Strong Average-Case Lower Bounds from Non-trivial Derandomization*

Lijie Chen†
lijieche@mit.edu
CSAIL, MIT
Cambridge, MA, USA

Hanlin Ren
rhl16@mails.tsinghua.edu.cn
IIIS, Tsinghua University
Beijing, China

ABSTRACT

We prove that for all constants a, $\mathsf{NQP} = \mathsf{NTIME}[n^{\mathrm{polylog}(n)}]$ cannot be $(1/2+2^{-\log^a n})$-approximated by $2^{\log^a n}$-size $\mathsf{ACC}^0 \circ \mathsf{THR}$ circuits (ACC^0 circuits with a bottom layer of THR gates). Previously, it was even open whether E^{NP} can be $(1/2 + 1/\sqrt{n})$-approximated by $\mathsf{AC}^0[\oplus]$ circuits. As a straightforward application, we obtain an infinitely often $(\mathsf{NE} \cap \mathsf{coNE})_{/1}$-computable pseudorandom generator for poly-size ACC^0 circuits with seed length $2^{\log^{\varepsilon} n}$, for all $\varepsilon > 0$.

More generally, we establish a connection showing that, for a typical circuit class $\mathscr{C}$, non-trivial nondeterministic algorithms estimating the acceptance probability of a given S-size $\mathscr{C}$ circuit with an additive error $1/S$ (we call it a CAPP algorithm) imply strong $(1/2 + 1/n^{\omega(1)})$ average-case lower bounds for nondeterministic time classes against $\mathscr{C}$ circuits. Note that the existence of such (deterministic) algorithms is much weaker than the widely believed conjecture PromiseBPP = PromiseP.

We also apply our results to several sub-classes of TC^0 circuits. First, we show that for all k, NP cannot be $(1/2 + n^{-k})$-approximated by n^k-size $\mathsf{Sum} \circ \mathsf{THR}$ circuits (exact $\mathbb{R}$-linear combination of threshold gates), improving the corresponding worst-case result in [Williams, CCC 2018]. Second, we establish strong average-case lower bounds and build $(\mathsf{NE} \cap \mathsf{coNE})_{/1}$-computable PRGs for $\mathsf{Sum} \circ \mathsf{PTF}$ circuits, for various regimes of degrees. Third, we show that non-trivial CAPP algorithms for $\mathsf{MAJ} \circ \mathsf{MAJ}$ indeed already imply worst-case lower bounds for TC^0_3 ($\mathsf{MAJ} \circ \mathsf{MAJ} \circ \mathsf{MAJ}$). Since exponential lower bounds for $\mathsf{MAJ} \circ \mathsf{MAJ}$ are already known, this suggests TC^0_3 lower bounds are probably within reach.

Our new results build on a line of recent works, including [Murray and Williams, STOC 2018], [Chen and Williams, CCC 2019], and [Chen, FOCS 2019]. In particular, it strengthens the corresponding $(1/2 + 1/\mathrm{polylog}(n))$-inapproximability average-case lower bounds in [Chen, FOCS 2019].

The two important technical ingredients are techniques from Cryptography in NC^0 [Applebaum et al., SICOMP 2006], and Probabilistic Checkable Proofs of Proximity with NC^1-computable proofs.

CCS CONCEPTS

• **Theory of computation → Circuit complexity**; *Pseudorandomness and derandomization.*

KEYWORDS

circuit complexity, average-case complexity, derandomization

ACM Reference Format:

Lijie Chen and Hanlin Ren. 2020. Strong Average-Case Lower Bounds from Non-trivial Derandomization. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC '20), June 22–26, 2020, Chicago, IL, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3357713.3384279

1 INTRODUCTION

1.1 Background and Motivation

A holy grail of theoretical computer science is to prove *unconditional* circuit lower bounds for explicit functions (such as $\mathsf{NP} \not\subset \mathsf{P}_{/\mathrm{poly}}$). To approach this notoriously hard central open problem, the first step is to understand the power of various *constant depth* circuit classes. Back in the 1980s, there was a lot of significant progress in proving lower bounds for constant depth circuits. A line of works [2, 22, 28, 53] established exponential lower bounds for AC^0 (constant depth circuits consisting of AND/OR gates of unbounded fan-in), and [36, 41] proved exponential lower bounds for $\mathsf{AC}^0[p]$ (AC^0 circuits extended with MOD_p gates) when p is a prime.

However, the progress had stopped there—the power of $\mathsf{AC}^0[m]$ for a composite m had been elusive, despite that it had been conjectured that they cannot even compute the majority function. In fact, it had been a notorious long-standing open question in computational complexity whether NEXP (nondeterministic exponential time) has polynomial-size ACC^0 circuits[1], until a seminal work by Williams [49] a few years ago, which proved NEXP does not have polynomial-size ACC^0 circuits, via a new *algorithmic* approach to circuit lower bounds [47].

Not only being an exciting new development after a long gap, the new circuit lower bound is also remarkable as it surpasses all previous known barriers for proving circuit lower bounds: relativization [11], algebrization [1], and natural proofs [37][2]. Moreover, the

*A full version of this paper is available online at ECCC [18].

†Supported by NSF CCF-1741615 and a Google Faculty Research Award.

STOC '20, June 22–26, 2020, Chicago, IL, USA

ACM ISBN 978-1-4503-6979-4/20/06…$15.00
https://doi.org/10.1145/3357713.3384279

[1]It had been stressed several times as one of the most *embarrassing* open questions in complexity theory, see [6]. ACC^0 denotes the union of $\mathsf{AC}^0[m]$ for all constant m.

[2]We remark that there is no consensus on whether the natural proof barrier applies to ACC^0: *i.e.*, there is no widely accepted construction of PRFs in ACC^0. A candidate

underlying method (the algorithmic method) puts many important classical complexity gems together, ranging from nondeterministic time hierarchy theorem [38, 54], IP = PSPACE [32, 40], hardness vs randomness [35], to PCP Theorem [7, 8].

Recent development of the algorithmic approach to circuit lower bounds. Recently, Murray and Williams [34] significantly advanced the algorithmic approach by proving that strong enough circuit-analysis (Gap-UNSAT)[3] algorithms can also imply circuit lower bounds for NQP (nondeterministic quasi-polynomial time) or NP, instead of the previous gigantic class NEXP. Building on the new connection and the corresponding algorithms for $\mathsf{ACC}^0 \circ \mathsf{THR}$ [48], they showed that $\mathsf{NQP} \not\subset \mathsf{ACC}^0 \circ \mathsf{THR}$.

Building on [34], [17] recently generalized the connection to the *average-case*, by showing that strong enough circuit-analysis algorithms also imply $(1/2 + o(1))$-inapproximability average-case lower bounds for NQP or NP. In particular, it was shown that NQP cannot be $(1/2 + 1/\mathrm{polylog}(n))$-approximated by $\mathsf{ACC}^0 \circ \mathsf{THR}$. This is very interesting for two reasons: first, average-case lower bounds tend to have other applications such as constructing unconditional PRGs; second, the proof techniques do not apply the easy-witness lemma of [34, 49], and follows a more direct approach.

Still, the $(1/2 + 1/\mathrm{polylog}(n))$-inapproximability result is not enough to get us a non-trivial (say, with $n^{o(1)}$ seed length) PRG construction for ACC^0, which requires at least a $(1/2 + 1/n^{\omega(1)})$-inapproximability bound.

The $1/2 + 1/\sqrt{n}$ Razborov-Smolensky barrier. Indeed, proving a non-trivial $(1/2 + n^{-\omega(1)})$-inapproximability result is even open for $\mathsf{AC}^0[\oplus]$ circuits (AC^0 circuits extended with parity gates). Using the renowned polynomial approximation method, [36, 41, 42] showed that the majority function cannot be $(1/2 + n^{1/2-\varepsilon})$-approximated by $\mathsf{AC}^0[\oplus]$. However, it is even open that whether E^{NP} can be $(1/2 + 1/\sqrt{n})$-approximated by $(\log n)$-degree $\mathbb{F}_2$-polynomials. Improving the $(1/2 + 1/\sqrt{n})$-bound (and constructing the corresponding PRGs) is recognized as a significant open question in circuit complexity [16, 21, 43, 44].

