Almost-Everywhere Circuit Lower Bounds from Non-Trivial Derandomization

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

Xin Lyu
Tsinghua University
Beijing, China
lvx17@mails.tsinghua.edu.cn

R. Ryan Williams
MIT
Cambridge, MA, USA
rrw@mit.edu

***Abstract*—In certain complexity-theoretic settings, it is notoriously difficult to prove complexity separations which hold *almost everywhere*, i.e., for all but finitely many input lengths. For example, a classical open question is whether NEXP is contained in i.o.-NP; that is, it is open whether nondeterministic exponential time computation can be simulated on infinitely many input lengths by an NP algorithm. This difficulty also applies to Williams' algorithmic method for circuit lower bounds [Williams, J. ACM 2014]. [Murray and Williams, STOC 2018] proved that nondeterminstic quasi-polynomial time is not contained in ACC^0, while it remained an open problem to show that E^NP (2^O(n) time with an NP oracle) is not contained in i.o.-ACC^0.**

In this paper, we show how many infinitely-often circuit lower bounds proved by the algorithmic method can be adapted to establish almost-everywhere lower bounds.

First, we show there is a function f in E^NP such that, *for all sufficiently large input lengths n*, f cannot be (1/2+exp(-n^e))-approximated by exp(n^e)-size ACC^0 circuits on inputs of length n (for all small e), improving lower bounds in [Chen and Ren, STOC 2020] and [Viola, ECCC 2020]. Second, we construct rigid matrices in P^NP for all but finitely many inputs, rather than infinitely often as in [Alman and Chen, FOCS 2019] and [Bhangale et al. 2020].

Third, we show there is a positive c such that E^NP has constant-error probabilistic degree at least cn/(log^2 n) for all large enough n, improving an infinitely-often separation by [Viola, ECCC 2020].

Our key to proving almost-everywhere worst-case lower bounds is a new "constructive" proof of an NTIME hierarchy theorem proved by [Fortnow and Santhanam, CCC 2016], where we show for every "weak" nondeterminstic algorithm, a "refuter algorithm" exists that can construct "bad" inputs for the hard language. We use this refuter algorithm to construct an almost-everywhere hard function. To extend our lower bounds to the average case, we prove a new XOR Lemma based on approximate linear sums, and combine it with PCP of proximity ideas developed in [Chen and Williams, CCC 2019] and [Chen and Ren, STOC 2020]. As a byproduct of our new XOR Lemma, we obtain a nondeterministic pseudorandom generator for poly-size ACC^0 circuits with seed length polylog(n), which resolves an open question in [Chen and Ren, STOC 2020].

***Keywords*-computational complexity**

I. Introduction

Proving unconditional circuit lower bounds for explicit functions (with the flagship problem of $\mathsf{NP} \not\subset \mathsf{P}_{/\mathsf{poly}}$) is one of the central problems in theoretical computer science. In the 1980s, considerable progress was made in proving lower bounds for constant-depth circuits, as first steps towards lower bounds for general circuits. The classical works [1], [2], [3], [4] culminated in exponential lower bounds for AC^0 (constant depth circuits consisting of unbounded fan-in AND/OR gates). The works [5], [6] established exponential lower bounds for $\mathsf{AC}^0[q]$ (AC^0 circuits with MOD_q gates) for prime power q.

Unfortunately, the progress in the 1980s did not go much further: lower bounds for $\mathsf{AC}^0[m]$ have been extremely difficult to establish for composite m, although it has been conjectured that $\mathsf{AC}^0[m]$ cannot compute the Majority function. In fact, it was a notorious open question whether NEXP (nondeterministic exponential time) has polynomial-size ACC^0 circuits.[1] Several years ago, Williams [8] finally proved such a lower bound, via an *algorithmic* approach to circuit lower bounds [9]. Combining many results from classical complexity, such as the nondeterministic time hierarchy theorem [10], [11], hardness vs randomness [12], and the PCP Theorem [13], [14], Williams' work shows how nontrivial circuit-analysis algorithms can be generically applied to prove circuit lower bounds.

Developments after $\mathsf{NEXP} \not\subset \mathsf{ACC}^0$. The separation $\mathsf{NEXP} \not\subset \mathsf{ACC}^0$ had several drawbacks compared to the classical lower bounds of the 80s. The most significant drawback was that NEXP is a much larger class than our ultimate goal NP (previous lower bounds for AC^0 or $\mathsf{AC}^0[p]$ usually work for functions in P). Murray and Williams improved this state of affairs significantly by showing $\mathsf{NQP} := \mathsf{NTIME}[2^{\mathrm{polylog}(n)}]$ is not contained in ACC^0 [15].

Another drawback is that the algorithmic approach [8], [15] only yielded worst-case lower bounds,

[1] This had been stressed several times as one of the most embarrassing open questions in complexity theory, see [7]. Note that ACC^0 denotes the union of $\mathsf{AC}^0[m]$ for all constant m.

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00009

while prior lower bounds for AC^0 or $\mathsf{AC}^0[p]$ can often be adapted to hold in the average case (e.g., [16]). A line of recent work [17], [18], [19], [20] generalizes the algorithmic approach to the average-case setting, culminating in the result that NQP cannot be $(1/2 + 1/\mathrm{poly}(n))$-approximated by ACC^0 circuits [20].

The Infinitely-Often Separation Drawback. All the aforementioned developments significantly expand the reach of the algorithmic method. However, there has remained a subtle but important drawback of the algorithmic method: it only achieves infinitely-often separations. For example, [15] shows there is an NQP function f such that, for every polynomial-size ACC^0 circuit family $\{C_n\}$, *there are infinitely many input lengths n such that C_n fails to compute f on n-bit inputs*. This certainly implies the separation $\mathsf{NQP} \not\subset \mathsf{ACC}^0$, but it could be the case that *for nearly every input length* NQP is easy for ACC^0, and NQP is only hard on extremely rare input lengths n, e.g., $n = 2^{2^{2^k}}$ for $k \in \mathbb{N}$. In a case where the hard input lengths are so far apart, *practically* the situation is not very different from $\mathsf{NQP} \subset \mathsf{ACC}^0$. In fact, it has remained open whether E^{NP} is contained *infinitely-often* in ACC^0.

The Infinitely-Often Barrier in Complexity Theory. Ideally, we desire *almost-everywhere separations*: we want a function $f = \{f_n : \{0,1\}^n \to \{0,1\}\}$ so that for all sufficiently large input lengths n, f_n cannot be computed by any ACC^0 circuit (in notation, we would say $f \notin$ i.o.-ACC^0). Most previous lower bounds for AC^0 and $\mathsf{AC}^0[p]$ are almost-everywhere: they show $f \notin$ i.o.-AC^0 or $f \notin$ i.o.-$\mathsf{AC}^0[p]$ for some f. Indeed, most combinatorial/algebraic lower bound approaches argue hardness for each input length separately, so they naturally give lower bounds for all input lengths. However, in structural complexity theory, arguments often involve different input lengths simultaneously, and it is common that in some settings almost-everywhere separations are much harder to achieve than corresponding infinitely-often separations. Two classical examples include:

- (An Almost-Everywhere NTIME Hierarchy is Open.) It is known that $\mathsf{NTIME}[2^n] \not\subset \mathsf{NTIME}[2^n/n]$ [10], [11], but it is open whether $\mathsf{NTIME}[2^n] \subset$ i.o.-$\mathsf{NTIME}[n \log n]$. (Indeed, there is an oracle O such that $\mathsf{NEXP}^O \subset$ i.o.-NP^O [21].)
- (An Almost-Everywhere Super-Linear Circuit Lower Bounds for $\mathsf{MATIME}[2^n]$ is Open.) It is known that $\mathsf{MA}_{/1} \not\subset \mathsf{SIZE}(n^k)$ for all k and $\mathsf{MATIME}[2^n] \not\subset \mathsf{P}_{/\mathsf{poly}}$, but it is open whether $\mathsf{MATIME}[2^n] \subset$ i.o.-$\mathsf{SIZE}(O(n))$. (Indeed, it is even open whether $\Sigma_2\mathsf{TIME}[2^n] \subset$ i.o.-$\mathsf{SIZE}(O(n))$.

Other examples include fixed-polynomial lower bounds for the complexity classes $\mathsf{NP}^{\mathsf{NP}}$ [22], $\mathsf{ZPP}^{\mathsf{NP}}$ [23], [24], $\mathsf{S}_2\mathsf{P}$ [25], [26], PP [27], [28], time-space trade-off for solving SAT [29], [30], and hierarchy theorems such as [31], [32], [33]. All of these lower bounds only provide an infinitely-often separation, and it is open to prove an almost-everywhere separation. There are also interesting algorithmic results motivated by complexity concerns, which are only guaranteed to work for infinitely many input lengths (*e.g.*, [34], [35], [36]).

A. Our Results

In this work, we achieve almost-everywhere circuit lower bounds with the algorithmic approach. To formally discuss our results, we briefly recall two circuit-analysis problems.

1) CAPP**:** Given a circuit C of size S, estimate the probability that C accepts a uniformly random input within an additive error of $1/S$.
2) Gap-UNSAT**:** Given a circuit C, distinguish between the case that C is unsatisfiable and the case that C has at least $2^n/3$ satisfying assignments.

