THEORY OF COMPUTING, Volume 22 (7), 2026, pp. 1–31
www.theoryofcomputing.org

New Lower Bounds for Probabilistic Degree and AC0 with Parity Gates

Emanuele Viola*

Received July 13, 2020; Revised January 13, 2021, January 29, 2026; Published October 11, 2026

Abstract.

We make the first progress on probabilistic-degree lower bounds and correlation bounds for polynomials since the papers by Razborov and Smolensky in the 80's. The bounds hold for computing some function $f : \{0,1\}^n \to \{0,1\}$ in E^{NP}, and include:

(1) $\Omega(n/\log^2 n)$ lower bounds probabilistic degree. This is optimal up to a factor $O(\log^2 n)$. The previous best lower bound was $\Omega(\sqrt{n})$ proved in the 80's by Razborov and Smolensky.

(2) $\exp(\Omega(n/\log^2 n)^{1/(h-1)})$ lower bounds on the size of depth-h $\mathrm{AC}^0[\oplus]$ circuits, for any h. This almost matches the $\exp(\Omega(n^{1/(h-1)}))$ lower bounds for AC^0 by Håstad. The previous best lower bound was $\exp(\Omega(n^{1/(h+1)}))$ by Rajgopal, Santhanam, and Srinivasan (MFCS'18) who recently improved Razborov and Smolensky's $\exp(\Omega(n^{1/(2h-2)}))$ bound.

(3) $(1/2 - (\log^{O(h)} s)/n)$ average-case hardness for size-s depth-h $\mathrm{AC}^0[\oplus]$ circuits under the uniform distribution. The previous best was $(1/2 - (\log^{O(h)} s)/\sqrt{n})$. A concurrent paper by Chen and Ren (STOC'20) obtains an incomparable result.

A prelimiary version of this paper appeared in ECCC as TR20-015 (2021) [77].
*Supported by NSF CCF award 1813930.

ACM Classification: F.2.3

AMS Classification: 68Q17

Key words and phrases: lower bound, probabilistic degree, AC0, PCP, polynomials

 DOI: 10.4086/toc.2026.v022a007

The previous best lower bounds in (1) and (3), mentioned above, held for the Majority function. Each of the new lower bounds in this paper is false for Majority. For (2) the previous best held for E^{NP}.

The proofs build on Williams's "guess-and-SAT" method. For (1) we show how to use a PCP by Ben-Sasson and Viola (ICALP'14) towards probabilistic-degree lower bounds. For (3) we combine a recent method by Alman and Chen (STOC'19, SICOMP 2025) with hardness amplification.

1 Introduction

1.1 History, motivation

Probabilistic degree is a fundamental complexity measure of boolean functions that has been intensely studied since it was introduced by Razborov [57]. Probabilistic degree measures how well a function can be computed by a "random" polynomial, from a suitable distribution. One can consider polynomials over different fields. For simplicity in this paper we focus on the field $\mathbb{F}_2$ with two elements, and we only mention here that the results can be extended to $\mathbb{F}_p$ for prime p.

Definition 1.1. The *ϵ-error degree* of a function $f : \{0,1\}^n \to \{0,1\}$ is the minimum d such that there is a distribution D on polynomials over $\mathbb{F}_2$ of degree $\le d$ such that $\mathbb{P}[D(x) \ne f(x)] \le \epsilon$ for every $x \in \{0,1\}^n$.

The parameter ϵ ranges from 0 to at most $1/2$, because outputting a random bit yields error $1/2$. In this article we will consider both the setting when ϵ is close to 0, and the setting when ϵ approaches $1/2$. The interplay between the two settings will be important.

Razborov and Smolensky [57, 62, 63] proved that the Majority function on n bits has constant-error degree $\Omega(\sqrt{n})$.[1] Better lower bounds have not been obtained despite much research. One difficulty is that many classes of functions do have $O(\sqrt{n})$ probabilistic degree. Alman and Williams show [5, Theorem 1.2] that any symmetric function on n bits has ϵ-error polynomials of degree $O(\sqrt{n\log(1/\epsilon)})$. Earlier, Srinivasan [64] proved a slightly weaker bound. The recent article [65] gives a nearly tight characterization of the probabilistic degree of symmetric functions. Srinivasan [64, Theorem 12] also proved that threshold functions (with arbitrary weights) have ϵ-error polynomials of degree $\sqrt{n}(\log n \cdot \log 1/\epsilon)^{O(1)}$. Golovnev, Kulikov, and Williams [32] conjecture that DeMorgan formulas of size s have probabilistic degree $O(\sqrt{s})$.

Razborov and Smolensky actually proved a tradeoff between the error and the degree. For some constant c and any degree $\le c\sqrt{n}$, we can get constant error as mentioned above, but for lower degree we can push the error to be close to $1/2$. Formally, the tradeoff gives that for any degree d the error is

$$\epsilon \ge 1/2 - O(d/\sqrt{n})\,. \tag{1.1}$$

[1] In this paper, the notation O and Ω implies positive absolute constants.

This tradeoff is tight for Majority [76], and is the best available for an explicit function, for any degree $d \geq \log_2 n$. In particular, conceivably every explicit function (say in P or even NP) has $(1/2 - 1/\sqrt{n})$-error degree $\log_2 n$. For small degrees, $d < \log_2 n$, the iterated Cauchy–Schwarz argument of [9] gives stronger error bounds. See [70, 78, 71] for direct arguments. However in this paper we are only concerned with $d \geq \log_2 n$.

We note that by Yao's duality principle [87], the claim that a function h has ϵ-error degree d is equivalent to the claim that there exists a distribution over the inputs such that any fixed degree-d polynomial fails to compute h with probability ϵ over inputs drawn from that distribution. In the special case where the latter distribution is uniform we call the function h *ϵ-hard* for polynomials of degree d.

Definition 1.2. Let $h : \{0,1\}^n \to \{0,1\}$ be a function, and let F be a set of boolean functions on n bits. We say that h is ϵ-hard for F if for every $f \in F$ we have $\mathbb{P}_x[f(x) \neq h(x)] \geq \epsilon$, where x is uniform in $\{0,1\}^n$.

We remark that the state of lower bounds is the same for probabilistic degree and hardness against polynomials; no better lower bounds are known if we allow a non-uniform distribution. Hardness results for polynomials are surveyed in [71, Chapter 1].

Probabilistic degree is linked to a wide range of other problems. In particular, progress on probabilistic degree is *necessary* for progress on circuit lower bounds, rigidity, and communication complexity. For some of these problems, it is also *sufficient*. We now elaborate on this. We begin by observing that, with regards to lower bounds, improving the tradeoff (1.1) for small d is easier than improving it for large d. Specifically, if a function f has $(1/2 - \epsilon)$-error degree d then it has constant-error degree $O(d/\epsilon)$ (see Lemma 3.4).

1.1.1 Circuit lower bounds for $\mathrm{AC}^0[\oplus]$

The circuits $\mathrm{AC}^0[\oplus]$ consist of And, Or, and Parity gates with unbounded fan-in, and have constant depth. This class lies at the frontier of the techniques in circuit lower bounds. The work by Razborov and Smolensky used probabilistic-degree lower bounds to obtain lower bounds of $\exp(\Omega(n^{1/(2h-2)}))$ on the size of $\mathrm{AC}^0[\oplus]$ circuits of depth h. This bound was recently improved to $\exp(\Omega(n^{1/(h+1)}))$ by Rajgopal, Santhanam, and Srinivasan [53], see also [51]. Still, these lower bounds are lower than the $\exp(\Omega(n^{1/(h-1)}))$ that can be obtained for AC^0 via Håstad's switching lemma [35].

The gap between AC^0 and $\mathrm{AC}^0[\oplus]$ lower bounds is closely related to the state of probabilistic-degree lower bounds: If we could prove linear lower bounds on constant-error degree, that is eliminate the square root in Equation (1.1), then we would obtain lower bounds for $\mathrm{AC}^0[\oplus]$ that are as strong as the AC^0 ones, using the same reasoning as in the proof of Corollary 1.5.

Also, we don't have strong average-case lower bounds. It is consistent with our knowledge that for every explicit function there is a polynomial-size $\mathrm{AC}^0[\oplus]$ circuit computing it correctly on $1/2 + 1/\sqrt{n}$ fraction of the inputs $\{0,1\}^n$. Note that degree-d polynomials are a special case of depth-2 $\mathrm{AC}^0[\oplus]$ circuits of size $n^{O(d)}$, so this is similar (but not equivalent to) the fact that we don't have such strong average-case lower bounds for degree $\log n$.

1.1.2 Rigidity

Nearly half a century ago Valiant [68] asked to construct matrices that cannot be approximated by matrices of low rank. We can say that a $\{0,1\}^n \times \{0,1\}^n$ matrix M is ϵ-error rigid for rank r if for any rank-r matrix F we have $\mathbb{P}_{i,j}[M_{i,j} \neq F_{i,j}] \geq \epsilon$, where i and j are uniform. Viewing matrices as truth-tables of functions, this is the same as being ϵ-hard for rank-r functions. Again this can be considered over different fields and we focus on $\mathbb{F}_2$ in this article. The rigidity question is motivated by a host of applications ranging from circuit lower bounds to communication complexity. Non-explicitly, there exist matrices which are constant-error rigid for $r = \Omega(2^n)$. But despite intense research the available constructions are far from this. Recently, an exciting paper by Alman and Chen [4] constructed in the complexity class $\mathrm{Time}(2^{O(n)})^{\mathrm{NP}} = \mathrm{E}^{\mathrm{NP}}$ matrices M which are constant-error rigid for rank $2^{n^{1/4-\epsilon}}$, for any $\epsilon > 0$. They also gave a conditional construction for higher rank $2^{n^{1-\alpha}}$ for all $\alpha > 0$. For a discussion of previous rigidity bounds and applications we refer the reader to [4] and Lokam's survey [44].

Probabilistic-degree lower bounds stand in the way of further progress. Indeed, as pointed out by Servedio and Viola [60], sparse polynomials are a special case of low-rank matrices. Specifically, write the truth-table of a polynomial on $2n$ variables as a $2^n \times 2^n$ matrix (partitioning the variables in half arbitrarily). Now observe that each monomial is a rank one matrix. Hence a polynomial with s monomials yields a rank s matrix. Because a polynomial of degree $\sqrt{2n}$ has $\leq n^{O(\sqrt{n})}$ monomials, we see that better probabilistic-degree lower bounds are necessary for improving the results in [4] to any rank $2^{\omega(\sqrt{n}\log n)}$. (We can remove the log factor when considering *sparsity*, see [60] for discussion. Here we focus on *degree* for simplicity.)

1.1.3 Communication complexity

In the influential number-on-forehead communication model introduced by Chandra, Furst, and Lipton [18], k parties collaboratively wish to compute a function on n bits. The bits are divided in k blocks (figuratively corresponding to the foreheads) and the twist is that Party i knows all blocks except the i-th. For this model, we have no lower bounds when $k \geq \log n$ (for $k < \log n$ they were first proved in [9], see [27, 56, 78] for expositions). For background on communication complexity see the books [43, 55].

Interestingly, it turns out that proving communication lower bounds for more parties requires better probabilistic-degree lower bounds. We include in this paper a stronger quantitative version of this connection from [73], see Theorem 6.1, which already kicks in for $k = O(\log^2 n)$ parties exchanging $O(\log^3 n)$ bits of communication. In particular, improving the tradeoff (1.1) is necessary to rule out such protocols.

1.1.4 Other circuit classes

Many papers in the literature study $\mathrm{AC}^0[\oplus]$ circuits augmented with other types of gates. In fact, average-case lower bounds for $\mathrm{AC}^0[\oplus]$ are closely related to worst-case lower bounds for the circuit class $\mathrm{Maj} \circ \mathrm{AC}^0[\oplus]$ of such circuits with a majority gate at the output [34]. Correspondingly, we only have lower bounds when the fan-in of the majority gate is $t < n/\mathrm{poly}\log(n)$. Such a lower bound follows simply by approximating the entire circuit by a polynomial of degree $\sqrt{n}/\operatorname{poly}\log(n)$, and then using the probabilistic degree lower bound. For the approximation, we proceed by approximating Majority using [5] and each $\mathrm{AC}^0[\oplus]$ circuit using [57], and then compose the polynomials.

Alman and Chen [4] consider the class of $\mathrm{AC}^0[\oplus] \circ \mathrm{Ltf}_t \circ \mathrm{AC}^0[\oplus] \circ \mathrm{Ltf}$ circuits, where Ltf are threshold functions with arbitrary weights, and prove that the fan-in t of the middle Ltf gates satisfies $t \geq n/\log^{O(h)} s$ for depth-h and size-s circuits that compute some function in E^{NP}. A lower bound for this class could not just rely on probabilistic degree, because the Ltf gates at the input already require $\sqrt{n}$ probabilistic degree.