1.2 Our Results

In this paper, we significantly improve the circuit-analysis-algorithms-to-average-case-lower-bounds connection in [17]. We first define the circuit-analysis task of our interest.

- CAPP[4] **for $\mathscr{C}$ circuits with inverse-circuit-size error**: Given a $\mathscr{C}$ circuit C of size S on n input bits, estimate
$$\Pr_{x \in \{0,1\}^n}[C(x) = 1]$$
within an additive error $1/S$.

For simplicity, throughout this paper, we will just refer to the above problem as CAPP. We remark that under the widely believed assumption PromiseBPP = PromiseP, this problem has a $\mathrm{poly}(S)$ time algorithm even for $\mathscr{C} = \mathsf{P}_{/\mathrm{poly}}$. In the following, we show that indeed a non-trivial improvement on the brute-force $2^n \cdot \mathrm{poly}(S)$-time algorithm already implies strong average-case lower bounds for $\mathscr{C}$.

From Non-trivial CAPP Algorithms to Strong Average-Case Circuit Lower Bounds.

Theorem 1.1. *Let $\mathscr{C}$ be a typical circuit class*[5] *such that $\mathscr{C}$ circuits of size S can be implemented by (general) circuits of depth $O(\log S)$. The following hold.*

(NP *Average-Case Lower Bound) Suppose there is a constant $\varepsilon > 0$ such that the* CAPP *problem of $\mathsf{AND}_4 \circ \mathscr{C}$ circuits of size $2^{\varepsilon n}$ can be solved in $2^{n-\varepsilon n}$ time. Then for every constant $k \geq 1$,* NP *cannot be $(1/2 + n^{-k})$-approximated by $\mathscr{C}$ circuits of n^k size.*

(NQP *Average-Case Lower Bound) Suppose there is a constant $\varepsilon > 0$ such that the* CAPP *problem of $\mathsf{AND}_4 \circ \mathscr{C}$ circuits of size 2^{n^ε} can be solved in 2^{n-n^ε} time. Then for every constant $k \geq 1$,* NQP *cannot be $(1/2 + 2^{-\log^k n})$-approximated by $\mathscr{C}$ circuits of $2^{\log^k n}$ size.*

(NEXP *Average-Case Lower Bound) Suppose the* CAPP *problem of $\mathsf{AND}_4 \circ \mathscr{C}$ circuits of size $\mathrm{poly}(n)$ can be solved in $2^n/n^{\omega(1)}$ time. Then* NE *cannot be $(1/2 + 1/\mathrm{poly}(n))$-approximated by $\mathscr{C}$ circuits of $\mathrm{poly}(n)$ size.*

By the standard Discriminator Lemma [27], we immediately obtain worst-case lower bounds for $\mathsf{MAJ} \circ \mathscr{C}$ circuits as well.

Corollary 1.2. *Under the algorithmic assumptions of Theorem 1.1, we obtain worst-case lower bounds for $\mathsf{MAJ} \circ \mathscr{C}$ circuits in the corresponding cases: (1)* NP *not in n^k-size $\mathsf{MAJ} \circ \mathscr{C}$ for all k; (2)* NQP *not in $2^{\log^k n}$-size $\mathsf{MAJ} \circ \mathscr{C}$ for all k; (3)* NE *not in $\mathrm{poly}(n)$-size $\mathsf{MAJ} \circ \mathscr{C}$.*

Remark 1.3. We remark that the conclusions of Theorem 1.1 still hold if the corresponding CAPP algorithms are *non-deterministic*. That is, on any computational branch, it either outputs a correct estimation[6] or rejects, and it does not reject all branches.

Remark 1.4. Theorem 1.1 assumes $\mathscr{C}$ is a sub-class of NC^1 (*e.g.*, $\mathsf{THR} \circ \mathsf{THR}$, TC^0, or ACC^0). On the other hand, if $\mathscr{C}$ is stronger than NC^1 (*e.g.*, NC^2, $\mathsf{P}_{/\mathrm{poly}}$), [17, Theorem 1.3] already showed that[7] even CAPP with constant error suffices to prove the stated average-case lower bounds in Theorem 1.1. Although we still left open the possible case that $\mathscr{C}$ is uncomparable to NC^1, our theorem together with [17] cover nearly all interesting circuit classes.

Comparison with [17]. Our Theorem 1.1 improves on the corresponding connection in [17] in two ways: (1) we get a much better inapproximability bound, which is crucial for our construction of nondeterministic PRGs; (2) we only need CAPP algorithms for $\mathsf{AND}_4 \circ \mathscr{C}$, while [17] requires algorithms for $\mathsf{AC}^0 \circ \mathscr{C}$. On the other hand, our requirement on the CAPP algorithms is stronger (additive error $1/S$) than that of [17] (constant additive error).

construction [15] is proposed recently, which still needs to be tested. But we can say that *if* there is a natural proof barrier for ACC^0, then this lower bound has surpassed it. (We also remark that there is a recent proposal on getting ACC^0 circuit lower bounds via torus polynomials [14].)

[3] The Gap-UNSAT problem asks one to distinguish between an unsatisfiable formula and a formula accepting a random input with probability $> 1/2$.

[4] The acronym CAPP denotes the Circuit Acceptance Probability Problem.

[5] A circuit class $\mathscr{C}$ is *typical* if it is closed under both negation and projection.

[6] It is allowed that on different branches it outputs different estimations as long as they are all within an additive error of $1/S$.

[7] [17, Theorem 1.3] only states the result with inapproximability $1/2 + n^{-c}$ for a constant c, but it is easy to see that its proof can be generalized to the inapproximability corresponding to Theorem 1.1.

More on our definition on CAPP. We remark that our definition of CAPP is a bit non-standard, comparing to the usual definition with a constant error. Nonetheless, such a CAPP algorithm is *much weaker* than a full-power #SAT algorithm, and (as discussed before) is widely believed to exist even for $\mathsf{P}_{/\mathrm{poly}}$ circuits.

Strong Average-Case Lower Bounds for $\mathsf{ACC}^0 \circ \mathsf{THR}$. Applying the non-trivial #SAT algorithms for $\mathsf{ACC}^0 \circ \mathsf{THR}$ circuits in [48], it follows that NQP cannot be even weakly approximated by $\mathsf{ACC}^0 \circ \mathsf{THR}$ circuits, and it is (worst-case) hard for $\mathsf{MAJ} \circ \mathsf{ACC}^0 \circ \mathsf{THR}$ circuits.

Theorem 1.5. *For every constant $k \geq 1$, NQP cannot be $(1/2 + 2^{-\log^k n})$-approximated by $\mathsf{ACC}^0 \circ \mathsf{THR}$ circuits of size $2^{\log^k n}$. Consequently, NQP cannot be computed by $\mathsf{MAJ} \circ \mathsf{ACC}^0 \circ \mathsf{THR}$ circuits of size $2^{\log^k n}$ (in the worst-case), for all $k \geq 1$.*

The same holds for $(\mathsf{N}\cap\mathsf{coN})\mathsf{QP}_{/1}$ in place of NQP.

Nondeterministic PRGs for ACC^0 *with Sub-Polynomial Seed Length.* As an important application of the above strong average-case lower bound, we also obtain the first PRG with $n^{o(1)}$ seed length for ACC^0 circuits (previous, this was open even for $\mathsf{AC}^0[\oplus]$ circuits), albeit it is nondeterministic and infinitely often.

Theorem 1.6. *For every constant $\varepsilon > 0$, there is an infinitely often, $(\mathsf{NE} \cap \mathsf{coNE})_{/1}$-computable PRG fooling polynomial size ACC^0 circuits with seed length $2^{(\log n)^{\varepsilon}}$.*[8]

Remark 1.7. We can indeed optimize the seed length to be the *inverse of any sub-fourth-exponential function.* See [18, Section 7.2] for details.

