Electronic Colloquium on Computational Complexity, Report No. 15 (2020)

New lower bounds for probabilistic degree and AC0 with parity gates

Emanuele Viola[*]

February 18, 2020

Abstract

We prove new lower bounds for computing some functions $f : \{0,1\}^n \to \{0,1\}$ in E^{NP} by polynomials modulo 2, constant-depth circuits with parity gates ($AC^0[\oplus]$), and related classes. Results 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 $AC^0[\oplus]$ 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 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 $AC^0[\oplus]$ circuits under the uniform distribution, for say polynomial or quasi-polynomial s, and any fixed h. The previous best was $(1/2 - (\log^{O(h)} s)/\sqrt{n})$.

(4) any majority of t $AC^0[\oplus]$ circuits, $MAJ_t \circ AC^0[\oplus]$, of size s and depth h, has $t \geq n^2/\log^{O(h)}(s)$, for any s, h. The previous best was $t \geq n/\log^{O(h)}(s)$.

(5) any $AC^0[\oplus] \circ LTF_t \circ AC^0[\oplus] \circ LTF$ circuit, where LTF are threshold functions, has $t \geq n/\log^{O(h)}(s)$, for any s, h. The previous best was $t \geq \sqrt{n}/\log^{O(h)}(s)$ recently proved by Alman and Chen.

The mentioned previous best lower bounds in (1), (3), and (4) held for the Majority function. Each of the new lower bounds in this paper is false for Majority. For (2) and (5) the previous best held for E^{NP}.

The proofs build on Williams' "guess-and-SAT" method. For (1) we show how to use a PCP by Ben-Sasson and Viola towards probabilistic-degree lower bounds. Results (2), (4), and (5) are then more or less automatic, as is (3) under a non-uniform distribution. To strengthen (3) to hold under the uniform distribution we use a different argument which combines the recent work by Alman and Chen with hardness amplification.

A concurrent work by Chen and Ren obtains a result stronger than (3).

[*] Supported by NSF CCF award 1813930.

ISSN 1433-8092

1 Introduction

Probabilistic degree is a fundamental complexity measure of boolean functions that has been intensely studied since it was introduced by Razborov [Raz87]. Probabilistic degree measures how well a function can be computed by a "random" polynomial. 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. 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 d such that $\mathbb{P}[D(x) \neq f(x)] \leq \epsilon$ 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 work 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.

Papers by Razborov and Smolensky [Raz87, Smo87, Smo93] proved that the Majority function on n bits has constant-error degree $\Omega(\sqrt{n})$. The notation O and Ω denotes absolute constants. 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 [AW15], Theorem 1.2, show that any symmetric function on n bits has ϵ-error polynomials of degree $O(\sqrt{n \log(1/\epsilon)})$. Earlier, Srinivasan [Sri13] proved a slightly weaker bound. The recent work [STV19] gives a nearly tight characterization of the probabilistic degree of symmetric functions. Srinivasan [Sri13], Theorem 12, also proved that threshold functions (with arbitrary weights) have ϵ-error polynomials of degree $\sqrt{n}(\log n \cdot \log 1/\epsilon)^{O(1)}$. [GKW18] 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 degree $\Omega(\sqrt{n})$ 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

$$\epsilon \geq 1/2 - \Omega(d/\sqrt{n}). \tag{1}$$

This tradeoff is tight for Majority [Vio19], and is the best available for an explicit function, for any degree $d \geq \log_2 n$. In particular, conceivably every explicit function has $(1/2-1/\sqrt{n})$-degree $\log_2 n$. For small degrees, $d < \log_2 n$, the iterated Cauchy-Schwarz argument of [BNS92] gives stronger error bounds. See [Vio06b, VW08, Vio09a] 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 [Yao77], 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 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 hardness lower bounds are known if we allow a non-uniform distribution. Hardness results for polynomials are surveyed in [Vio09a], 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 improving the trade-off (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 18).

Circuit lower bounds for $AC^0[\oplus]$. The circuits $AC^0[\oplus]$ 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 works by Razborov and Smolensky used probabilistic-degree lower bounds to obtain lower bounds of $\exp(\Omega(n^{1/(2h-2)}))$ on the size of $AC^0[\oplus]$ circuits of depth h. This bound was recently improved to $\exp(\Omega(n^{1/(h+1)}))$ by Rajgopal, Santhanam, and Srinivasan [RSS18], see also [OSS19]. Still, these lower bounds are lower than the $\exp(\Omega(n^{1/(h-1)}))$ that can be obtained for AC^0 via switching lemmas [Hås87].

The gap between AC^0 and $AC^0[\oplus]$ 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 (1), then we would obtain lower bounds for $AC^0[\oplus]$ that are as strong as the AC^0 ones.

