 3.8, there exists a polynomial $p(x) \in \mathbb{F}_q[x_1, \dots, x_n]$ with degree $O\left((k + \log(1/\varepsilon))(k + \log(s/\varepsilon))^{d-1}\right)$ such that

$$\Pr_{x \in \{0,1\}^n} [p(x) = \neg C(x)] \geq 1 - \varepsilon.$$

Then

$$\begin{aligned}
\Pr_{x \in \{0,1\}^n} [C(x) = F(x)] &= \Pr_{x \in \{0,1\}^n} [\neg C(x) \neq F(x)] \\
&\leq \Pr_{x \in \{0,1\}^n} [p(x) \neq F(x)] + \Pr_{x \in \{0,1\}^n} [p(x) \neq \neg C(x)] \\
&\leq \Pr_{x \in \{0,1\}^n} [1 - p(x) = F(x)] + \varepsilon \\
&\leq \frac{1}{2} + O\left(\frac{(k + \log(1/\varepsilon))(k + \log(s/\varepsilon))^{d-1}}{\sqrt{n}} \right) + \varepsilon,
\end{aligned}$$

where the second inequality follows from the fact that $\Pr_{x \in \{0,1\}^n} [p(x) = \neg C(x)] \geq 1 - \varepsilon$ and the third inequality follows from Proposition 4.1. The result follows from setting $\varepsilon = 1/n$. $\square$

As a corollary, we get a lower bound for $\mathsf{GC}^0(k)[q]$.

Corollary 4.3. *Let q and r be distinct prime numbers, let $F \in \{\mathsf{MAJ}, \mathsf{MOD}_r\}$, and let $k = \Theta(n^{1/2d})$. Any depth-d $\mathsf{GC}^0(k)[q]$ circuit that computes F must have size $2^{\Omega\left(n^{1/2(d-1)}\right)}$.*

Proof. Let C be a size-s, depth-d $\mathsf{GC}^0(k)[q]$ circuit C that can compute $F \in \{\mathsf{MAJ}, \mathsf{MOD}_r\}$. We have

$$1 = \Pr[C(x) = F(x)] \leq \frac{1}{2} + O\left(\frac{(k + \log n)(k + \log(ns))^{d-1}}{\sqrt{n}} \right) + \frac{1}{n}.$$

By solving for s, we can conclude that

$$s \geq 2^{\Omega\left(n^{1/2(d-1)} - k^{d/(d-1)}\right)}.$$

Plugging in k gives the desired result. $\square$

We can improve our average-case lower bounds for $\mathsf{GC}^0(k)[q]$ to average-case lower bounds for $\mathsf{GC}^0(k)[q]/\mathsf{rpoly}$. Recall that $/\mathsf{rpoly}$ means the circuit gets random advice as additional input. In other words, one gets to choose a probability distribution over polynomially many bits that depends on the input size (but not the specific input), and the circuit gets to draw one sample from this distribution.

Theorem 4.4 (Average-case lower bound for $\mathsf{GC}^0(k)[q]$)**.** *Let q and r be distinct prime numbers, and let $F \in \{\mathsf{MOD}_r, \mathsf{MAJ}\}$. There exists an input distribution on which any $\mathsf{GC}^0(k)[q]/\mathsf{rpoly}$ circuit of depth d, $k = O(n^{1/2d})$, and size at most $\exp\left(O\left(n^{1/2.01d}\right)\right)$ only computes F with probability $\frac{1}{2} + \frac{1}{n^{\Omega(1)}}$.*

Proof. Toward a contradiction, assume that for all input distributions, there exists a $\mathsf{GC}^0_d(k)[q]/\mathsf{rpoly}$ circuit with $k = O(n^{1/2d})$ and size $2^{\Omega(n^{1/2.01d})}$ that computes F with probability $1/2 + \varepsilon$ for $\varepsilon = 1/n^{o(1)}$. Then Yao's minimax principle implies that there exists a distribution over $\mathsf{GC}^0_d(k)[q]$ circuits that computes F with probability $1/2 + \varepsilon$ on every input. By drawing $O(1/\varepsilon^2)$ samples from this distribution and taking the majority vote of their outputs, we obtain a new circuit that computes

F with probability 0.99 on every input. Recall that one can compute majority on m bits with a size-$2^{O(n^{1/d})}$ AC^0 circuit [Hås14]. Therefore, since $O(1/\varepsilon^2) = n^{o(1)}$, the majority of the $\mathsf{GC}^0_d(k)[q]$ circuits can be computed in depth d and size $2^{n^{o(1)}}$, which doubles the depth of the original circuit and only increases the size by a negligible amount.

Next, we amplify the success probability from 0.99 to $1 - \exp(-n)$, for some $\exp(-n) < 2^{-n}$, by sampling $O(n)$ circuits that succeed with probability 0.99 and taking their majority vote. Since the circuits succeed with probability 0.99, it is easy to see that a 0.99-fraction of the votes will be 0's or 1's with high probability. Hence, the approximate majority construction of Ajtai and Ben-or [ABO84] suffices, which can be performed by a polynomial-size AC^0 circuit.[11]

Because this distribution over $\mathsf{GC}^0_d(k)[q]$ circuits fails to compute F with probability less than 2^{-n}, it follows by union bounding over all 2^n inputs that there exists one circuit in the distribution that computes F on all inputs. Hence, we have constructed a $\mathsf{GC}^0_d(k)[q]$ circuit of depth $2d + O(1)$, $k = O(n^{1/2d})$, and size $\exp(n^{1/2.01d})$, contradicting Corollary 4.3. ☐

4.2 Non-Uniform GCC^0 Lower Bounds

We prove that there are languages in E^{NP} that fail to have polynomial-size $\mathsf{GCC}^0(k)$ circuits for certain values of k (which are stated carefully in Theorem 4.13). Recall that E is the class of languages that can be decided by a Turing machine in time $2^{O(n)}$. This generalizes the breakthrough work of Williams [Wil14] who proved that there are languages in NEXP and E^{NP} that fail to have polynomial-size ACC^0 circuits. Here we focus on E^{NP} instead of NEXP because we get a stronger size-depth tradeoff. We note that similar arguments can show that NEXP fails to have $\mathsf{GCC}^0(k)$ circuits.

These lower bounds are based on Williams' algorithmic method, which, in short, connects the existence of fast algorithms for the CircuitSAT problem to circuit lower bounds.

Definition 4.5 ($\mathcal{C}$-CircuitSAT)**.** Given as input a description of a $\mathcal{C}$ circuit C, the $\mathcal{C}$-CircuitSAT problem is to decide whether there exists an input $x \in \{0,1\}^n$ such that $C(x) = 1$.

The algorithmic method only works for "nice" circuit classes.

Definition 4.6 (Nice circuits [Wil14])**.** A *nice* circuit class $\mathcal{C}$ is a collection of circuit families that:

- contain AC^0: for every circuit family in AC^0, there is an equivalent circuit family in $\mathcal{C}$, and
- is closed under composition: for $\{C_n\}, \{D_n\} \in \mathcal{C}$ and any integer c, the circuit family obtained by feeding n input bits to $n^c + c$ copies of C_n and feeding the outputs into D_{n^c+c} is also in $\mathcal{C}$.

Every well-studied circuit class is nice, and it is easy to see that GCC^0 is nice too.

We can now formally state the essence of the algorithmic method. Specifically, fast algorithms for $\mathcal{C}$-CircuitSAT imply circuit lower bounds for $\mathcal{C}$.

Theorem 4.7 ([Wil14, Theorem 3.2])**.** *Let $S(n) \leq 2^{n/4}$ and let $\mathcal{C}$ be a nice circuit class. There is a $c > 0$ such that, if $\mathcal{C}$-CircuitSAT instances with at most $n + c\log n$ variables, depth $2d + O(1)$, and $O(nS(2n) + S(3n))$ size can be solved in $O(2^n/n^c)$ time, then E^{NP} does not have non-uniform $\mathcal{C}$ circuits of depth d and $S(n)$ size.*

[11] For the unfamiliar reader, the approximate majority circuit will output "1" when at least a 0.75-fraction of the inputs are 1, "0" when at most a 0.25-fraction of the inputs are 0, and behave arbitrarily otherwise.

To apply Theorem 4.7 and obtain our GCC^0 lower bound, we will give fast algorithms for GCC^0-CircuitSAT, showing that the algorithmic method of Williams also lifts from ACC^0 to GCC^0. As a starting point, we will recall the ACC^0 satisfiability algorithm and then extend the necessary parts to $\mathsf{GCC}^0(k)$. Let SYM^+ be the class of depth-two circuits with a layer of AND gates at the bottom and some symmetric function at the top. The ACC^0-CircuitSAT algorithm can be modularized as follows. Given as input a description of a size-s depth-d ACC^0 circuit (that is comprised of AND, OR, NOT, and MOD_m gates for a fixed m), the algorithm performs the following four steps.

1. Turn each MOD_m gate into an AND of MOD_p's of AND's, where all gates have constant fan-in and p is some prime dividing m. This takes $s^{O(1)}$ time.

2. Replace each OR gate with a probabilistic circuit consisting of a MOD_p of $2^{\text{poly}(\log s)}$ ANDs, each of fan-in $\text{poly}(\log s)$. Call the resulting circuit C. C uses $\text{poly}(\log s)$ random bits.

3. Convert C into a SYM^+ circuit C' of size $2^{O(\text{poly}(\log s))}$ whose top symmetric gate can be evaluated in time $2^{O(\text{poly}(\log s))}$.

4. Run a SYM^+-CircuitSAT algorithm on C'.

To design a $\mathsf{GCC}^0(k)$-CircuitSAT algorithm, it suffices to modify only the second step in the above blueprint to handle $\mathsf{G}(k)$ gates. In particular, we will use our Theorem 3.12 to turn a $\mathsf{G}(k)$ gate into a probabilistic circuit with only MOD_p gates and bounded fan-in ANDs with comparable parameters to Step 2 above. (In particular, our circuit will have the same size and AND fan-in, but with $k \log s$ in place of $\log s$.)

Now we will prove that Step 2 above holds for $\mathsf{G}(k)$ gates. We first recall the ACC^0 theorems established in [Wil14] that we will use in a black-box manner. In these theorems, we will fix a function $f(d) := 2^{O(d)}$ that quantifies the size-depth tradeoffs in these theorems. This will be important to track the size-depth improvements we obtain in our $\mathsf{GCC}^0(k)$ lower bounds.

Theorem 4.8 ([AG94, Wil14]). *Let $f : \mathbb{N} \to \mathbb{N}$ be a function where $f(d) = 2^{O(d)}$ and let $t \in \mathbb{N}$. Let C be a probabilistic circuit with depth $2d = O(1)$, size 2^{t^4}, no OR or MOD_m gates for any composite m, and AND gates of fan-in at most t^4 that computes a function with t^3 probabilistic inputs and error probability $1/3$. There is an algorithm that, given C, outputs an equivalent SYM^+ circuit of size $2^{O(t^{f(d)})}$. The algorithm takes at most $2^{O(t^{f(d)})}$ time.*

Furthermore, if the number of ANDs in the SYM^+ circuit that evaluate to 1 is known, then the symmetric function in the SYM^+ circuit can be evaluated in $2^{O(t^{f(d)})}$ time.

Williams transforms a size-s, depth-d ACC^0 circuit into a SYM^+ circuit by replacing each OR/AND gate with a depth-2 probabilistic circuit with AND gates of bounded fan-in and then applying Theorem 4.8 with $t \leftarrow O(\log s)$. This is formalized in the following lemma.

Lemma 4.9 ([AH94, AG94, Wil14]). *Let $f : \mathbb{N} \to \mathbb{N}$ be a function where $f(d) = 2^{O(d)}$. There is an algorithm that, given an ACC^0 circuit of depth $d = O(1)$ and size s, outputs an equivalent SYM^+ circuit of size $2^{O(\log^{f(d)} s)}$. The algorithm takes $2^{O(\log^{f(d)} s)}$ time.*

Furthermore, if the number of ANDs in the SYM^+ circuit that evaluate to 1 is known, then the symmetric function in the SYM^+ circuit can be evaluated in $2^{O(\log^{f(d)} s)}$ time.

We will get a similar conversion for size-s depth-d GCC^0 circuits by replacing $\mathsf{G}(k)$ gates with our newly constructed depth-2 probabilistic circuits from Theorem 3.12, which are comparable in size and identical in depth to the AND/OR probabilistic circuit construction used to prove Lemma 4.9. This allows us to use Theorem 4.8 with $t \leftarrow O(k \log s)$.

Theorem 4.10. *Let $f : \mathbb{N} \to \mathbb{N}$ be a function where $f(d) = 2^{O(d)}$. There is an algorithm that, given a $\mathsf{GCC}^0(k)$ circuit of depth $d = O(1)$ and size s, outputs an equivalent SYM^+ circuit of size $2^{O((k \log s)^{f(d)})}$. The algorithm takes at most $2^{O((k \log s)^{f(d)})}$ time.*

Furthermore, if the number of ANDs in the SYM^+ circuit that evaluate to 1 is known, then the symmetric function in the SYM^+ circuit can be evaluated in $2^{O((k \log s)^{f(d)})}$ time.