Previously, the best PRG for ACC^0 is from [20], which is $(\mathsf{NE} \cap \mathsf{coNE})_{/1}$-computable and has seed length $n - n^{1-\beta}$ for any constant $\beta > 0$. Our construction significantly improves on that.

Lower Bounds and PRGs for $\mathsf{Sum} \circ \mathscr{C}$ *Circuits.* For a circuit class $\mathscr{C}$, a $\mathsf{Sum} \circ \mathscr{C}$ circuit is an $\mathbb{R}$-linear combination $C(x) := \sum_{i=1}^{t} \alpha_i C_i(x)$, such that each $\alpha_i \in \mathbb{R}$, each C_i is a $\mathscr{C}$ circuit on n input bits, and $C(x) \in \{0,1\}$ for all $x \in \{0,1\}^n$. We denote t as the *sparsity* of the circuit, and we define the size of C as the total size of all $\mathscr{C}$ sub-circuits C_i's.

We first show that if we have the corresponding non-trivial #SAT algorithms instead of the non-trivial CAPP algorithms, we would have average-case lower bounds for $\mathsf{Sum} \circ \mathscr{C}$ circuits. To avoid repetition, in the following we only state the version for NQP.

Corollary 1.8. *Let $\mathscr{C}$ be a typical circuit class such that $\mathscr{C}$ circuits of size S can be implemented by (general) circuits of depth $O(\log S)$. Suppose there is a constant $\varepsilon > 0$ such that the #SAT problem of $\mathsf{AND}_4 \circ \mathscr{C}$ circuits of size $2^{n^{\varepsilon}}$ can be solved in $2^{n-n^{\varepsilon}}$ time. Then for every constant $k \geq 1$, NQP cannot be $(1/2 + 2^{-\log^k n})$-approximated by $\mathsf{Sum} \circ \mathscr{C}$ circuits of $2^{\log^k n}$ size.*

This immediately implies a strong average-case lower bound for $\mathsf{Sum} \circ \mathsf{ACC}^0 \circ \mathsf{THR}$.

Corollary 1.9. *For every constant $k \geq 1$, NQP cannot be $(1/2 + 2^{-\log^k n})$-approximated by $\mathsf{Sum} \circ \mathsf{ACC}^0 \circ \mathsf{THR}$ circuits of size $2^{\log^k n}$. Consequently, NQP cannot be computed by $\mathsf{MAJ} \circ \mathsf{Sum} \circ \mathsf{ACC}^0 \circ \mathsf{THR}$ circuits of size $2^{\log^k n}$ (in the worst-case), for all $k \geq 1$.*

The same holds for $(\mathsf{N}\cap\mathsf{coN})\mathsf{QP}_{/1}$ in place of NQP.

Now we discuss some applications of our new techniques to some sub-classes of TC^0 circuits.

We begin with some notation. Recall that a degree-d PTF gate is a function defined by $\mathrm{sign}(p(x))$, where p is a degree-d polynomial on x over $\mathbb{R}$, and $\mathrm{sign}(z)$ outputs 1 if $z \geq 0$ and 0 otherwise. Clearly, a THR gate is simply a degree-1 PTF gate.

[51] proved that NP cannot be computed by n^k-size $\mathsf{Sum} \circ \mathsf{THR}$ circuits for all $k > 0$. With our improved connection, we apply the #SAT algorithm for $\mathsf{AND}_4 \circ \mathsf{THR}$ of [51] to improve it to a corresponding average-case lower bound.

Theorem 1.10. *For all constants k, NP cannot be $(1/2 + 1/n^k)$-approximated by n^k-size $\mathsf{Sum} \circ \mathsf{THR}$ circuits. Consequently, NP cannot be computed by n^k-size $\mathsf{MAJ} \circ \mathsf{Sum} \circ \mathsf{THR}$ circuits for all constants k.*[9]

We remark that $\mathsf{MAJ} \circ \mathsf{Sum} \circ \mathsf{THR}$ is a sub-class of $\mathsf{THR} \circ \mathsf{THR}$ with no previous known lower bounds. So Theorem 1.10 can be viewed as progress toward resolving the notorious open question of proving super-polynomial $\mathsf{THR} \circ \mathsf{THR}$ lower bounds.

Applying the non-trivial *zero-error* #SAT algorithm for PTF in [10], we also obtain NQP (NE) average-case lower bounds for $\mathsf{Sum} \circ \mathsf{PTF}_d$ circuits.

Theorem 1.11. *The following hold.*

- *For every constants $d, k \geq 1$, NQP cannot be $(1/2 + 2^{-\log^k n})$-approximated by $\mathsf{Sum} \circ \mathsf{PTF}_d$ circuits of sparsity $2^{\log^k n}$. Consequently, NQP does not have $2^{\log^k n}$-size $\mathsf{MAJ} \circ \mathsf{Sum} \circ \mathsf{PTF}_d$ circuits.*
- *Let $d(n) = 0.49 \frac{\log n}{\log \log n}$, then NE cannot be $(1/2 + 1/\mathrm{poly}(n))$-approximated by $\mathsf{Sum} \circ \mathsf{PTF}_{d(n)}$ circuits of sparsity $\mathrm{poly}(n)$. Consequently, $\mathsf{NE} \not\subset \mathsf{MAJ} \circ \mathsf{Sum} \circ \mathsf{PTF}_{d(n)}$.*

From the above theorem, we can also obtain non-trivial nondeterministic PRGs for $\mathsf{Sum} \circ \mathsf{PTF}$ circuits.

Theorem 1.12. *For every constants $d, k \geq 1$ and $\varepsilon > 0$, there is an $(\mathsf{NE} \cap \mathsf{coNE})_{/1}$-computable i.o. PRG with seed length $O(2^{\log^{\varepsilon} n})$ that $(1/n^k)$-fools $\mathsf{Sum} \circ \mathsf{PTF}_d$ circuits of sparsity n^k.*[10]

Previously, the best (constant-error) PRG for degree-d PTF has seed length $O(\log n \cdot 2^{O(d)})$ [33]. Our construction has a worse seed-length, is nondeterministic and infinitely often, but works for the larger class $\mathsf{Sum} \circ \mathsf{PTF}$.

Towards TC^0_3 *Lower Bounds.* In [19], it is shown that non-trivial CAPP algorithms for $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits with inverse-polynomial additive error would already imply $\mathsf{THR} \circ \mathsf{THR}$ circuit lower bounds. We significantly improve that connection by showing it would indeed imply TC^0_3 lower bounds!

[8] That is, this PRG G is computable by a nondeterministic machine M with one bit of advice such that for a seed $s \in \{0,1\}^{2^{(\log n)^{\varepsilon}}}$, $M(s)$ either outputs $G(s)$ or rejects on any computational branch, and it outputs $G(s)$ on some computational branches. See [18, Definition 2.7] for a formal definition.

[9] This average-case lower bound can also be extended to against $\mathsf{Sum} \circ \mathsf{ReLU}$ circuits, similar to the exact $\mathsf{Sum} \circ \mathsf{ReLU}$ lower bounds in [51].

[10] We did not attempt to optimize this seed length.

Theorem 1.13. *If there is a $2^n/n^{\omega(1)}$ time* CAPP *algorithm for* poly(n)*-size* MAJ ∘ MAJ *circuits. Then* NEXP ⊄ MAJ ∘ MAJ ∘ MAJ.

We remark that MAJ∘MAJ∘MAJ is actually equivalent to MAJ∘THR∘THR (since MAJ∘MAJ = MAJ∘THR [23]). Since exponential-size (worst-case) lower bounds against MAJ ∘ MAJ are already known. If only we can "mine" a non-trivial CAPP algorithm (which is widely believed to exist) for MAJ ∘ MAJ circuits from these lower bounds, we would have worst-case lower bounds against TC^0_3.