Also, we don't have strong hardness lower bounds. It is consistent with our knowledge that for every explicit function there is a polynomial-size $AC^0[\oplus]$ 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 $AC^0[\oplus]$ circuits of size $n^{O(d)}$, so this is similar (but not equivalent to) the fact that we don't have such strong hardness lower bounds for degree $\log n$.

Rigidity. More than forty years ago Valiant [Val77] asked to construct matrices that cannot be approximated by matrices with 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 ϵ-hard for rank-r functions. Again this can be considered over different fields and we focus on $\mathbb{F}_2$ in this work. 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 work by Alman and Chen [AC19] constructed in the complexity class $Time(2^{O(n)})^{NP} = E^{NP}$ matrices M with 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 [AC19] and Lokam's survey [Lok09].

Probabilistic-degree lower bounds stand in the way of further progress. Indeed, as pointed out by Servedio and Viola [SV12], 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 [AC19] to any rank $2^{\omega(\sqrt{n}\log n)}$. (We can remove the log factor when considering *sparsity,* see [SV12] for discussion. Here we focus on *degree* for simplicity.)

Communication complexity. In the influential number-on-forehead communication model introduced by Chandra, Furst, and Lipton [CFL83], 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. For this model, we have no lower bounds when $k \geq \log n$ (for $k \leq \log n$ they were first proved in [BNS92], see [CT93, Raz00, VW08] for expositions). For background on communication complexity see the books [KN97, RY19].

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 [Vio17], see Theorem 25, 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) is necessary to rule out such protocols.

Other circuit classes. Many papers in the literature study $AC^0[\oplus]$ circuits augmented with other types of gates. In fact, average-case lower bounds for $AC^0[\oplus]$ are closely related to worst-case lower bounds for the circuit class $Maj \circ AC^0[\oplus]$ of such circuits with a majority gate at the output [HMP+93]. Correspondingly, we only have lower bounds when the fan-in of the majority gate is $t < n/\text{poly}\log(n)$. Such a lower bound follows simply by approximating the entire circuit by a polynomial of degree $\sqrt{n}/\text{poly}\log(n)$, and then using the probabilistic degree lower bound. Thus this lower bound holds for Majority, and is in fact almost optimal for that function because obviously we can compute majority with $t = n$.

Alman and Chen [AC19] consider the class of $AC^0[\oplus] \circ Ltf_t \circ AC^0[\oplus] \circ Ltf$ 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 ($ACC^0 \circ LTF_{n^{2-\Omega(1)}} \circ LTF$, [ACW16], Corollary 1.1), and Tamaki (depth-2 circuits with $n^2/\text{poly}\log(n)$ gates, where each gate can be any symmetric function or an Ltf, [Tam16]).

Pseudorandom generators. Another important motivation for improving the tradeoff (1) and average-case lower bounds for $AC^0[\oplus]$ is the construction of *pseudorandom generators.* Nisan's landmark paper [Nis91] 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 would yield better generators for low-degree polynomials [BV10, Lov09, Vio09b], a long-standing problem for which several new approaches have recently been proposed [CHHL18, CHLT19]. Nisan's approach yields pseudorandom generators with small error, in which case improving probabilistic degree lower bounds is known to be necessary [Vio09b].

The situation with constant-error generators is less clear. We remark that for pseudorandom generators one needs hardness *under the uniform distribution,* and that other pseudorandom-generator constructions that have been proposed do not go through for classes such as $AC^0[\oplus]$, see [SV10, GSV18, Vio].

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.

1.1 Our results

In this work we show that E^{NP} requires nearly-linear constant-error degree, improving on the $\Omega(\sqrt{n})$ lower bounds established by Razborov and Smolensky [Raz87, Smo87, Smo93] for the majority function.

Theorem 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 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 3 is false for any symmetric or threshold function [Sri13, AW15].

Theorem 3 has several corollaries stated next. First, as mentioned earlier, it actually improves the tradeoff (1) for every setting of parameters (and not just constant error).

Corollary 4. *The function in Theorem 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) we could not set ϵ 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 $AC^0[\oplus]$.

Corollary 5. *Any depth-h $AC^0[\oplus]$ circuit computing f in Theorem 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 [RSS18]. The bound in Corollary 5 is false for any symmetric function [OSS19], and it almost matches the $\exp(\Omega(n^{1/(h-1)}))$ lower bounds that are available for AC^0 via switching lemmas [Hås87].

Next we prove a general lower bound for $AC^0[\oplus]$ 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 $n^2/poly\log(n)$. First we formally define the class.

Definition 6. The class t-SoT-$AC^0[\oplus]$ 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 7. *Any t-SoT-$AC^0[\oplus]$ circuit of depth h and size s computing the function f in Theorem 3 requires $t \geq n^2/\log^{O(h)} s$, for infinitely many n.*

In particular, $Maj_t \circ AC^0[\oplus]$ 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 [Raz87] and Smolensky [Smo87, Smo93] and the probabilistic degree of Majority [Sri13, AW15].