1) Almost-Everywhere Circuit Lower Bounds From Non-Trivial Derandomization: Our first result is that "non-trivial derandomization" for a circuit class $\mathscr{C}$ implies almost-everywhere $\mathscr{C}$-circuit lower bounds for E^{NP}. In the following, we say that a circuit class $\mathscr{C}$ is *typical* if $\mathscr{C}$ is closed under projections and negations. (See the full version of the paper for a formal definition.)

Theorem I.1. *There are universal constants $\varepsilon \in (0,1)$, $K \geq 1$ satisfying the following. Let $\mathscr{C}$ be typical, and let $s(n)$ be a nondecreasing time-constructible function with $n \leq s(n) \leq 2^{\varepsilon n}$ for all n. If* Gap-UNSAT *on $\mathsf{AND} \circ \mathsf{OR} \circ \mathscr{C}$-circuits of size $s(n)^K$ can be solved deterministically in $2^n/n^{\omega(1)}$ time, then there are functions in E^{NP} that do not have $\mathscr{C}$-circuits of size $s(n/2)$, for all sufficiently large n.*

An Extension to Average-Case Lower Bounds. Combining PCPs of Proximity and a new XOR Lemma (see Section I-B2), we can extend the above theorem to prove strong average-case lower bounds. Say that a function $f : \{0,1\}^n \to \{0,1\}$ *cannot be $(1/2+\varepsilon)$-approximated* by circuits of type $\mathcal{C}$, if every circuit from $\mathcal{C}$ computes f correctly on less than $(1/2+\varepsilon)2^n$ of the n-bit inputs. For a language $L : \{0,1\}^\star \to \{0,1\}$, we use L_n to denote its restriction to n-bit inputs.

Theorem I.2. *Let $\mathscr{C}$ be typical. Suppose there is an $\varepsilon > 0$ such that* CAPP *of 2^{n^ε}-size $\mathsf{AND}_4 \circ \mathscr{C}$ circuits can be deterministically solved in 2^{n-n^ε} time. Then there is a language $L \in \mathsf{E}^{\mathsf{NP}}$ and a constant $\delta > 0$ such that,*

for all large enough n, L_n cannot be $(1/2 + 2^{-n^\delta})$-approximated by $\mathscr{C}$-circuits of size 2^{n^δ}.

The above results have several applications to complexity lower bounds and pseudorandom generators. We will discuss them separately.

2) Applications in Complexity Lower Bounds: **Almost-Everywhere Strong Average-Case Exponential Lower Bounds for $\mathsf{ACC}^0 \circ \mathsf{THR}$.**[2] Combining Theorem I.2 and the corresponding #SAT algorithm from [37] for $\mathsf{ACC}^0 \circ \mathsf{THR}$, the almost-everywhere strongly average-case lower bound for E^{NP} against $\mathsf{ACC}^0 \circ \mathsf{THR}$ follows immediately.

Recall that for a given $d, m \in \mathbb{N}$, $\mathsf{AC}^0_d[m]$ is the class of circuit families of depth d, with unbounded fan-in $\mathsf{AND}, \mathsf{OR}, \mathsf{MOD}_m$ gates, and $\mathsf{ACC}^0 := \bigcup_{d,m} \mathsf{AC}^0_d[m]$.

Corollary I.3. *For every integer $d, m \geq 1$, there is an $\varepsilon = \varepsilon_{d,m}$ and a language $L \in \mathsf{E}^{\mathsf{NP}}$ such that L_n cannot be $(1/2 + 2^{-n^\varepsilon})$-approximated by $\mathsf{AC}^0_d[m] \circ \mathsf{THR}$ circuits of 2^{n^ε} size, for all sufficiently large n.*

Corollary I.3 compares favorably with prior circuit lower bounds for problems in E^{NP}. Williams [8], [37] proved that E^{NP} cannot be worst-case computed by $2^{n^{o(1)}}$ size $\mathsf{ACC}^0 \circ \mathsf{THR}$ circuits. Following the work of Rajgopal *et al.* [38], Viola [39] recently proved E^{NP} cannot be $(1/2 + 1/n^{1-\varepsilon})$-approximated by $2^{n^{o(1)}}$-size $\mathsf{AC}^0[\oplus]$ circuits. Chen and Ren [20] recently proved that E^{NP} cannot be $(1/2 + g(n)^{-1})$ approximated by $g(n)$-size ACC^0 circuits, where g is any sub-half-exponential function.[3] All of these lower bounds are only infinitely-often separations, and yield strictly weaker average-case lower bounds than Corollary I.3.

We also remark that [40] devised a notion of "significant separation", which is stronger than infinitely-often separation while weaker than almost-everywhere separation.[4] They showed a significant separation of NEXP and ACC^0. This is incomparable with almost-everywhere separation for E^{NP}.

Almost-Everywhere Construction of Rigid Matrices with an NP Oracle. The problem of efficiently constructing *rigid matrices* is a longstanding open problem in complexity theory [41], [42].

Definition I.4. Let $\mathbb{F}$ be a field. For $r, n \in \mathbb{N}$ and a matrix $M \in \mathbb{F}^{n \times n}$, the r-rigidity of M, denoted as $\mathcal{R}_M(r)$, is the minimum Hamming distance between M and a matrix of rank at most r.

Alman and Chen [43] recently showed that rigid matrices over the field $\mathbb{F}_2$ (similar results hold for all fields of constant order) with interesting parameters (considered by [44] for connections to communication complexity) could be constructed *infinitely often* in P^{NP} via the algorithmic method. Their proof has been simplified and improved by Bhangale *et al.* [45]. Formally, [45] construct a P^{NP} algorithm M which, for infinitely many n, $M(1^n)$ outputs a matrix H_n such that $\mathcal{R}_{H_n}(2^{\log^{1-\varepsilon} n}) \geq \delta n^2$ over $\mathbb{F}_2$.

Applying similar ideas from the proof of Theorem I.1 and Theorem I.2, we can strengthen their construction to an almost-everywhere one.

Theorem I.5. *There is a $\delta > 0$ such that, for all finite fields $\mathbb{F}$ and $\varepsilon > 0$, there is a P^{NP} algorithm which on input 1^n outputs an $n \times n$ matrix H satisfying $\mathcal{R}_H(2^{\log^{1-\varepsilon} n}) \geq \delta n^2$ over $\mathbb{F}$, for all large enough n.*

Almost-Everywhere Probabilistic Degree Lower Bounds. The notion of probabilistic degree for Boolean functions has been studied extensively for decades. Let us recall the definition.

Definition I.6. The *ε-error probabilistic degree* of a function $f : \{0,1\}^n \to \{0,1\}$ is the minimum d such that there is a distribution $\mathcal{D}$ on $\mathbb{F}_2$-polynomials of degree at most d such that $\Pr_{P \sim \mathcal{D}}[P(x) \neq f(x)] \leq \varepsilon$. When ε is not specified, it is assumed to be $1/3$ by default.

Very recently, Viola [39] proved an $\Omega(n/\log^2 n)$ probabilistic degree lower bound for E^{NP} using the algorithmic method. We extend his result to the almost-everywhere case.

Theorem I.7. *There is a language $L : \{0,1\}^\star \to \{0,1\}$ in E^{NP} such that L_n has $1/3$-error probabilistic degree $\Omega(n/\log^2 n)$, for all sufficiently large n.*

Almost-Everywhere Exponential Correlation Bounds against $n^{1/2-\delta}$-Degree $\mathbb{F}_2$-Polynomials. Combining the proof technique of the main theorem and an improved XOR Lemma (introduced in the next subsection), we can also prove a strong inapproximability result for low-degree polynomials for a problem in E^{NP}.

Theorem I.8. *For all $\delta > 0$, there is a language $L : \{0,1\}^\star \to \{0,1\}$ in E^{NP} such that L_n cannot be $(1/2 + 2^{-n^{\Omega(1)}})$-approximated by $n^{1/2-\delta}$ degree $\mathbb{F}_2$-polynomials for all sufficiently large n.*

The previous best known correlation bound against $n^{1/2-\delta}$-degree $\mathbb{F}_2$-polynomials was only $1/2 + n^{-\delta}$

[2] $\mathsf{ACC}^0 \circ \mathsf{THR}$ denotes the class of constant-depth circuits comprised of AND, OR and MOD_m gates (for a constant $m > 1$), with a bottom layer of gates computing arbitrary linear threshold functions.

[3] We say that g is *sub-half-exponential* if $g(g(n)) = 2^{n^{o(1)}}$.

[4] Roughly speaking, "significant separation" means that when the separation holds for an input length n, then there is another input length at most polynomially larger than n such that the separation also holds. That is, the hardness cannot be very "sparsely distributed".

for the Majority function [5], [6], [46], and this degree/approximation tradeoff is indeed tight for Majority [47].

3) Applications to Pseudorandom Generators: Following the known connection between average-case hardness and PRG construction, we obtain two different PRG constructions for ACC^0, both with the near-optimal $\mathrm{polylog}(n)$ seed length. Our first PRG works for all input lengths, while is only computably in E^{NP}. Our second construction obtains a *nondeterministic pseudorandom generator* (NPRG) (see the full version of the paper for a formal definition), which is a weaker class compared to E^{NP}-computable PRGs. But the NPRG only works for infinitely many input lengths.