Concurrent Works. A concurrent work by Viola [45] proved that E^{NP} cannot be $(1/2+\log^{O(h)} s/n)$-approximated by $\mathsf{AC}^0[\oplus]$ circuits of size s and depth h. This result is incomparable with ours. We proved that E^{NP} cannot be $(1/2+\varepsilon)$-approximated by ACC^0 circuits of polynomial size for some $\varepsilon \ll 1/n$, while the inapproximability result in [45] only achieves $\varepsilon > 1/n$. On the other hand, our paper does not prove anything about *exponential* (*e.g.* $2^{n^{0.01}}$) sized $\mathsf{AC}^0[\oplus]$ circuits, while the results in [45] bypass the $(1/2 + 1/\sqrt{n})$ barrier.

1.3 Intuition

In the following, we sketch the intuitions for our new average-case lower bounds.

In this section, we will aim for a simpler version that NQP cannot be $(1/2 + n^{-k})$-approximated by ACC^0 for a large constant k (say, $k = 10^3$) for simplicity. We believe this version already captures all important technical ideas of our new average-case circuit lower bounds.

1.3.1 Review of [17] and the Bottleneck. First, since our work crucially builds on [17] (which proved NQP cannot be $(1/2+1/\mathrm{polylog}(n))$-approximated by ACC^0), it would be very instructive to review the proof structure of [17], and understand what is the bottleneck of extending [17] to prove a $(1/2 + n^{-k})$-inapproximability bound.

A high-level overview of [17]: three steps. Suppose we are proving NQP cannot be $(1-\delta)$-approximated by ACC^0 for now, where δ is a small constant. On a very high level, the proof of [17] involves the following three steps.[11]

- Step I (Conditional collapse from NC^1 to ACC^0.) Assuming NQP can be $(1-\delta)$-approximated by ACC^0, [17] shows that NC^1 collapses to ACC^0, using the existence of self-reducible NC^1-complete languages [9, 12, 31].
- Step II (An NE algorithm certifying low depth hardness.) Next, making use of the non-trivial SAT algorithm for ACC^0 circuits [49], [17] shows that there is an NE algorithm $V(\cdot,\cdot)$ certifying n^{ε}-depth hardness. Formally, $V(x, y)$ takes inputs such that $|y| = 2^{|x|}$; for infinitely many n's, $V(1^n,\cdot)$ is satisfiable, and $V(1^n, y) = 1$ implies y, interpreted as a function $f_y : \{0,1\}^n \to \{0,1\}$, does not have n^{ε}-depth circuits.
- Step III (Certifying low depth hardness implies average-case lower bounds for low depth circuits.) Finally, [17] shows that the above algorithm V would be sufficient to imply that NQP cannot be $(1-\delta)$-approximated by NC^1 (and also ACC^0).

The bottleneck of the argument: Step I.. Suppose we are going to prove NQP cannot be $(1/2 + n^{-k})$-approximated by ACC^0, let us examine which one of the above three steps would break.

Clearly, Step II is unaffected (assuming Step I works). Another observation is that since NC^1 can compute majority[12], we can use the XOR Lemma [24, 29, 52] to show that NQP cannot be $(1/2+n^{-k})$-approximated by NC^1 circuits.[13] Therefore, Step III still works if we want to prove the stronger $(1/2+n^{-k})$-inapproximability result.

However, Step I completely breaks. Assuming NQP can be $(1/2+n^{-k})$-approximated by ACC^0 circuits, it seems hopeless to show that NC^1 collapse to ACC^0 using some random self-reducible languages. This is because the given circuit only $(1/2 + n^{-k})$-approximates the given random self-reducible language, and to the best of our knowledge, all known corrector for such languages in this error regime requires computing at least some variants of the majority function, while ACC^0 is conjectured not to be able to compute majority [41]!

1.3.2 A Detour: Chen and Williams [19] and $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ Circuit Lower Bounds. So it seems unlikely that we can show a collapse theorem from NC^1 to ACC^0 under the assumption that NQP can be $(1/2 + n^{-k})$-approximated by ACC^0. A natural idea to avoid this obstacle is to show NC^1 collapses to some other larger classes under the same assumption. Examining the proof idea of [17], it seems at least we can show NC^1 collapses to MAJ ∘ ACC^0 under the assumption. However, the issue is that then we don't know how to implement Step II, as we don't have a non-trivial SAT (or even Gap-UNSAT) algorithm for MAJ ∘ ACC^0 circuits.

So we indeed want a *collapse theorem which would collapse* NC^1 *to a circuit class* $\mathscr{C}$ *for which we at least know some non-trivial algorithms for, and of course* $\mathscr{C}$ *also has to contain* ACC^0. Perhaps the best choice for us is the $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuits which has recently been studied by [19]. So let us take a detour into this circuit class and the corresponding lower bounds in [19].

$\widetilde{\mathsf{Sum}}_\delta \circ \mathscr{C}$ Circuits. Let $\mathscr{C}$ be a class of functions from $\{0,1\}^n \to \{0,1\}$ and $\delta \in [0, 0.5)$. We say $f : \{0,1\}^n \to \{0,1\}$ admits a $\widetilde{\mathsf{Sum}}_\delta \circ \mathscr{C}$ circuit of sparsity S, if there are S functions $C_1, C_2, \ldots, C_S$ from $\mathscr{C}$, together with S coefficients $\alpha_1, \alpha_2, \ldots, \alpha_S$ in $\mathbb{R}$, such that for all $x \in \{0,1\}^n$,

$$\left|\sum_{i=1}^{S} \alpha_i \cdot C_i(x) - f(x)\right| \le \delta.$$

Given a valid $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuit C, we say $C(x) = 1$ if the corresponding output value $|\sum_i \alpha_i C_i(x) - 1| \le \delta$, and $C(x) = 0$ otherwise. [19] gives a $2^{n-n^{\varepsilon}}$-time Gap-UNSAT (in fact, constant-error CAPP) algorithm for $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ of $2^{n^{\varepsilon}}$-size when δ is small (the algorithm is indeed already implicit in [51]). Building on this algorithm (and more importantly, PCP of proximity), [19] proves that NQP ⊄ $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ for any constant $\delta \in [0, 1/2)$.

[11] Actually, in [17], Step III is much more complicated than the previous two steps, and Step II just follows from [50]. In the presentation of [17], Step III is decomposed into several sub-steps [17, Section 6.2, 7-9]. We choose to give the overview in this way because we essentially make use of Step III as a black box, and our improvement is mostly focusing on the first two steps. In particular, our improved Step II is much more involved than that of [17], and crucially builds on [19].

[12] It is proved that black-box hardness amplification requires majority [26, 39].

[13] Precisely speaking, we have to start with our $(\mathsf{N}\cap\mathsf{coN})\mathsf{QP}_{/1}$ lower bounds for that purpose.

1.3.3 *Key Technical Ingredient: A* ⊕L*-complete Language* CMD *with a* $\widetilde{\mathsf{Sum}}_\delta$ *Error Corrector.*

So given the result of [19], the question becomes:

> **A New Collapse Theorem?**: Can we show a collapse from NC^1 to $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuits, assuming NQP can be $(1/2 + n^{-k})$-approximated by ACC^0 circuits?

Our improvement of Step I answers the question affirmatively, by making use of a ⊕L-complete[14] language CMD [5, 25, 30] with very nice reducibility properties. We remark that the underlying techniques play a crucial part in the famous construction of NC^0-computable one-way functions (and low-stretch PRGs) [5] (see also the book [4]).

(1) (⊕L-completeness under projections.) That is, for every language $L \in \oplus\mathsf{L}$, there is a polynomial-time computable projection P such that $L(x) = \mathsf{CMD}(P(x))$.
(2) (Single-query error correctability with a randomized image DCMD.) For technical reasons, we also have to introduce another ⊕L-complete language DCMD, which is a "randomized image" of CMD under projections (when randomness is fixed) [25, Claim 2.19].
That is, given $n \in \mathbb{N}$, there is $m = \mathrm{poly}(n)$ and a randomized reduction $P(x, r)$ (r is the random bits) from CMD on input length n to DCMD on input length m, such that:
  (a) For all $x \in \{0, 1\}^n$, $P(x, \mathcal{U}_\ell)$ distributes uniformly on $\{0, 1\}^m$, where ℓ is the number of random bits involved, and $\mathcal{U}_\ell$ is the uniform distribution over $\{0, 1\}^\ell$.
  (b) For all fixed random bits r, $P(x, r)$ is a projection of x.
  (c) For all $x \in \{0, 1\}^n$, $\mathsf{CMD}_n(x) = \mathsf{DCMD}_m(P(x, r)) \oplus r_0$ for all r, where r_0 is the first bit of r.