Also, $AC^0[\oplus] \circ Ltf_t \circ AC^0[\oplus] \circ Ltf$ 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 [AC19]. 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.

Uniform distribution. Corollary 4 implies that for any distribution C on polynomial-size constant-depth $AC^0[\oplus]$ circuits one has $\mathbb{P}_C[C(x) \neq f(x)] \geq 1/2 - \text{poly}\log(n)/n$ for some fixed input x. As mentioned earlier, Yao's duality principle [Yao77] 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 8. *Let $s = s(n)$ and $h = h(n)$ be non-decreasing functions satisfying $s(O(n \log n)) \leq s^8(n)$ and $h(O(n \log n)) \leq h(n) + 9$.*

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 $AC^0[\oplus]$ circuits, for infinitely many n.

Note that in particular this holds for $s(n) = n^c$ and $h = c$ for any constant c. It also holds for quasi-polynomial $s(n) = 2^{\log^c n}$.

Again, the previous best was hardness $1/2 - (\log^{O(h)} s)/\sqrt{n}$, under any distribution. Theorem 8 gets us closer to a correlation bound that could be used to obtain new pseudorandom generators via Nisan's method [Nis91]. For example hardness $(1/2 - 1/n^{2.1})$ would already imply a new generator for $AC^0[\oplus]$. Perhaps, the result in Theorem 8 can already be used to improve on the seed length of the state-of-the-art generator from [FSUV13].

A concurrent paper by Chen and Ren [CR20] obtains a result stronger than Theorem 8. ([CR20] and the present paper were independently submitted to the same conference.)

1.2 Techniques

The results in this paper rely on a masterful method by Williams [Wil13a] 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' ACC^0 lower bound [Wil14b], see [Wil11, Wil13b, Wil14a, ACW16, Tam16, Che19, COS18, MW18, AC19].

To explain the proof of Theorem 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 may 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 efficiently to 3SAT with about 2^n clauses (and variables) using the Cook-Levin theorem. We then consider the function f that on input x and i computes the i bit of the satisfying assignment corresponding to x, if it is the case that $x \in L$. One can show $f \in E^{NP}$. 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 method does not work this way, due to the lack of reductions to 3SAT with efficient indexing circuit. This modular version of the method has been obtained in [JMV18].

Using PCP. In the original paper [Wil13a] 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 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 [BV14] building on [BGH$^+$05, BS08]. 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 [BV14]. 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.

Our approach. To go around these issues we begin by modifying slightly the PCP in [BV14]. 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 [BV14] 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

$\leq 1 - 0.5/n^b$ for some fixed b.

Now an important point is that we are allowed to *amplify the error* of the polynomial. 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 PCP cannot yet be written as 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, the acceptance probability of the PCP is now 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 3, we note that one can compute the number of satisfying assignments to a degree-d polynomial in n variables in time $2^{n-\Omega(n/d)+O(\log n)}$. The algorithm for this is a fairly straightforward combination of techniques that have been used in the corresponding algorithms for ACC^0, *Orthogonal Vectors,* and *systems of polynomial equations* [Wil14b, CW16, LPT$^+$17]. However we are not aware that this claim appears in the literature. A direct application of the Orthogonal Vectors algorithm gives slightly worse bounds (see Section 2.1). [LPT$^+$17] obtain running time $2^{n-\Omega(n/d)+o(n)}$ which is not sufficient for our purposes. This concludes the overview of the proof of Theorem 3.

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

Proof of Theorem 8. To prove the lower bound under the uniform distribution in Theorem 8 we use a different argument. Here our starting point is a *conditional* result by Alman and Chen [AC19]. Under the assumption that non-deterministic quasi-polynomial time has polynomial-size circuits (i.e., $NQP \subseteq P/\text{poly}$) 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 [AC19] (see Section 5).

At the high level, Theorem 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 [AB09], and to the discussion in [SV10].

However, the use of hardness amplification is not straightforward. One issue is that hardness amplification in general cannot be used for classes such as $AC^0[\oplus]$, see [GSV18] and the discussion in [SV10, Vio06a]. 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 $AC^0[\oplus]$ 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 [BFNW93, Imp95, STV01a], 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 $MAJ_{n^{2-2\alpha-o(1)}} \circ AC^0[\oplus]$ 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 [GNW95] which follows from Impagliazzo's theory of hard-core sets [Imp95] and essentially appears in a beautiful paper by Klivans [Kli01]. The analysis essentially says that if a function f is $\Omega(1)$-hard for $MAJ_{t^2} \circ C$ circuits then 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.)