The following E^{NP}-computable PRG is a direct consequence of our average-case lower bound for ACC^0 (Corollary I.3).

Theorem I.9. *For all constants d and m, there is $\delta = \delta(d,m)$ and an E^{NP}-computable PRG which takes n-bit seed and outputs 2^{n^δ}-bit strings fooling $\mathsf{AC}^0_d[m]$ circuits of size 2^{n^δ}, for all large enough n.*

Our proof technique can also be used to construct an infinitely-often PRG against ACC^0 circuits with $\mathrm{polylog}(n)$ seed length. This significantly improves upon the previous work By Chen and Ren [20], and answers one of their open questions.

Theorem I.10. *For all constants d and m, there is $\delta = \delta(d,m) > 0$ and an i.o.-NPRG which takes n-bit seeds and outputs 2^{n^δ}-bit strings fooling $\mathsf{AC}^0_d[m]$ circuits of size 2^{n^δ}.*

Remark I.11. We remark that our NPRG and E^{NP}-computable PRG also work for other circuit classes $\mathscr{C}$, given non-trivial CAPP algorithm for slightly larger circuit classes $\mathscr{C}'$. See the full version of the paper for the details.

B. Two Technical Tools

To achieve our almost-everywhere strongly average-case lower bounds, we develop two new technical tools. The first is a "constructive" proof of the almost-everywhere sublinear witness NTIME hierarchy of Fortnow and Santhanam [48] which builds a P^{NP} algorithm that can explicitly find inputs on which the weak algorithm make mistakes. The second is an XOR Lemma based on computations by approximate linear sums. We believe both results are interesting in their own right, and will likely have other applications in computational complexity. In the following we state both of them informally. Check Section II for a more in-depth discussion on these two new tools, and why they come up naturally in our lower bound proofs.

1) An Almost-Everywhere (Sublinear Witness) NTIME Hierarchy with Refuter: A critical piece of Williams' proof that $\mathsf{NEXP} \not\subset \mathsf{ACC}^0$ (and later work) is the NTIME hierarchy [10], [11]. However, as mentioned earlier, that hierarchy is only known to hold infinitely-often; consequently, the resulting circuit lower bounds fail to be almost-everywhere, and extending the NTIME hierarchy to hold almost-everywhere is notoriously open.

Nevertheless, Fortnow and Santhanam [48] managed to prove an almost-everywhere NTIME hierarchy for a restricted subclass of NTIME, where the "weak" nondeterministic machines (being diagonalized against) use witnesses of length less than n bits. Let $\mathsf{NTIMEGUESS}[t(n), g(n)]$ be the class of languages decided by nondeterministic algorithms running in $O(t(n))$ steps and guessing at most $g(n)$ bits. Fortnow and Santhanam proved there is a language L in nondeterministic $O(T(n))$ time that is not decidable, even infinitely-often, by nondeterministic $o(T(n))$-time $n/10$-guess machines:

Theorem I.12. *For every time-constructible function $T(n)$ such that $n \le T(n) \le 2^{\mathrm{poly}(n)}$, $\mathsf{NTIME}[T(n)] \not\subset$ i.o.-$\mathsf{NTIMEGUESS}[o(T(n)), n/10]$.*

Our most important new ingredient is the construction of a "refuter" for the hierarchy of Theorem I.12: an algorithm with an NP oracle which can efficiently find bad inputs for any $\mathsf{NTIMEGUESS}[o(T(n)), n/10]$ machine.

Theorem I.13 (Refuter with an NP Oracle, Informal)**.** *For every time-constructible function $T(n)$ such that $n \le T(n) \le 2^{\mathrm{poly}(n)}$, there is a language $L \in \mathsf{NTIME}[T(n)]$ and an algorithm $\mathcal{R}$ such that:*

1) ***Input.*** *The input to $\mathcal{R}$ is a pair $(M, 1^n)$, with the promise that M describes a nondeterministic Turing machine running in $o(T(n))$ time and guessing at most $n/10$ bits.*
2) ***Output.*** *For every fixed M and all sufficiently large n, $\mathcal{R}(M, 1^n)$ outputs a string $x \in \{0,1\}^n$ such that $M(x) \neq L(x)$.*
3) ***Complexity.*** *$\mathcal{R}$ runs in $\mathrm{poly}(T(n))$ time with adaptive access to an SAT oracle.*

Since $\mathcal{R}$ can find counterexamples to any faster algorithm attempting to decide L, we call $\mathcal{R}$ a refuter.

Applying the refuter construction of Theorem I.13 instead of the general NTIME hierarchy in the original proof of [8], we can achieve almost-everywhere circuit lower bounds.

Other Explicit Refuters for Complexity Separations: It is instructive to compare our refuter construction to other refuter constructions, such as [49], [50], [51]. They showed that, assuming certain complexity

separations ($\mathsf{NP} \neq \mathsf{P}$, $\mathsf{NP} \not\subset \mathsf{BPP}$ or $\mathsf{NP} \not\subset \mathsf{P}_{/\mathsf{poly}}$), one can construct a refuter which takes a corresponding algorithm A claimed to solve SAT, and outputs for infinitely many n a counter-example x_n of length n such that A fails to solve SAT on x_n. All these refuters are conditional, in the sense that they assumed the (unproven) hypothesis such as $\mathsf{NP} \neq \mathsf{P}$, while our refuter is designed to witness the already proven NTIME hierarchy theorem of [48].

2) An XOR Lemma Based on Approximate Linear Sums of Circuits: Our second important technical ingredient—critical to our average case lower bounds—is a new XOR Lemma based on approximate linear sums of circuits. The XOR Lemma (originally due to Yao [52]) says that if an n-bit Boolean function f cannot be *weakly approximated* (e.g., 0.99-approximated) by small circuits, then the kn-bit Boolean function $f^{\oplus k}$ cannot be *strongly approximated* (e.g., $(1/2+2^{-\Omega(k)})$-approximated) by smaller and simpler circuits.[5]

It is tempting to apply the XOR Lemma directly, to try to prove strong average-case lower bounds for ACC^0 (or $\mathsf{ACC}^0[2]$) given that weak bounds are known. However, when we apply the XOR Lemma to a restricted circuit class $\mathscr{C}$, the most refined analysis of the XOR Lemma [53], [54] still only shows that 0.99-inapproximability for $\mathsf{MAJORITY}_{t^2} \circ \mathscr{C}$ computing f implies $(1/2+1/t)$-inapproximability for $\mathscr{C}$ computing $f^{\oplus k}$, where $k = O(\log t)$. That is, applying the XOR Lemma to show strong-average case lower bounds for ACC^0, evidently requires proving weak average-case lower bounds for $\mathsf{MAJORITY} \circ \mathsf{ACC}^0$. But this task seems just as hard as proving strong average-case lower bounds in the first place!

We avoid this issue by proving a new kind of XOR Lemma, based on Levin's proof of the XOR Lemma [55]. For a circuit class $\mathscr{C}$, we define the class of linear combinations $[0,1]\mathsf{Sum} \circ \mathscr{C}$, where an n-input circuit C from the class has the form $C(x) := \sum_i \alpha_i \cdot C_i(x)$, where each $\alpha_i \in \mathbb{R}$ and $C_i \in \mathscr{C}$, and C satisfies the promise that $C(x) \in [0,1]$ for all $x \in \{0,1\}^n$. The *complexity* of C is defined to be the maximum of $\sum_i |\alpha_i|$ and the sum of the sizes of each C_i.

Our new XOR Lemma shows that if a Boolean function f cannot be well-approximated by linear combinations of $\mathscr{C}$-circuits *on average*, then $f^{\oplus k}$ is strongly average-case hard for $\mathscr{C}$-circuits. The flexibility afforded by linear combinations allows us to improve our results to strong average-case lower bounds.

Theorem I.14 (New XOR Lemma, Informal)**.** *For every $\delta < \frac{1}{2}$, for every function $f : \{0,1\}^n \to \{0,1\}$, if there is no $[0,1]\mathsf{Sum} \circ \mathscr{C}$ circuit $C : \{0,1\}^n \to [0,1]$ of complexity $\mathrm{poly}(s,n,1/\delta)$ such that $\mathbb{E}_{x\in\{0,1\}^n} |C(x) - f(x)| \leq \delta$, then $f^{\oplus k}$ cannot be $1/2+1/s$ approximated by s-size $\mathscr{C}$ circuits, for $k = \Theta\left(\delta^{-1}\log s\right)$.*

[5]The function $f^{\oplus k}$ partitions its kn-bit input into k blocks $x_1, x_2, \ldots, x_k$ of length n each, and outputs $\bigoplus_{i=1}^k f(x_i)$.

II. Technical Overview

In this section we provide more intuition behind our almost-everywhere lower bounds. We split the discussion into two parts, one for each main technical ingredient.

- In Section II-A, we demonstrate how to use our new refuter concept (and why it comes up naturally) to prove almost-everywhere E^{NP} lower bounds. With this powerful concept, we can automatically strengthen most of the previous E^{NP} lower bounds proved via the algorithmic method, except for the strong average-case lower bounds in [20].
- In Section II-B, we show how to use an XOR Lemma for approximate linear combinations of circuits, to prove a strong average-case almost-everywhere lower bounds for E^{NP}.