An error corrector in $\widetilde{\mathsf{Sum}}_\delta \circ f$. The second property of CMD stated above is *amazing*. It enables us to do the desired error correction with $\widetilde{\mathsf{Sum}}_\delta \circ f$ circuits (a linear sum of f functions composed with projections). See [18, Section 3] for the details. It then follows that if NQP can be $(1/2 + n^{-k})$-approximated by ACC^0 circuits, we would have the desired collapse from NC^1 to $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$.

1.3.4 *A Simpler Proof for a Worst-Case Lower Bound Against* $\mathsf{MAJ} \circ \mathsf{ACC}^0$.

With the improved collapse result, we can already prove worst-case lower bounds against $\mathsf{MAJ} \circ \mathsf{ACC}^0$. For simplicity, here we only show the following weaker version.

Theorem 1.14 (Toy Example). $\mathsf{NQP} \not\subset \mathsf{MAJ} \circ \mathsf{ACC}^0$.

Proof Sketch. There are two cases.

- First, we assume DCMD (which is in NQP) cannot be $(1/2 + 1/\mathrm{poly}(n))$-approximated by ACC^0. This implies that $\mathsf{NQP} \not\subset \mathsf{MAJ} \circ \mathsf{ACC}^0$, via the standard Discriminator Lemma [27].
- Second, suppose DCMD can be $(1/2 + 1/n^k)$-approximated by n^k-size ACC^0 circuits for a constant k. This implies that NC^1 collapses to $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$.
By [19], $\mathsf{NQP} \not\subset \widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$. This in turn implies that $\mathsf{NQP} \not\subset \mathsf{NC}^1$, and clearly also $\mathsf{NQP} \not\subset \mathsf{MAJ} \circ \mathsf{ACC}^0$. □

[14] Roughly speaking, ⊕L consists of languages L such that there is an $O(\log n)$ space nondeterministic Turing machine M, such that on every input x, $x \in L$ if and only if there is an odd number of computational paths making M accept on input x.

1.3.5 *Toward Average-Case Hardness: The Updated Three Steps Plan.*

Now we switch to the new average-case circuit lower bounds. With the new conditional collapse theorem, the following are our updated three steps plan for the new average-case lower bounds.

Step I' (Conditional collapse from NC^1 to $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$.)
Assuming NQP can be $(1/2 + n^{-k})$-approximated by ACC^0, we show that NC^1 collapses to $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$, utilizing the nice properties of the problems CMD and DCMD.

Step II' (An NE algorithm certifying low depth hardness.)
Next, making use of the non-trivial constant error CAPP algorithm for $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuits [19, 51], we show that there is an NE algorithm $V(\cdot, \cdot)$ certifying n^ε-depth hardness.

Step III' (Certifying low depth hardness implies average-case lower bounds for low depth circuits.)
Finally, we show that the above algorithm V would be sufficient to imply that NQP cannot be $(1/2 + n^{-k})$-approximated by NC^1 (and also ACC^0).

As previously discussed, Step III' can be achieved easily by combing [17] and the XOR Lemma [24, 29, 52]. It remains to implement Step II', which is the most technical part of this work.

1.3.6 *Review of Step II: Certifying Hardness via* PCP *and Nondeterministic Time Hierarchy.*

To implement Step II', the natural idea is to directly modify Step II ([17, Section 6.1]), and follow [50]. Now we briefly review the details of Step II and explain why it seems hard to adapt it directly.

Setting up the verifier V_{cert}. Let L be a unary language in $\mathsf{NTIME}[2^n] \setminus \mathsf{NTIME}[2^n/n]$ [54]. Fix an efficient PCP verifier V_{PCP} for L (such as [13]). That is, for a function $\ell := \ell(n) = n + O(\log n)$, $V_{\mathsf{PCP}}(1^n)$ takes ℓ random bits as input, runs in $\mathrm{poly}(n)$ time, is given access to an oracle $O : \{0, 1\}^\ell \to \{0, 1\}$, and satisfies the following conditions:

(1) (Completeness) if $1^n \in L$, then there exists an oracle O such that $V_{\mathsf{PCP}}(1^n)^O$ always accepts;
(2) (Soundness) if $1^n \notin L$, then for all oracles O, the probability $V_{\mathsf{PCP}}(1^n)^O$ accepts is $\le 1/n$.

Now, we define V_{cert} as follows: $V_{\mathrm{cert}}(1^n, y)$ treats y as the truth-table of an oracle $O_y : \{0, 1\}^\ell \to \{0, 1\}$, and verifies whether $V_{\mathsf{PCP}}(1^n)^{O_y}$ always accepts[15]. Clearly, V_{cert} runs in $\mathrm{poly}(n + |y|)$ time.

Since any depth-d circuit is equivalent to some $2^{O(d)}$-size $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuit (recall that now NC^1 collapses to $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$), to show that V_{cert} certifies n^{ε_1}-depth hardness, it suffices to show that V_{cert} certifies hardness for 2^{n^ε}-size $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuits for $\varepsilon > \varepsilon_1$.

Let us suppose the opposite that V_{cert} does not certify hardness for 2^{n^ε}-size $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuits. In particular, this means for all large enough n, if $V_{\mathrm{cert}}(1^n, \cdot)$ is satisfiable, then there is a 2^{n^ε}-size $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuit C such that $V_{\mathrm{cert}}(1^n, tt(C)) = 1$, where $tt(C)$ is the truth-table of C. Translating it to the setting of PCP, for large enough n, the following hold:

(1) (Succinct Completeness) if $1^n \in L$, then there exists a 2^{n^ε}-size $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuit $C : \{0, 1\}^\ell \to \{0, 1\}$ such that $V_{\mathsf{PCP}}(1^n)^C$ always accepts;

[15] Strictly speaking, here $|y| = 2^\ell = 2^n \cdot \mathrm{poly}(n)$ which is slightly larger than 2^n, but this slight difference does not really matter in the proof.

(2) (Soundness) if $1^n \notin L$, then for all oracles O, the probability $V_{\mathsf{PCP}}(1^n)^O$ accepts is $\le 1/n$.

The issue with the direct approach. Given the above two conditions, the natural idea for putting L in $\mathsf{NTIME}[2^n/n]$ to obtain a contradiction would be to try the following nondeterministic algorithm for L: Given an input 1^n, we (non-deterministically) guess a $2^{n^{\varepsilon}}$-size $\widetilde{\mathsf{Sum}}_{\delta} \circ \mathsf{ACC}^0$ circuit C[16], and try to estimate

$$p_{\mathrm{acc}}(V_{\mathsf{PCP}}(1^n)^C) = \Pr_{r \in \{0,1\}^{\ell}}[V_{\mathsf{PCP}}(1^n)^C(r)].$$

Let $D_C := V_{\mathsf{PCP}}(1^n)^C$. We would like to accept when $p_{\mathrm{acc}}(D_C) = 1$, and reject when $p_{\mathrm{acc}}(D_C) < 1/n$, so a constant additive error (say, 1/10) approximation to $p_{\mathrm{acc}}(D_C)$ would suffice.

The issue here is that, D_C is *not a* $\widetilde{\mathsf{Sum}}_{\delta} \circ \mathsf{ACC}^0$ *circuit anymore.* So we don't know how to estimate $p_{\mathrm{acc}}(D_C)$ using the constant error CAPP algorithm for $\widetilde{\mathsf{Sum}}_{\delta} \circ \mathsf{ACC}^0$ in [19, 51].

We remark that by [13], V_{PCP} can indeed be implemented by a 3-CNF, hence if C is only an ACC^0 circuit, $V_{\mathsf{PCP}}(1^n)^C$ is still an ACC^0 circuit. This is why this argument works in the original Step II, where we have a collapse from NC^1 to ACC^0 instead of $\widetilde{\mathsf{Sum}}_{\delta} \circ \mathsf{ACC}^0$.