Comparison with rigidity lower bound in [AC19]. We mentioned earlier that Alman and Chen gave [AC19] 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 [AC19] 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. [AC19]'s argument uses PCPs twice. The first application is similar to what we described above. The second application, following [CW19], 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 [AC19] 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 run-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 [AC19] also contains an argument which they call bootstrapping. As discussed in [AC19], bootstrapping runs into difficulties when the rank is larger than $\sqrt{n}$; the same difficulties arise for degree larger than $\sqrt{n}$.

1.3 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 2) for low-rank matrices. Razborov introduced [Raz89] 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 hardness with respect to *some* distribution. Following the literature we call this notion *probabilistic rank*. Probabilistic rank has been studied by several works including [Wun12, AW17]. 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 [Wun12], and for $AC^0[\oplus] \circ Ltf_t \circ AC^0[\oplus] \circ Ltf$ lower bounds, see [AC19, AW17]. For another application to circuits operating on matrices see Jukna's book [Juk12], Chapter 12.8.

It is natural to ask if Theorem 3 can be extended to bound probabilistic rank as well. From our approach it follows that *either* a strengthening of the PCP [BV14] *or* a strengthening of the algorithms for counting orthogonal vectors [AWY15, CW16] 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 [BV14] 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 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 9 considers the sum of three low-rank functions of *different projections of* $2n$ *bits.* As pointed out in [AC19], when the projections are the same this can be reduced to a single matrix, and for that one can apply algorithms for *orthogonal vectors* [CW16]. It is not clear what happens for different projections. 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 $2^{2n-\Omega(n/\log r)}$?

Organization. In Section 2 we prove the probabilistic-degree lower bound Theorem 3. The corollaries are then proved in Section 3. The hardness result under the uniform distribution, Theorem 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 [Vio17] between communication protocols and probabilistic degree.

2 Probabilistic degree lower bound

In this section we prove Theorem 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 [BV14]. This PCP in turn is a relatively minor modification of the PCP [BGH$^+$05] 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 [BV14].

First we state the result from [BV14], then state and prove our variant.

Lemma 10. *[BV14] 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 11. *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 10. 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 $R(\pi(q_1), \ldots, \pi(q_t)) = 0$, 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 10, 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 #SAT poly

In this subsection we show how to count efficiently the satisfying assignments of a low-degree polynomial.

Theorem 12. *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 ([CW16] 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 [CW16] Theorem 1.2.) Instead we use a classic dynamic-programming approach going back to Yates in the 30's, see [Wil14b].

Lemma 13. *[Modulus-amplifying polynomials] [Yao90, BT94] 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 $\{0, 1, \ldots, 2^\ell - 1\}$.

Lemma 14. *[Yates in the 30's, see [Wil14b]] Let $f : \{0,1\}^n \to \{0, 1, \ldots, 2^{\text{poly}(n)}\}$. Given the truth-table of f we can compute in time $2^n\text{poly}(n)$ 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 $\{1, 2, \ldots, n\}$.*

The proof of this lemma is an undergraduate exercise on dynamic programming.

Proof. [Of Theorem 12] Let $\ell := \alpha n/d$ for an $\alpha > 0$ to be determined later. View p as a polynomial over the integers (with coefficients in $\{0,1\}$). Let $F_{\ell+1}$ be the modulus amplifying polynomial of degree $2(\ell+1) - 1$. 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' is $M := \sum_{i=0}^{O(\alpha n)} \binom{n}{i} \leq 2^{H(O(\alpha))n}$ where H is the binary entropy function. The coefficients of p' have $O(\ell)$ bits.

Note that we can write down p' in time poly$(\ell) \cdot$ poly(M). For example we can perform the poly(ℓ) operations in the definition of F_ℓ to the polynomial p'.

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

We can write down p'' in time $2^\ell \cdot \text{poly}(M) \cdot \text{poly}(\ell)$. 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 not "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 $f : \{0,1\}^{n-\ell} \to \{0, 1, \dots, 2^{O(\ell)}\}$.Lemma 14 solves the evaluation problem in time $2^{n-\ell} \cdot \text{poly}(n)$.

Overall, the algorithm's running time is

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

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

2.3 Proof of Theorem 3

Assume towards a contradiction that every function in E^{NP} has 1/3-error polynomials of degree $d = \alpha n / \log^2 n$. 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 $f \in E^{NP}$ on n bits has $1/n^c$-error polynomials of degree $d' := O(dc \log n)$. We will set c later.

Let $L \in \text{NTime}(2^n) \setminus \text{NTime}(o(2^n))$ [Coo73, SFM78, Zák83]. We want to show that $L \in NTime(o(2^n))$ to reach a contradiction. Consider the algorithm $f(x, y)$ that on input $x \in \{0,1\}^n$ and $y \in \{1, 2, \dots, 2^n \text{poly}(n)\}$ constructs the circuits from the PCP in Lemma 11, 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.