A. Almost-Everywhere Lower Bounds for E^{NP} and the Refuter

To explain the intuition behind our almost-everywhere circuit lower bounds, it is instructive to first recall how Williams [8] proved that E^{NP} does not have $2^{n^{o(1)}}$-size ACC^0 circuits, and understand why that approach only achieves an infinitely-often separation.

1) Review of E^{NP} not in ACC^0: **A Nondeterministic Algorithm $\mathcal{A}_{L^{\mathsf{hard}}}$ That Can't Be Improved.** By the NTIME hierarchy [10], [11], we know there is a language $L^{\mathsf{hard}} \in \mathsf{NTIME}[2^n] \setminus \mathsf{NTIME}[2^n/n]$. Let $\mathcal{A}_{L^{\mathsf{hard}}}$ be a nondeterministic $O(2^n)$-time algorithm deciding L^{hard}.

A "Cheating" Algorithm $\mathcal{A}_{\mathsf{PCP}}$ Trying to Speed Up $\mathcal{A}_{L^{\mathsf{hard}}}$. Assume we have non-trivial derandomization algorithms for ACC^0, i.e., there is a $2^n/n^{\omega(1)}$-time algorithm for deciding $\mathsf{Gap\text{-}UNSAT}$ on ACC^0-circuits of n inputs and $2^{n^{o(1)}}$ size. A key idea in [8], [9] is to combine probabilistically checkable proofs (PCPs) and non-trivial $\mathsf{Gap\text{-}UNSAT}$ algorithms to design a nondeterministic algorithm $\mathcal{A}_{\mathsf{PCP}}$ that tries to "speed up" the algorithm $\mathcal{A}_{L^{\mathsf{hard}}}$, as follows:

- Given an input $x \in \{0,1\}^n$, $\mathcal{A}_{\mathsf{PCP}}$ applies an efficient PCP reduction (e.g., [56]) to $\mathcal{A}_{L^{\mathsf{hard}}}(x)$. For $\ell = n + O(\log n)$, we obtain a verifier oracle circuit $\mathsf{VPCP}_x(r) \colon \{0,1\}^\ell \to \{0,1\}$ with the following properties.
  - (Simplicity) VPCP_x is an oracle circuit with gates for a function $\mathcal{O} \colon \{0,1\}^\ell \to \{0,1\}$. VPCP_x has very simple structure, so that if $\mathcal{O}$

is an ACC^0-circuit, then the composed circuit $\mathsf{VPCP}_x^{\mathcal{O}}$ is also in ACC^0.
  - (Completeness) If $x \in L^{\mathsf{hard}}$, then there is an oracle $\mathcal{O}$ such that $\Pr_{r\in\{0,1\}^\ell}[\mathsf{VPCP}_x^{\mathcal{O}}(r) = 1] = 1$.
  - (Soundness) If $x \notin L^{\mathsf{hard}}$, then for all oracles $\mathcal{O}$, $\Pr_{r\in\{0,1\}^\ell}[\mathsf{VPCP}_x^{\mathcal{O}}(r) = 1] \leq 1/3$.
- Next, $\mathcal{A}_{\mathsf{PCP}}$ guesses an ACC^0-circuit C of size $2^{n^{o(1)}}$. By the simplicity property, VPCP_x^C (with oracle C) is also an ACC^0-circuit of $2^{n^{o(1)}}$ size. Running the assumed Gap-UNSAT algorithm for ACC^0 on the circuit $D = \neg\mathsf{VPCP}_x^C$, we can distinguish between the case that D is unsatisfiable (which happens for some $\mathcal{O}$, if $x \in L^{\mathsf{hard}}$) and the case that $\Pr_x[D(x)] \geq 2/3$ (which happens for all $\mathcal{O}$, if $x \notin L^{\mathsf{hard}}$). Therefore, we accept if and only if our Gap-UNSAT algorithm returns "unsatisfiable".

Intuitively, $\mathcal{A}_{\mathsf{PCP}}$ wants to "cheat" in the computation of L^{hard} by *only considering oracles of small circuit complexity*.

The E^{NP} Lower Bound. Note that the nondeterministic algorithm $\mathcal{A}_{\mathsf{PCP}}$ indeed runs faster than 2^n time: the running time of $\mathcal{A}_{\mathsf{PCP}}$ is dominated by the running time of the non-trivial derandomization algorithm, which is $o(2^n/n)$. Therefore, we know that $\mathcal{A}_{\mathsf{PCP}} \in \mathsf{NTIME}[o(2^n/n)]$, and hence it *cannot* compute L^{hard} by the NTIME hierarchy theorem.

We conclude that, for infinitely many n, there is a "bad input" $x_n \in \{0,1\}^n$ such that $\mathcal{A}_{\mathsf{PCP}}(x_n)$ rejects but $x_n \in L^{\mathsf{hard}}$.[6] For those $x_n \in L^{\mathsf{hard}}$, the completeness of the PCP implies there is an oracle $\mathcal{O}$ such that $\mathsf{VPCP}_{x_n}^{\mathcal{O}}(r) = 1$ for all r, but no such oracle admits $2^{n^{o(1)}}$-size ACC^0-circuit —otherwise, $\mathcal{A}_{\mathsf{PCP}}$ would have guessed it, and $\mathcal{A}_{\mathsf{PCP}}(x_n)$ would accept instead. Therefore, constructing a description of the oracle $\mathcal{O}$ is tantamount to constructing a function without small ACC^0 circuits.

We can now design the hard E^{NP} language as follows: on an input x of length $2n + O(\log n)$, split x into two parts x_1 and x_2 such that $|x_1| = n$. Using an NP oracle, we search for the lexicographically-first oracle $\mathcal{O}\colon \{0,1\}^\ell \to \{0,1\}$ for the verifier VPCP_{x_1} (that is, $\mathsf{VPCP}_{x_1}^{\mathcal{O}}(r) = 1$ for all r). Finally, we output $\mathcal{O}(x_2)$. If there is an $x_n \in \{0,1\}^n$ such that $\mathcal{A}_{\mathsf{PCP}}(x_n)$ rejects but $x_n \in L^{\mathsf{hard}}$, this E^{NP} algorithm has high ACC^0 circuit complexity on input length $2n + O(\log n)$.

Input and Advice. In the above, the input x is split into two parts x_1 and x_2. The part x_1 behaves as "advice" specifying the "bad input" on which $\mathcal{A}_{\mathsf{PCP}}$ and L^{hard} differ. The part x_2 is the input to the constructed oracle $\mathcal{O}$. Luckily for us, the advice is roughly the same length as the input length to the oracle, so it does not affect the hardness of the overall function. For example, a hardness result superpolynomial in $|x_2|$ is also superpolynomial in $|x_1| + |x_2|$, since we assumed $|x_1| = O(|x_2|)$.

The NTIME Hierarchy Barrier. Let us examine the above proof outline more carefully. The analysis shows that the language decided by our E^{NP} algorithm is hard on inputs of length $2n$, provided that $\mathcal{A}_{\mathsf{PCP}}$ and L^{hard} give different outputs when they are restricted to inputs of length n. From the NTIME hierarchy and the fact that $\mathcal{A}_{\mathsf{PCP}} \in \mathsf{NTIME}[O(2^n/n)]$, we conclude there must be infinitely many such n.

The above argument would yield an almost-everywhere separation, if we could show that L^{hard} and $\mathcal{A}_{\mathsf{PCP}}$ are different *on all sufficiently large input lengths*. However, this would apparently require showing $\mathsf{NTIME}[2^n] \not\subset$ i.o.-$\mathsf{NTIME}[2^n/n]$, and such an almost-everywhere separation is a notoriously hard open problem—it is even open whether $\mathsf{NEXP} \subset$ i.o.-NP! It seems hopeless to make progress using the above framework, without making breakthrough progress on an almost-everywhere NTIME hierarchy.

First Observation: $\mathcal{A}_{\mathsf{PCP}}$ Guesses Short Witnesses. Here we make an important observation that bypasses the above barrier. For the above proof to work, L^{hard} only needs to be hard for the specific algorithm $\mathcal{A}_{\mathsf{PCP}}$, not necessarily *all nondeterministic $o(2^n/n)$-time algorithms*. In other words, we do not need the full power of an almost-everywhere NTIME hierarchy. Therefore, it is natural to examine what properties of the specific algorithm $\mathcal{A}_{\mathsf{PCP}}$ we can exploit.

One way in which $\mathcal{A}_{\mathsf{PCP}}$ is very different from a general $O(2^n)$-time nondeterministic algorithm is that it makes a considerably smaller number of guesses: only $2^{n^{o(1)}}$. Such restricted versions of NTIME have been studied before, under the guise of *bounded nondeterminism*. We use $\mathsf{NTIMEGUESS}[t(n), g(n)]$ to denote the class of languages decidable by nondeterministic algorithms using $O(t(n))$ steps and guessing at most $g(n)$ witness bits. Fortnow and Santhanam [48] showed that when $g(n)$ is sublinear, one can establish an almost-everywhere NTIME hierarchy (Theorem I.12).