Related classes of circuits were studied by Alman, Chan, and Williams ($\mathrm{ACC}^0 \circ \mathrm{Ltf}_{n^{2-\Omega(1)}} \circ \mathrm{Ltf}$, [3, Corollary 1.1]), and Tamaki (depth-2 circuits with $n^2/\operatorname{poly}\log(n)$ gates, where each gate can be any symmetric function or an Ltf [67]).

1.1.5 Pseudorandom generators

Another important motivation for improving the tradeoff (1.1) and average-case lower bounds for $\mathrm{AC}^0[\oplus]$ is the construction of *pseudorandom generators.* Nisan's landmark paper [49] showed how to use a $(1/2-\epsilon)$-hard function to obtain approximately $1/\epsilon$ bits of pseudorandomness. Hence, a suitable improvement to the tradeoff – for the specific case of uniform distributions – would yield better generators for low-degree polynomials [16, 46, 72], a long-standing problem for which several new approaches have recently been proposed [20, 21, 19, 76]. Nisan's approach yields pseudorandom generators with small error, in which case improving probabilistic degree lower bounds is known to be necessary [72]. The situation with constant-error generators is less clear, cf. [74]. We also remark that other pseudorandom-generator constructions that have been proposed do not go through for classes such as $\mathrm{AC}^0[\oplus]$, see [61, 33, 74].

We consider it worthwhile to try to identify an open problem which is as simple as possible while at the same time being a bottleneck for as wide a range of complexity lower bounds as possible. The above discussion suggests that probabilistic degree may be a good candidate for this.

1.2 Our results

In this paper we show that E^{NP} requires nearly linear constant-error degree, improving on the $\Omega(\sqrt{n})$ lower bounds established by Razborov and Smolensky [57, 62, 63] for the majority function.

Theorem 1.3. There is a function $f : \{0,1\}^n \to \{0,1\}$ in E^{NP} with 1/3-error degree $\Omega(n/\log^2 n)$, for infinitely many n.

The bound in Theorem 1.3 is tight up to a factor of $O(\log^2 n)$ because every function on n bits has polynomials of degree n with error 0. We note that, as mentioned earlier, the bound in Theorem 1.3 is false for any symmetric or threshold function [64, 5].

Theorem 1.3 has several corollaries stated next. First, as mentioned earlier, it actually improves the tradeoff (1.1) for every setting of parameters (and not just constant error).

Corollary 1.4. The function in Theorem 1.3 has $(1/2-\epsilon)$-error degree $\Omega(\epsilon n/\log^2 n)$, for infinitely many n and any ϵ.

So in particular we can set $\epsilon = (\log^4 n)/n$ and still get degree $\Omega(\log^2 n)$ while from the tradeoff (1.1) we could not set $1/2-\epsilon$ larger than $1/2-1/\sqrt{n}$. Error larger than $1/2-1/\sqrt{n}$ was not even available for polynomials of degree $\log n$.

Then we improve the size lower bounds for $\mathrm{AC}^0[\oplus]$.

Corollary 1.5. Any depth-h $\mathrm{AC}^0[\oplus]$ circuit computing f in Theorem 1.3 has size

$$\geq \exp(\Omega(n/\log^2 n)^{1/(h-1)}),$$

for infinitely many n.

This improves on the $\exp(\Omega(n^{1/(h+1)}))$ lower bounds by Rajgopal, Santhanam, and Srinivasan [53]. The bound in Corollary 1.5 is false for any symmetric function [51], and it almost matches the $\exp(\Omega(n^{1/(h-1)}))$ lower bounds that are available for AC^0 via switching lemmas [35].

Next we prove a general lower bound for $\mathrm{AC}^0[\oplus]$ circuits suitably augmented with gates computing threshold or arbitrary symmetric functions. We show that we can allow such gates as long as the product of their fan-in's along any root-leaf path in the circuit is at most $n^2/\operatorname{poly}\log(n)$. First we formally define the class.

Definition 1.6. The class t-SoT-$\mathrm{AC}^0[\oplus]$ consists of circuits made of And, Or, Parity, Sym, and Ltf gates such that the product of the fan-in of the Sym and Ltf gates along any root-leaf path in the circuit is at most t. Here each Sym gate computes an arbitrary symmetric function, and each Ltf gate computes a threshold function with arbitrary weights.

Corollary 1.7. Any t-SoT-$\mathrm{AC}^0[\oplus]$ circuit of depth h and size s computing the function f in Theorem 1.3 requires $t \geq n^2/\log^{O(h)} s$, for infinitely many n.

In particular, $\mathrm{Maj}_t \circ \mathrm{AC}^0[\oplus]$ circuits of size s and depth h have $t \geq n^2/\log^{O(h)} s$, improving on the previous best of $t \geq n/\log^{O(h)} s$ which follows from Razborov [57] and Smolensky [62, 63] and the probabilistic degree of Majority [64, 5].

Also, $\mathrm{AC}^0[\oplus] \circ \mathrm{Ltf}_t \circ \mathrm{AC}^0[\oplus] \circ \mathrm{Ltf}$ circuits of size s and depth h have $t \geq n/\log^{O(h)} s$. The previous best was $t \geq \sqrt{n}/\log^{O(h)} s$ recently proved by Alman and Chen [4]. Our proof of this result simply approximates the circuit and then invokes the probabilistic degree lower bound. By contrast Alman and Chen rely on a combination of degree and rank arguments.

1.2.1 Uniform distribution

Corollary 1.4 implies that for any distribution C on polynomial-size constant-depth $\mathrm{AC}^0[\oplus]$ circuits one has $\mathbb{P}_C[C(x) \neq f(x)] \geq 1/2 - \operatorname{poly}\log(n)/n$ for some fixed input x. This is because as mentioned earlier the functions computable by such circuits of polynomial size have poly-logarithmic probabilistic degree [57]. As mentioned earlier, Yao's duality principle [87] then guarantees hardness under *some* distribution, not guaranteed to be uniform. With a different argument, we show how to strengthen this to hold under the uniform distribution.

Theorem 1.8. Let h be an integer and s a monotone increasing function satisfying $s(\Omega(n\log n)) < 2^{O(\log^2 s(n))}$ (such as $s(n) = 2^{n^\alpha}$ for any positive α).[2] Then there is a function $f : \{0,1\}^n \to \{0,1\}$ in E^{NP} that is $(1/2 - (\log^{O(h)} s)/n)$-hard for size-$s$ depth-h $\mathrm{AC}^0[\oplus]$ circuits, for infinitely many n.

Again, the previous best was hardness $1/2 - (\log^{O(h)} s)/\sqrt{n}$, under any distribution. Theorem 1.8 gets us closer to a correlation bound that could be used to obtain new pseudorandom generators via Nisan's method [49].

1.3 Techniques

The results in this paper rely on a masterful method by Williams [81] which we call the "guess-and-SAT" method. This method has been used to prove several lower bounds for computing functions in E^{NP}, NEXP, NQP, and related classes; lower bounds which we do not know how prove by other means. Researchers have established lower bounds against several non-uniform circuit classes starting with Williams's ACC^0 lower bound [84], see [80, 82, 83, 3, 67, 22, 26, 48, 53, 4, 79, 24].

To explain the proof of Theorem 1.3 let us quickly review the method. Suppose we aim to prove a lower bound for computing a function in E^{NP} by a circuit class CKT. We begin by considering a language L in non-deterministic time 2^n which cannot be solved in non-deterministic time $o(2^n)$. Such a language exists by the non-deterministic time-hierarchy theorem. We note that this hierarchy only holds for infinitely many input lengths, as opposed to all sufficiently large input lengths, which is why all these lower bounds, including the ones in this paper, only hold for infinitely many input lengths.

We reduce L to 3SAT with about 2^n clauses (and variables) using the Cook-Levin theorem. This is done in a specific way – using *indexing circuits* – which is clarified below. We then consider the function f that on input x and i computes the i bit of a canonical satisfying assignment corresponding to x, if it is the case that $x \in L$. One can show $f \in \mathrm{E}^{\mathrm{NP}}$. Towards a contradiction we assume that f has a small circuit $C \in$ CKT. The algorithm for L *guesses* C and then combines it with the *indexing circuit* that given z of about n bits outputs the variables in clause z of the 3CNF (and with a post-process circuit which possibly negates the outcomes, and outputs Or). This gives a new circuit C' such that it suffices to know if C' accepts every input z to decide membership in L. If this can be done in time $o(2^n)$, we contradict the hierarchy theorem.

Actually, the original proof of the ACC^0 lower bound does not work this way, due to the lack of reductions to 3SAT with ACC^0 indexing circuits. This modular version of the method has been obtained in [39].

1.3.1 Using PCP

In the original paper [81] it was already suggested that rather than reducing to 3SAT we can reduce to gap-3SAT, or in other words use *probabilistically checkable proofs* (PCP). In a PCP we

[2] When an inequality involves multiple occurrences of the O and Ω asymptotic notation, we interpret each such symbol as a positive constant, existentially quantified. For instance, for positive-valued functions f, g and monotone increasing positive-valued functions s, t, the inequality $s(\Omega(f(n))) \le t(O(g(n)))$ means $(\exists c_1, c_2 > 0)(s(c_1 f(n)) \le t(c_2 g(n))$ for all sufficiently large n.

refer to the input z of about n bits as the *randomness*.

Again, existing PCPs did not come equipped with an indexing circuit that was efficient enough for a modular version of the method as above. This was remedied by Ben-Sasson and Viola in [14] building on [12, 13]. Their PCP is critical for this work. In this PCP, the indexing circuit simply computes *projections* (a.k.a. 1-local functions). In other words, on input the randomness z, the PCP queries the proof at locations $q_1, q_2, \ldots, q_t$ where each q_i is about n bits, and each bit of q_i depends on only one bit of z.

Projections work well for probabilistic degree, because they do not increase degree. However, the PCP makes a polynomial number t of queries which are then post-processed by a 3CNF ϕ. It is not known if a similar PCP with only a constant number of queries exists, see discussion in [14]. This 3CNF ϕ on a polynomial number of variables is problematic, because it may not have small degree. (Note that we cannot approximate it by a polynomial, as we could get a bad approximation which accepts wrong proofs, and we would have no way of detecting it.) Another issue is that we are trying to prove a lower bound against a *distribution* on polynomials. Typical instantiations of the method work against *fixed* circuits.

1.3.2 Our approach

To go around these issues we begin by modifying slightly the PCP in [14]. Rather than making a polynomial number of queries and applying the 3CNF ϕ to their answers, the modification simply picks a random clause i in ϕ, queries its three variables, negates the outputs as needed, and outputs Or.

This PCP will still accept correct proofs with probability 1, because the 3CNF evaluates to 1 and so all clauses are true.

If on the other hand the proof is incorrect, by the soundness of the PCP [14] ϕ evaluates to 0 on at least say a 0.5 fraction of the choices for the randomness z. For each such z, at least one of the clauses is not satisfied. Because the number of clauses is a polynomial, we have at least a polynomial probability, over the choice of i, to pick a false clause. Hence, over the choice of both z and i, this PCP will accept an incorrect proof with probability $\le 1 - 0.5/n^b$ for some fixed b.

The gap between the accept and reject probabilities is small. However, it is going to be large enough for us. Recall that, towards a contradiction, we are assuming that the proof of the PCP is going to be computed by a low-degree polynomial. In fact, by a distribution on polynomials which errs with constant probability. Now an important point is that we are allowed to *reduce this error*. This is possible because we have a distribution on polynomials that has an advantage on every fixed input. So we can just sample many times from the distribution and compute majority to drive the error down exponentially. Specifically, by sampling $c \log n$ many times, for a constant c depending on b, we can drive the error to $\epsilon < 0.5/n^b$.

At this point by an averaging argument it is possible to fix a polynomial p that maintains a gap between the accept and reject probabilities of the PCP. Hence, for inputs in the language L, there exists a proof given by a polynomial p such that the PCP will still accept with probability $> 1 - 0.5/n^b$, while for inputs not in L, the PCP will accept any proof with probability $< 1 - 0.5/n^b$.

The algorithm for L proceeds by guessing the polynomial p and combining it with the PCP. So if we can give an algorithm running in time $o(2^n)$ to compute the acceptance probability of this PCP exactly, we derive the desired contradiction.