1.3.7 Getting Around of the Issue with PCP *of Proximity.* To avoid the aforementioned issue, we would like to adopt the PCP of Proximity framework introduced in [19], which also plays a crucial part in the P^{NP} construction of rigid matrices in [3]. For more intuition on this framework and how it compares to and improves on the earlier works [47, 49], one is referred to [19, Section 1.6].

For a SAT instance F, Y a subset of its variables, and $y \in \{0,1\}^{|Y|}$, we use $F_{Y=y}$ to denote the resulting instance obtained by assigning the Y variables in F to y.[17] We also use $\mathsf{OPT}(F)$ to denote the maximum fraction of clauses that can be satisfied by any assignment.

The following transformation is the key technical part of [19].[18]

Theorem 1.15 (Implicit in [19]). *Let* Enc *be the encoder of some constant-rate error correcting code. There is a polynomial-time transformation that, given a circuit D on n inputs of size $m \ge n$, outputs a 2-SAT instance F on variable set $Y \cup Z$, where $|Y| = O(n)$, $|Z| \le \mathrm{poly}(m)$ and F has* $\mathrm{poly}(m)$ *clauses, such that for two constants $c_{\mathsf{PCPP}} > s_{\mathsf{PCPP}}$, the following hold for all $x \in \{0,1\}^n$.*

- *If $D(x) = 1$, then $\mathsf{OPT}(F_{Y=\mathsf{Enc}(x)}) \ge c_{\mathsf{PCPP}}$. Furthermore, there is a* $\mathrm{poly}(m)$*-time algorithm computing a corresponding $z_D(x)$ given x which satisfies at least a c_{PCPP} fraction of clauses.*
- *If $D(x) = 0$, then $\mathsf{OPT}(F_{Y=\mathsf{Enc}(x)}) \le s_{\mathsf{PCPP}}$.*

The key idea of [19] is to apply the above transformation on the obtained circuit D_C, and *guess* the corresponding $\mathscr{C}$ circuits for each output bit of the function $z_{D_C}(x)$. In [19], the focus is to prove worst-case lower bounds like $\mathsf{NQP} \not\subset \mathscr{C}$ for a circuit class $\mathscr{C}$. Therefore, we can safely assume $\mathsf{P} \subseteq \mathscr{C}$ and there exist corresponding $\mathscr{C}$ circuits for each output bit of $z_{D_C}(x)$.

[16] Note that here we are waiving the very important issue of *how to test whether the guessed* $\widetilde{\mathsf{Sum}}_{\delta} \circ \mathsf{ACC}^0$ *is valid*. We will discuss this issue at the end of the section.

[17] Here we don't remove the already satisfied clauses or the clauses which cannot be satisfied after the partial assignment.

[18] This formulation is due to [46].

However, in our case, we only have the collapse from NC^1 to $\widetilde{\mathsf{Sum}}_{\delta} \circ \mathsf{ACC}^0$. So we need the following adaption with the proof computable by a formula.

Theorem 1.16. *Let* Enc *be the encoder of some constant-rate error correcting code. There is a polynomial-time transformation that, given a formula D on n inputs of size $m \ge n$, outputs a 2-SAT instance F on variable set $Y \cup Z$, where $|Y| = O(n)$, $|Z| \le \mathrm{poly}(m)$ and F has* $\mathrm{poly}(m)$ *clauses, such that for two constants $c_{\mathsf{PCPP}} > s_{\mathsf{PCPP}}$, the following hold for all $x \in \{0,1\}^n$.*

- *If $D(x) = 1$, then $\mathsf{OPT}(F_{Y=\mathsf{Enc}(x)}) \ge c_{\mathsf{PCPP}}$. Furthermore, there is a* $\mathrm{poly}(m)$*-size formula computing a corresponding $z_D(x)$ given x which satisfies at least a c_{PCPP} fraction of clauses.*
- *If $D(x) = 0$, then $\mathsf{OPT}(F_{Y=\mathsf{Enc}(x)}) \le s_{\mathsf{PCPP}}$.*

The algorithm. Again, suppose for the sake of contradiction that V_{cert} does not certify n^{ε}-depth hardness. In particular, this means for all large enough n, it follows that if $V_{\mathrm{cert}}(1^n, \cdot)$ is satisfiable, then there is an n^{ε}-depth circuit C such that $V_{\mathrm{cert}}(1^n, tt(C)) = 1$. Translating it to the setting of PCP, the following hold for large enough n:

(1) (Low Depth Completeness) if $1^n \in L$, then there exists an n^{ε}-depth circuit $C : \{0,1\}^{\ell} \to \{0,1\}$ such that $V_{\mathsf{PCP}}(1^n)^C$ always accepts;
(2) (Soundness) if $1^n \notin L$, then for all oracles O, the probability that $V_{\mathsf{PCP}}(1^n)^O$ accepts is $\le 1/n$.

Recall that we set $D_C := V_{\mathsf{PCP}}(1^n)^C$. Our goal now is still to accept when $p_{\mathrm{acc}}(D_C) = 1$, and reject when $p_{\mathrm{acc}}(D_C) \le 1/n$.

By previous discussions, V_{PCP} can be taken as a 3-CNF, so D_C is indeed a circuit of depth $n^{\varepsilon} + O(\log n) = O(n^{\varepsilon})$, and therefore it is also a formula of size $2^{O(n^{\varepsilon})}$. Now we apply Theorem 1.16 to the formula D_C to obtain a 2-SAT instance F with $n_{\mathrm{clause}} = 2^{O(n^{\varepsilon})}$ clauses on variable set $Y \cup Z$.

Now we guess $|Z|$ $\mathsf{Sum}_{\delta} \circ \mathsf{ACC}^0$ circuits $T_1, T_2, \dots, T_{|Z|}$ and use $\widetilde{\pi}(x)$ to denote the concatenation of $T_1(x), T_2(x), \dots, T_{|Z|}(x)$. Then we estimate the following quantity

$$\begin{aligned} p_{\mathrm{key}} &:= \mathop{\mathbb{E}}_{x \in \{0,1\}^{\ell}} \mathop{\mathbb{E}}_{i \in [n_{\mathrm{clause}}]} F_i(\mathsf{Enc}(x), \widetilde{\pi}(x)) \\ &= \mathop{\mathbb{E}}_{i \in [n_{\mathrm{clause}}]} \mathop{\mathbb{E}}_{x \in \{0,1\}^{\ell}} F_i(\mathsf{Enc}(x), \widetilde{\pi}(x)), \end{aligned} \tag{1}$$

where F_i is the i-th clause in the 2-SAT instance F, so it only depends on two bits in $\mathsf{Enc}(x) \circ \widetilde{\pi}(x)$. By a simple manipulation, one can show that $F_i(\mathsf{Enc}(x), \widetilde{\pi}(x))$ also has a $\mathsf{Sum}_{O(\delta)} \circ \mathsf{ACC}^0$ circuit. Therefore, setting δ to be a small enough constant, we can apply the constant error CAPP algorithm from [19, 51] to estimate p_{key} in $2^{n-n^{\varepsilon}}$ time. Now we verify the correctness of the algorithm.

(1) When $p_{\mathrm{acc}}(D_C) = 1$, on the correct guess that $\widetilde{\pi}(x) = z_{D_C}(x)$ for all x, by Item (1) of Theorem 1.16, it follows $p_{\mathrm{key}} \ge c_{\mathsf{PCPP}}$.
(2) When $p_{\mathrm{acc}}(D_C) \le 1/n$, on all possible guesses, by Item (2) of Theorem 1.16, we have $p_{\mathrm{key}} \le s_{\mathsf{PCPP}} + 1/n$.

Therefore, to distinguish the above two cases, it suffices to estimate p_{key} within an additive error of $\frac{c_{\mathsf{PCPP}} - s_{\mathsf{PCPP}}}{10}$, and accept if our estimation is $\ge \frac{c_{\mathsf{PCPP}} + s_{\mathsf{PCPP}}}{2}$. Putting everything together, this puts $L \in \mathsf{NTIME}[2^n/n]$, contradiction.