Trying a New Approach. A natural proposal is to apply Theorem I.12 instead of the general NTIME hierarchy. For that purpose, we have to choose our parameters carefully so that $\mathcal{A}_{\mathsf{PCP}}$ makes few guesses. Let us check what happens when we rely on Fortnow and Santhanam's almost-everywhere hierarchy instead in our design of $\mathcal{A}_{\mathsf{PCP}}$. Our pseudocode below will fail,

[6] Note it is impossible that $\mathcal{A}_{\mathsf{PCP}}(x_n)$ accepts but $x_n \notin L^{\mathsf{hard}}$, as $\mathcal{A}_{\mathsf{PCP}}$ only guesses over a proper subset of all possible witnesses.

but studying how to fix it will lead to a correct lower bound proof.

1) Suppose we have a non-trivial circuit analysis algorithm (for CAPP or Gap-UNSAT) for ACC^0-circuits of size 2^{n^ε}. Set $k < \frac{1}{\varepsilon}$ and $T(n) = 2^{\log^k n}$. Let L^{hard} be a language in $\mathsf{NTIME}[T(n)]$ but not in i.o.-$\mathsf{NTIMEGUESS}[o(T(n)), n/10]$. Let $\mathcal{A}_{L^{\mathsf{hard}}}$ be an $O(T(n))$-time nondeterministic algorithm deciding L.
2) Given an input x of length n, $\mathcal{A}_{\mathsf{PCP}}$ applies the PCP reduction (e.g. [56]) to $\mathcal{A}_{L^{\mathsf{hard}}}(x)$. For $\ell = \log^k n + O(\log\log n)$, we obtain a verifier oracle circuit $\mathsf{VPCP}_x\colon \{0,1\}^\ell \to \{0,1\}$, with three properties.
   - (Simplicity) VPCP_x calls an oracle $\mathcal{O}\colon \{0,1\}^\ell \to \{0,1\}$. VPCP_x has very simple structure such that if $\mathcal{O}$ is an ACC^0-circuit, then so is the composed circuit $\mathsf{VPCP}_x^{\mathcal{O}}$.
   - (Completeness) If $x \in L^{\mathsf{hard}}$, then there is an oracle $\mathcal{O}$ such that $\Pr_{r\in\{0,1\}^\ell}[\mathsf{VPCP}_x^{\mathcal{O}}(r) = 1] = 1$.
   - (Soundness) If $x \notin L^{\mathsf{hard}}$, then for all oracle $\mathcal{O}$, $\Pr_{r\in\{0,1\}^\ell}[\mathsf{VPCP}_x^{\mathcal{O}}(r) = 1] \le 1/3$.
3) Next, $\mathcal{A}_{\mathsf{PCP}}$ guesses an ACC^0-circuit C of size $2^{\ell^\varepsilon} = o(n)$ (since $k < \varepsilon^{-1}$). Note that VPCP_x^C is also an ACC^0-circuit of $2^{O(\ell^\varepsilon)}$ size, by the simplicity property. Then, we use our non-trivial circuit analysis algorithm to estimate the acceptance probability of $\mathsf{VPCP}_x^C(\cdot)$, and **accept** if and only if the estimate is $\ge 1/2$.

This new $\mathcal{A}_{\mathsf{PCP}}$ runs in $o(T(n))$ time, and guesses at most $n/10$ bits of witness, so its language is in $\mathsf{NTIMEGUESS}[o(T(n)), n/10]$. Hence, for all large enough n, there is an $x_n \in \{0,1\}^n$ such that $x_n \in L^{\mathsf{hard}}$ while $\mathcal{A}_{\mathsf{PCP}}$ rejects x_n. Consequently, by a similar analysis as in Section II-A1, we have: (1) there is an oracle $\mathcal{O}$ such that $\mathsf{VPCP}_x^{\mathcal{O}}(r) = 1$ for all r, and (2) no such oracle has 2^{ℓ^ε} size ACC^0 circuits.

Therefore, for all large enough n, there is an $x_n \in \{0,1\}^n$ such that the lexicographically-first correct oracle $\mathcal{O}_n\colon \{0,1\}^{\ell(n)} \to \{0,1\}$ for VPCP_x does not have $2^{\ell(n)^\varepsilon}$ size ACC^0 circuits. On input x of length m, we can set $n \approx 2^{m^{1/k}}$ so that $\ell(n) = m$, use an NP oracle to find the $\mathcal{O}_n$, and output $\mathcal{O}_n(x)$.

An Input and Advice Problem. The above plan sounds nice, but there is a major problem: we now need a much longer advice! Following the Section II-A1, on an input x of length n, we must split $x = x_1x_2$ such that $|x_2| \approx \log T(|x|) \approx \log^k(n)$, use x_1 as advice to specify the "bad input", construct the oracle $\mathcal{O}$ for PCP and finally outputs $\mathcal{O}(x_2)$. In this way, we can only obtain a hardness of $2^{|x_2|^\varepsilon} < n/10$, which becomes a trivial lower bound compared to the input length n.

Solution: A 'Refuter' Algorithm for Theorem I.12. As discussed above, we cannot afford appending x_n to the end of the input as in Section II-A1. Therefore, in order to make sense of the above proposal, we have to *generate the required x_n ourselves*. Our key observation is that, since we are proving lower bounds for E^{NP} anyway, we can also try to use an NP oracle to *algorithmically* find the desired x_n such that $\mathcal{A}_{\mathsf{PCP}}(x_n) \neq L^{\mathsf{hard}}(x_n)$, given n.

To this end, we introduce the concept of a *refuter* $\mathcal{R}$. For our purpose, $\mathcal{R}$ is a deterministic algorithm with access to an NP oracle which takes as input the (code of) a nondeterministic algorithm $\mathcal{A}$, and 1^n for $n \in \mathbb{N}$, with the promise that $\mathcal{A}$ runs in $o(T(n))$ time and guesses at most $n/10$ bits of witness. For all large enough n, $\mathcal{R}$ outputs a string $x_n \in \{0,1\}^n$ such that $\mathcal{A}(x_n) \neq L^{\mathsf{hard}}(x_n)$. We call such an algorithm a *refuter* for L^{hard}, since it can explicitly refute any faster algorithm $\mathcal{A}$ attempting to decide L^{hard}.

How can we construct a refuter $\mathcal{R}$? A natural idea is to enumerate all input strings of length n, then use an NP oracle to find the first $x \in \{0,1\}^n$ such that $\mathcal{A}(x) \neq L^{\mathsf{hard}}(x)$. This algorithm can find the required input x_n correctly since such an x_n exists by Theorem I.12. However, this method is extremely inefficient, having running time $\Omega(2^n)$.

Open Up The Black Box! We observed that the hard language L^{hard} established by Theorem I.12 is quite special. Given its structure, we can indeed design a algorithm which binary-searches over all inputs of length n to find the desired x_n. With a careful analysis of the L^{hard} language, we design a refuter for L^{hard} that runs in time $O(T(n)\cdot\log(2^n)) \le O(n\cdot T(n))$.

Our final E^{NP} algorithm with a SAT oracle works as follows. On an input of y length m, set $n \approx 2^{m^{1/k}}$ so that $\ell(n) = m$, invoke the refuter to find an input $x_n \in \{0,1\}^n$ such that $\mathcal{A}_{\mathsf{PCP}}(x_n) \neq L^{\mathsf{hard}}(x_n)$, then use the SAT oracle to find the lexicographically-first oracle $\mathcal{O}\colon \{0,1\}^m \to \{0,1\}$ such that $\mathsf{VPCP}_{x_n}^{\mathcal{O}}(r) = 1$ for all r. Finally, our algorithm outputs $\mathcal{O}(y)$. The running time can be bounded by $O(T(n)\cdot n) \approx 2^m$ with the help of the SAT oracle. It is not hard to see that the language decided by this E^{NP} algorithm will be almost-everywhere hard for $2^{n^{o(1)}}$-size ACC^0-circuits, which completes the proof.

Finally, we end this subsection by providing some intuitions on how the refuter for Theorem I.12 is constructed.

The A.E. NTIME Hierarchy. Before explaining our refuter, it's instructive to review the proof ideas of Theorem I.12.

Let $\mathcal{A}$ be an $\mathsf{NTIMEGUESS}[o(T(n)), n/10]$ machine. We define a new algorithm $\mathcal{A}'$ based on $\mathcal{A}$ and show that $\mathcal{A}$ fails to compute $\mathcal{A}'$ on all large enough input lengths. Specially, $\mathcal{A}'$ works as follows: On an input x of length n, $\mathcal{A}'$ rejects immediately if $\mathcal{A}$ rejects the witness (the $n/10$-bit prefix of) x on input 0^n. Otherwise, $\mathcal{A}'$ simply outputs $\mathcal{A}(x+1)$. Here $x+1$ denotes the lexicographically next string after x. If there is no such $x+1$ (*i.e.*, $x = 1^n$), $\mathcal{A}'$ just outputs 1.