This acceptance probability cannot yet be written as the number of satisfying assignments to a low-degree polynomial. This is because the map from i to the clause is not known to be of low degree. However, there are only a polynomial number of choices for i. So we simply enumerate over all of them. For a fixed i, we can write the acceptance probability of the PCP as the number of satisfying assignments to a low-degree polynomial in z.

Summarizing, we have reduced proving probabilistic-degree lower bounds to computing the number of satisfying assignments to a polynomial.

To finish the proof of Theorem 1.3, we need to compute the number of satisfying assignments to a degree-d polynomial in n variables in time $2^{n-\Omega(n/d)+O(\log n)}$. Such an algorithm was recently obtained by Williams: it is a special case of Theorem 26 in [85]. [45] obtain running time $2^{n-\Omega(n/d)+o(n)}$ which is not quite sufficient for our purposes. However the results in these papers are more general and so the proofs are somewhat involved. We give a streamlined presentation of the algorithm that we need, using a fairly straightforward combination of techniques that have been used in the corresponding algorithms for ACC^0, *Orthogonal Vectors*, and *systems of polynomial equations* [84, 17, 45]. This concludes the overview of the proof of Theorem 1.3.

The corollaries then follow using known simulations of circuit classes by probabilistic polynomials.

1.3.3 Proof of Theorem 1.8

To prove the lower bound under the uniform distribution in Theorem 1.8 we use a different argument. Here our starting point is a *conditional* result by Alman and Chen [4]. Under the assumption that non-deterministic quasi-polynomial time has polynomial-size circuits (i. e., $NQP \subseteq P/\mathrm{poly}$) they show how to construct $2^n \times 2^n$ matrices which are constant-error rigid for rank $2^{n^{1-\alpha}}$ for any $\alpha > 0$. Recall that (the functions whose truth tables are) such matrices are also $\Omega(1)$-hard for polynomials of degree $n^{1-\alpha-o(1)}$.

In this paper we also give a streamlined exposition of their result for the case of polynomials, achieving slightly better parameters than what claimed in [4] (see Section 5).

At the high level, Theorem 1.8 is established by combining this result by Alman and Chen with *hardness amplification*, a technique to transform hard functions into harder functions. For background we refer the reader to Chapter 17 "Hardness Amplification and Error Correcting Codes" in the textbook [7], and to the discussion in [61].

However, the use of hardness amplification is not straightforward. One issue is that hardness amplification in general cannot be used for classes such as $\mathrm{AC}^0[\oplus]$, see [33] and the discussion in [61, 69]. Our proof will use hardness amplification *twice*, for rather different purposes. Let us give an overview. Towards a contradiction let us assume that every function in E^{NP} has $\mathrm{AC}^0[\oplus]$ circuits of size n^c which compute the function correctly on a $1/2 + 1/n^{0.9}$ fraction of the inputs. By worst-case to average-case hardness amplification [8, 38, 66], this means that E^{NP} has small *circuits*, not necessarily of constant depth. However, this is sufficient to satisfy the assumption in the result by Alman and Chen, which then yields a function which is $\Omega(1)$-hard for polynomials of degree $n^{1-\alpha-o(1)}$, for any α. In particular, this function is $\Omega(1)$-hard for $\mathrm{MAJ}_{n^{2-2\alpha-o(1)}} \circ \mathrm{AC}^0[\oplus]$

circuits of size $n^{c'}$. An important point is that here we can set $c' > c$, which is necessary for the final contradiction. Now in our second application of hardness amplification we use a fine analysis of Yao's famous XOR lemma [31] which follows from Impagliazzo's theory of hard-core sets [38] and essentially appears in a beautiful paper by Klivans [41]. The analysis essentially says that if a function f is $\Omega(1)$-hard for $\mathrm{MAJ}_{t^2} \circ C$ circuits then XOR of $O(\log t)$ independent copies of f is $(1/2 - 1/t)$-hard for circuits C. Applying this analysis we produce another function in E^{NP} which violates our initial assumption and concludes the proof.

We note that these uses of hardness amplification change the input length, and so we require some mild conditions on the size function $s = s(n)$ to finish the argument. (One can actually relax the conditions at the price of having a more complicated statement.)

1.3.4 Comparison with rigidity lower bound in [4]

We mentioned earlier that Alman and Chen [4] gave an unconditional construction of $2^n \times 2^n$ matrices which are constant-error rigid for rank $2^{n^{1/4-\epsilon}}$. This result does not imply new probabilistic-degree lower bounds. For the latter, we would need to obtain rank larger than $2^{\sqrt{n}}$.

It is natural to ask whether the approach in [4] can yield new probabilistic-degree lower bounds. We now explain what the problem is and why rank $2^{\sqrt{n}}$ is in fact a natural barrier for their approach. [4]'s argument uses PCPs twice. The first application is similar to what we described above. The second application, following [25], is a PCP of proximity that helps in computing the acceptance probability of the first PCP.

The critical point is this: when using this approach to prove a lower bound against a set of functions (for example polynomials of degree d) *this second PCP is applied to evaluate a circuit which contains, and so is at least as big as, one of these functions*. This makes the proof length of this PCP at least as long as the description of the functions. In the case of polynomials of degree d, this would be at least $n^d \geq 2^d$. In particular, the randomness used by the second verifier has length u at least d. Now, the algorithm in [4] involves enumerating over all 2^u choices for this randomness, and running for each choice an algorithm that counts the number of satisfying assignments. For the latter problem, there are algorithms running in time $2^{n-\Omega(n/d)}$ (see Section 2.2). The overall running time accounting for going over the 2^u choices and running the counting algorithm on each is then $2^d \cdot 2^{n-\Omega(n/d)}$. However, this is only better than 2^n when $d \leq \sqrt{n}$.

The paper [4] also contains an argument which they call bootstrapping. As discussed in [4], bootstrapping runs into difficulties when the rank is larger than $\sqrt{n}$; the same difficulties arise for degree larger than $\sqrt{n}$.

1.3.5 Comparison with the work by Rajgopal, Santhanam, and Srinivasan [53]

As mentioned earlier, Rajgopal, Santhanam, and Srinivasan [53] recently proved $\exp(\Omega(n^{1/(h+1)}))$ lower bounds against depth-h $\mathrm{AC}^0[\oplus]$ circuits. Their proof does not rely on the PCP from [14], instead it uses a reduction to 3SAT from the same paper which refines [39] (the latter paper suffices if we replace $h+1$ with $h+2$ in the bound). The use of [14] is straightforward while the bulk of the proof in [53] lies in obtaining new satisfiability algorithms for $\mathrm{AC}^0[\oplus]$ circuits. It is interesting to note that the balance in our proof of Corollary 1.5 is opposite. We rely on the PCP from [14] to establish a new reduction from probabilistic-degree lower bounds to computing the number of satisfying assignments to a polynomial, while the algorithm for the latter task is fairly straightforward.

1.4 Towards probabilistic rank

As mentioned earlier, probabilistic degree is a special case of rigidity. The notion of rigidity is in fact the same as that of hardness (Definition 1.2) for low-rank matrices. Razborov introduced [58] a variant of rigidity which is to rigidity what probabilistic degree is to hardness for low-degree polynomials: we have a distribution on low-rank matrices that on every entry has the correct value except with probability ϵ. By Yao's duality principle, this is the same as correlating with respect to *every* distribution. Following the literature we call this notion *probabilistic rank*. Probabilistic rank has been studied by several works including [86, 6]. This notion is particularly interesting because lower bounds for probabilistic rank, besides being a prerequisite for rigidity lower bounds, also suffice for some of the applications of rigidity lower bounds. For example they suffice for communication lower bounds, see [86], and for $\mathrm{AC}^0[\oplus] \circ \mathrm{Ltf}_t \circ \mathrm{AC}^0[\oplus] \circ \mathrm{Ltf}$ lower bounds, see [4, 6]. For an application to circuits operating on matrices see Jukna's book [40], Chapter 12.8. With a simple proof, in Section 7 we also show that sufficiently strong lower bounds on probabilistic rank imply new *data-structure* lower bounds, a connection which as far as we know is new.

It is natural to ask if Theorem 1.3 can be extended to bound probabilistic rank as well. From our approach it follows that *either* a strengthening of the PCP [14] *or* a strengthening of the algorithms for counting orthogonal vectors [1, 17] would give improved lower bounds for probabilistic rank.

On the PCP side we would need a PCP that on input randomness z makes queries of the type $z \oplus a_i$ where a_i are fixed strings, and $\oplus$ is bit-wise XOR. One of the components of the PCP [14] does have this form, but another (the PCP of proximity for Reed Solomon codes, which remains the most intricate part of the proof) does permute bits.

On the side of orthogonal vectors, it would suffice to solve the following problem.

Problem 1.9. Given six functions $f_1', f_1'', f_2', f_2'', f_3', f_3'' : \{0,1\}^n \to \{0,1\}^r$ and six subsets $s_1', s_1'', s_2', s_2'', s_3', s_3'' \subseteq \{1, 2, \ldots, 2n\}$ of size n each, compute

$$\mathbb{P}_{z \in \{0,1\}^{2n}} \left[\sum_{i=1}^{3} \langle f_i'(z|_{s_i'}), f_i''(z|_{s_i''}) \rangle = 1 \mod 2 \right]$$

in time $2^{2n - \Omega(n/\log r)}$ when $r \leq 2^{\alpha n}$ for a small enough $\alpha > 0$. Here $z|_s$ denotes the n bits of z indexed by s.

The connection to rank is that a rank-r $2^n \times 2^n$ matrix can be written as the truth-table of a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^r$ defined as $g(x, y) := \langle f'(x), f''(y) \rangle$ where $\langle , \rangle$ is inner product. Thus Problem 1.9 considers the sum of three low-rank functions of *different projections of* $2n$ *bits.* As pointed out in [4], when the projections are the same this can be reduced to a single matrix, and for that one can apply algorithms for *orthogonal vectors* [17]. It is not clear what happens for different projections, which could arise in the PCP [14].

Here's a special case that we can't yet solve. Divide the $2n$ bits z into 4 blocks $z = (x, y, v, w)$ of $n/2$ bits each. Can one compute

$$\mathbb{P}_{z\in\{0,1\}^{2n}}[\langle f(x,y), f(v,w)\rangle + \langle f(x,v), f(y,w)\rangle + \langle f(x,w), f(y,v)\rangle = 1 \mod 2]$$

in time $2^{2n-\Omega(n/\log r)}$?

1.5 Concurrent and subsequent work

1.5.1 Concurrent work

A concurrent paper by Chen and Ren [24] obtains a result incomparable to Theorem 1.8. They obtain functions in E^{NP} that are $(1/2 - 1/f(n))$-hard for $\mathrm{AC}^0[\oplus]$ (or even ACC^0) circuits of size $f(n)$. They implicitly prove this for up to *sub-half-exponential* functions $f(n)$, and the techniques in [24] do not apply to circuits of larger size. In particular, for exponential-size circuits [24] do not obtain better hardness than the classic $1/2 - 1/\sqrt{n}$ result. Our techniques apply even to exponential-size circuits, though our improvement on hardness is smaller. For example, Theorem 1.8 yields hardness $1/2 - 1/n^{0.99}$ for circuits of size $2^{n^{\Omega(1)}}$. This state of affairs raises the exciting possibility of making both the hardness and the size parameter exponential.

1.5.2 Subsequent work

The paper [23] simultaneously improves the parameters in [24] and in Theorem 1.8 (improving the function $f(n)$ above to exponential). It also shows how to make the lower bounds "almost everywhere" instead of "infinitely often." The paper [15] constructs rectangular (and smooth) PCPs and uses them to show the existence of rigid matrices in E^{NP}. The PCP we asked about above are in particular rectangular. The results in this paper are obtained using as black-box the PCP [14]. Both [15] and [14] in turn rely on PCPs in [12]. Quantitatively, [15] prove hardness $1/2 - \Omega(1)$ against functions whose corresponding $2^n \times 2^n$ matrix has rank $2^{n/O(\log n)}$. Recall (see for example the argument in Section 1.1.2) that degree-d polynomials have rank $\le O(\binom{n}{d}) \le O((en/d)^d)$. Hence [15] implies constant-error probabilistic-degree lower bounds for degree up to $\Omega(n/(\log n \cdot \log\log n))$, improving the degree bound in our main Theorem 1.3 by almost a $\log n$ factor.

Building on both [23] and [15], the paper [37] proves strong average-case hardness $1/2-2^{-\Omega(k)}$ against functions whose corresponding $2^n \times 2^n$ matrix has rank $2^{n/O(k(k+\log n))}$ (for $k \le c\sqrt{n}$). This generalizes both [23] and [15]. The latter can be recovered setting $k = O(1)$.