Proof. Let C be the given circuit. As in the ACC^0 case, we will identically use Step 1 to convert all MOD_m gates into MOD_p gates, with p prime, in $s^{O(1)}$ time (see [Wil14, Appendix A] for specific details). Denote this new circuit C'. At this point we will now use Theorem 3.12 to replace each $\mathsf{G}(k)$ gate with a probabilistic circuit that computes the gate except with probability $\varepsilon := 1/3s$ and uses the *same random bits* (versus having a fresh supply per gate), which can be done in time $s \cdot 2^{O(k^3 \log^3 s)}$. Since the fan-in of each $\mathsf{G}(k)$ gate is at most s and $\varepsilon = 1/3s$, it follows that each $\mathsf{G}(k)$ gate is replaced by a depth-2 probabilistic circuit of size $2^{O(k^3 \log^3 s)}$ consisting of MOD_p gates with p prime, and AND gates of fan-in $O((k \log s)^3)$. Furthermore, the circuit uses $O(k^2 \log^3 s)$ random bits altogether. Notice by a union bound, there is at most $s(1/3s) = 1/3$ probability that one of the s probabilistic subcircuits substituted in is faulty. Therefore, the resulting circuit computes C with probability $\geq 2/3$. We finally apply Theorem 4.8 to construct the desired SYM^+ circuit in the desired time complexity. $\square$

The algorithm in Theorem 4.10 is the transformation in Step 2 above. Hence, all that remains to get our lower bound is to put the pieces together. To do so, we need the following evaluation algorithm, which takes a SYM^+ circuit as input and outputs its truth table.

Lemma 4.11 ([Wil14])**.** *There is an algorithm that, given a SYM^+ circuit of size $s \leq 2^{0.1n}$ and n inputs with a symmetric function that can be evaluated in $\mathrm{poly}(s)$ time, runs in $(2^n + \mathrm{poly}(s))\mathrm{poly}(n)$ time and prints a 2^n-bit vector V which is the truth table of the function represented by the given circuit. That is, $V[i] = 1$ iff the SYM^+ circuit outputs 1 on the ith variable assignment.*

This gives us our fast $\mathsf{GCC}^0(k)$-CircuitSAT algorithm. Recall that $f : \mathbb{N} \to \mathbb{N}$ in the theorems below is a function $f(d) = 2^{O(d)}$.

Theorem 4.12. *For every $d > 1$ and $\varepsilon = \varepsilon(d) := .99/f(d)$, the satisfiability of depth-d $\mathsf{GCC}^0(k)$ circuits with n inputs and $2^{n^\varepsilon/k}$ size can be determined in time $2^{n-\Omega(n^\delta/k)}$ for some $\delta > \varepsilon$.*

Proof. Consider C, a depth-d GCC^0 circuit of size $2^{n^\varepsilon/k}$. For any $\ell \in [n]$, we can create circuit C' of depth $d+1$, size $s2^\ell$ over $n - \ell$ inputs by taking 2^ℓ copies of C, plugging in a distinct assignment of the first ℓ bits into each copy, and then taking the OR of them. Notice that C is satisfiable iff C' is.

We now apply Theorem 4.10 on C' to get an equivalent SYM^+ circuit C'', which is a symmetric function of $s'' \leq 2^{(k(\ell+\log s))^{f(d)}}$ ANDs. By Lemma 4.11 and the fact the symmetric function can be computed in $\mathrm{poly}(s'')$ time, it follows that upon setting $\ell := \log s = n^\varepsilon/k$, we get an algorithm that runs in $O(2^{n-\ell}\mathrm{poly}(n)) = 2^{n-\Omega(n^\delta/k)}$ for some $\delta > \varepsilon$. $\square$

Our circuit satisfiability algorithm implies the following lower bound.

Theorem 4.13 ($\mathsf{E}^{\mathsf{NP}} \not\subseteq \mathsf{GCC}^0$)**.** *For every d, there is a constant $C > 1$ and $\delta = \delta(d) := 1/Cf(2d)$, such that for all $k \leq O(n^\delta/\log n)$, there exists a language in E^{NP} that fails to have $\mathsf{GCC}^0(k)$ circuits of depth d and size $\exp\left(\Omega(n^\delta/k)\right)$.*

Proof. By Theorem 4.12, we know for every d, the satisfiability of depth-d $\mathsf{GCC}^0(k)$ of size $2^{O(n^{.99/f(d)})}$ on n inputs can be solved in $2^{n-\Omega(n^{\varepsilon}/k)}$ time for some $\varepsilon > 1/4f(d)$. Now by Theorem 4.7, we know there exists a constant $c > 0$ such that if $\mathsf{GCC}^0(k)$-Circuit SAT instances with $n + c\log n$ variables, depth $2d + O(1)$, and size $s = n2^{(2n)^{\delta}} + 2^{(3n)^{\delta}}$ can be solved in time $O(2^n/n^c)$, then E^{NP} doesn't have non-uniform $\mathsf{GCC}^0(k)$ circuits of depth d and size $2^{n^{\delta}}$. Since $f(d) = 2^{O(n)}$, we know $f(2d+O(1)) \le Cf(2d)$ for some constant C. Consequently, for $\delta = 1/Cf(2d)$, we can indeed solve depth $2d+O(1)$ and size $n2^{(2n)^{\delta}}+2^{(3n)^{\delta}} \le \exp\left(O(n^{\frac{.99}{f(2d+O(1))}})\right)$ GCC^0 circuits over $n+c\log n$ inputs in time $2^{(n+c\log n)-\Omega((n+c\log n)^{\varepsilon}/k)} = O(2^n/n^c)$ for small enough constant c (by using the assumption $n^{\delta}/k = \Omega(\log n)$), yielding the desired lower bound. □

We conclude with some remarks about the extent of our contribution. The Williams lower bound of $\mathsf{E}^{\mathsf{NP}} \not\subseteq \mathsf{ACC}^0$ suffices to prove that there exist languages in E^{NP} that fail to have polynomial-size GCC^0 circuits (or even exponential-size GCC^0 circuits for some small enough exponential function). This is achieved by naïvely transforming the GCC^0 circuit to an ACC^0 circuit. Specifically, suppose we have a size-s depth-d $\mathsf{GCC}^0(k)$ circuit, and then we transform each $\mathsf{G}(k)$ gate into a CNF (or DNF, it does not matter). The resulting circuit will be a size-s^k depth-$2d$ ACC^0 circuit. Then, after applying the lower bound for depth-d size-$\exp(\Omega(n^{1/f(2d)}))$ ACC^0 circuits[12], we obtain a separation between E^{NP} and depth-d $\mathsf{GCC}^0(k)$ circuits of size $\exp(O(n^{1/Cf(4d)}/k))$.

In our Theorem 4.13, we get a separation between E^{NP} and depth-d $\mathsf{GCC}^0(k)$ circuits of size $\exp(O(n^{1/Cf(2d)}/k))$. The difference is the $f(2d)$ in Theorem 4.13 vs. $f(4d)$ in the naïve approach that appear in the exponent of the exponent of the circuit size. Because f is an exponential function as well, the difference is then a factor of 2 in the exponent of the exponent of the exponent. Hence, using our result yields a *triply exponential improvement* in the size-depth tradeoff compared to the naïve approach.

4.3 PAC Learning $\mathsf{GC}^0[p]$

Carmosino, Impagliazzo, Kabanets, and Kolokolova [CIKK16] gave a quasipolynomial time learning algorithm for $\mathsf{AC}^0[p]$ in the PAC model over the uniform distribution with membership queries. We recall their result in more detail and argue that there is a quasipolynomial time learning algorithm for $\mathsf{GC}^0(k)[p]$.

To begin, we establish some notation and define the learning model. For a circuit class Λ and a set of size functions $\mathcal{S}$, $\Lambda[\mathcal{S}]$ denotes the set of size-$\mathcal{S}$ n-input circuits of type Λ. For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and $\varepsilon \in [0,1]$, $\widetilde{\mathsf{CKT}}_n(f,\varepsilon)$ denotes the set of all circuits that compute f on all but an ε fraction of inputs.

Definition 4.14 (Learning model)**.** Let $\mathcal{C}$ be a class of Boolean functions. An algorithm A PAC-learns $\mathcal{C}$ if for any n-variate $f \in \mathcal{C}$ and for any $\varepsilon, \delta > 0$, given membership query access to f, algorithm A prints with probability at least $1-\delta$ over its internal randomness a circuit $C \in \widetilde{\mathsf{CKT}}_n(f,\varepsilon)$. The runtime of A is measured as a function of $T(n, 1/\varepsilon, 1/\delta, \mathrm{size}(f))$.

Carmosino et al. establish a connection between learning and natural proofs [RR97]. We recall the definition of natural proofs here for convenience. Let F_n be the collection of all Boolean functions on n variables. Λ and Γ denote complexity classes. A *combinatorial property* is a sequence of subsets of F_n for each n.

Definition 4.15 (Natural property [RR97])**.** A combinatorial property R_n is Γ-natural against Λ with density δ_n if it satisfies the following three conditions:

[12]This is the lower bound proved by Williams [Wil14]. It is also a special case of Theorem 4.13 with $k = 1$.

- **Constructivity:** The predicate $f_n \stackrel{?}{\in} R_n$ is computable in Γ.

- **Largeness:** $|R_n| \geq \delta_n |F_n|$.

- **Usefulness:** For any sequence of functions f_n, if $f_n \in \Lambda$ then $f_n \notin R_n$, almost everywhere.

A proof that some explicit function is not in Λ is called Γ-natural against Λ with density δ_n when it involves a Γ-natural property R_n that is useful against Λ with density δ_n. Razborov and Rudich [RR97] showed that the Razborov-Smolensky lower bound proofs are NC^2-natural against $\mathsf{AC}^0[p]$, where, roughly speaking, the natural property contains functions that cannot be approximated by low-degree polynomials (see [RR97, Section 3] and [CIKK16, Section 5] for further details). An immediate implication of our lower bounds (Corollary 4.3 and Theorem 4.4) is that the same property is NC^2-natural against $\mathsf{GC}^0(k)[p]$.

Theorem 4.16. *For every prime p, there is an NC^2-natural property of n-variate Boolean functions, with largeness at least $1/2$, that is useful against $\mathsf{GC}^0(k)[p]$ circuits of depth d and of size up to $\exp\left(\Omega(n^{1/2d})\right)$ where $k = O(n^{1/2d})$.*

Carmosino et al. [CIKK16] prove the following connection between natural properties and PAC learning algorithms over the uniform distribution with membership queries.

Theorem 4.17 ([CIKK16, Theorem 5.1])**.** *Let Λ be any circuit class containing $\mathsf{AC}^0[p]$ for some prime p. Let R be a P-natural property, with largeness at least $1/5$, that is useful against $\Lambda[u]$, for some size function $u : \mathbb{N} \to \mathbb{N}$. Then there is a randomized algorithm that, given oracle access to any function $f : \{0,1\}^n \to \{0,1\}$ from $\Lambda[s_f]$, produces a circuit $C \in \widetilde{\mathsf{CKT}}(f, \varepsilon)$ in time $\mathrm{poly}(n, 1/\varepsilon, 2^{u^{-1}\mathrm{poly}(n,1/\varepsilon,s_f)})$.*

By combining Theorem 4.16, Theorem 4.17, and the basic fact that $\mathsf{AC}^0[p] \subseteq \mathsf{GC}^0(k)[p]$ for all primes (and prime powers) p, we get the following learning algorithm for $\mathsf{GC}^0(k)[p]$.

Corollary 4.18 (Learning $\mathsf{GC}^0(k)[p]$ in quasipolynomial time)**.** *Let $k = O(n^{1/2d})$. For every prime p, there is a randomized algorithm that, using membership queries, learns a given n-variate Boolean function $f \in \mathsf{GC}^0(k)[p]$ of size s_f to within error ε over the uniform distribution, in time $\mathrm{quasipoly}(n, s_f, 1/\varepsilon)$.*

5 Applications to Quantum Complexity

We study the implications of our lower bounds for $\mathsf{GC}^0[p]$ and GC^0 on quantum complexity theory. Specifically, we show exponential separations between shallow quantum circuits and both $\mathsf{GC}^0[p]$ and GC^0, surpassing all previously known separations between quantum and classical circuits. We emphasize that these separations are *unconditional* and our results generalize the prior work in this area [BGK18, WKST19, BGKT20, GJS21, RT22, GKMdO24].

For convenience, we summarize the separations we obtain in this section. We say a separation is exponential when polynomial-size quantum circuits can solve a certain problem but even some exponential-size classical circuits cannot. In this section, we exhibit (formal definitions and arguments are given within the corresponding subsection):

- A promise problem separating $\mathsf{BQLOGTIME}$ and $\mathsf{GC}^0(k)$ (Corollary 5.7).

- A relation problem separating QNC^0 and $\mathsf{GC}^0(k)$ (Theorem 5.15).

- A relation problem separating $\mathsf{QNC}^0/\mathsf{qpoly}$ and $\mathsf{GC}^0(k)[p]$ for any prime p (Theorems 5.24 and 5.34).
- An interactive problem separating QNC^0 and $\mathsf{GC}^0(k)[p]$ for any prime p (Theorem 5.41).

Our separations are all exponential (i.e., the problems can be solved by polynomial-size QNC^0 circuits but are hard for exponential-size classical circuits), and Theorems 5.15, 5.24 and 5.34 prove average-case lower bounds.

In addition to our results in Sections 3 and 4, our quantum-classical separations require a few new classical ingredients. We prove a *multi-output* multi-switching lemma for GC^0 (Theorem 5.20), which generalizes the multi-switching lemma proved by Kumar [Kum23] to multi-output GC^0 circuits. Our result is based on the multi-switching lemmas for AC^0 that were proven by Håstad [Hås14] and Rossman [Ros17], and is based on the proof of the AC^0 multi-output multi-switching lemma established in [WKST19].

We also prove that a single $\mathsf{G}(k)$ gate can compute functions that are not computable in $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$ when $k = \log^{\omega(1)} n$ (Theorem 5.38). We use this to show that certain $\mathsf{GC}^0(k)[p]$ circuits are incomparable to NC^1 (Corollary 5.40), which is needed in the proof of Theorem 5.41.