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 $x \in L$:

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

The bound $1 - 3/n^c$ follows by a union bound; but 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 and so in particular for $p = \pi$ we have:

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

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 $\ell = \sum_{i=0}^{d'} \binom{n'}{i} \leq (en'/d')^{d'} \leq n^{d'} = n^{O(dc \log n)} = 2^{O(\alpha cn)}$ 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 sum the values. Because the number of i is only $\text{poly}(r) = \text{poly} \log T = \text{poly}(n)$, 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 $\text{poly}(n)$ by Lemma 11. Composing these with p' can be done in time polynomial in the length of the representation of p' which is $\text{poly}(\ell)$. 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 12. The degree of q is $O(d') \leq O(\alpha cn/ \log n)$. The running time of the algorithm in Lemma 12 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 4, 5, and 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 [Raz87]. We use the recent tighter version by Kopparty and Srinivasan [KS18], who also show their bound is nearly tight.

Lemma 15. *[KS18] Any $AC^0[\oplus]$ circuit of depth d and size s has ϵ-error degree $\leq O(\log s)^{d-1} \log(1/\epsilon)$.*

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

Lemma 16. *([AW15], Theorem 1.2). Any symmetric function on n bits has ϵ-error polynomials of degree $O(\sqrt{n \log(1/\epsilon)})$.*

Lemma 17. *([Sri13], 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 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 7.

Lemma 18. *Suppose that $f : \{0,1\}^n \to \{0,1\}$ has $(1/2 - \epsilon)$-error 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 $O(\sqrt{t})$ [AW15]. 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 4 is immediate. Now we prove the other two corollaries.

Proof. [of Corollary 5] We apply Lemma 15 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 3 we have $O(\log s)^{h-1} \geq n/\log^2 n$ and the result follows. □

Proof. [of Corollary 7] We approximate each And and Or gate by a degree-$\log^{O(1)}(s)$ polynomial with error $0.1/s$ by Lemma 15. 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 16 and 17. By a union bound, the error is as desired. 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 $h' = 1$, using the fact that the fan-in of a gate at this level is at most n. 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 $t'' \leq t'/m$. 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 8

In this section we prove Theorem 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 19. *[BFNW93, Imp95, STV01b] Let $s = s(n)$ satisfy $s(O(n)) \leq s^{O(1)}(n)$. 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 $s(n) \geq n$ such that $\mathbb{P}_{x \in \{0,1\}^n}[C(x) = f(x)] \geq 1/2 + 1/n$. 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 20. For the proof we can use Theorem 24 and Lemma 28 in [STV01b]. That gives the size bound $s' = n^{O(1)} \cdot s(O(n))$ which is $\leq s^{O(1)}(n)$ 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 [COS18], 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 [STV01b] 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 [Imp95] of Yao's XOR lemma (see [GNW95]), as was made explicit in a beautiful paper by Klivans [Kli01]. 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' 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 [Vio09a]). 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 21. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function that is $\Omega(1)$-hard for $Maj_t \circ AC^0[\oplus]$ circuits of size s and depth h. Then the function $f' : (\{0,1\}^n)^k \to \{0,1\}$ defined as $f'(x_1, x_2, \ldots, x_k) := \oplus_{i \leq k} f(x_i)$ is $(1/2 - O(1/\sqrt{t}) - (1 - \Omega(1))^k)$-hard for $AC^0[\oplus]$ circuits of size s/t and depth $h - 1$.*

Remark 22. The combination of Theorem 8 and Fact 9 in [Kli01] show that there exists a "hard-core" set $S \subseteq \{0,1\}^n$ of size $\Omega(2^n)$ such that for every $AC^0[\oplus]$ circuit C of size s/t and depth $h - 1$ one has $\mathbb{P}_{x \in S}[C(x) \neq h(x)] \geq 1/2 - O(1/\sqrt{t})$. 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 [Imp95] or Theorem 10 in [Kli01]).

4.2 A result by Alman and Chen

We need the following result which is essentially Theorem 1.9 by Alman and Chen [AC19].

Lemma 23. *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 [AC19] claims $\Omega(1)$-hardness for rank $n^{1-\alpha}$ for any $\alpha > 0$. As mentioned earlier and in [SV12], 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 8

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

By Lemma 19, E^{NP} has circuits of size $s^{O(1)}(n)$.

By Lemma 23, 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 $MAJ_{n^2/\log^{O(h)} s} \circ AC^0[\oplus]$ circuits of size say $s^{10}(n)$ and depth $h + 10$. This is because the latter circuits have $o(1)$-error degree $d := \sqrt{n^2/\log^{O(h)} s} \cdot \log^{h+O(1)} s \leq n/\log^{O(h)} s \leq n/\log s$ by Lemma 16 and 15. 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 21 to construct a function on nk bits for $k = O(\log n)$ which, for infinitely many n, has hardness $1/2 - 1/\sqrt{n^2/\log^{O(h)} s} - 1/n \geq 1/2 - \log^{O(h)} s/n$ for circuits of size $s^{10}/(n^2/\log^{O(h)} s) \geq s^8$, where the inequality holds without loss of generality, and depth $(h + 10) - 1 = h + 9$. Here we are using that $(1 - \Omega(1))^k \leq 1/n$ for $k = O(\log n)$.