1.5.3 Organization

In Section 2 we prove the probabilistic-degree lower bound Theorem 1.3. The corollaries are then proved in Section 3. The hardness result under the uniform distribution, Theorem 1.8 is proved in Section 4. This latter theorem relies on a slightly stronger version of a result by Alman and Chen specialized to polynomials. We give a streamlined exposition of this in Section 5. In Section 6 we include a quantitative version of a connection from [73] between communication protocols and probabilistic degree. In section 7 we describe the connection to data-structure lower bounds.

2 Probabilistic degree lower bound

In this section we prove Theorem 1.3. We begin with two subsections, then prove the main theorem.

2.1 PCP tools

We need a simple variant of a PCP by Ben-Sasson and the author [14]. This PCP in turn is a relatively minor modification of the PCP [12] in which the verifier's query indexes are just projections of the verifier's randomness bits. In fact, for the application in this paper it is sufficient to have soundness polynomially bounded away from 1, so one can dispense with the soundness amplification step in [14].

First we state the result from [14], then state and prove our variant. Here "algorithm" refers to any standard model of computation, such as random-access multi-tape Turing machines, cf. [84].

Lemma 2.1 ([14])**.** Let M be an algorithm running in time $T = T(n) \geq n$ on inputs of the form (x, y) where $|x| = n$. Given $x \in \{0,1\}^n$ one can output in time $\mathrm{poly}(n, \log T)$ circuits $Q : \{0,1\}^r \to (\{0,1\}^r)^t$ for $t = \mathrm{poly}(r)$ and $R : \{0,1\}^t \to \{0,1\}$ such that:

Proof length. $2^r \leq T \cdot \mathrm{poly}\log T$,

Completeness. If there exists y such that $M(x, y)$ accepts then there exists a map $\pi : \{0,1\}^r \to \{0,1\}$ such that for any $z \in \{0,1\}^r$ we have $R(\pi(q_1), \ldots, \pi(q_t)) = 1$ where $(q_1, \ldots, q_t) = Q(z)$,

Soundness. If no y causes $M(x, y)$ to accept, then for every map $\pi : \{0,1\}^r \to \{0,1\}$, at most $1/n^{10}$ fraction of the $z \in \{0,1\}^r$ have $R(\pi(q_1), \ldots, \pi(q_t)) = 1$ where $(q_1, \ldots, q_t) = Q(z)$,

Complexity. Q is a projection (a.k.a. 1-local), i. e., each output bit of Q_i is one input bit, the negation of an input bit, or a constant; R is a 3CNF.

Next is our variant.

Lemma 2.2. Let M be an algorithm running in time $T = T(n) \geq n$ on inputs of the form (x, y) where $|x| = n$. Given $x \in \{0,1\}^n$ one can output in time $\mathrm{poly}(n, \log T)$ a collection of $\mathrm{poly}(r)$ circuits $Q_i : \{0,1\}^r \to (\{0,1\}^r)^3$ and $R_i : \{0,1\}^3 \to \{0,1\}$ such that:

Proof length. $2^r \leq T \cdot \mathrm{poly}\log T$,

Completeness. If there exists y such that $M(x, y)$ accepts then there exists a map $\pi : \{0,1\}^r \to \{0,1\}$ such that for any $z \in \{0,1\}^r$ and any $i \leq \mathrm{poly}(r)$ we have $R_i(\pi(q_1), \pi(q_2), \pi(q_3)) = 1$ where $(q_1, q_2, q_3) = Q_i(z)$,

Soundness. If no y causes $M(x, y)$ to accept, then for every map $\pi : \{0,1\}^r \to \{0,1\}$, at most a $1 - 1/r^{O(1)}$ fraction of the pairs $(z, i) \in \{0,1\}^r \times [\mathrm{poly}(r)]$ have $R_i(\pi(q_1), \pi(q_2), \pi(q_3)) = 1$ where $(q_1, q_2, q_3) = Q_i(z)$,

Complexity. Each Q_i is a projection (a.k.a. 1-local), i. e., each output bit of Q is one input bit, the negation of an input bit, or a constant; each R_i is an Or of three literals.

Proof. We start with the PCP in Lemma 2.1. Rather than computing all of the 3CNF R, the verifier only checks that clause i is satisfied. The function R_i is therefore an Or of three literals. Completeness is immediate. To argue soundness, note that for every choice of z such that $R(\pi(q_1), \dots, \pi(q_t)) = 0$, there is at least one clause of R that is not satisfied. Because the number of clauses is poly(r), and by the soundness of the PCP in Lemma 2.1, we have that the probability over (z, i) that $R_i(\pi(q_1), \pi(q_2), \pi(q_3)) = 0$ is at least $(1 - 1/n^{10}) \cdot (1/r^{O(1)}) \geq 1/r^{O(1)}$. □

2.2 Counting roots of polynomials

In this subsection we show how to count efficiently the satisfying assignments (a.k.a. roots) of a low-degree polynomial. We give a streamlined exposition of an algorithm which is a special case of Theorem 26 in [85].

Theorem 2.3 ([85, Theorem 26])**.** Given an $\mathbb{F}_2$-polynomial p of degree d in n variables we can compute the number of satisfying assignments (a.k.a. non-roots) in time $2^{n-\Omega(n/d)+O(\log n)}$.

For the proof we use *modulus-amplifying polynomials* from the 90's. We use them in a way similar to their use for orthogonal-vector algorithms ([17] Theorem 1.2). However using the latter algorithms directly appears to give a slightly worse bound. (For example, the dimension of the corresponding vectors is $\Omega(n^d)$ and so the running time would be $2^{n-\Omega(n/d \log n)}$ according to [17] Theorem 1.2.) Instead we use a classic dynamic-programming approach going back to Yates in the 1930's, see [84].

Lemma 2.4 (Modulus-amplifying polynomials, [88, 11])**.** For every integer ℓ there is a univariate polynomial F_ℓ of degree $2\ell - 1$ over the integers such that for every input y:
- $y \equiv 0 \mod 2 \Rightarrow F_\ell(y) \equiv 0 \mod 2^\ell$, and
- $y \equiv 1 \mod 2 \Rightarrow F_\ell(y) \equiv 1 \mod 2^\ell$.

Without loss of generality, the coefficients of F_ℓ are in $\{0, 1, \dots, 2^\ell - 1\}$. The coefficients of the polynomials can be computed in time poly(ℓ).

Lemma 2.5 (Yates in the 1930's, see [84])**.** Let $f : \{0,1\}^n \to \{0, 1, \dots, 2^{\mathrm{poly}(n)}\}$. Given the truth-table of f we can compute in time 2^npoly(n) the truth-table of the function $g : \{0,1\}^n \to \mathbb{Z}$ defined as $g(x) = \sum_{y \subseteq x} f(y)$, where we identify $\{0,1\}^n$ with the subsets of $\{1, 2, \dots, n\}$.

A proof of this lemma via dynamic programming can be found in [84].

Proof of Theorem 2.3. Let $\ell := \alpha n/d$ for an $\alpha > 0$ to be determined later. View p as a polynomial over the integers modulo $2^{\ell+1}$ (with coefficients in $\{0,1\}$). Let $F_{\ell+1}$ be the modulus amplifying polynomial of degree $2(\ell+1) - 1$. The idea is to compose $F_{\ell+1}$ and p to obtain polynomial p' of degree $O(d\ell) = O(\alpha n)$. Because we are only interested in the values $\{0,1\}$ for the variables, we can simplify p' replacing terms x_i^j with $j \geq 1$ with x_i. With this simplification, the number of monomials in p' will be $M := \sum_{i=0}^{O(\alpha n)} \binom{n}{i} \leq 2^{H(O(\alpha))n}$ where H is the binary entropy function. The coefficients of p' have $O(\ell)$ bits.

Note that we can write down p' in time $\mathrm{poly}(\ell) \cdot \mathrm{poly}(M)$. For example we can perform the $\mathrm{poly}(\ell)$ operations corresponding to F_ℓ to the polynomial p' working modulo $2^{\ell+1}$. After each operation, we simplify p' as mentioned above.

Now define $p''(x_1, x_2, \ldots, x_{n-\ell}) := \sum_{x_{n-\ell+1}, x_{n-\ell+2}, \ldots, x_n} p'(x_1, x_2, \ldots, x_n)$.

We can write down p'' in time $2^\ell \cdot \mathrm{poly}(M) \cdot \mathrm{poly}(\ell)$. Note p'' has again $\leq M$ monomials, and coefficients of $O(\ell)$ bits.

Note that for any input $x \in \{0,1\}^{n-\ell}$, $p''(x)$ counts the number of satisfying assignments among the 2^ℓ completions of x to an input of length n. This is because p' has been amplified to work modulo $2^{\ell+1}$, so there is no "wrap-around" when summing over 2^ℓ inputs. Hence we only need to evaluate p'' on all inputs. At this point we forget that p'' has low degree and we just consider p'' as a function $f : \{0,1\}^{n-\ell} \to \{0, 1, \ldots, 2^{O(\ell)}\}$ mapping monomials to coefficients. More in detail, we consider p'' as a polynomial with exactly $2^{n-\ell}$ monomials, with possibly zero coefficient. We view p'' as the function f whose input is interpreted as a monomial and whose output is the corresponding coefficient. Evaluating p'' on every input x is the same as computing the truth-table of the function $g(x) = \sum_{y \subseteq x} f(y)$. Lemma 2.5 solves the evaluation problem in time $2^{n-\ell} \cdot \mathrm{poly}(n)$.

Overall, the algorithm's running time is

$$2^\ell \mathrm{poly}(M) \cdot \mathrm{poly}(\ell) + 2^{n-\ell} \cdot \mathrm{poly}(n) \leq 2^{O(H(O(\alpha)))n} + 2^{n-\alpha n/d + O(\log n)}.$$

For a small enough α the first term is less than the second, and the result follows. □

2.3 Proof of Theorem 1.3

Assume towards a contradiction that every function in E^{NP} has 1/3-error polynomials of degree $d = \alpha n/\log^2 n$, where α is a constant to be determined later. First, we reduce the error. Consider the majority of t independent polynomials. This is a probabilistic polynomial of degree $t \cdot d$, simply writing the majority exactly as a degree-t polynomial and composing polynomials. On every input, by a Chernoff bound the error of this new polynomial is $2^{-\Omega(t)}$. Taking $t = O(c \log n)$ we obtain that every $f \in \mathrm{E}^{\mathrm{NP}}$ on n bits has $1/n^c$-error polynomials of degree $d' := O(dc \log n)$. We will set c later.

Let $L \in \mathrm{NTime}(2^n) \setminus \mathrm{NTime}(o(2^n))$ [28, 59, 89]. We want to show that $L \in \mathrm{NTime}(o(2^n))$ to reach a contradiction. Consider the algorithm $f(x, y)$ that on input $x \in \{0,1\}^n$ and $y \in \{1, 2, \ldots, 2^n \mathrm{poly}(n)\}$ constructs the circuits from the PCP in Lemma 2.2 corresponding to the verifier for L, computes the first satisfying assignment if one exists (otherwise computes say the all zero string), and outputs its bit y. This algorithm can be implemented in E^{NP} by computing the satisfying assignment one bit at the time.

Recalling $(q_1, q_2, q_3) = Q_i(z)$, where $z \in \{0,1\}^r$ and $r = n + O(\log n)$, we have for every $x \in L$:

$$\mathbb{P}_{z \in \{0,1\}^r, i \in [r^{O(1)}]}[R_i(f(x, q_1), f(x, q_2), f(x, q_3)) = 1] = 1;$$

while for every $x \notin L$ and every function π

$$\mathbb{P}_{z,i}[R_i(\pi(q_1), \pi(q_2), \pi(q_3)) = 1] \leq 1 - 1/n^b$$

for a constant $b = O(1)$.

By the argument above, f has a $1/n^c$-error polynomial of degree d'. We pick $c = b + 1$. For a fixed input $x \in L$ we can hard-wire the input x in the probabilistic polynomials for f and write $P = P_x$ for the corresponding distribution of polynomials. Because P has error $1/n^c$ on every fixed input, and because each constraint is an Or of three literals, we have for $x \in L$:

$$\mathbb{P}_{z,i,P}[R_i(P(q_1), P(q_2), P(q_3)) = 1] \geq 1 - 1/n^c .$$

Note that the weaker bound $1 - 3/n^c$ follows by a union bound. The stronger bound holds because it suffices that one input to R_i is 1 to have $R_i = 1$. Also we can take each occurrence of P inside the probability to be the same sample from the distribution of polynomials. Neither observation is crucial here but may be useful elsewhere.