5.1 Pushing Raz & Tal: $\mathsf{BQLOGTIME} \not\subseteq \mathsf{GC}^0$

In a breakthrough work, Raz and Tal [RT22] showed that BQP is not in PH relative to an oracle. An unconditional separation between $\mathsf{BQLOGTIME}$ and AC^0 is at the core of their result. Specifically, they give a distribution that is *pseudorandom* (i.e., cannot be distinguished from the uniform distribution) for AC^0 circuits, but not for $\mathsf{BQLOGTIME}$ circuits. By well-known reductions, this implies their oracle and circuit separations. We show that their distribution is also pseudorandom for GC^0 circuits. Hence, by the same reductions, we can conclude that $\mathsf{BQLOGTIME} \not\subseteq \mathsf{GC}^0$. We begin with a formal definition of $\mathsf{BQLOGTIME}$.

Definition 5.1. $\mathsf{BQLOGTIME}$ is the class of promise problems $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$ that are decidable, with bounded error probability, by a $\mathsf{LOGTIME}$-uniform family of quantum circuits $\{C_n\}_{n\in\mathbb{N}}$, where each C_n is an n-qubit quantum circuit with $O(\log n)$ gates that are either (i) input query gates (i.e., gates that map $|i\rangle|z\rangle$ to $|i\rangle|z \oplus x_i\rangle$ where $x = x_1 \dots x_n$ is the input string) or (ii) standard quantum gates from a fixed, finite gate set.

Let $\mathcal{D}_{\text{RAZ-TAL}}$ denote the distribution over $\{-1,1\}^{2N}$ described in [RT22, Section 4] (see also [Wu22, Section 2]). Raz and Tal showed that if $\mathcal{D}_{\text{RAZ-TAL}}$ is sufficiently pseudorandom, then one can obtain separations from $\mathsf{BQLOGTIME}$.

Lemma 5.2 ([RT22])**.** *Let $\mathcal{F}$ be a class of Boolean functions $f : \{\pm1\}^{2N} \to \{0,1\}$. Suppose that for each $f \in \mathcal{F}$,*

$$\left|\mathbf{E}[f(\mathcal{D}_{\text{RAZ-TAL}})] - \mathbf{E}[f(U_{2N})]\right| \le \left(\frac{1}{\log N}\right)^{\omega(1)}.$$

Then $\mathsf{BQLOGTIME} \not\subseteq \mathcal{F}$.

Furthermore, Raz and Tal showed that the desired pseudorandomness property follows from understanding the *second-level Fourier growth*, i.e., the ℓ_1-norm of the Fourier coefficients on the second level.

Lemma 5.3 ([RT22], [Wu22, Theorem 4.4])**.** *Let $f : \{\pm 1\}^{2N} \to \{0,1\}$ be a Boolean function. For $L > 0$, suppose that for any restriction ρ,*

$$\sum_{\substack{S \subseteq [2N] \\ |S|=2}} |\widehat{f_\rho}(S)| \leq L.$$

Then,

$$\left| \mathbf{E}[f(\mathcal{D}_{\text{Raz-Tal}})] - \mathbf{E}[f(U_n)] \right| \leq \frac{2\varepsilon L}{\sqrt{N}}.$$

In prior work, Kumar [Kum23] gave upper bounds on the Fourier growth of GC^0-computable functions.

Lemma 5.4 ([Kum23, Theorem 5.14])**.** *Let $f : \{\pm 1\}^n \to \{\pm 1\}$ be computable by a size-m $\mathsf{GC}^0_d(k)$ circuit. Then, for all $\ell \in \mathbb{N}$, the following is true for some universal constants $C_1, C_2 > 0$:*

$$\sum_{\substack{S \subseteq [n] \\ |S|=\ell}} |\widehat{f}(x)| \leq C_1 (C_2 \cdot k(k + \log m)^{d-1})^{\ell}.$$

In particular, for some universal constant $C > 0$,

$$\sum_{\substack{S \subseteq [n] \\ |S|=2}} |\widehat{f}(x)| \leq C k^2 (k + \log m)^{2(d-1)}.$$

We can now start combining these ingredients to obtain the claimed separation.

Proposition 5.5 (Generalization of [RT22, Theorem 7.4])**.** *Let $f : \{\pm 1\}^{2N} \to \{\pm 1\}$ be a size-m $\mathsf{GC}^0_d(k)$ circuit. Then there is a universal constant $C > 0$ such that*

$$\left| \mathbf{E}[f(\mathcal{D}_{\text{Raz-Tal}})] - \mathbf{E}[f(U_n)] \right| \leq \frac{C \varepsilon k^2 (k + \log m)^{2(d-1)}}{\sqrt{N}}.$$

Proof. Combine Lemmas 5.3 and 5.4. ☐

Combining Lemma 5.2 and Proposition 5.5 yields the following two corollaries.

Corollary 5.6 (Generalization of [RT22, Corollary 7.5])**.** *Let $f : \{\pm 1\}^{2N} \to \{\pm 1\}$ be a $\mathsf{GC}^0(k)$ circuit of constant depth and size* quasipoly(N)*. For $\varepsilon = O\left(\frac{1}{\log N}\right)$ and $k = \frac{O(N^{1/4d})}{\log^{\omega(1)} N}$,*

$$\left| \mathbf{E}[f(\mathcal{D}_{\text{Raz-Tal}})] - \mathbf{E}[f(U_{2N})] \right| \leq \frac{1}{\log^{\omega(1)} N}.$$ [13]

Corollary 5.7 (Generalization of [RT22, Corollary 1.6])**.** *There is a promise problem in $\mathsf{BQLOGTIME}$ that is not solvable by constant-depth $\mathsf{GC}^0(k)$ for $k = \frac{O(n^{1/4d})}{\log^{\omega(1)} n}$ and size* quasipoly(n)*, where n is the input size.*

[13] Note that $\varepsilon \in \Omega(1/\log N)$ is necessary for the $\mathsf{BQLOGTIME}$ to succeed with a large enough probability. See [RT22, Section 6] for further detail.

Our circuit separation also says something about oracle separations. By standard techniques, Corollaries 5.6 and 5.7 imply an oracle A relative to which $\mathsf{BQP}^A \not\subseteq \mathsf{C}^A$ for any class of languages C that can be decided by a uniform family of constant-depth, exponential-size GC^0 circuits.[14]

Corollary 5.8 (Generalization of [RT22, Corollary 1.5])**.** *There is an oracle relative to which* BQP *is not contained in the class of languages decidable by uniform families of circuits* $\{C_n\}$*, where for all* $n \in \mathbb{N}$*,* C_n *is a size-*$2^{n^{O(1)}}$ *depth-*d $\mathsf{GC}^0(k)$ *circuit with* $k \in \frac{2^{n/4d}}{n^{\omega(1)}}$*.*

The proof is the same as [RT22, Appendix A] but the step where they apply their Theorem 1.2 should be replaced with our Corollary 5.6. Hence, we omit the details. Similar proofs were also given by Aaronson [Aar10] and Fefferman, Shalteil, Umans, and Viola [FSUV13], which were based on an earlier work of Bennett and Gill [BG81].

It is well-known that PH is the class of languages decided by uniform families of size-$2^{n^{O(1)}}$ constant-depth AC^0 circuits (see e.g., [AB09, Theorem 6.29]). Therefore, the separation of BQP and PH is a special case of our theorem, because $\mathsf{AC}^0 \subseteq \mathsf{GC}^0(k)$ for all $k \geq 0$.

Because $\mathsf{G}(k)$ gates can compute many functions, Corollary 5.8 can be instantiated in many ways. We give one concrete example separating BQP from a biased version of the counting hierarchy, which we now define. First, we define existential and universal counting quantifiers. Similar definitions date back to [Wag86, Tor91, AW93]. For a bit string x, let $\mathrm{len}(x)$ denote the length of x.

Definition 5.9 (Counting quantifiers)**.** Let C be a class of languages, and let $k : \mathbb{N} \to \mathbb{N}$ be a function. Define $\exists_k \cdot \mathsf{C}$ to be the set of all languages L such that there is some polynomial p and a language $C \in \mathsf{C}$ such that $x \in L \iff$

$$|\{y \in \{0,1\}^{p(\mathrm{len}(x))} : \langle x, y\rangle \in C\}| > k(\mathrm{len}(x)).$$

Define $\forall_k \cdot \mathsf{C}$ to be the set of all languages L such that there is some polynomial p and a language $C \in \mathsf{C}$ such that for $x \in \{0,1\}^n$, $x \in L \iff$

$$|\{y \in \{0,1\}^{p(\mathrm{len}(x))} : \langle x, y\rangle \notin C\}| \leq k(\mathrm{len}(x)).$$

We note that $\exists_0 = \exists$ and $\forall_0 = \forall$.

We can now define the k-biased counting hierarchy. For two functions $f_1, f_2 : \mathbb{N} \to \mathbb{N}$, we say $f_1 \leq f_2$ when $\forall n, f_1(n) \leq f_2(n)$.

Definition 5.10 (Biased counting hierarchy)**.** Let $k : \mathbb{N} \to \mathbb{N}$ be a function. The k-biased counting hierarchy $\mathsf{CH}(k)$ is the smallest family of language classes satisfying:

(i) $\mathsf{P} \in \mathsf{CH}(k)$,

(ii) If $L \in \mathsf{CH}(k)$, then $\exists_{k'} \cdot L$ and $\forall_{k'} \cdot L \in \mathsf{CH}(k)$ for all $k' : \mathbb{N} \to \mathbb{N}$, $k' \leq k$.

It is a well-known fact that the polynomial hierarchy PH can be characterized by alternating $\exists_0$ and $\forall_0$ quantifiers, so $\mathsf{CH}(0) = \mathsf{PH}$. As mentioned above, there is also a well-known characterization of PH by AC^0 circuits. Roughly speaking, the $\exists_0$ quantifiers are replaced by OR gates, and the $\forall_0$ quantifiers are replaced by AND gates. Let k-OR be the gate that is 1 iff $> k$ input bits are 1. Similarly, let k-AND be the gate that is 0 iff $> k$ input bits are 0. Observe OR = 0-OR and AND = 0-AND, and that k-AND and k-OR are $\mathsf{G}(k)$ gates up to negations (specifically, one can

[14] The notion of uniformity here is sometimes called direct connect uniform [AB09, Definition 6.28] or highly uniform [Gol08, Exercise 3.8].

construct k-AND with NOT and k-OR via De Morgan's law). So, in *exactly* the same manner as PH, for any class $\mathsf{C} \in \mathsf{CH}(k)$, one can construct a $\mathsf{GC}^0(k)$ circuit that decides an $L \in \mathsf{C}$ by replacing the $\exists_k$ quantifiers with k-OR gates and the $\forall_k$ quantifiers with k-AND gates. By doing this standard construction, one obtains the following corollary of Corollary 5.8.

Corollary 5.11. *There is an oracle A relative to which $\mathsf{BQP}^A \not\subseteq \mathsf{CH}(k)^A$ for $k(n) = \frac{2^{\Theta(n)}}{n^{\omega(1)}}$.*

This result is perhaps surprising considering that $\mathsf{BQP} \subseteq \mathsf{PP}$ relative to all oracles [ADH97] and PP is the first level of the standard counting hierarchy. Thus, our Corollary 5.11 shows that a relativizing simulation of BQP requires being able to count a larger number of witnesses (exponential in the input instance size), as one can in PP.

More broadly, Corollary 5.8 separates BQP from many complexity classes that contain PH and are incomparable with PP; the specific complexity classes one gets depends on how the $\mathsf{G}(k)$ gates are used in the uniform circuit families. Above we gave an example where the $\mathsf{G}(k)$ gates are all k-AND and k-OR gates.

5.2 Separation Between QNC^0 and GC^0

We exhibit a search problem with input size n that can be solved by QNC^0 circuits (i.e., polynomial-size, constant-depth quantum circuits with bounded fan-in gates), but not by size-s $\mathsf{GC}^0(k)$ circuits for $s \leq \exp(n^{1/4d})$ and $k = O(\log s)$. In particular, we show strong average-case lower bounds against GC^0 for this search problem, i.e., that any GC^0 circuit can only succeed on an $\exp(-n^c)$ fraction of input strings for some $c > 0$. In Section 5.2.3, we show that our separation holds even when the quantum circuits are subject to noise.

Our result builds on prior work of Bravyi, Gosset, and König [BGK18] and Watts, Kothari, Schaeffer, and Tal [WKST19]. In particular, we use the same search problems introduced in these works and prove that they are average-case hard for GC^0. To prove our lower bound, we prove a new multi-switching lemma for multi-output GC^0 circuits in Section 5.2.1.

Bravyi et al. introduced the 2D Hidden Linear Function Problem and showed that it can be solved by QNC^0 circuits.

Definition 5.12 (2D Hidden Linear Function Problem, 2D HLF [BGK18])**.** Let $b \in \{0,1,2,3\}^n$ be a vector and let $A \in \{0,1\}^n$ be a symmetric matrix describing an $n \times n$ 2D grid, i.e., $A_{ij} = 1$ when vertices i and j are connected on the 2D grid. Define $q : \mathbb{F}_2^n \to \mathbb{Z}_4$ as $q(u) := u^T A u + b^T u \pmod 4$. Define $\mathcal{L}_q$ as

$$\mathcal{L}_q := \{u \in \mathbb{F}_2^n : \forall v \in \mathbb{F}_2^n,\, q(u \oplus v) = q(u) + q(v) \pmod 4\}\,.$$

$\oplus$ denotes the addition of binary vectors modulo two, and the addition $q(u) + q(v)$ is modulo four. Bravyi, Gosset, and König [BGK18] show that the restriction of q onto $\mathcal{L}_q$ is a linear form, i.e., there exists a $z \in \mathbb{F}_2^n$ such that $q(x) = 2z^T x \pmod 4$ for all $x \in \mathcal{L}_q$. Given $A \in \{0,1\}^{n \times n}$ and $b \in \{0,1,2,3\}^n$ as input, the 2D Hidden Linear Function (2D HLF) problem is to output one such $z \in \mathbb{F}_2^n$.