To contradict our initial assumption, for a large enough constant c, it suffices that $s(O(n \log n)) \leq s^8(n)$ and $h(O(n \log n)) \leq h(n) + 9$. Indeed, these are our assumptions.

5 Proof of Lemma 23

We need the following recent construction of *smooth* PCPs of proximity by Paradise [Par19], which we state using notation similar to that in Section 2.1.

Lemma 24. *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 $R(y(q'_1), \ldots, y(q'_c), \pi(q_1), \ldots, \pi(q_c)) = 1$ where $(q_1, \ldots, q_c) = Q(z)$, $(q'_1, \ldots, q'_c) = Q'(z)$, 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 $R(y(q'_1), \ldots, y(q'_c), \pi(q_1), \ldots, \pi(q_c)) = 1$ where $(q_1, \ldots, q_c) = Q(z)$, $(q'_1, \ldots, q'_c) = Q'(z)$,*

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 23] As in the proof of Theorem 3, let $L \in \mathrm{NTime}(2^n) \setminus \mathrm{NTime}(o(2^n))$ [Coo73, SFM78, Zák83]. We want to show that $L \in NTime(o(2^n))$ 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 $y \in \{1,2,\ldots,2^n\mathrm{poly}(n)\}$ constructs the circuits from the PCP in Lemma 10, 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 $L \in NTime(o(2^n))$ proceeds by guessing this circuit C, and composing it with the PCP in Lemma 10 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 24 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 24 to circuit C'' we see that if $x \in L$ then there are proofs π_z such that

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

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

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

where $(q_1,\ldots,q_c) = Q(z')$, $(q_1',\ldots,q_c') = Q'(z')$, and picking the soundness in the PCPs in Lemmas 10 and 24 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 $\mathbb{P}_{z,q}[p(z,q) \neq f(z,q)] \leq \beta$. Because the queries q are smooth, and by a union bound, we have that if $x \in L$ then

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

while if $x \notin L$ then the above probability is $\leq 1/2$ 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.

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

$$R(ECC(z)(q_1'),\ldots,ECC(z)(q_c'),p(z,q_1),\ldots,p(z,q_c))$$

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 12 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 [Vio17]. We present next a stronger quantitative relationship.

Theorem 25. *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 $f(x) = Maj(p_1(x), \ldots, p_t(x))$ 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 13. 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 $Maj(p_1(x), \ldots, p_t(x))$ is determined by

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

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

By Håstad and Goldmann's simulation [HG91], $q(x)$ can be computed by a number-on-forehead protocol with $O(d \log t)$ parties. Each party just sends the sum of a subset of the monomials in q. This sum has magnitude at most the magnitude of one coefficient times the number of monomials in q, which is $\le \text{poly}(t) \cdot n^{O(d \log t)}$. The number of bits is then $O(d \log(n/\epsilon) \cdot \log n)$. □

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity.* Cambridge University Press, 2009. A modern approach.

[AC19] Josh Alman and Lijie Chen. Efficient construction of rigid matrices using an NP oracle. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 2019.

[ACW16] Josh Alman, Timothy M. Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 2016.

[AW15] Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 136–150, 2015.

[AW17] Josh Alman and R. Ryan Williams. Probabilistic rank and matrix rigidity. In *ACM Symp. on the Theory of Computing (STOC)*, pages 641–652, 2017.

[AWY15] Amir Abboud, Richard Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *SODA*, pages 218–230. SIAM, 2015.

[BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1993.

[BGH$^+$05] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Short PCPs verifiable in polylogarithmic time. In *IEEE Conf. on Computational Complexity (CCC)*, pages 120–134, 2005.

[BNS92] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *J. of Computer and System Sciences*, 45(2):204–232, 1992.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM J. on Computing*, 38(2):551–607, 2008.

[BT94] Richard Beigel and Jun Tarui. On ACC. *Computational Complexity*, 4(4):350–366, 1994.

[BV10] Andrej Bogdanov and Emanuele Viola. Pseudorandom bits for polynomials. *SIAM J. on Computing*, 39(6):2464–2486, 2010.

[BV14] Eli Ben-Sasson and Emanuele Viola. Short PCPs with projection queries. In *Coll. on Automata, Languages and Programming (ICALP)*, 2014.

[CFL83] Ashok K. Chandra, Merrick L. Furst, and Richard J. Lipton. Multi-party protocols. In *15th ACM Symp. on the Theory of Computing (STOC)*, pages 94–99, 1983.