One can check that $\mathcal{A}'$ runs in $\mathsf{NTIME}[T(n)]$. Now we fix a large enough n, and suppose for the sake of contradiction that $\mathcal{A}(x) = \mathcal{A}'(x)$ for every $x \in \{0,1\}^n$. (That is, $\mathcal{A}$ is a speed up version of $\mathcal{A}'$.) There are two cases now depending on the value of $\mathcal{A}(0^n)$:

1) $\mathcal{A}(0^n) = 1$. Consequently, we also have $\mathcal{A}'(0^n) = 1$. This is only possible if $\mathcal{A}'$ accepts every input of length n, which implies, by the definition of $\mathcal{A}'$, that $\mathcal{A}$ reject every witness on the input 0^n. Hence it follows that $\mathcal{A}(0^n) = 0$ and consequently $\mathcal{A}'(0^n) = 0$ as well, contradiction!
2) $\mathcal{A}(0^n) = 0$. Since now $\mathcal{A}$ rejects every witness on the input 0^n, we have $\mathcal{A}'(0^n) = \mathcal{A}'(0^n+1) = \cdots = \mathcal{A}'(1^n) = 1$ by the definition $\mathcal{A}'$, contradiction to the assumption that $\mathcal{A}'(0^n) = \mathcal{A}(0^n)$.

The above $\mathcal{A}'$ is only hard for $\mathcal{A}$. To design a hard language against every $\mathsf{NTIMEGUESS}[o(T(n)), n/10]$ algorithm, we can add the description of that algorithm as part of input, which only adds a constant overhead. Now, the resulting algorithm $\mathcal{A}_{\mathsf{HARD}}$ interprets the first $\log n$ bits as the code of a $\mathsf{NTIMEGUESS}[o(T(n)), n/10]$ machine, and the rest being the witness mentioned in the definition of $\mathcal{A}'$ above.

Constructing the Refuter. The above proof is nonconstructive (in the sense that it does not tell us on which input $\mathcal{A}$ and $\mathcal{A}'$ differ) since it is a proof by contradiction. Our observation here is that the definition of the algorithm $\mathcal{A}'$ allows us to consider a linear ordering of all inputs of length n (formed lexicographically, string x is followed by $x+1$). Let us focus on the second case above that $\mathcal{A}(0^n) = 0$ (the first case can be handled similarly, check the full version of the paper for details). Since $\mathcal{A}$ rejects every witness on input 0^n, we have $\mathcal{A}'(x) = \mathcal{A}(x+1)$ for every x, except for $\mathcal{A}'(1^n) = 1$.

Consider the following list of outputs $\mathcal{A}(0^n), \mathcal{A}(0^n+1), \ldots, \mathcal{A}(1^n), \mathcal{A}'(1^n)$. Since the first and last outputs differ, one can use a binary search to find two adjacent different elements with $O(\log(2^n)) = O(n)$ queries to the list (check the full version of the paper for details). This is exactly what we want, since $\mathcal{A}(x) \neq \mathcal{A}(x+1)$ means $\mathcal{A}(x) \neq \mathcal{A}'(x)$. Finally, an access to the above list can be simulated by an NP query, and we obtain the desired P^{NP} refuter.

Generalization to Other Lower Bounds. Our refuter framework is general enough that many similar E^{NP} lower bound proofs (based on Williams' algorithmic paradigm) can be adapted to the almost-everywhere setting as well, e.g., the construction of rigid matrices in [43], [45] and the probabilistic degree lower bound in [39]. See the full version of the paper for details.

B. Strong Average-Case Hardness Lower Bounds via a New XOR Lemma

In this subsection, we first explain why it is difficult to prove strong average-case lower bounds for ACC^0, and then show how we get around previous barriers with an improved XOR Lemma based on approximate linear combinations of circuits.

The Hardness Amplification Barrier. The traditional approach to average-case lower bounds is through *hardness amplification*, which ultimately aims to show how worst-case lower bounds imply average-case lower bounds. The key barrier to proving stronger average-case lower bounds for ACC^0 (or even $\mathsf{AC}^0[2]$) is the lack of an appropriate hardness amplification theorem for both classes. It is known that proving such an amplification theorem requires computing *majority* [57], [58]. That is, to establish strong average-case lower bounds for ACC^0, one apparently needs to start with at least a worst-case $\mathsf{MAJORITY} \circ \mathsf{ACC}^0$ lower bound, which seems as hard as proving strong the average-case lower bounds for ACC^0 itself.

In a recent work, Chen and Ren [20] managed to bypass the above barrier and prove a strong average-case lower bounds for NQP against ACC^0, via a sophisticated win-win argument. However, this argument fails to achieve either almost-everywhere separations or sub-exponential hardness due to inherent limitation. Since the knowledge of [20] is not required to understand our new approach, see the full version of this paper for a discussion on the limitations of the techniques of [20].

We show it is possible to remove the win-win argument. By directly applying a new XOR Lemma, we can prove almost-everywhere strong average-case lower bounds of sub-exponential hardness.

Algorithms and Lower Bounds for (Approximate) Linear Sums of $\mathscr{C}$. The key concept behind the new XOR Lemma is that of linear combinations of circuits. Here we recall their definition (from [59], [18], [20]). Let $\mathscr{C}$ be a class of functions from $\{0,1\}^n \to \{0,1\}$. We say $L\colon \{0,1\}^n \to \mathbb{R}$ is a $\mathsf{Sum} \circ \mathscr{C}$ circuit, if $L(x) := \sum_{i=1}^{t} \alpha_i \cdot C_i(x)$, where each $\alpha_i \in \mathbb{R}$ and each $C_i \in \mathscr{C}$. We define the *complexity* of L to be $\max\left(\sum_{i=1}^{t} |\alpha_i|, \sum_{i=1}^{t} \mathsf{SIZE}(C_i)\right)$.

We say $f\colon \{0,1\}^n \to \{0,1\}$ admits a $\widetilde{\mathsf{Sum}}_\delta \circ \mathscr{C}$-circuit, if there is a $\mathsf{Sum} \circ \mathscr{C}$-circuit $L\colon \{0,1\}^n \to \mathbb{R}$

such that $|L(x) - f(x)| \leq \delta$ for all x. We set $\delta = 1/3$ by default when it is omitted. We also use the notation $\widetilde{\mathsf{Sum}} \circ \mathsf{ACC}^0$ to denote the class of languages which can be computed by a $\widetilde{\mathsf{Sum}} \circ \mathsf{ACC}^0$-circuit family of polynomial complexity.

For us, the most important aspect of linear combinations of circuits is that they can **preserve algorithms**. For example, if there are non-trivial algorithms solving #SAT for $\mathscr{C}$, then we can compute $\mathbb{E}_x[L(x)]$ as $\sum_{i=1}^{t} \alpha_i \cdot \mathbb{E}_x[C_i(x)]$, where $\mathbb{E}_x[C_i(x)]$ is computed by solving #SAT for $C_i \in \mathscr{C}$. Hence, if a Boolean function f has a $\widetilde{\mathsf{Sum}}_\delta \circ \mathscr{C}$-circuit L, then

$$\mathop{\mathbb{E}}_x[f(x)] - \mathop{\mathbb{E}}_x[L(x)] \leq \max_x |f(x) - L(x)| \leq \delta. \quad (1)$$

Therefore, we are able to estimate the acceptance probability of f in non-trivial time using non-trivial #SAT algorithms for $\mathscr{C}$. With the above observation, [18] applied the non-trivial #SAT algorithms for ACC^0 in [8] to obtain a non-trivial CAPP algorithm for $\widetilde{\mathsf{Sum}} \circ \mathsf{ACC}^0$, from which they proved E^{NP} cannot be computed by $\widetilde{\mathsf{Sum}} \circ \mathsf{ACC}^0$ circuits of $2^{n^{o(1)}}$ complexity.

Our Contribution: Direct Hardness Amplification With a Non-Standard XOR Lemma: As mentioned earlier, prior win-win arguments fail to achieve almost-everywhere separations. Can one avoid a win-win argument and establish average-case hardness *directly*?

We show this is possible! Starting from a "ℓ_1-approximation lower bound" against $\mathsf{Sum} \circ \mathsf{ACC}^0$-circuits (slightly stronger than ℓ_∞-inapproximability), we show how strong average-case lower bounds can be established directly by applying a new non-standard version of the XOR Lemma (recalled below).

A Non-Standard XOR Lemma: For a Boolean function f and $\delta < 1/2$, if there is no $\mathsf{Sum} \circ \mathscr{C}$-circuit $C\colon \{0,1\}^n \to [0,1]$ of complexity $\mathrm{poly}(s, 1/\delta, n)$ such that $\mathbb{E}_{x \in \{0,1\}^n} |C(x) - f(x)| \leq \delta$, then $f^{\oplus k}$ cannot be $1/2 + 1/s$ approximated by s-size $\mathscr{C}$ circuits, for $k = \Theta\left(\delta^{-1} \log s\right)$.

Note that we only consider circuits C such that $C(x) \in [0,1]$ for all x, which is crucial in our proof. We denote the class of such circuits as $[0,1]\mathsf{Sum} \circ \mathscr{C}$.

ℓ_1 Approximation Bounds Against $\mathsf{Sum} \circ \mathscr{C}$. Building on prior work on $\widetilde{\mathsf{Sum}} \circ \mathscr{C}$-circuits ([18], [20]) with modifications, we show there is a function $f \in \mathsf{E}^{\mathsf{NP}}$ that cannot be approximated by $[0,1]\mathsf{Sum} \circ \mathsf{ACC}^0$ within a small constant ℓ_1 distance. Applying the new XOR Lemma above, we can establish an almost-everywhere strong average-case lower bound for E^{NP} against ACC^0.