Subsequently, Watts et al. [WKST19] introduced the Parallel Parity Halving Problem and showed that it reduces to 2D HLF.

Definition 5.13 (Parallel Parity Halving Problem, $\mathrm{PHP}^r_{n,m}$ [WKST19])**.** Given r length-n strings $x_1, \ldots, x_r \in \{0,1\}^n$ as input, promised that each x_i has even parity, output r length-m strings $y_1, \ldots, y_r \in \{0,1\}^m$ such that, for all $i \in [p]$,

$$|y_i| \equiv \frac{1}{2}|x_i| \pmod 2.$$

Lemma 5.14 ([WKST19, Theorem 26, Corollary 30])**.** PHP$^n_{m,n}$ *reduces to* 2D HLF*.*

Hence, to obtain our separation between QNC^0 and $\mathsf{GC}^0(k)$ it suffices to prove that PHP is hard for GC^0, which we do in the remainder of this subsection. Doing so yields the following result.

Theorem 5.15 (Generalization of [WKST19, Theorem 1])**.** *The* 2D HLF *problem on n bits cannot be solved by a size-*$\exp(O(n^{1/4d}))$ $\mathsf{GC}^0_d(k)$ *circuit with* $k = O(n^{1/4d})$*. Furthermore, there exists an (efficiently samplable) input distribution on which any* $\mathsf{GC}^0_d(k)$ *circuit (or* $\mathsf{GC}^0_d(k)/\mathsf{rpoly}$ *circuit) of size at most* $\exp(n^{1/4d})$ *only solves the* 2D HLF *problem with probability at most* $\exp(-n^c)$ *for some* $c > 0$*.*

In Section 5.2.1, we prove a multi-switching lemma for multi-output GC^0 circuits necessary for our lower bound. In Section 5.2.2, we prove that PHP is average-case hard to compute for GC^0 circuits, yielding Theorem 5.15. Finally, in Section 5.2.3, we generalize our result further to obtain an exponential separation between *noisy* QNC^0 circuits and $\mathsf{GC}^0(k)$, building on the work of Bravyi, Gosset, König, and Temamichel [BGKT20] and Grier, Ju, and Schaeffer [GJS21].

5.2.1 A Multi-Switching Lemma for GC^0

We prove a multi-output multi-switching lemma for $\mathsf{GC}^0(k)$, building on prior works of Rossman [Ros17], Håstad [Hås14], and Kumar [Kum23]. We must first establish some notation, following Rossman [Ros17] and Watts et al. [WKST19, Appendix A]. Our proof involves the following classes of functions:

- $\mathsf{DT}(w)$ is the class of depth-w decision trees.
- $\mathsf{CKT}(k; d; s_1, \dots, s_d)$ is the class of depth-d $\mathsf{GC}^0(k)$ circuits with s_i gates at the ith layer of the circuit for all $i \in \{1, \dots, d\}$. Note that these circuits have s_d many output bits.
- $\mathsf{CKT}(k; d; s_1, \dots, s_d) \circ \mathsf{DT}(w)$ is the class of circuits in $\mathsf{CKT}(k; d; s_1, \dots, s_d)$ whose inputs are labeled by depth-w decision trees. Note that these functions have s_d many output bits.
- $\mathsf{DT}(t) \circ \mathsf{CKT}(k; d; s_1, \dots, s_d) \circ \mathsf{DT}(w)$ is the class of depth-t decision trees whose leaves are labeled by functions in $\mathsf{CKT}(k; d; s_1, \dots, s_d) \circ \mathsf{DT}(k)$. Note that these functions have s_d many output bits.
- $\mathsf{DT}(w)^m$ is the class of m-tuples of depth-k decision trees. This function has m many output bits, where each output bit is computed by an element of $\mathsf{DT}(w)$.
- $\mathsf{DT}(t) \circ \mathsf{DT}(w)^m$ is the class of depth-t decision trees where each leaf is labeled by an m-tuple of depth-k decision trees. Note that these functions have m many output bits.

In the remainder of this subsection, we will build to the multi-switching lemma by combining ingredients from Rossman [Ros17] and Kumar [Kum23]. To begin, we need the following lemma that says, with high probability, a depth-ℓ decision tree will reduce in depth under random restriction.

Lemma 5.16 ([Ros17, Lemma 20])**.** *For a depth-ℓ decision tree $T \in \mathsf{DT}(\ell)$,*

$$\Pr_{\rho \sim \mathcal{R}_p}[T|_\rho \text{ has depth} \geq t] \leq (2ep\ell/t)^t.$$

We also need the multi-switching lemma for GC^0.

Lemma 5.17 ([Kum23, Theorem 4.8, Lemma 4.9]). *Let* $f \in \mathsf{CKT}(k; d; s_1, \ldots, s_d) \circ \mathsf{DT}(w)$*, then*

$$\Pr_{\rho \sim \mathcal{R}_p}[f|_\rho \notin \mathsf{DT}(t-1) \circ \mathsf{CKT}(k; d-1; s_2, \ldots, s_d) \circ \mathsf{DT}(r)] \leq 4(64(2^k s_1)^{1/r} pw)^t.$$

Proof. This follows immediately from [Kum23, Theorem 4.8, Lemma 4.9]. We include the details for completeness. The bottom two layers of f are s_1 elements of $\mathsf{G}(k) \circ \mathsf{DT}(w)$, i.e., $\mathsf{G}(k)$ gates whose inputs are labeled by depth-w decision trees. [Kum23, Lemma 4.9] shows that $\mathsf{G}(k) \circ \mathsf{DT}(w)$ is equivalent to $\mathsf{G}(k) \circ \mathsf{AND}_w$, i.e., a depth-2 circuit whose bottom layer has fan-in-w AND gates that feed into a $\mathsf{G}(k)$ gate one the top layer. Hence, the s_1 $\mathsf{G}(k) \circ \mathsf{DT}(w)$ substructures in f can be viewed as s_1 $\mathsf{G}(k) \circ \mathsf{AND}_w$ subcircuits. To complete the proof, apply [Kum23, Theorem 4.8] to these s_1 subcircuits. □

We can now show that under random restriction elements of $\mathsf{DT}(t-1) \circ \mathsf{CKT}(k; d; s_1, \ldots, s_d) \circ \mathsf{DT}(w)$ simplify to elements of $\mathsf{DT}(t-1) \circ \mathsf{CKT}(k; d-1; s_2, \ldots, s_d) \circ \mathsf{DT}(r)$ with high probability.

Lemma 5.18 (Generalization of [Ros17, Lemma 24]). *Let* $f \in \mathsf{DT}(t-1) \circ \mathsf{CKT}(k; d; s_1, \ldots, s_d) \circ \mathsf{DT}(w)$*, then*

$$\Pr_{\rho \sim \mathcal{R}_p}[f|_\rho \notin \mathsf{DT}(t-1) \circ \mathsf{CKT}(k; d-1; s_2, \ldots, s_d) \circ \mathsf{DT}(r)] \leq 5(64(2^k s_1)^{1/r} pw)^t.$$

Proof. Say f is computed by a depth-$(t-1)$ decision tree T, where each leaf ℓ is labeled by a circuit $C_\ell \in \mathsf{CKT}(k; d; s_1, s_2, \ldots, s_d) \circ \mathsf{DT}(w)$. Let $\mathcal{E}_1$ be the event $T|_\rho$ has depth $\leq \lfloor t/2 \rfloor - 1$, and let $\mathcal{E}_2$ be the event $C_\ell|_\rho \in \mathsf{DT}(\lceil t/2 \rceil - 1) \circ \mathsf{CKT}(k; d-1; s_2, \ldots, s_d) \circ \mathsf{DT}(r)$ for all leaves ℓ of T. Note that

$$\mathcal{E}_1 \wedge \mathcal{E}_2 \implies f|_\rho \in \mathsf{DT}(t-1) \circ \mathsf{CKT}(k; d-1; s_2, \ldots, s_d) \circ \mathsf{DT}(r).$$

By Lemma 5.16, we know

$$\Pr_{\rho \sim \mathcal{R}_p}[\neg \mathcal{E}_1] \leq (2ep(t-1)/\lceil t/2 \rceil)^{\lceil t/2 \rceil} \leq (4ep)^{t/2}.$$

By Lemma 5.17 and a union bound, we have

$$\begin{aligned}
\Pr_{\rho \sim \mathcal{R}_p}[\neg \mathcal{E}_2] &\leq \sum_{\text{leaves } \ell} \Pr[C_\ell|_\rho \notin \mathsf{DT}(\lceil t/2 \rceil - 1) \circ \mathsf{CKT}(k; d-1; s_2, \ldots, s_d) \circ \mathsf{DT}(r)] \\
&\leq \sum_\ell 4(64(2^k s_1)^{1/r} pw)^t \\
&\leq 2^t \cdot 4(64(2^k s_1)^{1/r} pw)^t \\
&= 4(128(2^k s_1)^{1/r} pw)^t.
\end{aligned}$$

Therefore, we can finally bound

$$\begin{aligned}
\Pr_\rho[f|_\rho \notin \mathsf{DT}(t-1) \circ \mathsf{CKT}(k; d-1; s_2, \ldots, s_d) \circ \mathsf{DT}(r)] &\leq \Pr_\rho[\neg \mathcal{E}_1] + \Pr_\rho[\neg \mathcal{E}_2] \\
&\leq (4ep)^{t/2} + 4(128(2^k s_1)^{1/r} pw)^t \\
&\leq 5(128(2^k s_1)^{1/r} pw)^t.
\end{aligned}$$

□

Lemma 5.18 shows a depth reduction by 1 under random restriction. At a high level, our argument will repeat this process d times to simplify the depth of the circuit to 1 with high probability. When the depth has simplified to 1, we will need the following form of the multi-switching lemma for GC^0 to complete our argument.

Theorem 5.19 ([Kum23, Theorem 4.8, Lemma 4.9] restated)**.** *Let $f \in \mathsf{CKT}(k;1;m) \circ \mathsf{DT}(w)$. Then*

$$\Pr_{\rho \sim \mathcal{R}_p}[f|_\rho \notin \mathsf{DT}(t-1) \circ \mathsf{DT}(r-1)^m] \leq 4(64(2^k m)^{1/r} pw)^t.$$

Proof. Like Lemma 5.17, this follows immediately from [Kum23, Theorem 4.8, Lemma 4.9]. □

We are now ready to prove our multi-output multi-switching lemma for $\mathsf{GC}^0(k)$, the main theorem of this subsection.

Theorem 5.20 (Multi-Output Multi-Switching Lemma for GC)**.** *Let $f \in \mathsf{CKT}(k;d;s_1,\ldots,s_{d-1},m)$ with n inputs and m outputs. Let $s = s_1 + \cdots + s_{d-1} + m$. Let $p = p_1 \cdot p_2 \cdots p_d$ and $w := \lceil \log s \rceil + 1$. Then*

$$\Pr_{\rho \sim \mathcal{R}_p}[f|_\rho \notin \mathsf{DT}(2t-2) \circ \mathsf{DT}(r-1)^m] \leq 5(128 \cdot 2^{k/w} p_1)^t + \sum_{i=2}^{d-1} 5(128 \cdot 2^{k/w} p_i w)^t + 4(128(2^k m)^{1/r} pw)^t.$$

Proof. Let $s_d := m$. Notice we can factor $\rho \sim \mathcal{R}_p$ as $\rho_1 \circ \cdots \circ \rho_d$, where each $\rho_i \sim \mathcal{R}_{p_i}$. Now for each $i \in [d-1]$, define the event

$$\mathcal{E}_i \iff f|_{\rho_1 \cdot \cdots \cdot \rho_i} \in \mathsf{DT}(t-1) \circ \mathsf{CKT}(d-i; s_{i+1}, \ldots, s_d) \circ \mathsf{DT}(w),$$

and define

$$\mathcal{E}_d \iff f|_{\rho_1 \circ \cdots \circ \rho_d} \in \mathsf{DT}(2t-2) \circ \mathsf{DT}(r-1)^m.$$

Notice that

$$\bigwedge_{i=1}^{d} \mathcal{E}_i \implies \mathcal{E}_d \iff f|_\rho \in \mathsf{DT}(2t-2) \circ \mathsf{DT}(q-1)^m.$$

We will bound the complement of this event. Notice that since

$$f \in \mathsf{CKT}(k;d;s_1,\ldots,s_d) \subset \mathsf{DT}(t-1) \circ \mathsf{CKT}(k;d;s_1,\ldots,s_d) \circ \mathsf{DT}(1),$$

we have by Lemma 5.18 that

$$\Pr_\rho[\neg\mathcal{E}_1] \leq 5(64(2^k s_1)^{1/w} p_1)^t \leq 5(128 \cdot 2^{k/w} p_1)^t.$$

For $i = 2, \ldots, d-1$, Lemma 5.18 gives us

$$\Pr[\neg\mathcal{E}_i | \mathcal{E}_1, \ldots, \mathcal{E}_{i-1}] \leq 5(64(2^k s_i)^{1/w} p_i w)^t \leq 5(128 \cdot 2^{k/w} p_i w)^t.$$

We now bound $\Pr[\neg\mathcal{E}_d | \mathcal{E}_1, \ldots, \mathcal{E}_{d-1}]$. Let $g := f|_{\rho_1 \circ \cdots \circ \rho_{d-1}}$. Conditioning on $\mathcal{E}_1, \ldots, \mathcal{E}_{d-1}$, we have

$$g \in \mathsf{DT}(t-1) \circ \mathsf{CKT}(k;1;m) \circ \mathsf{DT}(w).$$

For each leaf ℓ of the partial decision tree of depth $t-1$ for g, define g_ℓ to be g restricted by the root-to-leaf path in the tree to ℓ. It follows that each g_ℓ, by definition, is $\mathsf{CKT}(k;1;m) \circ \mathsf{DT}(w)$. Consequently, by Theorem 5.19, we have for each ℓ,

$$\Pr[g_\ell|_{\rho_d} \notin \mathsf{DT}(t-1) \circ \mathsf{DT}(r-1)^m] \leq 4(64(2^k m)^{1/r} pw)^t.$$