[Che19] Lijie Chen. Non-deterministic quasi-polynomial time is average-case hard for ACC circuits. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 2019.

[CHHL18] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. In *CCC*, volume 102 of *LIPIcs*, pages 1:1–1:21. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[CHLT19] Eshan Chattopadhyay, Pooya Hatami, Shachar Lovett, and Avishay Tal. Pseudorandom generators from the second fourier level and applications to AC0 with parity gates. In *ACM Innovations in Theoretical Computer Science conf. (ITCS)*, pages 22:1–22:15, 2019.

[Coo73] Stephen A. Cook. A hierarchy for nondeterministic time complexity. *J. of Computer and System Sciences*, 7(4):343–353, 1973.

[COS18] Ruiwen Chen, Igor Carboni Oliveira, and Rahul Santhanam. An average-case lower bound against \mathsf acc^0 ACC 0. In *LATIN*, volume 10807 of *Lecture Notes in Computer Science*, pages 317–330. Springer, 2018.

[CR20] Lijie Chen and Hanlin Ren. Strong average-case circuit lower bounds from nontrivial derandomization. In *ACM Symp. on the Theory of Computing (STOC)*, 2020.

[CT93] Fan R. K. Chung and Prasad Tetali. Communication complexity and quasi randomness. *SIAM Journal on Discrete Mathematics*, 6(1):110–123, 1993.

[CW16] Timothy M. Chan and Ryan Williams. Deterministic APSP, orthogonal vectors, and more: Quickly derandomizing Razborov-Smolensky. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1246–1255. SIAM, 2016.

[CW19] Lijie Chen and R. Ryan Williams. Stronger connections between circuit analysis and circuit lower bounds, via PCPs of proximity. In *CCC*, volume 137 of *LIPIcs*, pages 19:1–19:43. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.

[FSUV13] Bill Fefferman, Ronen Shaltiel, Christopher Umans, and Emanuele Viola. On beating the hybrid argument. *Theory of Computing*, 9:809–843, 2013.

[GKW18] Alexander Golovnev, Alexander S. Kulikov, and Ryan Williams. Circuit depth reductions. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:192, 2018.

[GNW95] Oded Goldreich, Noam Nisan, and Avi Wigderson. On Yao's XOR lemma. Technical Report TR95–050, *Electronic Colloquium on Computational Complexity*, March 1995. www.eccc.uni-trier.de/.

[GSV18] Aryeh Grinberg, Ronen Shaltiel, and Emanuele Viola. Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 2018. Available at http://www.ccs.neu.edu/home/viola/.

[Hås87] Johan Håstad. *Computational limitations of small-depth circuits.* MIT Press, 1987.

[HG91] Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1(2):113–129, 1991.

[HMP$^+$93] András Hajnal, Wolfgang Maass, Pavel Pudlák, Márió Szegedy, and György Turán. Threshold circuits of bounded depth. *J. of Computer and System Sciences*, 46(2):129–154, 1993.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 538–545, 1995.

[JMV18] Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local reductions. *Information and Computation*, 261(2), 2018. Available at http://www.ccs.neu.edu/home/viola/.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers.* Springer, 2012.

[Kli01] Adam R. Klivans. On the derandomization of constant depth circuits. In *Workshop on Randomization and Computation (RANDOM).* Springer, 2001.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, 1997.

[KS18] Swastik Kopparty and Srikanth Srinivasan. Certifying polynomials for \$\mathrm{AC}^0[\oplus]\$ circuits, with applications to lower bounds and circuit compression. *Theory of Computing*, 14(1):1–24, 2018.

[Lok09] Satyanarayana V. Lokam. Complexity lower bounds using linear algebra. *Foundations and Trends in Theoretical Computer Science*, 4(1-2):1–155, 2009.

[Lov09] Shachar Lovett. Unconditional pseudorandom generators for low degree polynomials. *Theory of Computing*, 5(1):69–82, 2009.

[LPT$^+$17] Daniel Lokshtanov, Ramamohan Paturi, Suguru Tamaki, R. Ryan Williams, and Huacheng Yu. Beating brute force for systems of polynomial equations over finite fields. In Philip N. Klein, editor, *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 2190–2202. SIAM, 2017.

[MW18] Cody Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In *STOC*, pages 890–901. ACM, 2018.

[Nis91] Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica. An Journal on Combinatorics and the Theory of Computing*, 11(1):63–70, 1991.

[OSS19] Igor Carboni Oliveira, Rahul Santhanam, and Srikanth Srinivasan. Parity helps to compute majority. In *Conf. on Computational Complexity (CCC)*, volume 137 of *LIPIcs*, pages 23:1–23:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.

[Par19] Orr Paradise. Smooth and strong pcps. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:23, 2019.