By an averaging argument, there exists a fixed polynomial p of degree d such that

$$\mathbb{P}_{z,i}[R_i(p(q_1), p(q_2), p(q_3)) = 1] \geq 1 - 1/n^c .$$

On the other hand for any $x \notin L$ and for every proof π we have:

$$\mathbb{P}_{z,i}[R_i(\pi(q_1), \pi(q_2), \pi(q_3)) = 1] \leq 1 - 1/n^b < 1 - 1/n^c .$$

In particular this holds if we set π to be any polynomial p.

Hence, there is a gap between the probabilities. We now give an algorithm to detect the gap. Specifically, we give an efficient non-deterministic algorithm that decides if there exists a polynomial p such that $\mathbb{P}_{z,i}[R_i(p(q_1), p(q_2), p(q_3)) = 1] \geq 1 - 1/n^c$.

On input x, the algorithm guesses a polynomial p' of degree d' on $n' = n + O(\log n)$ variables. The number of bits required to specify such a polynomial is $\ell = \sum_{i=0}^{d'} \binom{n'}{i} \leq (\mathrm{e}n'/d')^{d'} \leq n^{d'} = n^{O(dc \log n)} = 2^{O(\alpha c n)}$ which is $o(2^n)$ for a small enough α depending only on c. The algorithm then computes the corresponding probability, and accepts if it is at least $1 - 1/n^c$.

Computing the probability. The algorithm will compute the probability for every fixed value of i, and then average the values. Because the number of i is only $\mathrm{poly}(r) = \mathrm{poly} \log T = \mathrm{poly}(n)$, this only incurs a polynomial slow-down.

Once i is fixed, the queries q_1, q_2, q_3 are linear functions of the input z. Composing these functions with p' and writing the output R_i exactly as a degree-3 polynomial, we obtain a polynomial q of degree $O(d')$. To write down the polynomial we need first to compute the query circuits Q_i and the circuit R_i. This can be done in time $\mathrm{poly}(n)$ by Lemma 2.2. Composing these with p' can be done in time polynomial in the length of the representation of p' which is $\mathrm{poly}(\ell)$. As above, the latter quantity can be made say $\leq 2^{n/2}$, again for α small enough depending on c. We then compute the number of inputs that cause q to accept using Lemma 2.3. The degree of q is $O(d') \leq O(\alpha c n / \log n)$. The running time of the algorithm in Lemma 2.3 is $2^{n - \Omega(\log n)/(\alpha c) + O(\log n)}$. For a small enough α the overall running time of the algorithm is $o(2^n)$, concluding the proof.

3 Proofs of corollaries

In this section we prove Corollaries 1.4, 1.5, and 1.7. We need several upper bounds on probabilistic degree from the literature.

3.1 Probabilistic degree upper bounds

We rely on Razborov's seminal approximation result [57]. We use the recent tighter version by Kopparty and Srinivasan [42], who also show their bound is nearly tight.

Lemma 3.1 ([42])**.** Any $\mathrm{AC}^0[\oplus]$ circuit of depth d and size s has ϵ-error degree at most

$$O(\log s)^{d-1}\log(1/\epsilon)\,.$$

We also need upper bounds on the probabilistic degree of symmetric and threshold functions, by Alman and Williams, and Srinivasan.

Lemma 3.2 ([5, Theorem 1.2])**.** Any symmetric function on n bits has ϵ-error degree at most $O(\sqrt{n\log(1/\epsilon)})$.

Lemma 3.3 ([64, Theorem 12])**.** Any threshold function (a.k.a. Ltf) has ϵ-error degree $\leq \sqrt{n}(\log n\cdot\log 1/\epsilon)^{O(1)}$.

3.2 Proofs of corollaries

First we prove Corollary 1.4 by connecting two settings of parameters for probabilistic degree, showing that improving the state-of-the-art for small degree and error approaching 1/2 is easier than improving it for large degree and constant error. Then we prove Corollary 1.7.

Lemma 3.4. Suppose that $f:\{0,1\}^n\to\{0,1\}$ has $(1/2-\epsilon)$-error degree d. Then f has 1/10-error degree $O(d/\epsilon)$.

In particular, for $\epsilon=1/\sqrt{n}$ and $d=\log n$ we obtain degree $O(\log n\cdot\sqrt{n})$.

Proof. Let P be a distribution witnessing $(1/2-\epsilon)$-error degree d. We take the majority of $t=O(1/\epsilon^2)$ independent copies of P. By a Chernoff bound this majority has error $\leq 1/20$. Moreover, Majority on t bits has 1/20-error polynomials of degree $O(\sqrt{t})$ [5]. So replacing the majority with a probabilistic polynomial gives error $1/20+1/20\leq 1/10$ and degree $O(d/\epsilon)$. □

From this the proof of Corollary 1.4 is immediate. Now we prove the other two corollaries.

Proof of Corollary 1.5. We apply Lemma 3.1 with $\epsilon=1/3$ to obtain that the function computed by the circuit has 1/3-error degree $O(\log s)^{h-1}$. By Theorem 1.3 we have $O(\log s)^{h-1}\geq n/\log^2 n$ and the result follows. □

Proof of Corollary 1.7. We show that these circuits have small probabilistic degree. We approximate each And and Or gate by a degree-$\log^{O(1)}(s)$ polynomial with error $0.1/s$ by Lemma 3.1. We approximate each Sym and Ltf gate with fan-in m by a degree $\sqrt{m}\cdot\log^{O(1)}(s)$ polynomial with error $0.1/s$ by Lemmas 3.2 and 3.3. We compose all these polynomials. By a union bound, the error is $< 1/3$.

It remains to bound the degree. We prove by induction that for every gate g at distance h' from the input, the composed polynomial (obtained by composing the polynomial for each gate) has degree $\sqrt{t'}\cdot\log^{O(h')} s$, where t' is the least value such that the sub-circuit obtained by considering g as output is t'-SoT-AC$^0[\oplus]$. This is obvious for $h' = 1$ by the above approximations. For larger h', suppose g is an And or Or gate. Then by induction the sub-circuits have degree $\sqrt{t'}\cdot\log^{O(h'-1)} s$ and the result follows. Otherwise, g is either a Sym or an Ltf gate with fan-in m. Hence some child of g is t'/m-SoT-AC$^0[\oplus]$, and others are t''-SoT-AC$^0[\oplus]$ for $t'' \le t'/m$. By induction they all have degree $\sqrt{t'/m}\cdot\log^{O(h'-1)} s$ and so the degree of g is $\sqrt{m}\cdot\log^{O(1)}(s)\cdot\sqrt{t'/m}\cdot\log^{O(h'-1)} s$ as desired. ◻

4 Uniform distribution: Proof of Theorem 1.8

In this section we prove Theorem 1.8. We need some tools from the theory of hardness amplification.

4.1 Tools from hardness amplification

First we need a relatively standard connection between average-case and worst-case lower bounds for general circuits. The important point here is that we are not claiming that the circuits are of constant depth or have any other restriction except size.

Lemma 4.1 ([8, 38, 66])**.** Let $s = s(n)$ satisfy $s(O(n)) \le s^{O(1)}(n)$. There is a constant c such that the following holds.

Suppose that for every function $f : \{0,1\}^n \to \{0,1\}$ in E^{NP} there is a circuit of size $s(n) \ge n$ such that $\mathbb{P}_{x\in\{0,1\}^n}[C(x) = f(x)] \ge 1/2 + 1/n$. Then for every function $f : \{0,1\}^n \to \{0,1\}$ in E^{NP} there is a circuit of size $s^{O(1)}(n)$ such that $C = f$.

Remark 4.2. For the proof we can use Theorem 24 and Lemma 28 in [66]. That gives the size bound $s' = n^{O(1)}\cdot s(O(n))$ which is $\le s^{O(1)}(n)$ by the assumptions on s. The only thing that is left to verify is that their encoding procedure can be implemented in E^{NP}. Indeed, as observed already in [26], given a function f in E^{NP} we can construct another E^{NP} function which first writes down the truth table of f by calling the algorithm for f 2^n times, using the NP oracle for each call. Then we can run the encoding procedure in [66] which runs in polynomial time in the length 2^n of the truth table.

We need another result from the hardness amplification literature. This time we need a much finer bound on the complexity of the associated decoding procedure. We need that the proof of a hardness amplification from constant to $1/2 - \epsilon$ can be implemented essentially by a majority on $1/\epsilon^2$ bits. Such a fine result actually follows from Impagliazzo's hard-core-set proof [38] of Yao's XOR lemma (see [31]), as was made explicit in a beautiful paper by Klivans [41]. Klivans actually was working in a slightly different context and did not need to bound the fan-in of the majority gate, but the bound is evident in his analysis. We note that Klivans's application (a switching-lemma free proof of the average-case lower bounds for AC^0) was later simplified by Klivans and Vadhan (see the exposition in [71]). This simplification bypasses the XOR lemma, using instead the random self-reducibility of parity. In our application, the function may not be randomly self-reducible.

Lemma 4.3. Let $f : \{0,1\}^n \to \{0,1\}$ be a function that is $\Omega(1)$-hard for $\mathrm{Maj}_t \circ \mathrm{AC}^0[\oplus]$ circuits of size s and depth h. Then the function $f' : (\{0,1\}^n)^k \to \{0,1\}$ defined as $f'(x_1, x_2, \ldots, x_k) := \oplus_{i \le k} f(x_i)$ is $(1/2 - O(1/\sqrt{t}) - (1-\Omega(1))^k)$-hard for $\mathrm{AC}^0[\oplus]$ circuits of size s/t and depth $h-1$.

Remark 4.4. The combination of Theorem 8 and Fact 9 in [41] shows that there exists a "hard-core" set $S \subseteq \{0,1\}^n$ of size $\Omega(2^n)$ such that for every $\mathrm{AC}^0[\oplus]$ circuit C of size s/t and depth $h-1$ one has $\mathbb{P}_{x\in S}[C(x) \neq h(x)] \ge 1/2 - O(1/\sqrt{t})$. Otherwise, one can take the majority of t such circuits to obtain a contradiction. This suffices for a proof of the XOR lemma (see Lemma 4 in [38] or Theorem 10 in [41]).

4.2 A result by Alman and Chen

We need the following result which is essentially Theorem 1.9 by Alman and Chen [4].

Lemma 4.5. Suppose that E^{NP} has circuits of size s. Then there is a function in E^{NP} that is $\Omega(1)$-hard for $\mathbb{F}_2$-polynomials of degree $\Omega(n/\log s)$, infinitely often.

Theorem 1.9 by Alman and Chen [4] claims $\Omega(1)$-hardness for rank $n^{1-\alpha}$ for any $\alpha > 0$. As mentioned earlier and in [60], this implies the same hardness for degree $n^{1-\alpha}$ for any $\alpha > 0$. We note here slightly stronger parameters, and we give a streamlined proof for polynomials, following closely theirs, in Section 5.

4.3 Proof of Theorem 1.8

Towards a contradiction, suppose that for every $f : \{0,1\}^n \to \{0,1\}$ in E^{NP} and for all sufficiently large input lengths n there are $\mathrm{AC}^0[\oplus]$ circuits C of size $s = s(n)$ and depth h such that $\mathbb{P}_{x\in\{0,1\}^n}[f(x) \neq C(x)] \le 1/2 - (\log^{9h} s)/n$.

By Lemma 4.1, E^{NP} has circuits of size $s^{O(1)}(n)$.

By Lemma 4.5, there is a function in E^{NP} that is $\Omega(1)$-hard under the uniform distribution for polynomials of degree $\Omega(n/\log s)$, for infinitely many n. In particular, for those n, the latter function is $\Omega(1)$-hard under the uniform distribution for $\mathrm{MAJ}_{n^2/\log^{8h} s} \circ \mathrm{AC}^0[\oplus]$ circuits where the $\mathrm{AC}^0[\oplus]$ circuits have size $2^{\log^2 s(n)}$ and depth $h+1$. This is because the latter circuits have $o(1)$-error degree $d := O(\sqrt{n^2/\log^{8h} s} \cdot \log^{2h} s \cdot \log n) \le o(n/\log s)$. This is proved by applying Lemma 3.2 to MAJ and 3.1 with error $1/n^2$ to the $\mathrm{AC}^0[\oplus]$ circuits. So if such a circuit computed

h correctly on a δ fraction of the inputs there would be a distribution P on polynomials of degree d such that $\mathbb{P}_{x,P}[P(x) = h(x)] \geq \delta - o(1)$, where x is uniformly distributed. And so by an averaging argument we could fix a specific polynomial to contradict the claim above.