As there are 2^{t-1} leaves, by a union bound it follows that the probability *some* g_ℓ doesn't simplify is at most

$$4(128(2^k m)^{1/r} pw)^t.$$

In the complementary event, we have

$$g|_{\rho_d} = f|_{\rho_1\circ\cdots\circ\rho_d} \in \mathsf{DT}(t-1)\circ\mathsf{DT}(t-1)\circ\mathsf{DT}(q-1)^m = \mathsf{DT}(2t-2)\circ\mathsf{DT}(q-1)^m,$$

so event $\mathcal{E}_d$ holds. We now finally bound

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\rho\sim\mathcal{R}_p}[f|_\rho \notin \mathsf{DT}(2t-2)\circ\mathsf{DT}(r-1)^m] &= \mathbf{Pr}[\neg\mathcal{E}_1,\ldots,\neg\mathcal{E}_d] \\ &= \sum_{i=1}^{d}\mathbf{Pr}[\neg\mathcal{E}_i|\mathcal{E}_1,\ldots,\mathcal{E}_{i-1}] \\ &\le 5(128\cdot 2^{k/w}p_1)^t + \sum_{i=2}^{d-1}5(128\cdot 2^{k/w}p_i w)^t + 4(128(2^k m)^{1/r}pw)^t.\end{aligned}$$

□

Corollary 5.21. *Let $f:\{-1,1\}^n\to\{-1,1\}^m$ be computable by a $\mathsf{GC}^0(k)$ circuit of size s, depth d, and $k = O(\log s)$. Let $p = \frac{1}{m^{1/q}O(\log s)^{d-1}}$. Then, for all $t\in\mathbb{N}$,*

$$\mathop{\mathbf{Pr}}_{\rho\sim\mathcal{R}_p}[f|_\rho \notin \mathsf{DT}(2t-2)\circ\mathsf{DT}(q-1)^m] \le 2^{-t}.$$

Proof. For $w := \lceil\log s\rceil$, we have that $2^{k/w} = O(1)$. Using this and applying Theorem 5.20 with $p_1 = \Omega(1)$, $p_2 = \cdots = p_{d-1} = \Omega(1/w)$, and $p_d = 1/O(m^{1/q}w)$ yields the desired result. □

5.2.2 GC^0 Lower Bound

We can now use our multi-output multi-switching lemma to prove that PHP (Definition 5.13) is hard GC^0 circuits.

Theorem 5.22 ($\mathrm{PHP}^r_{n,m}\notin\mathsf{GC}^0(k)$, Generalization of [WKST19, Theorem 25])**.** *Let $r = n$ and $m\in[n,n^2]$. Any $\mathsf{GC}^0_d(k)$ circuit $F:\{0,1\}^{nr}\to\{0,1\}^{mr}$ with size $s\le\exp\left((nr)^{\frac{1}{2d}}\right)$ and $k = O((nr)^{\frac{1}{2d}})$ solves $\mathrm{PHP}^r_{n,m}$ with probability at most $\exp\left(-n^2/(m^{1+o(1)}O(\log s)^{2(d-1)}\right)$.*

Proof. Set $q = \sqrt{\log(mr)}$, $p = 1/(O(\log s)^{d-1}(mr)^{1/q})$, and $t = pnr/8$. Let ρ be a p-random restriction. The only fact about AC^0 used in the proof of [WKST19, Theorem 25] is that a function F computable by a size-s AC^0 circuit simplifies to an element of $\mathsf{DT}(2t)\circ\mathsf{DT}(q)^m$ under ρ with probability at least $1-\exp(-\Omega(pnr))$. By Corollary 5.21, this holds for functions computable by size-s $\mathsf{GC}^0(k)$ circuits with $k = O(\log s)$. Hence, the rest of the argument in [WKST19, Theorem 25] goes through. □

With that, the main result follows.

Proof of Theorem 5.15. The result follows from combining Lemma 5.14 and Theorem 5.22. □

5.2.3 Separation Between Noisy QNC^0 and GC^0

Our separation between GC^0 and QNC^0 holds even when the QNC^0 circuits are subjected to noise. The noise model considered is the *local stochastic quantum noise model* [FGL20, BGKT20] (see also [GJS21, Section 2.2]). As in prior works, the noise rate is assumed to be below some constant threshold. Here and throughout, "noisy QNC^0" refers to QNC^0 subjected to local stochastic quantum noise with a certain constant noise rate.

Bravyi, Gosset, König, and Temamichel [BGKT20] show that for any relation problem solvable by QNC^0, one can construct a "noisy version" of that relation problem that is solvable by noisy QNC^0 ([BGKT20, Definition 15, Theorem 17], [GJS21, Definition 14, Theorem 15]. Additionally, [GJS21, Lemma 16] implies that any classical circuit solving the noisy version of the relation problem can solve the original relation problem with the overhead of first running a quasipolynomial-size AC^0 circuit.

We can apply this framework to separate $\mathsf{GC}^0(k)$ and noisy QNC^0.

Theorem 5.23 (Generalization of [GJS21, Proposition 18, Theorem 19])**.** *There is a search problem that is solvable by noisy* QNC^0 *with probability* $1 - \exp(-\Omega(\mathrm{polylog}(n)))$*, but any size-*$s$ *depth-*d $\mathsf{GC}^0(k)$ *circuit with* $k = O(\log s)$ *cannot solve the search problem with probability exceeding*

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

Proof. Let the noisy 2D HLF be the relation problem obtained from applying [GJS21, Definition 14] to the 2D HLF (Definition 5.12). The quantum upper bound is precisely [GJS21, Proposition 18].

Towards the classical lower bound, assume there exists a size-s, depth-d $\mathsf{GC}^0(k)$ circuit with $k = O(\log s)$ that solves noisy 2D HLF with probability at most ε. Then, by [GJS21, Lemma 16], there exists a size-$(s + \exp(\mathrm{polylog}(n)))$, depth-$(d + O(1))$ $\mathsf{GC}^0(k)$ circuit with $k = O(\log(s + \exp(\mathrm{polylog}(n)))$ that solves 2D HLF with probability at most ε. But, by Theorem 5.22 and [WKST19, Theorem 26, Corollary 30], any $\mathsf{GC}^0(k)$ circuit of size s, depth d, and $k = O(\log s)$ for 2D HLF succeeds with probability at most

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

Therefore, we can conclude that

$$\varepsilon \le \exp\left(\frac{-n^{1/2-o(1)}}{O\left(\log\left(s + \exp(\mathrm{polylog}(n))\right)\right)^{2d+O(1)}}\right).$$

□

5.3 Separation Between $\mathsf{QNC}^0/\mathsf{qpoly}$ and $\mathsf{GC}^0(k)[2]$

We exhibit a relation problem that can be solved with high probability by a $\mathsf{QNC}^0/\mathsf{qpoly}$ circuit but is average-case hard for $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$. Recall that $\mathsf{QNC}^0/\mathsf{qpoly}$ is the class of QNC^0 circuits with quantum advice, i.e., polynomial-size, constant-depth quantum circuits with bounded fan-in gates that can start with any quantum state as long as it is independent of the input.

Our argument follows the same structure as Watts et al. [WKST19]. However, we obtain an *exponential separation* between $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ and $\mathsf{QNC}^0/\mathsf{qpoly}$. Previously, the best separation was between $\mathsf{QNC}^0/\mathsf{qpoly}$ and *polynomial-size* $\mathsf{AC}^0[2]$ circuits.

Theorem 5.24 (Generalization of [WKST19, Theorem 6])**.** *There is a search problem that is solvable by* $\mathsf{QNC}^0/\mathsf{qpoly}$ *with probability* $1 - o(1)$*, but any* $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ *circuit of depth* d *and size at most* $\exp\left(O\left(n^{1/2.01d}\right)\right)$ *with* $k = O(n^{1/2d})$ *cannot solve the search problem with probability exceeding* $n^{-\Omega(1)}$*.*

The remainder of this subsection is devoted to proving Theorem 5.24. The quantum upper bound is given in [WKST19, Section 6.1, Section 6.3]. We will show an average-case lower bound for the following problem.

Definition 5.25 (r-Parallel Parity Bending Problem [WKST19, Problem 8])**.** Given inputs $x_1, \ldots, x_r$ with $x_i \in \{0,1,2\}^n$ for all $i \in [r]$, produce outputs $y_1, \ldots, y_r \in \{0,1\}^n$ such that y_i satisfies:

$$\begin{aligned} |y_i| &\equiv 0 \pmod 2 \quad \text{if} \quad |x_i| \equiv 0 \pmod 3 \quad \text{or} \\ |y_i| &\equiv 1 \pmod 2 \quad \text{if} \quad |x_i| \not\equiv 0 \pmod 3). \end{aligned}$$

for at least a $\frac{2}{3} + 0.005$ fraction of the $i \in [k]$.

Note that this problem takes input over $\{0,1,2\}$. Ultimately we are studying Boolean circuits, so, technically speaking, trits are encoded with two bits (e.g., $0 \mapsto 00$, $1 \mapsto 01$, $2 \mapsto 10$). We use $\{0,1,2\}$ for notational convenience.

On the way to our lower bound, we first prove lower bounds for the following relation problem.

Definition 5.26 (3 Output Mod 3 [WKST19, Problem 9])**.** Given an input $x \in \{0,1,2\}^n$, output a trit $y \in \{0,1,2\}$ such that $y \equiv |x| \pmod 3$.

To prove 3 Output Mod 3 is hard for $\mathsf{GC}^0(k)[2]$, we use the following worst-case to average-case reduction, given in [WKST19].

Lemma 5.27. *Suppose there is a $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit of size s and depth d that solves 3 Output Mod 3 (Definition 5.26) on a uniformly random input with probability $1/3 + \varepsilon$ for some $\varepsilon > 0$. Then there exists a $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit C of depth $d + O(1)$ and size $s + O(n)$ such that for any $x \in \{0,1,2\}^n$,*

$$\begin{aligned} &\mathbf{Pr}[C(x) \equiv |x| \pmod 3] = \frac{1}{3} + \varepsilon, \text{ and} \\ &\mathbf{Pr}[C(x) \equiv |x| + 1 \pmod 3] = \mathbf{Pr}[C(x) \equiv |x| + 2 \pmod 3] = \frac{1}{3} - \frac{\varepsilon}{2}. \end{aligned}$$

Proof. The proof is exactly the same as [WKST19, Lemma 35]. ☐

We can now show that 3 Output Mod 3 is average-case hard for exponential-size $\mathsf{GC}^0(k)[2]$ circuits.

Lemma 5.28 (Generalization of [WKST19, Lemma 36])**.** *Let $k = O(n^{1/2d})$. Any $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit of depth d and size $s \le \exp\left(O\left(n^{1/2.01d}\right)\right)$ solves 3 Output Mod 3 (Definition 5.26) on the uniform distribution with probability at most $\frac{1}{3} + \frac{1}{n^{\Omega(1)}}$.*

Proof. Let C be the $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit that solves 3 Output Mod 3 on the uniform distribution with probability $\frac{1}{3} + \varepsilon$. Lemma 5.27 implies that there is a circuit C' that succeeds with probability $\frac{1}{3} + \varepsilon$ and outputs each wrong answer with probability $\frac{1}{3} - \frac{\varepsilon}{2}$.

Let $E : \{0,1,2\} \to \{0,1\}$ be the circuit that maps 0 to 0 and everything else to 1. Define C'' to be the circuit that, given input x, outputs 0 with probability $\frac{1}{4}$, and outputs $E(C''(x))$ otherwise. Observe that, if $|x| \equiv 0 \pmod 3$, then C'' correctly outputs 0 with probability $\frac{1}{4} + \frac{3}{4}(\frac{1}{3} + \varepsilon) = \frac{1}{2} + \frac{3\varepsilon}{4}$. Similarly, if $|x| \not\equiv 0 \pmod 3$, then C'' correctly outputs 1 with probability $\frac{3}{4}(\frac{1}{3} + \varepsilon + \frac{1}{3} - \frac{\varepsilon}{2}) = \frac{1}{2} + \frac{3\varepsilon}{8}$. Hence C'' computes MOD_3 with probability $\frac{1}{2} + \frac{3\varepsilon}{8}$, so Theorem 4.4 implies that $\varepsilon \in \frac{1}{n^{\Omega(1)}}$. ☐

The average-case lower bound in Lemma 5.28 implies the following corollary.