The proof is involved, but the key insight is to see that inapproximability lower bound argument in (1) *does not* require the circuit L to approximate f on every point. That is, suppose f is a Boolean function and $L\colon \{0,1\}^n \to \mathbb{R}$ is a $\mathsf{Sum} \circ \mathscr{C}$ circuit satisfying $\mathbb{E}_x |L(x) - f(x)| \leq \delta$. In such a case, we say that f can be ℓ_1-approximated by L within distance δ. Note we still have that

$$\mathop{\mathbb{E}}_x[f(x)] - \mathop{\mathbb{E}}_x[L(x)] \leq \mathop{\mathbb{E}}_x |f(x) - L(x)| \leq \delta.$$

Let us use $\widehat{\mathsf{Sum}}_\delta \circ \mathscr{C}$ to denote the class of Boolean functions which can be ℓ_1-approximated by a family of $\mathsf{Sum} \circ \mathscr{C}$ circuits within distance δ. By the above reasoning, non-trivial CAPP algorithms for $\widehat{\mathsf{Sum}}_\delta \circ \mathscr{C}$ still follow from non-trivial #SAT algorithms for $\mathscr{C}$ circuits! In spirit, this means we should be able to prove lower bounds against $\widehat{\mathsf{Sum}}_\delta \circ \mathsf{ACC}^0$, as we have a non-trivial CAPP algorithm for them (thanks to the #SAT algorithm for ACC^0 of Williams [8]).

Intuition for the new XOR Lemma. Now we sketch how the new XOR Lemma is proved. It is based on a careful examination of Levin's proof of the XOR Lemma [55]. Let $\varepsilon = \varepsilon_k = \frac{1}{2} \cdot (1-\delta)^k$. We will prove the contrapositive, i.e., if $f^{\oplus k}$ can be $(1/2 + \varepsilon)$-approximated by a $\mathscr{C}$-circuit C of size s, then we can construct a $[0,1]\mathsf{Sum} \circ \mathscr{C}$-circuit of complexity $O(s \cdot n \cdot \varepsilon^{-2})$ which ℓ_1-approximates f within error $O(\delta)$.

The proof is by induction on k. For the case $k = 1$, we can take the $[0,1]\mathsf{Sum} \circ \mathscr{C}$-circuit to be the circuit C itself. For the case $k > 1$, we argue as follows. For an input x to $f^{\oplus k}$, we write $x = yz$ such that $|y| = n$ and $|z| = (k-1)n$. For each $y \in \{0,1\}^n$, we consider the quantity:

$$\begin{aligned} T(y) &:= \Pr_z[f^{\oplus k}(y,z) = C(y,z)] && (2)\\ &= \Pr_z[f^{\oplus(k-1)}(z) = f(y) \oplus C(y,z)] && (3)\\ &= \Pr_z[f(y) = C(y,z) \oplus f^{\oplus(k-1)}(z)]. && (4) \end{aligned}$$

That is, $T(y)$ measures (2) how well $C(y,z)$ approximates $f^{\oplus k}$ when the first input is fixed to y; (3) how well $f(y) \oplus C(y,z)$ approximates $f^{\oplus(k-1)}$; (4) how often does $C(y,z) \oplus f^{\oplus(k-1)}(z)$ correctly predict $f(y)$, when z is uniformly random. By (2) and our assumption on C, we have

$$\mathop{\mathbb{E}}_y[T(y)] = \Pr_{y,z}[f^{\oplus k}(y,z) = C(y,z)] \geq 1/2 + \varepsilon.$$

Now there are two cases.

1. Some $T(y)$ is far from $1/2$. Suppose there is a $y \in \{0,1\}^n$ satisfying $|T(y) - 1/2| > \varepsilon/(1-\delta)$. By (3), $f^{\oplus(k-1)}(z)$ can be computed correctly by either $f(y) \oplus C(y,z)$ or by its negation on at least a $1/2 + \varepsilon_k/(1-\delta) = 1/2 + \varepsilon_{k-1}$ fraction of inputs (we treat y as fixed in the circuit). That is, this case can be reduced to the case of $k-1$.

2. All $T(y)$'s are close to $1/2$. This is the more interesting case. Suppose $|T(y) - 1/2| \leq \varepsilon/(1-\delta)$ for all y. Consider the following algorithm $\mathcal{A}$ trying to predict $f(y)$ for every y: take a uniformly random sample $z \in \{0,1\}^{(k-1)\cdot n}$, and output $C(y,z) \oplus f^{\oplus(k-1)}(z)$. For each y, by the interpretation (4), it follows that $\mathcal{A}(y)$ predicts $f(y)$ correctly with probability $T(y)$.

To ease our later discussion, in the following we specify a natural bijection between functions from $\{0,1\}^n \to [0,1]$ and probabilistic functions on $\{0,1\}^n$ with outputs in $\{0,1\}$. For a probabilistic function $\mathcal{F}\colon \{0,1\}^n \to \{0,1\}$, we define $\mathbb{I}_{\mathcal{F}}$ such that $\mathbb{I}_{\mathcal{F}}(x) := \Pr[\mathcal{F}(x) = 1]$. Conversely, for a function $f\colon \{0,1\}^n \to [0,1]$, we define $\mathbb{P}_f$ such that $\mathbb{P}_f(x)$ outputs 1 with probability $f(x)$, and 0 otherwise.

In the following, we define the correlation of two functions $g, h\colon \{0,1\}^n \to [0,1]$ as

$$\mathrm{Cor}(g,h) := \mathop{\mathbb{E}}_{x \leftarrow \{0,1\}^n}[(2g(x)-1)\cdot(2h(x)-1)]. \quad (5)$$

When viewing g, h as two probabilistic functions, the quantity above is exactly

$$\begin{aligned}&\Pr_x[\mathbb{P}_g(x) = \mathbb{P}_h(x)] - \Pr_x[\mathbb{P}_g(x) \neq \mathbb{P}_h(x)]\\ =&2\Pr_x[\mathbb{P}_g(x) = \mathbb{P}_h(x)] - 1.\end{aligned}$$

The following two properties of $\mathcal{A}$ are crucial for us:

1) The algorithm $\mathcal{A}$ has a non-trivial correlation with $f(y)$. More precisely, $\mathrm{Cor}(\mathbb{I}_{\mathcal{A}}, f) > 2\varepsilon$ since $\mathrm{Cor}(\mathbb{I}_{\mathcal{A}}, f) = 2\cdot\Pr_y[\mathcal{A}(y) = f(y)] - 1 = 2\,\mathbb{E}_y[T(y)] - 1 \geq 2\varepsilon$.
2) On each input y, the algorithm $\mathcal{A}$ has little bias on its output. That is, $\mathbb{I}_{\mathcal{A}}(y) \in [1/2 - \varepsilon/(1-\delta), 1/2 + \varepsilon/(1-\delta)]$ for every y.

By a random sampling and applying a Chernoff bound, we can use a Sum of $O((\varepsilon\delta)^{-2}\cdot n)$-many $\mathscr{C}$-circuits to approximate $\mathbb{I}_{\mathcal{A}}(x)$ within error $\varepsilon\delta$ for every input x. We call this new $\mathsf{Sum}\circ\mathscr{C}$-circuit D. The following two properties of D follow from the above two properties of $\mathcal{A}$.

1) The circuit D and the function $f(y)$ have a correlation larger than $2\varepsilon\cdot(1-\delta)$. That is, $\mathrm{Cor}(D,f) > 2\varepsilon\cdot(1-\delta)$.
2) For every input y, $D(y) \in [1/2 - \varepsilon, 1/2 + \varepsilon]$. (Note that $\varepsilon\delta + \varepsilon/(1-\delta) = \varepsilon$.)

Now, observe that both of the bias on every input and the total correlation are all roughly ε, we can finally scale D properly to obtain a $[0,1]\mathsf{Sum}\circ\mathscr{C}$-circuit D', which has a correlation larger than $(1-\delta)$ with f. Formally, we define D' so that

$$D'(y) = \frac{1 + (D(y) - 1/2)\varepsilon^{-1}}{2}.$$

The second property of D implies that D' is a $[0,1]\mathsf{Sum}\circ\mathscr{C}$-circuit. And the first property of D implies that D' has a correlation $(1-\delta)$ with f (due to the definition (5)), which further implies that D' ℓ_1-approximates f with error $O(\delta)$. For the precise calculation, we refer to the full version of the paper.

Acknowledgments

The first author wants to thank Shuichi Hirahara for pointing out that Levin's proof of the XOR Lemma has a close connection with approximate linear sum (which was later pointed out again to the first author by Ronen Shaltiel).

We are grateful to Josh Alman, Hanlin Ren and Roei Tell for detailed comments on an early version of the draft, and helpful discussions.

L. Chen and R. Williams are supported by NSF grants CCF-1909429 and CCF-1741615.

References

[1] M. Ajtai, "Σ^1_1-formulae on finite structures," *Annals of Pure and Applied Logic*, vol. 24, no. 1, pp. 1–48, 1983.