Now we can apply Lemma 4.3 to construct a function on nk bits for $k = O(\log n)$ which, for infinitely many n, has hardness $1/2 - O(1)\Big/\sqrt{n^2/\log^{8h} s(n)} - 1/n \geq 1/2 - \log^{2h} s(n)/n$ for circuits of size $2^{\log^2 s(n)}/(n^2/\log^{O(h)} s(n)) \geq 2^{\Omega(\log^2 s(n))}$, where the inequality holds without loss of generality, and depth $(h+1)-1 = h$. Here we are using that $(1-\Omega(1))^k \leq 1/n$ for $k = O(\log n)$.

To contradict our initial assumption, it suffices that $s(O(n \log n)) < 2^{\Omega(\log^2 s(n))}$. This is indeed our assumption.

5 Proof of Lemma 4.5

We need the following recent construction of *smooth* PCPs of proximity by Paradise [52], which we state using notation similar to that in Section 2.1.

Lemma 5.1. For any constants $\delta, \sigma \in (0, 1/3]$ there is a constant c and an algorithm M with the following properties. On input a circuit C of size s on $\ell \leq s$ inputs, ℓ a power of 2, the algorithm $M(C)$ computes in time s^c circuits $Q : \{0,1\}^r \to (\{0,1\}^r)^c$, $Q' : \{0,1\}^r \to (\{0,1\}^{\log \ell})^c$, and $R : \{0,1\}^{2c} \to \{0,1\}$ such that:

Proof length. $2^r \leq s^c$,

Completeness. If y makes $C(y) = 1$ then there exists a map $\pi : \{0,1\}^r \to \{0,1\}$ such that for any $z \in \{0,1\}^r$ we have $R(y(q'_1), \dots, y(q'_c), \pi(q_1), \dots, \pi(q_c)) = 1$ where $(q_1, \dots, q_c) = Q(z)$, $(q'_1, \dots, q'_c) = Q'(z)$, and $y(i)$ is the i bit of y,

Soundness. If y is δ-far in Hamming distance from any z such that $C(z) = 1$, then for every map $\pi : \{0,1\}^r \to \{0,1\}$, at most a σ fraction of the $z \in \{0,1\}^r$ have

$$R(y(q'_1), \dots, y(q'_c), \pi(q_1), \dots, \pi(q_c)) = 1$$

where $(q_1, \dots, q_c) = Q(z)$, $(q'_1, \dots, q'_c) = Q'(z)$,

Smoothness. For every $i \leq c$, each query q_i is uniformly distributed in $\{0,1\}^r$ over the choice of $z \in \{0,1\}^r$.

Proof of Lemma 4.5. As in the proof of Theorem 1.3, let $L \in \mathrm{NTime}(2^n) \setminus \mathrm{NTime}(o(2^n))$ [28, 59, 89] be a unary language. We want to show that $L \in \mathrm{NTime}(o(2^n))$ to reach a contradiction (assuming the opposite of the conclusion, which we will do shortly). Consider the algorithm $f(x, y)$ that on input $x \in \{0,1\}^n$ and $y \in \{1, 2, \dots, 2^n \mathrm{poly}(n)\}$ constructs the circuits from the PCP in Lemma 2.1, computes the first satisfying assignment if one exists, and outputs its bit y. This algorithm can be implemented in E^{NP} by computing the satisfying assignment one bit at the time. By the assumption, this algorithm can be implemented by a circuit C of size $s \geq n$.

The algorithm to show $L \in \mathrm{NTime}(o(2^n))$ proceeds by guessing this circuit C, and composing it with the PCP in Lemma 2.1 to obtain another circuit C' of size $s^{O(1)}$. By the properties of the PCP, it then suffices to efficiently compute

$$\mathbb{P}_{z \in \{0,1\}^r}[C'(z) = 1] .$$

We use the PCP in Lemma 5.1 to aid in this. We first have to encode z. Fix any efficient, linear, binary error-correcting code $ECC : \{0,1\}^r \to \{0,1\}^{r_\delta}$ with Hamming distance δ and efficient encoding and decoding, where r_δ is linear in r for any fixed δ. By composing the decoding procedure with C' we obtain another circuit C'' of size $s^{O(1)}$ and see that it suffices to compute

$$\mathbb{P}_{z\in\{0,1\}^r}[C''(ECC(z)) = 1]\,.$$

Applying the PCP in Lemma 5.1 to circuit C'' we see that if $x \in L$ then there are proofs π_z such that

$$\mathbb{P}_{z\in\{0,1\}^r, z'\in\{0,1\}^{O(\log s)}}[R(ECC(z)(q'_1),\dots,ECC(z)(q'_c),\pi_z(q_1),\dots,\pi_z(q_c)) = 1] = 1\,,$$

while if $x \notin L$ then for any proofs π_z we have

$$\mathbb{P}_{z\in\{0,1\}^r, z'\in\{0,1\}^{O(\log s)}}[R(ECC(z)(q'_1),\dots,ECC(z)(q'_c),\pi_z(q_1),\dots,\pi_z(q_c)) = 1] \le 1/2\,,$$

where $(q_1,\dots,q_c) = Q(z')$, $(q'_1,\dots,q'_c) = Q'(z')$, and picking the soundness in the PCPs in Lemmas 2.1 and 5.1 to be a small enough constant.

The proofs $\pi_z(q)$ can be computed as a function of z and q by an E^{NP} algorithm $f(z,q)$. Now let us assume the opposite of the conclusion of the lemma. Then for every $\beta > 0$ there are polynomials p of degree βd such that $\mathbb{P}_{z,q}[p(z,q) \ne f(z,q)] \le \beta$. Because the queries q are smooth, and by a union bound, we have that if $x \in L$ then

$$\mathbb{P}_{z\in\{0,1\}^r, z'\in\{0,1\}^{O(\log s)}}[R(ECC(z)(q'_1),\dots,ECC(z)(q'_c),p(z,q_1),\dots,p(z,q_c)) = 1] \ge 1 - \beta c\,,$$

while if $x \notin L$ then the above probability is $\le 1/2$ as above. Thus picking $\beta < c/2$ there is a gap in the above probabilities. If we can detect this gap efficiently, we are done. (Recall that L is unary, so membership of x in L is solely determined by its length.)

The algorithm will guess the polynomial p and then compute the above probability. To compute the probability, it enumerates over all $s^{O(1)}$ choices for z'. For any fixed z', the function

$$R(ECC(z)(q'_1),\dots,ECC(z)(q'_c),p(z,q_1),\dots,p(z,q_c))$$

is a polynomial of degree $2c\beta d$. Here we simply write R as a polynomial of degree $2c$, and use that ECC is linear. Applying Theorem 2.3 allows us to compute this probability in time $2^{n-\Omega(n/\beta d)+O(\log n)}$. If $d = n/\log s$ the running time is $2^{n-\Omega(\log s)/\beta+O(\log n)}$. Picking β small enough the saving in time can compensate for the $s^{O(1)}$ slow-down that comes from enumerating over all z', and the proof is concluded. □

6 Number-on-forehead

The connection between probabilistic degree and number-on-forehead lower bounds was pointed out in [73]. We present next a stronger quantitative relationship.

Theorem 6.1. Suppose that $f : \{0,1\}^n \to \{0,1\}$ has $(1/2-\epsilon)$-error degree d. Then f has number-on-forehead protocols with $O(d\log(n/\epsilon))$ parties and communication $O(d\log(n/\epsilon)\cdot\log n)$, under any partition of the input.

To illustrate, if $d = \log n$ and $\epsilon = 1/\sqrt{n}$ (something which we cannot rule out for any function in NP) then we obtain $O(\log^2 n)$ parties and $O(\log^3 n)$ communication.

Proof. Let D be the distribution on degree-d polynomials witnessing the $(1/2-\epsilon)$-error degree of f. By taking the majority of $t = O(n/\epsilon^2)$ independent copies of D, we can drive the error to $< 2^{-n}$. By the probabilistic method we can fix the values of the D and obtain that there are t polynomials $p_1, \ldots, p_t$ of degree d such that $f(x) = \mathrm{Maj}(p_1(x), \ldots, p_t(x))$ for every input $x \in \{0,1\}^n$.

Now view the p_i as integer polynomials, and compose each with a modulus amplifying polynomial F of degree $O(\log t)$ given by Lemma 2.4. This guarantees that the value of each polynomial modulo 2 is the same as the value modulo $2^{O(\log t)} > t$. Hence, the value of $\mathrm{Maj}(p_1(x), \ldots, p_t(x))$ is determined by

$$q(x) := \sum_{i\le t} F(p_i(x))\,.$$

Each $F(p_i(x))$ is a polynomial of degree $D = O(d\log t)$ with $O(\log t)$-bit coefficients. Hence the same is true about $q(x)$.

By Håstad and Goldmann's simulation [36], $q(x)$ can be computed by a number-on-forehead protocol with $D+1$ parties, under any partition of the input with $D+1$ sets. Recall in the protocol each party just sends the sum of the monomials in q assigned to them. Monomials are arbitrarily assigned to parties who know their values. (For every monomial there exists at least one party who knows all the bits in the monomial, since each party misses one of $D+1$ sets and the monomial has degree D.) The sum sent by a party has magnitude at most the magnitude of one coefficient times the number of monomials in q, which is $\le \mathrm{poly}(t)\cdot n^{O(d\log t)}$. The number of bits is then $O(d\log(n/\epsilon)\cdot\log n)$. □

7 Data structures

In this section we show that sufficiently strong lower bounds on probabilistic rank imply new data-structure lower bounds. We say that a function $f : \{0,1\}^n \times \{0,1\}^\mu \to \{0,1\}$ has a space-s t-local data structure if there exists a function $E : \{0,1\}^n \to \{0,1\}^s$ and $m = 2^\mu$ functions g_q such that for every x, q we have

$$f(x,q) = g_q(E(x))\,,$$

where the functions g_q are t-local, that is they depend on only t bits of their input at fixed locations. Here x is the input data of length n, s is the *space* of the data structure, q is the query, and t is the time of the data structure. The data structure is supposed to encode x via $E(x)$ so that queries can be answered fast. Such problems are known as *static* (because the data does not change over time) and *non-adaptive* (because g_q reads bits at fixed locations).

Despite intense research, the state of known lower bounds can be summarized with the following inequality

$$t \geq \log(m/n)/\log(s/n)\,. \tag{7.1}$$

Specifically, no explicit function for which a bound better than Equation (7.1) is known, for any setting of parameters, and even for functions in E^{NP}. We refer to [75] for additional discussion. We raise the question of improving this state of affairs using the "guess-and-sat" method.

We note that several connections between data-structure lower bounds and other challenges in computational complexity appear in the literature: [47, 75, 30, 29, 54]. Some of these works explain the failure to prove stronger lower bounds by showing that such lower bounds would also solve other long-standing problems in complexity theory. However in some cases the latter lower bounds were actually found. For some settings of parameters, the connection in [47] has been subsumed by the lower bounds in [2] and [10], and the connection in [30] by those in [4, 15].

Here we consider the reverse direction and show that data-structure lower bounds follow from sufficiently strong probabilistic-rank lower bounds. Specifically, it suffices to show that f cannot be approximated with error about 2^{-t} by matrixes with rank s. And in fact it suffices to consider matrices $M = AB$ where A is fixed and B is sparse.

Claim 7.1. *Suppose a function $f : \{0,1\}^n \times \{0,1\}^\mu \longrightarrow \{0,1\}$ has a space-s t-local data structure. Then there exists a $2^n \times (s+1)$ matrix A and a distribution on $(s+1) \times 2^\mu$ matrices B such that any column of B has at most $t+1$ non-zero elements, and for every (x,q) we have*

$$\mathbb{P}_B[(AB)_{x,q} = f(x,q)] \geq 1/2 + \Omega(\sqrt{2^{-t}})\,,$$

where the matrices are considered over $\mathbb{F}_2$.

To illustrate, let us consider the setting $m = n^{10}$, so $\mu = 10 \log n$ and $s = n^{1.1}$. In this case the bound in Equation (7.1) becomes

$$t \geq \log(n^9)/\log(n^{0.1})$$

which is constant. To improve this bound it would be enough to bound above the probability in the claim above by $1/2 + o(1)$, however the corresponding matrix is highly unbalanced. At the other end of the spectrum, consider the case $m = 2^n$, so $\mu = n$, and $s = 2n$. In this case the bound in Equation (7.1) becomes

$$t \geq \Omega(\log 2^n) = \Omega(n)\,.$$

To improve on this we would need to bound the probability by a quantity exponentially close to $1/2$.