[Raz87] Alexander Razborov. Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function. *Akademiya Nauk SSSR. Matematicheskie Zametki*, 41(4):598–607, 1987. English translation in Mathematical Notes of the Academy of Sci. of the USSR, 41(4):333-338, 1987.

[Raz89] Alexander Razborov. On rigid matrices (russian), 1989.

[Raz00] Ran Raz. The BNS-Chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.

[RSS18] Ninad Rajgopal, Rahul Santhanam, and Srikanth Srinivasan. Deterministically counting satisfying assignments for constant-depth circuits with parity gates, with implications for lower bounds. In Igor Potapov, Paul G. Spirakis, and James Worrell, editors, *Symp. on Math. Foundations of Computer Science (MFCS)*, volume 117 of *LIPIcs*, pages 78:1–78:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[RY19] Anup Rao and Amir Yehudayoff. *Communication complexity.* 2019. https://homes.cs.washington.edu/ anuprao/pubs/book.pdf.

[SFM78] Joel I. Seiferas, Michael J. Fischer, and Albert R. Meyer. Separating nondeterministic time complexity classes. *J. of the ACM*, 25(1):146–167, 1978.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *19th ACM Symp. on the Theory of Computing (STOC)*, pages 77–82. ACM, 1987.

[Smo93] Roman Smolensky. On representations by low-degree polynomials. In *34th IEEE IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 130–138, 1993.

[Sri13] Srikanth Srinivasan. On improved degree lower bounds for polynomial approximation. In *FSTTCS*, volume 24 of *LIPIcs*, pages 201–212. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[STV01a] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *J. of Computer and System Sciences*, 62:236–266, 2001.

[STV01b] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *J. of Computer and System Sciences*, 62(2):236–266, 2001.

[STV19] Srikanth Srinivasan, Utkarsh Tripathi, and S. Venkitesh. On the probabilistic degrees of symmetric boolean functions, 2019.

[SV10] Ronen Shaltiel and Emanuele Viola. Hardness amplification proofs require majority. *SIAM J. on Computing*, 39(7):3122–3154, 2010.

[SV12] Rocco A. Servedio and Emanuele Viola. On a special case of rigidity. Available at http://www.ccs.neu.edu/home/viola/, 2012.

[Tam16] Suguru Tamaki. A satisfiability algorithm for depth two circuits with a subquadratic number of symmetric and threshold gates. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:100, 2016.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *6th Symposium on Mathematical Foundations of Computer Science*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977.

[Vio] Emanuele Viola. Constant-error pseudorandomness proofs from hardness require majority. *ACM Trans. Computation Theory.* Available at http://www.ccs.neu.edu/home/viola/.

[Vio06a] Emanuele Viola. The complexity of hardness amplification and derandomization. *Ph.D. thesis, Harvard University*, 2006.

[Vio06b] Emanuele Viola. New correlation bounds for GF(2) polynomials using Gowers uniformity. *Electronic Colloquium on Computational Complexity*, Technical Report TR06-097, 2006. www.eccc.uni-trier.de/.

[Vio09a] Emanuele Viola. On the power of small-depth computation. *Foundations and Trends in Theoretical Computer Science*, 5(1):1–72, 2009.

[Vio09b] Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. *Computational Complexity*, 18(2):209–217, 2009.

[Vio17] Emanuele Viola. Challenges in computational lower bounds. *SIGACT News, Open Problems Column*, 48(1), 2017.

[Vio19] Emanuele Viola. Matching Smolensky's correlation bound with majority. Available at http://www.ccs.neu.edu/home/viola/, 2019.

[VW08] Emanuele Viola and Avi Wigderson. Norms, XOR lemmas, and lower bounds for polynomials and protocols. *Theory of Computing*, 4:137–168, 2008.

[Wil11] Ryan Williams. Guest column: a casual tour around a circuit complexity bound. *SIGACT News*, 42(3):54–76, 2011.

[Wil13a] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM J. on Computing*, 42(3):1218–1244, 2013.

[Wil13b] Ryan Williams. Natural proofs versus derandomization. In *ACM Symp. on the Theory of Computing (STOC)*, 2013.

[Wil14a] Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *ACM Symp. on the Theory of Computing (STOC)*, 2014.

[Wil14b] Ryan Williams. Nonuniform ACC circuit lower bounds. *J. of the ACM*, 61(1):2:1–

2:32, 2014.

[Wun12] Henning Wunderlich. On a theorem of razborov. *Computational Complexity*, 21(3):431–477, 2012.

[Yao77] Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 222–227. IEEE Computer Society, 1977.

[Yao90] Andrew Chi-Chih Yao. On ACC and threshold circuits. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 619–627, 1990.

[Zák83] Stanislav Zák. A turing machine time hierarchy. *Theoretical Computer Science*, 26:327–333, 1983.