Corollary 5.29 (Generalization of [WKST19, Corollary 37])**.** *Let $k = O(n^{1/2d})$. Let C be a $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit of depth d and size $s \le \exp\left(O\left(n^{1/2.01d}\right)\right)$ outputting a trit. Then, for all $i \in \{0,1,2\}$,*

$$\frac{1}{3} - \frac{1}{n^{\Omega(1)}} \le \mathop{\mathbf{Pr}}_{x \in \{0,1,2\}^n}[C(x) - |x| \equiv i \pmod 3] \le \frac{1}{3} + \frac{1}{n^{\Omega(1)}}.$$

Proof. Because

$$\sum_{i\in\{0,1,2\}} \Pr_{x\in\{0,1,2\}^n}[C(x)-|x|\equiv i \pmod 3] = 1,$$

it suffices to prove

$$\Pr_{x\in\{0,1,2\}^n}[C(x)-|x|\equiv i \pmod 3] \le \frac{1}{3}+\frac{1}{n^{\Omega(1)}}$$

for each $i\in\{0,1,2\}$. For $i=0$, the desired bound is exactly shown in Lemma 5.28. For $i\in\{1,2\}$, observe that if there is a $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit D of depth d and size at most $\exp\left(O\left(n^{1/2.01d}\right)\right)$ for which

$$\Pr[D(x)-|x|\equiv i \pmod 3] \ge \frac{1}{3}+\frac{1}{n^{o(1)}},$$

then one could construct a circuit D' for which

$$\Pr[D'(x)\equiv |x| \pmod 3] \ge \frac{1}{3}+\frac{1}{n^{o(1)}}$$

by subtracting by the trit i at the end of the circuit. Subtracting by a fixed trit only adds a constant overhead to the size and depth of the circuit, so such a D' contradicts Lemma 5.28. $\square$

We note that [WKST19, Corollary 37] is only stated for polynomial-size $\mathsf{AC}^0[2]/\mathsf{rpoly}$ circuits. However, we observe the statement also holds for exponential-size circuits, as demonstrated in Corollary 5.29. This allows us to obtain exponentially stronger lower bounds than the ones obtained in [WKST19].

Now we study the difficulty of solving r instances of the 3 Output Mod 3 Problem.

Definition 5.30 (r-Parallel 3 Output Mod 3)**.** Given inputs $x_1,\dots,x_r\in\{0,1,2\}^n$, output a vector $\vec{y}\in\{0,1,2\}^r$ such that

$$y_i\equiv |x_i| \pmod 3$$

for at least a $\frac{1}{3}+0.01$ fraction of the $i\in[k]$.

To prove lower bounds for this problem, we use the XOR lemma for finite abelian groups.

Lemma 5.31 ([Rao07, Lemma 4.2], XOR lemma for finite abelian groups)**.** *Let $\mathcal{D}$ be a distribution over a finite abelian group G such that $|\mathbf{E}[\psi(X)]|\le\varepsilon$ for every non-trivial character ψ. Then $\mathcal{D}$ is $\varepsilon\sqrt{|G|}$-close (in total variation distance) to the uniform distribution over G.*

Theorem 5.32 (Generalization of [WKST19, Theorem 39])**.** *Let $k=O(n^{1/2d})$. There exists an $r\in\Theta(\log n)$ for which any $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit of depth d and size $s\le\exp\left(O\left(n^{1/2.01d}\right)\right)$ solves the r-Parallel 3 Output Mod 3 Problem (Definition 5.30) with probability at most $n^{-\Omega(1)}$.*

Proof. For $k=O(n^{1/2d})$, let C be a $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit of depth d and size at most $\exp\left(O\left(n^{1/2.01d}\right)\right)$ that solves the r-Parallel 3 Output Mod 3 problem with probability ε. Throughout this proof, let $x_1,\dots,x_r\in\{0,1,2\}^n$ be chosen uniformly at random, and let $(y_1,\dots,y_r)$ be the output trits of the circuit C. Let $\mathcal{D}$ be the distribution over r trits defined by

$$\bigotimes_{i=1}^{r}(|x_i|-y_i \pmod 3).$$

We begin by showing that $\mathcal{D}$ is close to the uniform distribution over $\{0,1,2\}^r$ in total variation distance. Let χ_a be the character of $\mathbb{F}_3^r$ corresponding to $a\in\mathbb{F}_3^r$. Recall that $\chi_a(z) := \omega^{\sum_{i=1}^r a_i z_i}$, where $z \in \mathbb{F}_3^r$ and ω is a third root of unity. To show that $\mathcal{D}$ is close to uniform, it suffices to show that $|\mathbf{E}[\chi_a(\mathcal{D})]|$ is small for all nonzero a.

For $a \in \mathbb{F}_3^r$, let S denote the set of indices on which $a_i \neq 0$. Consider the problem where, given a nonzero $a \in \mathbb{F}_3^r$ and strings $x_1, \ldots, x_r \in \{0,1,2\}^n$, the goal is to find trits $y_1, \ldots, y_r$ such that

$$\sum_{i \in S} a_i |x_i| \equiv \sum_{i \in S} a_i y_i \pmod{3}.$$

This problem reduces to 3 Output Mod 3 on the concatenated input $\widetilde{x} \coloneqq (a_i x_{j,i})_{i \in S, j \in [r]} \in \{0,1,2\}^{n|S|}$. Specifically, given any circuit A solving the former problem, one can solve the latter problem by first running the circuit A to obtain the trits $y_1, \ldots, y_r$. Then, add a circuit to compute the sum $\sum_{i \in S} a_i y_i \pmod{3}$, which is the correct answer to the 3 Output Mod 3 problem on input $\widetilde{x}$. This last step can be done with a depth-2 AC^0 circuit with $\exp(|S|) \leq \exp(r) \leq \mathrm{poly}(n)$ many gates.

Now, because we are choosing $x_1, \ldots, x_r$ uniformly at random, the concatenated input $\widetilde{x} \in \{0,1,2\}^{n|S|}$ is uniformly random. Therefore, Corollary 5.29 implies that the distribution

$$\sum_{i \in S} a_i (|x_i| - y_i) \pmod{3}$$

is at most $n^{-\Omega(1)}$-far from the uniform distribution over a trit $\{0,1,2\}$ in total variation distance. Hence, $|\mathbf{E}[\chi_a(\mathcal{D})]| \leq n^{-\Omega(1)}$ for each nonzero a. Then, Lemma 5.31 implies that $\mathcal{D}$ is $n^{-\Omega(1)}\sqrt{3^r}$-close to the uniform distribution on $\{0,1,2\}^r$.

Because $\mathcal{D}$ is close to uniform, the probability ε that the circuit C solves the r-Parallel 3 Output Mod 3 problem is (almost) equivalent to the probability that a uniformly random string in $\{0,1,2\}^r$ has more than a $\frac{1}{3} + 0.01$ fraction of its trits set to 0. By a Chernoff bound, this probability is bounded above by $\exp(-\Omega(r))$. More carefully, we see that the probability of C solving the r-Parallel 3 Output Mod 3 problem is at most

$$n^{-\Omega(1)}\sqrt{3^r} + \exp(-\Omega(r)),$$

which is bounded above by $n^{-\Omega(1)}$ for some $r \in \Theta(\log n)$. $\square$

In [WKST19, Theorem 40] they show that the r-Parallel Parity Bending Problem (Definition 5.25) is as hard as the r-Parallel 3 Output Mod 3 Problem (Definition 5.30). Their reduction and Theorem 5.32 imply the following corollary.

Corollary 5.33. *Let $k = O(n^{1/2d})$. There exists an $r \in \Theta(\log n)$ for which any $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit of depth d and size at most $\exp\left(O\left(n^{1/2.01d}\right)\right)$ solves the r-Parallel Parity Bending Problem with probability at most $n^{-\Omega(1)}$.*

Combining Corollary 5.33 with the quantum upper bound in [WKST19, Section 6] implies Theorem 5.24.

5.4 Separation Between $\mathsf{QNC}^0/\mathsf{qpoly}$ and $\mathsf{GC}^0(k)[p]$

We exhibit relation problems that can all be solved by $\mathsf{QNC}^0/\mathsf{qpoly}$ but each one is average-case hard for $\mathsf{GC}^0(k)[p]$ for some prime $p \neq 2$. Since we proved a separation when $p = 2$ in the previous subsection (Theorem 5.24), we have an exponential separation between $\mathsf{QNC}^0/\mathsf{qpoly}$ and $\mathsf{GC}^0(k)[p]$ for all primes p.

Theorem 5.34. *For any prime p, there is a search problem that is solvable by $\mathsf{QNC}^0/\mathsf{qpoly}$ with probability $1-o(1)$, but any $\mathsf{GC}^0(k)[p]/\mathsf{rpoly}$ circuit of depth d and size at most $\exp\left(O(n^{1/2.01d})\right)$ with $k = O(n^{1/2d})$ cannot solve the search problem with probability exceeding $n^{-\Omega(1)}$.*

We also note that we use the case where $p = 2$ to obtain separations for primes $p \neq 2$, which is why the $p = 2$ case is handled in a separate subsection.

Previously, the best separation known was between $\mathsf{QNC}^0/\mathsf{qpoly}$ and polynomial-size $\mathsf{AC}^0[p]$ circuits, which was shown in the recent work of Grilo, Kashefi, Markham, and Oliveira [GKMdO24]. The case where $p = 2$ was shown in Section 5.3. We handle all other primes in this subsection. We will show lower bounds for the following problem, which is a natural generalization of the r-Parallel Parity Bending Problem introduced by [WKST19, Problem Problem 8].

Definition 5.35 ((q, r)-Parallel Parity Bending Problem [GKMdO24, Definition 4])**.** Given inputs $x_1, \ldots, x_r$ with $x_i \in \{0,1\}^n$ for all $i \in [r]$, produce outputs $y_1, \ldots, y_r \in \{0,1\}^n$ such that y_i satisfies:

$$\begin{aligned} |y_i| &\equiv 0 \pmod q \quad \text{if} \quad |x_i| \equiv 0 \pmod 2 \quad \text{or} \\ |y_i| &\not\equiv 0 \pmod q \quad \text{if} \quad |x_i| \equiv 1 \pmod 2. \end{aligned}$$

for at least a $\frac{2}{3} + 0.005$ fraction of the $i \in [k]$.

Grilo et al. [GKMdO24] prove that $\mathsf{QNC}^0/\mathsf{qpoly}$ can solve this problem. We prove that the problem is average-case hard for $\mathsf{GC}^0(k)[p]$ for all primes $p \neq 2$. We begin with the following corollary of Theorem 4.4.

Corollary 5.36. *Let $k = O(n^{1/2d})$. For a prime $p \neq 2$, let C be a $\mathsf{GC}^0(k)[p]/\mathsf{rpoly}$ circuit of depth d and size $s \le \exp\left(O\left(n^{1/2.01d}\right)\right)$. Then, for all $i \in \{0,1\}$,*

$$\frac{1}{2} - \frac{1}{n^{\Omega(1)}} \le \mathop{\mathbf{Pr}}_{x \in \{0,1\}^n}\left[C(x) - |x| \equiv i \pmod 2\right] \le \frac{1}{2} + \frac{1}{n^{\Omega(1)}}.$$

Proof. The proof is similar to Corollary 5.29. Because

$$\sum_{i \in \{0,1\}} \mathop{\mathbf{Pr}}_{x \in \{0,1\}^n}\left[C(x) - |x| \equiv i \pmod 2\right] = 1,$$

it suffices to prove

$$\mathop{\mathbf{Pr}}_{x \in \{0,1\}^n}\left[C(x) - |x| \equiv i \pmod 2\right] \le \frac{1}{2} + \frac{1}{n^{\Omega(1)}}$$

for each $i \in \{0,1\}$. For $i = 0$, the desired bound is exactly shown in Theorem 4.4. For $i = 1$, observe that if there is a $\mathsf{GC}^0(k)[p]/\mathsf{rpoly}$ circuit D of depth d and size at most $\exp\left(O\left(n^{1/2.01d}\right)\right)$ for which

$$\mathbf{Pr}[D(x) - |x| \equiv 1 \pmod 2] \ge \frac{1}{2} + \frac{1}{n^{o(1)}},$$

then one could construct a circuit D' for which

$$\mathbf{Pr}[D'(x) \equiv |x| \pmod 2] \ge \frac{1}{2} + \frac{1}{n^{o(1)}}$$

adding a NOT gate to the end of the circuit. However, such a D' cannot exist as it contradicts Theorem 4.4. ∎

We now prove our average-case lower bound.

Theorem 5.37. *Let $p \neq 2$ be a prime, and let $k = O(n^{1/2d})$. There exists an $r \in \Theta(\log n)$ for which any $\mathsf{GC}^0(k)[p]/\mathsf{rpoly}$ circuit of depth d and size at most $\exp\left(O\left(n^{1/2.01d}\right)\right)$ solves the (q, r)-Parallel Parity Bending Problem (Definition 5.35) with probability at most $n^{-\Omega(1)}$.*

Proof. The proof is similar to Theorem 5.32. For $k = O(n^{1/2d})$, let C be a $\mathsf{GC}^0(k)[2]/\mathsf{rpoly}$ circuit of depth d and size at most $\exp\left(O\left(n^{1/2.01d}\right)\right)$ that, on input $x_1, \ldots, x_r \in \{0,1\}^n$, outputs $y_1, \ldots, y_r \in \{0,1\}$ such that, for at least a $\frac{1}{2} + 0.01$ fraction of $i \in [r]$, $y_i \equiv |x_i| \pmod 2$. Let ε denote the probability that C succeeds at this task. Throughout this proof, consider $x_1, \ldots, x_r$ to be chosen uniformly at random. Let $\mathcal{D}$ be the distribution over r bits defined by

$$\bigotimes_{i=1}^{r} (|x_i| - y_i \pmod 2).$$

We will show that $\mathcal{D}$ is close to the uniform distribution over $\{0,1\}^r$ in total variation distance. Let χ_a be the character of $\mathbb{F}_2^r$ corresponding to $a \in \mathbb{F}_2^r$. Recall that $\chi_a(z) \coloneqq (-1)^{\sum_{i=1}^r a_i z_i}$, where $z \in \mathbb{F}_2^n$. We will show that $|\mathbf{E}[\chi_a(\mathcal{D})]|$ is small for all nonzero a, which implies that $\mathcal{D}$ is close to the uniform distribution in total variation distance.