Proof. We rely on the following basic fact. Let $h : \{0,1\}^t \to \{0,1\}$ be a function on t bits, and let $g(x) := (-1)^{h(x)}$. Then there is a joint distribution (S, b) where S is a subset of $\{1, 2, \dots, n\}$ and $b \in \{0,1\}$ such that for every x the expectation $\mathbb{E}_{S,b}[(-1)^{b+\sum_{i\in S} x_i}]$ is $g(x)/\sum_\alpha |\hat{g}_\alpha|$ where $g(x) = \sum_\alpha \hat{g}(\alpha)(-1)^{b+\sum_{i\in\alpha} x_i}$ is the Fourier expansion of g (see [50] for background). To verify this, define the distribution so that with probability $|\hat{g}_\alpha|/\sum_\alpha |\hat{g}_\alpha|$ it yields outcome $(-1)^b = \operatorname{sign} \hat{g}_\alpha)$ and $S = \alpha$. Then the expectation is as claimed, since

$$\mathbb{E}_{S,b}[(-1)^{b+\sum_{i\in S} x_i}] = \sum_\alpha |\hat{g}_\alpha| \operatorname{sign} \hat{g}_\alpha)(-1)^{\sum_{i\in\alpha} x_i} \Big/ \sum_\beta |\hat{g}_\beta|$$

and $|\hat{g}_\alpha| \operatorname{sign} \hat{g}_\alpha) = \hat{g}_\alpha$.

Also note $\sum_\beta |\hat{g}_\beta| \le \sqrt{2^t} \sum_S |\hat{g}_\beta|^2 = \sqrt{2^t}$ by Cauchy–Schwarz and Parseval, using that g has range $\{-1,1\}$. From this we deduce that $\mathbb{E}_{(S,b)}[(-1)^{b+\sum_{i\in S} x_i} g(x)] \ge 1/\sqrt{2^t}$. And so $\mathbb{P}[b + \sum_{i\in S} x_i = h(x) \mod 2] \ge 1/2 + 0.5/\sqrt{2^t}$.

Row x of the matrix A is $E(x)$ with an extra 1 appended at the end. Column q of B is obtained by applying the fact above to the function g_q, setting to 1 the bits corresponding to S and to 0 the others, possibly except the last which is used to implement the bit b. □

With this approach, the "+1" is unavoidable: the data structure may choose to encode x with the all-zero string, so we can never get a non-zero value if our row is just $E(x)$.

Acknowledgments

We thank the anonymous referees for the detailed and helpful feedback.

References

[1] AMIR ABBOUD, R. RYAN WILLIAMS, AND HUACHENG YU: More applications of the polynomial method to algorithm design. In *Proc. 26th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'15)*, pp. 218–230. SIAM, 2015. [doi:10.1137/1.9781611973730.17] 11

[2] MIKLÓS AJTAI: A non-linear time lower bound for Boolean branching programs. *Theory of Computing*, 1(8):149–176, 2005. [doi:10.4086/toc.2005.v001a008] 23

[3] JOSH ALMAN, TIMOTHY M. CHAN, AND R. RYAN WILLIAMS: Polynomial representations of threshold functions and algorithmic applications. In *Proc. 57th FOCS*, pp. 467–476. IEEE Comp. Soc., 2016. [doi:10.1109/FOCS.2016.57] 5, 7

[4] JOSH ALMAN AND LIJIE CHEN: Efficient construction of rigid matrices using an NP oracle. *SIAM J. Comput.*, 54(4):102–134, 2025. Preliminary version in FOCS'19. [doi:10.1137/20M1322297] 4, 5, 6, 7, 9, 10, 11, 19, 23

[5] JOSH ALMAN AND R. RYAN WILLIAMS: Probabilistic polynomials and Hamming nearest neighbors. In *Proc. 56th FOCS*, pp. 136–150. IEEE Comp. Soc., 2015. [doi:10.1109/FOCS.2015.18] 2, 5, 6, 17

[6] JOSH ALMAN AND R. RYAN WILLIAMS: Probabilistic rank and matrix rigidity. In *Proc. 49th STOC*, pp. 641–652. ACM Press, 2017. [doi:10.1145/3055399.3055484] 11

[7] SANJEEV ARORA AND BOAZ BARAK: *Computational Complexity: A modern approach*. Cambridge Univ. Press, 2009. [doi:10.1017/CBO9780511804090] 9

[8] LÁSZLÓ BABAI, LANCE FORTNOW, NOAM NISAN, AND AVI WIGDERSON: BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complexity*, 3(4):307–318, 1993. [doi:10.1007/BF01275486] 9, 18

[9] LÁSZLÓ BABAI, NOAM NISAN, AND MÁRIÓ SZEGEDY: Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. Comput. System Sci.*, 45(2):204–232, 1992. [doi:10.1016/0022-0000(92)90047-M] 3, 4

[10] PAUL BEAME, MICHAEL SAKS, XIAODONG SUN, AND ERIK VEE: Time-space trade-off lower bounds for randomized computation of decision problems. *J. ACM*, 50(2):154–195, 2003. [doi:10.1145/636865.636867] 23

[11] RICHARD BEIGEL AND JUN TARUI: On ACC. *Comput. Complexity*, 4(4):350–366, 1994. [doi:10.1007/BF01263423] 14

[12] ELI BEN-SASSON, ODED GOLDREICH, PRAHLADH HARSHA, MADHU SUDAN, AND SALIL P. VADHAN: Short PCPs verifiable in polylogarithmic time. In *Proc. 20th IEEE Conf. on Comput. Complexity (CCC'05)*, pp. 120–134. IEEE Comp. Soc., 2005. [doi:10.1109/CCC.2005.27] 8, 12, 13

[13] ELI BEN-SASSON AND MADHU SUDAN: Short PCPs with polylog query complexity. *SIAM J. Comput.*, 38(2):551–607, 2008. [doi:10.1137/050646445] 8

[14] ELI BEN-SASSON AND EMANUELE VIOLA: Short PCPs with projection queries. In *picalp14*, pp. 163–173, 2014. [doi:10.1007/978-3-662-43948-7_14] 8, 10, 11, 12, 13

[15] AMEY BHANGALE, PRAHLADH HARSHA, ORR PARADISE, AND AVISHAY TAL: Rigid matrices from rectangular PCPs. In *Proc. 61st FOCS*, pp. 858–869. IEEE Comp. Soc., 2020. [doi:10.1109/FOCS46700.2020.00084] 12, 23

[16] ANDREJ BOGDANOV AND EMANUELE VIOLA: Pseudorandom bits for polynomials. *SIAM J. Comput.*, 39(6):2464–2486, 2010. Preliminary version in FOCS'07. [doi:10.1137/070712109] 5

[17] TIMOTHY M. CHAN AND R. RYAN WILLIAMS: Deterministic APSP, orthogonal vectors, and more: Quickly derandomizing Razborov-Smolensky. In *Proc. 27th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'16)*, pp. 1246–1255. SIAM, 2016. [doi:10.1137/1.9781611974331.87] 9, 11, 12, 14

[18] ASHOK K. CHANDRA, MERRICK L. FURST, AND RICHARD J. LIPTON: Multi-party protocols. In *Proc. 15th STOC*, pp. 94–99. ACM Press, 1983. [doi:10.1145/800061.808737] 4

[19] ESHAN CHATTOPADHYAY, JASON GAITONDE, CHIN HO LEE, SHACHAR LOVETT, AND ABHISHEK SHETTY: Fractional pseudorandom generators from any Fourier level, 2021. [doi:10.4230/LIPIcs.CCC.2021.10, arXiv:2008.01316] 5

[20] ESHAN CHATTOPADHYAY, POOYA HATAMI, KAAVE HOSSEINI, AND SHACHAR LOVETT: Pseudorandom generators from polarizing random walks. In *Proc. 33rd Comput. Complexity Conf. (CCC'18)*, pp. 1:1–21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.CCC.2018.1, arXiv:2008.01316] 5

[21] ESHAN CHATTOPADHYAY, POOYA HATAMI, SHACHAR LOVETT, AND AVISHAY TAL: Pseudorandom generators from the second fourier level and applications to AC0 with parity gates. In *Proc. 10th Innovations in Theoret. Comp. Sci. Conf. (ITCS'19)*, pp. 22:1–15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.ITCS.2019.22] 5

[22] LIJIE CHEN: Non-deterministic quasi-polynomial time is average-case hard for ACC circuits. In *Proc. 60th FOCS*, pp. 1281–1304. IEEE Comp. Soc., 2019. [doi:10.1109/FOCS.2019.00079] 7

[23] LIJIE CHEN, XIN LYU, AND R. RYAN WILLIAMS: Almost everywhere circuit lower bounds from non-trivial derandomization. In *Proc. 61st FOCS*. IEEE Comp. Soc., 2020. [doi:10.1109/FOCS46700.2020.00009] 12

[24] LIJIE CHEN AND HANLIN REN: Strong average-case circuit lower bounds from non-trivial derandomization. *SIAM J. Comput.*, 51(3):115–173, 2022. Preliminary version in STOC'20. [doi:10.1137/20M1364886] 7, 12

[25] LIJIE CHEN AND R. RYAN WILLIAMS: Stronger connections between circuit analysis and circuit lower bounds, via PCPs of proximity. In *Proc. 34th Comput. Complexity Conf. (CCC'19)*, pp. 19:1–43. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.CCC.2019.19] 10

[26] RUIWEN CHEN, IGOR CARBONI OLIVEIRA, AND RAHUL SANTHANAM: An average-case lower bound against $\mathsf{a}cc^0$. In *Proc. Latin American Symp. on Theoretical Informatics (LATIN'18)*, pp. 317–330. Springer, 2018. [doi:10.1007/978-3-319-77404-6_24] 7, 18

[27] FAN R. K. CHUNG AND PRASAD TETALI: Communication complexity and quasi randomness. *SIAM J. Discr. Math.*, 6(1):110–123, 1993. [doi:10.1137/0406009] 4

[28] STEPHEN A. COOK: A hierarchy for nondeterministic time complexity. *J. Comput. System Sci.*, 7(4):343–353, 1973. [doi:10.1016/S0022-0000(73)80028-5] 15, 20

[29] HENRY CORRIGAN-GIBBS AND DMITRY KOGAN: The function-inversion problem: Barriers and opportunities. *Electron. Colloq. Comput. Complexity*, TR18-182, 2018. [ECCC] 23

[30] ZEEV DVIR, ALEXANDER GOLOVNEV, AND OMRI WEINSTEIN: Static data structure lower bounds imply rigidity. In *Proc. 51st STOC*, pp. 967–978. ACM Press, 2019. [doi:10.1145/3313276.3316348] 23

[31] ODED GOLDREICH, NOAM NISAN, AND AVI WIGDERSON: On Yao's XOR lemma. In *Studies in Complexity and Cryptography*, LNCS 6650, pp. 273–301. Springer, 2011. Preliminary version ECCC TR95–050 (1995). [doi:10.1007/978-3-642-22670-0_23] 10, 19

[32] ALEXANDER GOLOVNEV, ALEXANDER S. KULIKOV, AND R. RYAN WILLIAMS: Circuit depth reductions. In *Proc. 12th Innovations in Theoret. Comp. Sci. Conf. (ITCS'21)*, pp. 24:1–20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2021. Preliminary version in ECCC TR18–192 (2018–2020). [doi:10.4230/LIPIcs.ITCS.2021.24] 2

[33] ARYEH GRINBERG, RONEN SHALTIEL, AND EMANUELE VIOLA: Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs. In *Proc. 59th FOCS*, pp. 956–966. IEEE Comp. Soc., 2018. Available at author's website. [doi:10.1109/FOCS.2018.00094] 5, 9

[34] ANDRÁS HAJNAL, WOLFGANG MAASS, PAVEL PUDLÁK, MÁRIÓ SZEGEDY, AND GYÖRGY TURÁN: Threshold circuits of bounded depth. *J. Comput. System Sci.*, 46(2):129–154, 1993. [doi:10.1016/0022-0000(93)90001-D] 4

[35] JOHAN HÅSTAD: *Computational Limitations of Small-Depth Circuits*. MIT Press, 1987. Available at DSpace@MIT. 3, 6

[36] JOHAN HÅSTAD AND MIKAEL GOLDMANN: On the power of small-depth threshold circuits. *Comput. Complexity*, 1(2):113–129, 1991. [doi:10.1007/BF01272517] 22

[37] XUANGUI HUANG AND EMANUELE VIOLA: Average-case rigidity lower bounds. In *Proc. 16th Comp. Sci. Symp. in Russia (CSR'21)*, pp. 186–205. Springer, 2021. Available at author's website. 12