Checking the guessed $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ *circuits.* Finally, as we have remarked briefly before, we waived an important issue on checking whether the guessed $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuits are *valid* (that is, whether the linear sum is close to either 0 or 1 on all inputs x). This is because in the algorithm described above, when $x \notin L$, it is still possible that we guess some *invalid* $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuits $T_1, T_2, \dots, T_{|Z|}$ and conclude that $p_{\mathsf{key}} > \frac{c_{\mathsf{PCPP}} + s_{\mathsf{PCPP}}}{2}$, as the constant error CAPP algorithm for $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ may behave arbitrarily on invalid $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuits.

More formally, given a presumed $\widetilde{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$ circuit C, let $f(x)$ be the corresponding $\sum_i \alpha_i C_i(x)$, and

$$\mathsf{bin}_f(x) := \begin{cases} 1 & f(x) > 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

To test whether C is valid, we want to check whether $\|\mathsf{bin}_f - f\|_\infty \le \delta$. Ideally, we want a test which accepts when $\|\mathsf{bin}_f - f\|_\infty \le \delta$ and reject when (say) $\|\mathsf{bin}_f - f\|_\infty \ge 3\delta$. But this turns out to be too hard.

Luckily, a careful examination shows that we only have to reject when $\|\mathsf{bin}_f - f\|_2 \ge 3\delta$, and this can be solved by a careful polynomial manipulation as in [19]. See [18, Section 5] for the details.

2 OPEN PROBLEMS

We conclude with several interesting open problems stemming from our work.

(1) The most exciting open question would be to apply Theorem 1.13 to prove super-polynomial lower bounds for TC^0_3.
(2) Are there P-complete problems with similar random-reducibility properties of CMD and DCMD? Besides being an interesting problem in its own right, the existence of such a problem would greatly simplify our framework for strong average-case lower bounds. In particular, we will no longer need hard MA problems with *low depth* predicates, and PCPP with *low depth* computable proofs.
(3) The seed length of our i.o. NPRG fooling ACC^0 circuits is only inverse sub-half-exponential. Can we obtain an i.o. NPRG with polylog(n) seed length? As a related question, can we show that there is a constant $\varepsilon > 0$ such that E^{NP} cannot be $(1/2 + 1/2^{n^\varepsilon})$-approximated by ACC^0 circuits of 2^{n^ε} size? (This paper only implicitly proves that E^{NP} cannot be $(1/2 + 1/f(n))$-approximated by ACC^0 circuits of $f(n)$ size for sub-half-exponential $f(n)$.)
(4) Since we have proved lower bounds for $\mathsf{MAJ} \circ \mathsf{ACC}^0$, the natural next step would be to prove lower bounds for $\mathsf{THR} \circ \mathsf{ACC}^0$. Can we formulate any *algorithmic approach* to prove such a lower bound? That is, are there certain non-trivial circuit-analysis algorithms for $\mathscr{C}$ which would imply $\mathsf{THR} \circ \mathscr{C}$ lower bounds?
It seems plausible to us that non-trivial #SAT algorithms would suffice (note that that we already proved non-trivial #SAT algorithms for $\mathscr{C}$ imply $\mathsf{MAJ} \circ \mathsf{Sum} \circ \mathscr{C}$ lower bounds, which is a non-trivial sub-class of $\mathsf{THR} \circ \mathscr{C}$). Such a connection would also imply lower bounds for $\mathsf{THR} \circ \mathsf{ACC}^0 \circ \mathsf{THR}$, which is (much) stronger than the already notorious circuit class $\mathsf{THR} \circ \mathsf{THR}$.
(5) Is THR contained in $\mathsf{MAJ} \circ \mathsf{ACC}^0$? (Or even $\mathsf{MAJ} \circ \mathsf{Sum} \circ \mathsf{ACC}^0$?) We don't have an inclination on the answer. But if it is contained in $\mathsf{MAJ} \circ \mathsf{ACC}^0$, it would immediately imply super-polynomial lower bounds for $\mathsf{THR} \circ \mathsf{THR}$.
(6) Vyas and Williams [46] conjectured that $\mathsf{SYM} \circ \mathscr{C}$ lower bounds should follow from #SAT algorithms for $\mathscr{C}$, where SYM denotes arbitrary symmetric functions. Can the new techniques in this paper help to prove this conjecture?

ACKNOWLEDGMENTS

The first author wants to thank his advisor Ryan Williams for basically everything: introducing the problem of proving lower bounds for $\mathsf{MAJ} \circ \mathsf{ACC}^0$ to him and being very optimistic about its resolution, countless encouraging and valuable discussion during the project, and also suggestions and comments on an early draft of this paper. The second author also wants to thank Ryan Williams for hosting him during the summer of 2019 and many inspiring discussions during the period.

We would like to thank Nikhil Vyas for helpful discussions.

REFERENCES

[1] Scott Aaronson and Avi Wigderson. 2009. Algebrization: A New Barrier in Complexity Theory. *TOCT* 1, 1 (2009), 2:1–2:54. https://doi.org/10.1145/1490270.1490272
[2] Miklós Ajtai. 1983. Σ^1_1-Formulae on finite structures. *Annals of Pure and Applied Logic* 24, 1 (1983), 1–48. https://doi.org/10.1016/0168-0072(83)90038-6
[3] Josh Alman and Lijie Chen. 2019. Efficient Construction of Rigid Matrices Using an NP Oracle. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, David Zuckerman (Ed.). IEEE Computer Society, 1034–1055. https://doi.org/10.1109/FOCS.2019.00067
[4] Benny Applebaum. 2014. *Cryptography in Constant Parallel Time.* Springer. https://doi.org/10.1007/978-3-642-17367-7
[5] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. 2006. Cryptography in NC^0. *SIAM J. Comput.* 36, 4 (2006), 845–888. https://doi.org/10.1137/S0097539705446950
[6] Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity - A Modern Approach.* Cambridge University Press. http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264
[7] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. 1998. Proof Verification and the Hardness of Approximation Problems. *J. ACM* 45, 3 (1998), 501–555. https://doi.org/10.1145/278298.278306
[8] Sanjeev Arora and Shmuel Safra. 1998. Probabilistic Checking of Proofs: A New Characterization of NP. *J. ACM* 45, 1 (1998), 70–122. https://doi.org/10.1145/273865.273901
[9] László Babai. 1987. Random Oracles Separate PSPACE from the Polynomial-Time Hierarchy. *Inf. Process. Lett.* 26, 1 (1987), 51–53. https://doi.org/10.1016/0020-0190(87)90036-6
[10] Swapnam Bajpai, Vaibhav Krishan, Deepanshu Kush, Nutan Limaye, and Srikanth Srinivasan. 2019. A #SAT Algorithm for Small Constant-Depth Circuits with PTF Gates. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA.* 8:1–8:20. https://doi.org/10.4230/LIPIcs.ITCS.2019.8
[11] Theodore P. Baker, John Gill, and Robert Solovay. 1975. Relativizations of the P =?NP Question. *SIAM J. Comput.* 4, 4 (1975), 431–442. https://doi.org/10.1137/0204037
[12] David A. Mix Barrington. 1989. Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1. *J. Comput. Syst. Sci.* 38, 1 (1989), 150–164. https://doi.org/10.1016/0022-0000(89)90037-8
[13] Eli Ben-Sasson and Emanuele Viola. 2014. Short PCPs with Projection Queries. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I.* 163–173. https://doi.org/10.1007/978-3-662-43948-7_14
[14] Abhishek Bhrushundi, Kaave Hosseini, Shachar Lovett, and Sankeerth Rao. 2019. Torus Polynomials: An Algebraic Approach to ACC Lower Bounds. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA.* 13:1–13:16. https://doi.org/10.4230/LIPIcs.ITCS.2019.13