For $a \in \mathbb{F}_2^r$, let S denote the set of indices on which $a_i \neq 0$. Consider the problem where, given a nonzero $a \in \mathbb{F}_2^r$ and strings $x_1, \ldots, x_r \in \{0,1\}^n$, the goal is to find $y_1, \ldots, y_r \in \{0,1\}$ such that

$$\sum_{i \in S} a_i |x_i| \equiv \sum_{i \in S} a_i y_i \pmod 2.$$

Let $\widetilde{x} \coloneqq (a_i x_{j,i})_{i \in S, j \in [r]} \in \{0,1\}^{n|S|}$, i.e., the bits chosen by a for each x_i for $i \in [r]$. The problem above reduces to computing MOD_2 on $\widetilde{x}$. Specifically, let $y_1, \ldots, y_r$ be the output of a circuit solving the former problem. Then, add a circuit that computes the sum $\sum_{i \in S} a_i y_i \pmod 2$, which is equal to $\mathsf{MOD}_2(\widetilde{x})$. Note this last step requires at most a depth-2 AC^0 circuit with $\exp(|S|) \leq \exp(r) \leq \mathrm{poly}(n)$ many gates.

Next, because $x_1, \ldots, x_r$ are uniformly random, so too is the concatenated input $\widetilde{x}$. Therefore, Corollary 5.36 implies that the distribution

$$\sum_{i \in S} a_i (|x_i| - y_i) \pmod 2$$

is at most $n^{-\Omega(1)}$-far from the uniform distribution over a single bit in total variation distance. Hence, $|\mathbf{E}[\chi_a(\mathcal{D})]| \leq n^{-\Omega(1)}$. Then, Lemma 5.31 implies that $\mathcal{D}$ is $n^{-\Omega(1)}\sqrt{2^r}$-close to the uniform distribution on $\{0,1\}^r$.

Note that for a sample drawn from $\mathcal{D}$, the bits that are 0 correspond to the circuit successfully computing MOD_2 on the corresponding input. Hence, the success probability ε of C is precisely the probability that a sample drawn from $\mathcal{D}$ has more than a $\frac{1}{2} + 0.01$ fraction of the bits set to 0. By a Chernoff bound, the probability that a uniformly random string in $\{0,1\}^r$ has more than a $\frac{1}{2} + 0.01$ of its bits set to 0 is at most $\exp(-\Omega(r))$. Because $\mathcal{D}$ is $n^{-\Omega(1)}\sqrt{2^r}$-close to uniform (in variation distance), we have that ε (i.e., the probability that the number of bits in a sample drawn from $\mathcal{D}$ has more than a $\frac{1}{2} + 0.01$ fraction of its bits set to 0) is at most

$$n^{-\Omega(1)}\sqrt{3^r} + \exp(-\Omega(r)),$$

which is bounded above by $n^{-\Omega(1)}$ for some $r \in \Theta(\log n)$.

At this point, we have shown that any $\mathsf{GC}^0(k)[p]/\mathsf{rpoly}$ circuit of depth d and size at most $\exp(O(n^{1/2.01d}))$ trying to compute $\mathsf{MOD}_2(x_i)$ on a $\frac{1}{2}+0.01$ fraction of inputs $x_1,\ldots,x_r \in \{0,1\}^n$ will succeed with probability at most $n^{-\Omega(1)}$. To complete the proof, we give a reduction from this problem to the (q,r)-Parallel Parity Bending Problem, following the reduction given in [WKST19, Theorem 40]. Suppose we have a solution $y_1,\ldots,y_r$ to (q,r)-Parallel Parity Bending Problem. Then we can output $y'_1,\ldots,y'_r$ solving the above problem as follows. For y_i, set $y'_i = 0$ when $|y_i| \equiv 0 \pmod{q}$, and set $y'_i = 1$ otherwise. This transformation preserves the number of successes, i.e., if y_i is correct for the (q,r)-Parallel Parity Bending Problem, then y'_i will equal $\mathsf{MOD}_2(x_i)$. ☐

5.5 On Interactive QNC^0 Circuits

Grier and Schaeffer [GS20] obtain quantum-classical separations for two-round interactive problems. We provide a high-level overview of their interactive problems and refer readers to [GS20] for further detail. The problems involve a simple quantum state $|G\rangle$ that is fixed (independent of the input). In the first round, the input specifies a sequence of Clifford gates to be applied to $|G\rangle$, along with a subset of $n - O(1)$ qubits to measure in the standard basis. A valid output for this round is any measurement outcome that could have been observed if the measurement was performed on an actual quantum computer.

In the second round, a similar process occurs: the input specifies a sequence of Clifford gates to be applied to the $O(1)$ qubits that were not measured in the first round. Again, a valid output is any measurement outcome that could have been observed if the measurement was performed on a quantum computer.

To summarize, all the interactive problems in [GS20] revolve around simulating a Clifford circuit on n qubits, and the simulation is broken into two rounds. The *point* is that this problem caters to quantum devices, and the interactive aspect is crucial for proving lower bounds.

In more detail, Grier and Schaeffer give three different interactive tasks $\mathsf{T}_1, \mathsf{T}_2$, and T_3 that follow the above structure. The differences between the three tasks come from, e.g., the geometry of the starting state $|G\rangle$. It is not too surprising that Grier and Schaeffer show that QNC^0 can solve their interactive tasks. On the other hand, they prove that any classical model that can solve these interactive tasks (i.e., *simulate* the action on the fixed state $|G\rangle$) must be fairly powerful. A bit more carefully, [GS20, Theorem 1] shows that $\mathsf{AC}^0[6] \subseteq \left(\mathsf{AC}^0\right)^{\mathsf{T}_1}$, $\mathsf{NC}^1 \subseteq \left(\mathsf{AC}^0\right)^{\mathsf{T}_2}$, and $\oplus\mathsf{L} \subseteq \left(\mathsf{AC}^0\right)^{\mathsf{T}_3}$.

To illustrate the usefulness of their theorem, let us explain how it implies a separation between $\mathsf{AC}^0[2]$ and QNC^0. For the upper bound, they show that QNC^0 can solve any of the tasks T_i. For the lower bound, suppose towards a contradiction that $\mathsf{AC}^0[2]$ can solve T_2. Then, by Grier and Schaeffer's theorem, this implies that $\mathsf{NC}^1 \subseteq \left(\mathsf{AC}^0\right)^{\mathsf{AC}^0[2]} = \mathsf{AC}^0[2]$, but this is a contradiction because the containment of $\mathsf{AC}^0[2]$ in NC^1 is known to be strict.

The remainder of this subsection will use Grier and Schaeffer's framework to show that there is an interactive task that QNC^0 circuits can solve but $\mathsf{GC}^0(k)[p]$ circuits cannot. We begin by showing that even a single $\mathsf{G}(k)$ gate can compute functions that are not computable by $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$.

Theorem 5.38. *There is a function $f : \{0,1\}^n \to \{0,1\}$ computable by a single $\mathsf{G}(k)$ gate that is not computable in NC^i for any constant i and $k = \omega(\log^{i-1}(n))$. When $k \in \log^{\omega(1)}(n)$, then there are functions $f : \{0,1\}^n \to \{0,1\}$ that are computable by a single $\mathsf{G}(k)$ gate that cannot be computed in $\mathsf{NC} = \mathsf{AC} = \mathsf{TC}$.*

Proof. We count the functions computable by NC^i and a single $\mathsf{G}(k)$. For NC^i, since the circuit has depth $O(\log^i n)$ with fan-in 2, there are $\leq 2^{O(\log^i n)}$ gates in any NC^i circuit. Furthermore, all fan-in points of these gates are connected by a wire to the fan-out of another gate or an input bit.

Each gate can be one of {AND, OR, NOT}, giving $6^{O(\log^i n)}$ many options. For each fan-in point of a gate, there exists $\leq 2^{O(\log^i n)} + n + 2$ many choices of wires that will connect this fan-in point to either the fan-out of another gate, an input variable, or a constant $0/1$ bit. This gives a total of $6^{O(\log^i n)}(2^{O(\log^i n)} + n + 2)^{2^{O(\log^i n)}} = 2^{\widetilde{O}(2^{\log^i n})}$ NC^1 circuits. Meanwhile, the number of $\mathsf{G}(k)$ gates of fan-in n is at least $2^{\binom{n}{\leq k}}$. To see this, note that $\binom{n}{\leq k}$ many inputs can be assigned arbitrarily, giving $2^{\binom{n}{\leq k}}$ many options.[15] This number exceeds $2^{\widetilde{O}(2^{\log^i n})}$ as long as $k = \omega(\log^{i-1}(n))$. The final part of the theorem follows from setting $k = \log^{\omega(1)}(n)$. ☐

As another form of Theorem 5.38, we can also show that, e.g., a single $\mathsf{G}(k)$ gate can compute functions that require exponential-size TC^0 circuits. We find this interesting in its own right, because proving lower bounds for TC^0 is currently beyond our techniques.

Theorem 5.39. *There is a function $f : \{0,1\}^n \to \{0,1\}$ computable by a single $\mathsf{G}(k)$ gate that requires $2^{\widetilde{\Omega}(n^\varepsilon)}$-size TC^0 circuits for $k = \Omega(n^\varepsilon)$ and $\varepsilon > 0$.*

Proof. We use a counting argument. In the proof of Theorem 5.38, we showed that there are at least $2^{\binom{n}{\leq k}}$ functions computable by $\mathsf{G}(k)$ gates. We will give a loose upper bound on the number of size-s TC^0 circuits, which suffices for our purposes. There are 4 choices each gate could be (from {AND, OR, NOT, MAJ}), and each gate has $\leq s$ choices on its fan-in. Thus the total number of ways to pick our s gates are at most $(4s)^s$. Each fan-in wire of each gate can be connected to the fan-out of another gate, an input bit, or a constant $0/1$ bit, so there are $s + n + 2$ many options. Thus the total number of size-s TC^0 circuits is at most $(4s)^s(s + n + 2)^s \leq (s(s + n))^{O(s)}$. For $s = \Omega(n^{k-1})$ and $k \geq 2$, this quantity is $\leq (s(s + n))^{O(s)} = 2^{O(s \log s)} = 2^{\widetilde{O}(n^{k-1})}$. But $2^{\widetilde{O}(n^{k-1})} = o(2^{\binom{n}{\leq k}})$, so the number of size-$s$ TC^0 circuits is smaller than the number of $\mathsf{G}(k)$ gates for $s = \Omega(n^{k-1})$ and $k \geq 2$. Hence, there exists a $\mathsf{G}(k)$ gate that cannot be computed by size n^{k-1} circuits. In particular, by setting $k = n^\varepsilon$, we see that $\mathsf{GC}^0(n^\varepsilon)$ requires $2^{\widetilde{\Omega}(n^\varepsilon)}$-size TC^0 circuits. ☐

We say two circuit classes C and D are incomparable when there are functions $f, g : \{0,1\}^n \to \{0,1\}$ such that $f \in \mathsf{C}$ but $f \notin \mathsf{D}$ and $g \notin \mathsf{C}$ but $g \in \mathsf{D}$.

Corollary 5.40. *Let p be a prime number. For $k \in \omega(1)$, the class of depth-d $\mathsf{GC}^0(k)[p]$ circuits of size at most $\exp\left(O(n^{1/2.01d})\right)$ is incomparable to NC^1.*

Proof. Theorem 4.4 says that MAJ cannot be computed by $\mathsf{GC}^0(k)[p]$ for any prime p. MAJ can be computed by NC^1 because $\mathsf{NC}^1 \supseteq \mathsf{TC}^0$. Theorem 5.38 implies that there is a function that can be computed by $\mathsf{GC}^0(k)[p]$ but not NC^1. ☐

We can now use Grier and Schaeffer's framework to get a separation between QNC^0 and $\mathsf{GC}^0(k)[p]$ for an interactive problem.

Theorem 5.41 (Generalization of [GS20, Corollary 2])**.** *Let $k = O(n^{1/2d})$. There is an interactive task that QNC^0 circuits can solve that depth-d, size-s $\mathsf{GC}^0(k)[p]$ circuits cannot for $s \leq \exp\left(O(n^{1/2.01d})\right)$.*

Proof. Grier and Schaeffer's task T_2 ([GS20, Problem 12]) can be solved by QNC^0. Suppose it can be solved by $\mathsf{GC}^0(k)[p]$ circuits for some prime p. Then, by [GS20, Theorem 1], $\mathsf{NC}^1 \subseteq (\mathsf{AC}^0)^{\mathsf{GC}^0(k)[p]} = \mathsf{GC}^0(k)[p]$ but this contradicts Corollary 5.40. ☐

[15] By counting carefully, one can show that the number of $\mathsf{G}(k)$ gates is $2 \cdot 2^{\binom{n}{\leq k}}$ for $0 \leq k \leq n-1$, and 2^{2^n} for $k = n$. We do not need this for our argument.

Acknowledgements

We thank Scott Aaronson, Srinivasan Arunachalam, Anna Gál, Jesse Goodman, Daniel Grier, Siddhartha Jain, Nathan Ju, William Kretschmer, Shyamal Patel, Ryan Williams, and Justin Yirka for helpful conversations. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing, supported by NSF QLCI Grant No. 2016245, and in part while SG was an intern at IBM Quantum.

SG is supported by the NSF QLCI Award OMA-2016245 (Scott Aaronson). VMK is supported by NSF Grants CCF-2008076 and CCF-2312573, and a Simons Investigator Award (#409864, David Zuckerman).