[38] RUSSELL IMPAGLIAZZO: Hard-core distributions for somewhat hard problems. In *Proc. 36th FOCS*, pp. 538–545. IEEE Comp. Soc., 1995. [doi:10.1109/SFCS.1995.492584] 9, 10, 18, 19

[39] HAMID JAHANJOU, ERIC MILES, AND EMANUELE VIOLA: Local reductions. *Inform. Comput.*, 261(2):281–295, 2018. Preliminary version in ICALP15, available at author's website. [doi:10.1016/j.ic.2018.02.009] 7, 10

[40] STASYS JUKNA: *Boolean Function Complexity: Advances and Frontiers*. Springer, 2012. [doi:10.1007/978-3-642-24508-4] 11

[41] ADAM R. KLIVANS: On the derandomization of constant depth circuits. In *Proc. 5th Internat. Workshop on Randomization and Computation (RANDOM'01)*, pp. 249–260. Springer, 2001. [doi:10.1007/3-540-44666-4_28] 10, 19

[42] SWASTIK KOPPARTY AND SRIKANTH SRINIVASAN: Certifying polynomials for $AC^0[\oplus]$ circuits, with applications to lower bounds and circuit compression. *Theory of Computing*, 14(12):1–24, 2018. [doi:10.4086/toc.2018.v014a012] 17

[43] EYAL KUSHILEVITZ AND NOAM NISAN: *Communication Complexity*. Cambridge Univ. Press, 1997. [doi:10.1017/CBO9780511574948] 4

[44] SATYANARAYANA V. LOKAM: Complexity lower bounds using linear algebra. *Found. Trends Theor. Comp. Sci.*, 4(1–2):1–155, 2009. [doi:10.1561/0400000011] 4

[45] DANIEL LOKSHTANOV, RAMAMOHAN PATURI, SUGURU TAMAKI, R. RYAN WILLIAMS, AND HUACHENG YU: Beating brute force for systems of polynomial equations over finite fields. In *Proc. 28th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'17)*, pp. 2190–2202. SIAM, 2017. [doi:10.1137/1.9781611974782.143] 9

[46] SHACHAR LOVETT: Unconditional pseudorandom generators for low-degree polynomials. *Theory of Computing*, 5(3):69–82, 2009. [doi:10.4086/toc.2009.v005a003] 5

[47] PETER BRO MILTERSEN, NOAM NISAN, SHMUEL SAFRA, AND AVI WIGDERSON: On data structures and asymmetric communication complexity. *J. Comput. System Sci.*, 57(1):37–49, 1998. [doi:10.1006/jcss.1998.1577] 23

[48] CODY D. MURRAY AND R. RYAN WILLIAMS: Circuit lower bounds for nondeterministic quasi-polytime from a new easy witness lemma. *SIAM J. Comput.*, 49(5):300–322, 2020. Preliminary version in STOC'18. [doi:10.1137/18M1195887] 7

[49] NOAM NISAN: Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991. [doi:10.1007/BF01375474] 5, 7

[50] RYAN O'DONNELL: *Analysis of Boolean Functions*. Cambridge Univ. Press, 2014. [doi:10.1017/CBO9781139814782] 24

[51] IGOR CARBONI OLIVEIRA, RAHUL SANTHANAM, AND SRIKANTH SRINIVASAN: Parity helps to compute majority. In *Proc. 34th Comput. Complexity Conf. (CCC'19)*, pp. 23:1–17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.CCC.2019.23] 3, 6

[52] ORR PARADISE: Smooth and strong PCPs. *Comput. Complexity*, 30(1):1–77, 2021. Preliminary version in ITCS'20 and ECCC TR19-023. [doi:10.1007/s00037-020-00199-3] 20

[53] NINAD RAJGOPAL, RAHUL SANTHANAM, AND SRIKANTH SRINIVASAN: Deterministically counting satisfying assignments for constant-depth circuits with parity gates, with implications for lower bounds. In *Proc. Internat. Symp. Math. Foundations of Comp. Sci. (MFCS'18)*, pp. 78:1–15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.MFCS.2018.78] 3, 6, 7, 10

[54] SIVARAMAKRISHNAN NATARAJAN RAMAMOORTHY AND CYRUS RASHTCHIAN: Equivalence of systematic linear data structures and matrix rigidity. In *Proc. 11th Innovations in Theoret. Comp. Sci. Conf. (ITCS'20)*, pp. 35:1–20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020. [doi:10.4230/LIPIcs.ITCS.2020.35] 23

[55] ANUP RAO AND AMIR YEHUDAYOFF: *Communication Complexity*. Cambridge Univ. Press, 2019. [doi:10.1017/9781108671644] 4

[56] RAN RAZ: The BNS-Chung criterion for multi-party communication complexity. *Comput. Complexity*, 9(2):113–122, 2000. [doi:10.1007/PL00001602] 4

[57] ALEXANDER RAZBOROV: Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Math. Notes. Acad. Sci. USSR (English translation)*, 41(4):333–338, 1987. Russian original available at Mathnet.ru. [doi:10.1007/BF01137685] 2, 5, 6, 17

[58] ALEXANDER RAZBOROV: On rigid matrices (Russian), 1989. Available in Russian at author's website. 11

[59] JOEL I. SEIFERAS, MICHAEL J. FISCHER, AND ALBERT R. MEYER: Separating nondeterministic time complexity classes. *J. ACM*, 25(1):146–167, 1978. [doi:10.1145/322047.322061] 15, 20

[60] ROCCO A. SERVEDIO AND EMANUELE VIOLA: On a special case of rigidity. *Electron. Colloq. Comput. Complexity*, TR12-144, 2012. Update available at author's website. [ECCC] 4, 19

[61] RONEN SHALTIEL AND EMANUELE VIOLA: Hardness amplification proofs require majority. *SIAM J. Comput.*, 39(7):3122–3154, 2010. Preliminary version in STOC'08. [doi:10.1137/080735096] 5, 9

[62] ROMAN SMOLENSKY: Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. 19th STOC*, pp. 77–82. ACM Press, 1987. [doi:10.1145/28395.28404] 2, 5, 6

[63] ROMAN SMOLENSKY: On representations by low-degree polynomials. In *Proc. 34th FOCS*, pp. 130–138. IEEE Comp. Soc., 1993. [doi:10.1109/SFCS.1993.366874] 2, 5, 6

[64] SRIKANTH SRINIVASAN: On improved degree lower bounds for polynomial approximation. In *Proc. 33rd Found. Softw. Techn. Theoret. Comp. Sci. Conf. (FSTTCS'13)*, pp. 201–212. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013. [doi:10.4230/LIPIcs.FSTTCS.2013.201] 2, 5, 6, 17

[65] SRIKANTH SRINIVASAN, UTKARSH TRIPATHI, AND S. VENKITESH: On the probabilistic degrees of symmetric Boolean functions. *SIAM J. Discr. Math.*, 35(3):2070–2092, 2021. [doi:10.1137/19M129416, arXiv:1910.02465] 2

[66] MADHU SUDAN, LUCA TREVISAN, AND SALIL VADHAN: Pseudorandom generators without the XOR lemma. *J. Comput. System Sci.*, 62(2):236–266, 2001. [doi:10.1006/jcss.2000.1730] 9, 18

[67] SUGURU TAMAKI: A satisfiability algorithm for depth two circuits with a sub-quadratic number of symmetric and threshold gates. *Electron. Colloq. Comput. Complexity*, TR16-100, 2016. [ECCC] 5, 7

[68] LESLIE G. VALIANT: Graph-theoretic arguments in low-level complexity. In *Proc. Internat. Symp. Math. Foundations of Comp. Sci. (MFCS'77)*, pp. 162–176. Springer, 1977. [doi:10.1007/3-540-08353-7_135] 4

[69] EMANUELE VIOLA: The complexity of hardness amplification and derandomization. *Electron. Colloq. Comput. Complexity*, 2006. PhD Thesis, Harvard University. Available at ECCC and author's website. 9

[70] EMANUELE VIOLA: New correlation bounds for GF(2) polynomials using Gowers uniformity. *Electron. Colloq. Comput. Complexity*, TR06-097, 2006. [ECCC] 3

[71] EMANUELE VIOLA: On the power of small-depth computation. *Found. Trends Theor. Comp. Sci.*, 5(1):1–72, 2009. [doi:10.1561/0400000033] 3, 19

[72] EMANUELE VIOLA: The sum of d small-bias generators fools polynomials of degree d. *Comput. Complexity*, 18(2):209–217, 2009. Preliminary version in CCC'08. [doi:10.1007/s00037-009-0273-5] 5

[73] EMANUELE VIOLA: Challenges in computational lower bounds. *SIGACT News, Open Problems Column*, 48(1), 2017. [doi:10.1145/3061640.3061648] 4, 13, 21

[74] EMANUELE VIOLA: Constant-error pseudorandomness proofs from hardness require majority. *ACM Trans. Comput. Theory*, 11(4):19:1–11, 2019. Available at author's website. [doi:10.1145/3322815] 5

[75] EMANUELE VIOLA: Lower bounds for data structures with space close to maximum imply circuit lower bounds. *Theory of Computing*, 15(18):1–9, 2019. [doi:10.4086/toc.2019.v015a018] 23

[76] EMANUELE VIOLA: Fourier conjectures, correlation bounds, and Majority. In *Proc. 48th Internat. Colloq. on Automata, Languages, and Programming (ICALP'21)*, pp. 111:1–15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2021. [doi:10.4230/LIPICS.ICALP.2021.111] 3, 5

[77] EMANUELE VIOLA: New lower bounds for probabilistic degree and ac0 with parity gates. *Electron. Colloq. Comput. Complexity*, TR20-015, 2021. [ECCC] 1

[78] EMANUELE VIOLA AND AVI WIGDERSON: Norms, XOR lemmas, and lower bounds for polynomials and protocols. *Theory of Computing*, 4(7):137–168, 2008. Preliminary version in CCC'07. [doi:10.4086/toc.2008.v004a007] 3, 4

[79] NIKHIL VYAS AND R. RYAN WILLIAMS: Lower bounds against sparse symmetric functions of ACC circuits: Expanding the reach of #SAT algorithms. In *Proc. 37th Symp. Theoret. Aspects of Comp. Sci. (STACS'20)*, pp. 59:1–17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020. [doi:10.4230/LIPIcs.STACS.2020.59] 7

[80] R. RYAN WILLIAMS: Guest column: A casual tour around a circuit complexity bound. *SIGACT News*, 42(3):54–76, 2011. [doi:10.1145/2034575.2034591] 7

[81] R. RYAN WILLIAMS: Improving exhaustive search implies superpolynomial lower bounds. *SIAM J. Comput.*, 42(3):1218–1244, 2013. [doi:10.1137/10080703X] 7

[82] R. RYAN WILLIAMS: Natural proofs versus derandomization. In *Proc. 45th STOC*, pp. 21–30. ACM Press, 2013. [doi:10.1145/2488608.2488612] 7

[83] R. RYAN WILLIAMS: New algorithms and lower bounds for circuits with linear threshold gates. In *Proc. 46th STOC*, pp. 194–202. ACM Press, 2014. [doi:10.1145/2591796.2591858] 7

[84] R. RYAN WILLIAMS: Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2:1–32, 2014. [doi:10.1145/2559903] 7, 9, 13, 14

[85] R. RYAN WILLIAMS: Limits on representing Boolean functions by linear combinations of simple functions: Thresholds, relus, and low-degree polynomials. In *Proc. 33rd Comput. Complexity Conf. (CCC'18)*, pp. 6:1–24. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.CCC.2018.6] 9, 14

[86] HENNING WUNDERLICH: On a theorem of Razborov. *Comput. Complexity*, 21(3):431–477, 2012. [doi:10.1007/s00037-011-0021-5] 11

[87] ANDREW CHI-CHIH YAO: Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *Proc. 18th FOCS*, pp. 222–227. IEEE Comp. Soc., 1977. [doi:10.1109/SFCS.1977.24] 3, 6

[88] ANDREW CHI-CHIH YAO: On ACC and threshold circuits. In *Proc. 31st FOCS*, pp. 619–627. IEEE Comp. Soc., 1990. [doi:10.1109/FSCS.1990.89583] 14

[89] STANISLAV ŽÁK: A Turing machine time hierarchy. *Theoret. Comput. Sci.*, 26:327–333, 1983. [doi:10.1016/0304-3975(83)90015-4] 15, 20

AUTHOR

Emanuele Viola
Associate professor
Khoury College of Computer Sciences
Northeastern University
Boston, MA
viola@ccs.neu.edu
http://www.ccs.neu.edu/~viola

ABOUT THE AUTHOR

EMANUELE VIOLA is happy that *Theory of Computing* exists.