[15] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. 2018. Exploring Crypto Dark Matter: - New Simple PRF Candidates and Their Applications. In *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part II*. 699–729. https://doi.org/10.1007/978-3-030-03810-6_25

[16] Eshan Chattopadhyay, Pooya Hatami, Shachar Lovett, and Avishay Tal. 2019. Pseudorandom Generators from the Second Fourier Level and Applications to AC^0 with Parity Gates. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*. 22:1–22:15. https://doi.org/10.4230/LIPIcs.ITCS.2019.22

[17] Lijie Chen. 2019. Non-deterministic Quasi-Polynomial Time is Average-Case Hard for ACC Circuits. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, David Zuckerman (Ed.). IEEE Computer Society, 1281–1304. https://doi.org/10.1109/FOCS.2019.00079

[18] Lijie Chen and Hanlin Ren. 2020. Strong Average-Case Circuit Lower Bounds from Non-trivial Derandomization. *Electronic Colloquium on Computational Complexity (ECCC)* 27 (2020), 10. https://eccc.weizmann.ac.il/report/2020/010

[19] Lijie Chen and R. Ryan Williams. 2019. Stronger Connections Between Circuit Analysis and Circuit Lower Bounds, via PCPs of Proximity. In *34th Computational Complexity Conference (CCC 2019) (Leibniz International Proceedings in Informatics (LIPIcs))*, Amir Shpilka (Ed.), Vol. 137. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 19:1–19:43. https://doi.org/10.4230/LIPIcs.CCC.2019.19

[20] Ruiwen Chen, Igor Carboni Oliveira, and Rahul Santhanam. 2018. An Average-Case Lower Bound Against ACC^0. In *LATIN 2018: Theoretical Informatics - 13th Latin American Symposium, Buenos Aires, Argentina, April 16-19, 2018, Proceedings*. 317–330. https://doi.org/10.1007/978-3-319-77404-6_24

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

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

[23] Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. 1992. Majority Gates VS. General Weighted Threshold Gates. *Computational Complexity* 2 (1992), 277–300. https://doi.org/10.1007/BF01200426

[24] Oded Goldreich and Leonid A. Levin. 1989. A Hard-Core Predicate for all One-Way Functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*. 25–32. https://doi.org/10.1145/73007.73010

[25] Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N. Rothblum. 2007. Verifying and decoding in constant depth. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*. 440–449. https://doi.org/10.1145/1250790.1250855

[26] Aryeh Grinberg, Ronen Shaltiel, and Emanuele Viola. 2018. Indistinguishability by Adaptive Procedures with Advice, and Lower Bounds on Hardness Amplification Proofs. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. 956–966. https://doi.org/10.1109/FOCS.2018.00094

[27] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. 1993. Threshold Circuits of Bounded Depth. *J. Comput. Syst. Sci.* 46, 2 (1993), 129–154. https://doi.org/10.1016/0022-0000(93)90001-D

[28] Johan Håstad. 1989. Almost Optimal Lower Bounds for Small Depth Circuits. *Advances in Computing Research* 5 (1989), 143–170.

[29] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. 2010. Uniform Direct Product Theorems: Simplified, Optimized, and Derandomized. *SIAM J. Comput.* 39, 4 (2010), 1637–1665. https://doi.org/10.1137/080734030

[30] Yuval Ishai and Eyal Kushilevitz. 2002. Perfect Constant-Round Secure Computation via Perfect Randomizing Polynomials. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*. 244–256. https://doi.org/10.1007/3-540-45465-9_22

[31] Joe Kilian. 1988. Founding Cryptography on Oblivious Transfer. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*. 20–31. https://doi.org/10.1145/62212.62215

[32] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. 1992. Algebraic Methods for Interactive Proof Systems. *J. ACM* 39, 4 (1992), 859–868. https://doi.org/10.1145/146585.146605

[33] Raghu Meka and David Zuckerman. 2013. Pseudorandom Generators for Polynomial Threshold Functions. *SIAM J. Comput.* 42, 3 (2013), 1275–1301. https://doi.org/10.1137/100811623

[34] Cody Murray and R. Ryan Williams. 2018. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. 890–901. https://doi.org/10.1145/3188745.3188910

[35] Noam Nisan and Avi Wigderson. 1994. Hardness vs Randomness. *J. Comput. Syst. Sci.* 49, 2 (1994), 149–167. https://doi.org/10.1016/S0022-0000(05)80043-1

[36] Alexander A Razborov. 1987. 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 (1987), 333–338.

[37] Alexander A. Razborov and Steven Rudich. 1997. Natural Proofs. *J. Comput. Syst. Sci.* 55, 1 (1997), 24–35. https://doi.org/10.1006/jcss.1997.1494

[38] Joel I. Seiferas, Michael J. Fischer, and Albert R. Meyer. 1978. Separating Nondeterministic Time Complexity Classes. *J. ACM* 25, 1 (1978), 146–167. https://doi.org/10.1145/322047.322061

[39] Ronen Shaltiel and Emanuele Viola. 2010. Hardness Amplification Proofs Require Majority. *SIAM J. Comput.* 39, 7 (2010), 3122–3154. https://doi.org/10.1137/080735096

[40] Adi Shamir. 1992. IP = PSPACE. *J. ACM* 39, 4 (1992), 869–877. https://doi.org/10.1145/146585.146609

[41] Roman Smolensky. 1987. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*. 77–82. https://doi.org/10.1145/28395.28404

[42] Roman Smolensky. 1993. On Representations by Low-Degree Polynomials. In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*. 130–138. https://doi.org/10.1109/SFCS.1993.366874

[43] Salil P. Vadhan. 2012. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science* 7, 1-3 (2012), 1–336. https://doi.org/10.1561/0400000010

[44] Emanuele Viola. 2009. On Approximate Majority and Probabilistic Time. *Computational Complexity* 18, 3 (2009), 337–375. https://doi.org/10.1007/s00037-009-0267-3

[45] Emanuele Viola. 2020. New lower bounds for probabilistic degree and AC0 with parity gates. *Electronic Colloquium on Computational Complexity (ECCC)* 27 (2020), 15. https://eccc.weizmann.ac.il/report/2020/015

[46] Nikhil Vyas and R. Ryan Williams. 2020. Lower Bounds Against Sparse Symmetric Functions of ACC Circuits: Expanding the Reach of #SAT Algorithms. In *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020) (Leibniz International Proceedings in Informatics (LIPIcs))*, Christophe Paul and Markus Bläser (Eds.), Vol. 154. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 59:1–59:17. https://doi.org/10.4230/LIPIcs.STACS.2020.59

[47] Ryan Williams. 2013. Improving Exhaustive Search Implies Superpolynomial Lower Bounds. *SIAM J. Comput.* 42, 3 (2013), 1218–1244. https://doi.org/10.1137/10080703X

[48] Ryan Williams. 2014. New algorithms and lower bounds for circuits with linear threshold gates. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*. 194–202. https://doi.org/10.1145/2591796.2591858

[49] Ryan Williams. 2014. Nonuniform ACC Circuit Lower Bounds. *Journal of the ACM (JACM)* 61, 1 (2014), 2:1–2:32. https://doi.org/10.1145/2559903

[50] R. Ryan Williams. 2016. Natural Proofs versus Derandomization. *SIAM J. Comput.* 45, 2 (2016), 497–529. https://doi.org/10.1137/130938219

[51] Richard Ryan Williams. 2018. Limits on Representing Boolean Functions by Linear Combinations of Simple Functions: Thresholds, ReLUs, and Low-Degree Polynomials. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*. 6:1–6:24. https://doi.org/10.4230/LIPIcs.CCC.2018.6

[52] Andrew C Yao. 1982. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*. IEEE, 80–91.

[53] Andrew Chi-Chih Yao. 1985. Separating the Polynomial-Time Hierarchy by Oracles (Preliminary Version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*. 1–10. https://doi.org/10.1109/SFCS.1985.49

[54] Stanislav Žák. 1983. A Turing Machine Time Hierarchy. *Theor. Comput. Sci.* 26, 3 (1983), 327–333. https://doi.org/10.1016/0304-3975(83)90015-4