References

[AA15] Scott Aaronson and Andris Ambainis. FORRELATION: A Problem that Optimally Separates Quantum from Classical Computing. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 307–316, 2015. arXiv:1411.5729, doi:10.1145/2746539.2746547. [p. 10]

[Aar10] Scott Aaronson. BQP and the Polynomial Hierarchy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, pages 141–150, 2010. arXiv:2010.05846, doi:10.1145/1806689.1806711. [pp. 3, 29]

[Aar16] Scott Aaronson. $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$. *Open Problems in Mathematics*, pages 1–122, 2016. scottaaronson.com. doi:10.1007/978-3-319-32162-2_1. [p. 3]

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. doi:10.5555/1540612. [pp. 13, 29]

[ABO84] Miklós Ajtai and Michael Ben-Or. A Theorem on Probabilistic Constant Depth Computations. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pages 471–474, 1984. doi:10.1145/800057.808715. [p. 22]

[ADH97] Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh A. Huang. Quantum Computability. *SIAM Journal on Computing*, 26(5):1524–1540, 1997. doi:10.1137/S0097539795293639. [pp. 11, 30]

[AG94] Eric Allender and Vivek Gore. A Uniform Circuit Lower Bound for the Permanent. *SIAM Journal on Computing*, 23(5):1026–1049, 1994. doi:10.1137/S0097539792233907. [pp. 8, 23]

[AGS21] Srinivasan Arunachalam, Alex Bredariol Grilo, and Aarthi Sundaram. Quantum Hardness of Learning Shallow Classical Circuits. *SIAM Journal on Computing*, 50(3):972–1013, 2021. arXiv:1903.02840, doi:10.1137/20M1344202. [p. 5]

[AH94] Eric Allender and Ulrich Hertrampf. Depth Reduction for Circuits of Unbounded Fan-in. *Information and Computation*, 112(2):217–238, 1994. doi:10.1006/inco.1994.1057. [pp. 17, 19, 20, 23]

[AIK22] Scott Aaronson, DeVon Ingram, and William Kretschmer. The Acrobatics of BQP. In *37th Computational Complexity Conference (CCC 2022)*, volume 234 of *Leibniz*

International Proceedings in Informatics (LIPIcs), pages 20:1–20:17, 2022. doi:10.4230/LIPIcs.CCC.2022.20. [p. 11]

[Ajt83] Miklós Ajtai. Σ_1-Formulae on Finite Structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983. doi:10.1016/0168-0072(83)90038-6. [p. 3]

[AW93] Eric W. Allender and Klaus W. Wagner. Counting Hierarchies: Polynomial Time and Constant Depth Circuits. In *Current Trends in Theoretical Computer Science: Essays and Tutorials*, pages 469–483. World Scientific, 1993. doi:10.1142/9789812794499_0035. [p. 29]

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A New Barrier in Complexity Theory. *ACM Transactions on Computation Theory*, 1(1):1–54, 2009. doi:10.1145/1490270.1490272. [pp. 3, 4]

[BG81] Charles H. Bennett and John Gill. Relative to a Random Oracle A, $\mathsf{P}^A \neq \mathsf{NP}^A \neq \mathsf{coNP}^A$ with Probability 1. *SIAM Journal on Computing*, 10(1):96–113, 1981. doi:10.1137/0210008. [p. 29]

[BGK18] Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018. arXiv:1704.00690, doi:10.1126/science.aar3106. [pp. 3, 10, 11, 26, 30]

[BGKT20] Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, 2020. arXiv:1904.01502, doi:10.1038/s41567-020-0948-z. [pp. 10, 11, 12, 26, 31, 34, 35]

[BGS75] Theodore Baker, John Gill, and Robert Solovay. Relativizations of the $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ Question. *SIAM Journal on computing*, 4(4):431–442, 1975. doi:10.1137/0204037. [pp. 3, 4]

[Blu81] Norbert Blum. *A 2.75n-lower bound on the network complexity of Boolean functions*. Technical Report A81/05, Universität des Saarlandes, 1981. [p. 3]

[Blu83] Norbert Blum. A Boolean function requiring $3n$ network size. *Theoretical Computer Science*, 28(3):337–345, 1983. doi:10.1016/0304-3975(83)90029-4. [p. 3]

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921. [p. 10]

[CHO+22] Lijie Chen, Shuichi Hirahara, Igor Carboni Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. Beyond natural proofs: Hardness magnification and locality. *Journal of the ACM*, 69(4), August 2022. doi:10.1145/3538391. [pp. 3, 4]

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning Algorithms from Natural Proofs. In *31st Conference on Computational Complexity (CCC 2016)*, 2016. doi:10.4230/LIPIcs.CCC.2016.1. [pp. 8, 9, 20, 25, 26]

[CSV19] Matthew Coudron, Jalex Stark, and Thomas Vidick. Trading Locality for Time: Certifiable Randomness from Low-Depth Circuits. *Communications in Mathematical Physics*, 2019. arXiv:1810.04233, doi:10.1007/s00220-021-03963-w. [p. 10]

[DK11] Evgeny Demenkov and Alexander S Kulikov. An Elementary Proof of a $3n - o(n)$ Lower Bound on the Circuit Complexity of Affine Dispersers. In *International Symposium on Mathematical Foundations of Computer Science*, pages 256–265. Springer, 2011. eccc:TR11-026. doi:10.1007/978-3-642-22993-0_25. [p. 3]

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S Kulikov. A Better-Than-$3n$ Lower Bound for the Circuit Complexity of an Explicit Function. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science*, pages 89–98, 2016. eccc:TR15-166. doi:10.1109/FOCS.2016.19. [p. 3]

[FGL20] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Constant overhead quantum fault tolerance with quantum expander codes. *Communications of the ACM*, 64(1):106–114, 2020. arXiv:1808.03821, doi:10.1145/3434163. [p. 34]

[FSS84] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, Circuits, and the Polynomial-Time Hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. doi:10.1007/BF01744431. [p. 3]

[FSUV13] Bill Fefferman, Ronen Shaltiel, Christopher Umans, and Emanuele Viola. On Beating the Hybrid Argument. *Theory of Computing*, 9(26):809–843, 2013. doi:10.4086/toc.2013.v009a026. [p. 29]

[GJS21] Daniel Grier, Nathan Ju, and Luke Schaeffer. Interactive quantum advantage with noisy, shallow Clifford circuits, 2021. arXiv:2102.06833. [pp. 10, 11, 12, 26, 31, 34, 35]

[GK16] Alexander Golovnev and Alexander S. Kulikov. Weighted Gate Elimination: Boolean Dispersers for Quadratic Varieties Imply Improved Circuit Lower Bounds. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 405–411, 2016. doi:10.1145/2840728.2840755. [p. 3]

[GKMdO24] Alex Bredariol Grilo, Elham Kashefi, Damian Markham, and Michael de Oliveira. The power of shallow-depth Toffoli and qudit quantum circuits, 2024. arXiv:2404.18104. [pp. 10, 11, 26, 39]

[Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008. doi:10.1017/CBO9780511804106. [pp. 13, 29]

[GS20] Daniel Grier and Luke Schaeffer. Interactive shallow Clifford circuits: Quantum advantage against NC^1 and beyond, 2020. arXiv:1911.02555, doi:10.1145/3357713.3384332. [pp. 3, 10, 11, 41, 42]

[Hås86] Johan Håstad. *Computational limitations for small depth circuits*. PhD thesis, Massachusetts Institute of Technology, 1986. [p. 3]

[Hås14] Johan Håstad. On the Correlation of Parity and Small-Depth Circuits. *SIAM Journal on Computing*, 43(5):1699–1708, 2014. doi:10.1137/120897432. [pp. 22, 27, 31]

[HMdOS24] Min-Hsiu Hsieh, Leandro Mendes, Michael de Oliveira, and Sathyawageeswar Subramanian. Unconditionally separating noisy QNC^0 from bounded polynomial threshold circuits of constant depth, 2024. [p. 12]

[HRST17] Johan Håstad, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An Average-Case Depth Hierarchy Theorem for Boolean Circuits. *Journal of the ACM*, 64(5), 2017. doi:10.1145/3095799. [p. 3]

[IW97] Russell Impagliazzo and Avi Wigderson. $\mathsf{P} = \mathsf{BPP}$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 220–229, 1997. doi:10.1145/258533.258590. [p. 3]

[Kum23] Vinayak M. Kumar. Tight Correlation Bounds for Circuits Between AC^0 and TC^0. In *38th Computational Complexity Conference*, volume 264, pages 18:1–18:40, 2023. arXiv:2304.02770, doi:10.4230/LIPIcs.CCC.2023.18. [pp. 3, 4, 5, 7, 9, 12, 13, 27, 28, 31, 32, 33]

[LG19] François Le Gall. Average-Case Quantum Advantage with Shallow Circuits. In *34th Computational Complexity Conference*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:20, 2019. arXiv:1810.12792, doi:10.4230/LIPICS.CCC.2019.21. [p. 10]

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant Depth Circuits, Fourier Transform, and Learnability. *Journal of the ACM*, 40(3):607–620, 1993. doi:10.1145/174130.174138. [p. 8]

[LY22] Jiatu Li and Tianqi Yang. $3.1n - o(n)$ Circuit Lower Bounds for Explicit Functions. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1180–1193, 2022. eccc:TR21-023. doi:10.1145/3519935.3519976. [p. 3]

[MP43] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5:115–133, 1943. doi:10.1007/BF02478259. [p. 5]

[MSS91] Wolfgang Maass, Georg Schnitger, and Eduardo D. Sontag. On the Computational Power of Sigmoid versus Boolean Threshold Circuits. In *Proceedings of the Thirty Second Annual Symposium of Foundations of Computer Science*, pages 767–776, 1991. doi:10.1109/SFCS.1991.185447. [p. 5]

[Mur71] Saburo Muroga. Threshold Logic and Its Applications. *John Wiley & Sons, Inc.*, 1971. [p. 5]

[NC02] Michael A. Nielsen and Isaac Chuang. Quantum Computation and Quantum Information, 2002. doi:10.1017/CBO9780511976667. [p. 13]

[NW94] Noam Nisan and Avi Wigderson. Hardness vs. Randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1. [p. 3]

[Pau75] Wolfgang J Paul. A $2.5n$-lower bound on the combinational complexity of Boolean functions. In *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, pages 27–36, 1975. doi:10.1145/800116.803750. [p. 3]

[Rao07] Anup Rao. An Exposition of Bourgain's 2-Source Extractor. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2007. eccc:TR07-034. [p. 37]

[Raz87] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41(4):598–607, 1987. doi:10.1007/BF01137685. [pp. 3, 5, 6, 14, 20]

[Reg24] Oded Regev. An Efficient Quantum Factoring Algorithm, 2024. arXiv:2308.06572. [p. 10]

[Ros17] Benjamin Rossman. An entropy proof of the switching lemma and tight bounds on the decision-tree size of AC^0 , 2017. URL: https://users.cs.duke.edu/~br148/logsize.pdf. [pp. 12, 27, 31, 32]

[RR97] Alexander A. Razborov and Steven Rudich. Natural Proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997. doi:10.1006/jcss.1997.1494. [pp. 3, 4, 9, 25, 26]

[RT22] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. *Journal of the ACM*, 69(4):1–21, 2022. eccc:TR18-107. doi:10.1145/3313276.3316315. [pp. 3, 10, 11, 26, 27, 28, 29]

[Sch74] Claus-Peter Schnorr. Zwei lineare untere Schranken fur die komplexitat Boolescher funktionen. *Computing*, 13:155–171, 1974. doi:10.1007/BF02246615. [p. 3]

[Sch80] Claus-Peter Schnorr. A $3n$-lower bound on the network complexity of Boolean functions. *Theoretical Computer Science*, 10(1):83–92, 1980. doi:10.1016/0304-3975(80)90074-2. [p. 3]

[Sho97] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. arXiv:quant-ph/9508027, doi:10.1137/S0097539795293172. [p. 10]

[Sim97] Daniel R. Simon. On the Power of Quantum Computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. doi:10.1137/S0097539796298637. [p. 10]

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 77–82, 1987. doi:10.1145/28395.28404. [pp. 3, 5, 6, 14, 20]

[Sto76] Larry J. Stockmeyer. On the combinational complexity of certain symmetric Boolean functions. *Mathematical Systems Theory*, 10(1):323–336, 1976. doi:10.1007/BF01683282. [p. 3]

[STV21] Srikanth Srinivasan, Utkarsh Tripathi, and S. Venkitesh. On the Probabilistic Degrees of Symmetric Boolean Functions. *SIAM Journal on Discrete Mathematics*, 35(3):2070–2092, 2021. doi:10.1137/19M1294162. [p. 15]

[Tor91] Jacobo Torán. Complexity Classes Defined by Counting Quantifiers. *Journal of the ACM*, 38(3):752–773, 1991. doi:10.1145/116825.116858. [p. 29]

[VV85] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, page 458–463. Association for Computing Machinery, 1985. doi:10.1145/22145.22196. [p. 17]

[Wag86] Klaus W. Wagner. The Complexity of Combinatorial Problems with Succinct Input Representation. *Acta Informatica*, 23:325–356, 1986. doi:10.1007/BF00289117. [p. 29]

[Wil14] Ryan Williams. Nonuniform ACC Circuit Lower Bounds. *Journal of the ACM*, 61(1), 2014. doi:10.1145/2559903. [pp. 3, 5, 7, 14, 17, 20, 22, 23, 24, 25]

[WKST19] Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 515–526, 2019. arXiv:1906.08890, doi:10.1145/3313276.3316404. [pp. 3, 10, 11, 12, 26, 27, 30, 31, 34, 35, 36, 37, 38, 39, 41]

[WP24] Adam Bene Watts and Natalie Parham. Unconditional Quantum Advantage for Sampling with Shallow Circuits, 2024. arXiv:2301.00995. [p. 10]

[Wu22] Xinyu Wu. A Stochastic Calculus Approach to the Oracle Separation of BQP and PH. *Theory of Computing*, 18(17):1–11, 2022. arXiv:2007.02431, doi:10.4086/toc.2022.v018a017. [pp. 27, 28]

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles. In *26th Annual Symposium on Foundations of Computer Science*, pages 1–10, 1985. doi:10.1109/SFCS.1985.49. [p. 3]

[Yao89] A. C. Yao. Circuits and local computation. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, page 186–196. Association for Computing Machinery, 1989. doi:10.1145/73007.73025. [pp. 3, 4]

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il