[2] M. L. Furst, J. B. Saxe, and M. Sipser, "Parity, circuits, and the polynomial-time hierarchy," *Mathematical Systems Theory*, vol. 17, no. 1, pp. 13–27, 1984.

[3] A. C. Yao, "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*, 1985, pp. 1–10.

[4] J. Håstad, "Almost optimal lower bounds for small depth circuits," *Advances in Computing Research*, vol. 5, pp. 143–170, 1989.

[5] A. 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*, vol. 41, no. 4, pp. 333–338, 1987.

[6] R. Smolensky, "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*, 1987, pp. 77–82.

[7] S. Arora and B. Barak, *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[8] R. Williams, "Nonuniform ACC circuit lower bounds," *Journal of the ACM (JACM)*, vol. 61, no. 1, p. 2, 2014.

[9] ——, "Improving exhaustive search implies superpolynomial lower bounds," *SIAM J. Comput.*, vol. 42, no. 3, pp. 1218–1244, 2013.

[10] J. I. Seiferas, M. J. Fischer, and A. R. Meyer, "Separating nondeterministic time complexity classes," *J. ACM*, vol. 25, no. 1, pp. 146–167, 1978.

[11] S. Žák, "A Turing machine time hierarchy," *Theoretical Computer Science*, vol. 26, no. 3, pp. 327–333, 1983.

[12] N. Nisan and A. Wigderson, "Hardness vs randomness," *J. Comput. Syst. Sci.*, vol. 49, no. 2, pp. 149–167, 1994.

[13] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems," *J. ACM*, vol. 45, no. 3, pp. 501–555, 1998.

[14] S. Arora and S. Safra, "Probabilistic checking of proofs: A new characterization of NP," *J. ACM*, vol. 45, no. 1, pp. 70–122, 1998.

[15] C. Murray and R. R. Williams, "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*, 2018, pp. 890–901.

[16] J. Håstad, B. Rossman, R. A. Servedio, and L. Tan, "An average-case depth hierarchy theorem for boolean circuits," *J. ACM*, vol. 64, no. 5, pp. 35:1–35:27, 2017.

[17] R. Chen, I. C. Oliveira, and R. Santhanam, "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*, 2018, pp. 317–330.

[18] L. Chen and R. R. Williams, "Stronger Connections Between Circuit Analysis and Circuit Lower Bounds, via PCPs of Proximity," in *34th Computational Complexity Conference (CCC 2019)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 137. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 19:1–19:43.

[19] L. Chen, "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*. IEEE Computer Society, 2019, pp. 1281–1304.

[20] L. Chen and H. Ren, "Strong average-case lower bounds from non-trivial derandomization," in *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, 2020, pp. 1327–1334.

[21] H. Buhrman, L. Fortnow, and R. Santhanam, "Unconditional lower bounds against advice," in *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 5555. Springer, 2009, pp. 195–209.

[22] R. Kannan, "Circuit-size lower bounds and non-reducibility to sparse sets," *Information and Control*, vol. 55, no. 1-3, pp. 40–56, 1982.

[23] N. H. Bshouty, R. Cleve, R. Gavaldà, S. Kannan, and C. Tamon, "Oracles and queries that are sufficient for exact learning," *J. Comput. Syst. Sci.*, vol. 52, no. 3, pp. 421–433, 1996.

[24] J. Köbler and O. Watanabe, "New collapse consequences of NP having small circuits," *SIAM J. Comput.*, vol. 28, no. 1, pp. 311–324, 1998.

[25] J. Cai, "$\mathsf{S}_2^{\mathsf{p}}$ is subset of $\mathsf{ZPP}^{\mathsf{np}}$," *J. Comput. Syst. Sci.*, vol. 73, no. 1, pp. 25–35, 2007.

[26] J. Cai, V. T. Chakaravarthy, L. A. Hemaspaandra, and M. Ogihara, "Competing provers yield improved Karp-Lipton collapse results," *Inf. Comput.*, vol. 198, no. 1, pp. 1–23, 2005.

[27] N. V. Vinodchandran, "A note on the circuit complexity of PP," *Theor. Comput. Sci.*, vol. 347, no. 1-2, pp. 415–418, 2005.

[28] S. Aaronson, "Oracles are subtle but not malicious," in *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, 2006, pp. 340–354.

[29] L. Fortnow, R. Lipton, D. van Melkebeek, and A. Viglas, "Time-space lower bounds for satisfiability," *Journal of the ACM (JACM)*, vol. 52, no. 6, pp. 835–865, 2005.

[30] R. Williams, "Algorithms and resource requirements for fundamental problems," Ph.D. dissertation, Ph. D. Thesis, Carnegie Mellon University, CMU-CS-07-147, 2007.

[31] B. Barak, "A probabilistic-time hierarchy theorem for "slightly non-uniform" algorithms," in *Randomization and Approximation Techniques, 6th International Workshop, RANDOM 2002, Cambridge, MA, USA, September 13-15, 2002, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2483. Springer, 2002, pp. 194–208.

[32] L. Fortnow and R. Santhanam, "Hierarchy theorems for probabilistic polynomial time," in *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, 2004, pp. 316–324.

[33] D. van Melkebeek and K. Pervyshev, "A generic time hierarchy for semantic models with one bit of advice," in *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*. IEEE Computer Society, 2006, pp. 129–144.

[34] V. Kabanets, "Easiness assumptions and hardness tests: Trading time for zero error," *J. Comput. Syst. Sci.*, vol. 63, no. 2, pp. 236–252, 2001.

[35] R. R. Williams, "Natural proofs versus derandomization," *SIAM J. Comput.*, vol. 45, no. 2, pp. 497–529, 2016.

[36] I. C. Oliveira and R. Santhanam, "Pseudodeterministic constructions in subexponential time," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. ACM, 2017, pp. 665–677.

[37] R. Williams, "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*, 2014, pp. 194–202.

[38] N. Rajgopal, R. Santhanam, and S. Srinivasan, "Deterministically counting satisfying assignments for constant-depth circuits with parity gates, with implications for lower bounds," in *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, ser. LIPIcs, vol. 117. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 78:1–78:15.

[39] E. Viola, "New lower bounds for probabilistic degree and AC0 with parity gates," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 27, p. 15, 2020.

[40] L. Fortnow and R. Santhanam, "Robust simulations and significant separations," *Inf. Comput.*, vol. 256, pp. 149–159, 2017.

[41] L. G. Valiant, "Graph-theoretic arguments in low-level complexity," in *Mathematical Foundations of Computer Science 1977, 6th Symposium, Tatranska Lomnica, Czechoslovakia, September 5-9, 1977, Proceedings*, ser. Lecture Notes in Computer Science, vol. 53. Springer, 1977, pp. 162–176.

[42] S. V. Lokam, "Complexity lower bounds using linear algebra," *Foundations and Trends in Theoretical Computer Science*, vol. 4, no. 1-2, pp. 1–155, 2009.

[43] J. Alman and L. Chen, "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*. IEEE Computer Society, 2019, pp. 1034–1055.

[44] A. A. Razborov, "On rigid matrices (in Russian)," 1989.

[45] A. Bhangale, P. Harsha, O. Paradise, and A. Tal, "Rigid matrices from rectangular PCPs," 2020, to appear in FOCS 2020.

[46] R. Smolensky, "On representations by low-degree polynomials," in *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, 1993, pp. 130–138.

[47] E. Viola, "Matching smolensky's correlation bound with majority," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 26, p. 175, 2019.

[48] L. Fortnow and R. Santhanam, "New non-uniform lower bounds for uniform classes," in *31st Conference on Computational Complexity (CCC 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[49] A. Bogdanov, K. Talwar, and A. Wan, "Hard instances for satisfiability and quasi-one-way functions," in *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, 2010, pp. 290–300.

[50] D. Gutfreund, R. Shaltiel, and A. Ta-Shma, "If NP languages are hard on the worst-case, then it is easy to find their hard instances," *Comput. Complex.*, vol. 16, no. 4, pp. 412–441, 2007.

[51] A. Atserias, "Distinguishing SAT from polynomial-size circuits, through black-box queries," in *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, 2006, pp. 88–95.

[52] O. Goldreich, N. Nisan, and A. Wigderson, "On yao's xor-lemma," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 2, no. 50, 1995.

[53] R. Impagliazzo, "Hard-core distributions for somewhat hard problems," in *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*. IEEE Computer Society, 1995, pp. 538–545.

[54] A. R. Klivans, "On the derandomization of constant depth circuits," in *APPROX 2001 and RANDOM 2001, Berkeley, CA, USA, August 18-20, 2001, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2129. Springer, 2001, pp. 249–260.

[55] L. A. Levin, "One-way functions and pseudorandom generators," *Combinatorica*, vol. 7, no. 4, pp. 357–363, 1987.

[56] E. Ben-Sasson and E. Viola, "Short PCPs with projection queries," in *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, 2014, pp. 163–173.

[57] R. Shaltiel and E. Viola, "Hardness amplification proofs require majority," *SIAM J. Comput.*, vol. 39, no. 7, pp. 3122–3154, 2010.

[58] A. Grinberg, R. Shaltiel, and E. Viola, "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*, 2018, pp. 956–966.

[59] R. R. Williams, "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*, 2018, pp. 6:1–6:24.