Electronic Colloquium on Computational Complexity, Report No. 191 (2016)

Improved Bounds for Quantified Derandomization of Constant-Depth Circuits and Polynomials

Roei Tell [*]

November 24, 2016

Abstract

Goldreich and Wigderson (STOC 2014) initiated a study of *quantified derandomization*, which is a relaxed derandomization problem: For a circuit class $\mathcal{C}$ and a parameter $B = B(n)$, the problem is to decide whether a circuit $C \in \mathcal{C}$ rejects all of its inputs, or accepts all but $B(n)$ of its inputs.

In this work we make progress on several frontiers that they left open. Specifically, for **constant-depth circuits**, we construct an algorithm for quantified derandomization that is significantly faster than the best currently-known algorithms for standard derandomization, and works for a parameter $B(n)$ that is only *slightly smaller* than a "barrier" parameter that was shown by Goldreich and Wigderson. For **constant-depth circuits with parity gates**, we tighten a "barrier" of Goldreich and Wigderson (from depth five to depth four), and construct algorithms for quantified derandomization of a remaining type of layered depth-3 circuit that they did not handle and left as an open problem (i.e., circuits with a top $\oplus$ gate, a middle layer of $\wedge$ gates, and a bottom layer of $\oplus$ gates).

In addition, we extend Goldreich and Wigderson's study of multivariate **polynomials that vanish rarely** to the setting of large finite fields. We prove two lower bounds on the seed length of hitting-set generators for polynomials over large fields that vanish rarely. As part of the proofs, we show a form of "error reduction" for polynomials (i.e., a reduction of the task of hitting arbitrary polynomials to the task of hitting polynomials that vanish rarely) that causes only a mild increase in the degree.

[*] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel. Email: roei.tell@weizmann.ac.il

ISSN 1433-8092

Contents

1 Introduction

For a circuit class $\mathcal{C}$, the standard one-sided error derandomization problem is the following: Given a circuit $C \in \mathcal{C}$, distinguish in deterministic polynomial time between the case that C rejects all of its inputs and the case that C accepts most of its inputs. Impagliazzo and Wigderson [IW99], following Nisan and Wigderson [NW94], showed that under reasonable complexity-theoretic assumptions, the standard derandomization problem can be solved even for a class as large as $\mathcal{C} = \mathcal{P}/\text{poly}$. However, at this moment, we do not know how to unconditionally solve this problem even when $\mathcal{C}$ is the class of polynomial-sized CNFs.

A couple of years ago, Goldreich and Wigderson [GW14] suggested a potentially easier problem, which they call *quantified* derandomization. Given a class $\mathcal{C}$ and a parameter $B = B(n)$, the problem is to decide whether a circuit $C \in \mathcal{C}$ over n input bits rejects all of its inputs, or accepts *all but* $B(n)$ *of its inputs* (rather than just "most" of its inputs). We call $B(n)$ the "badness" parameter, since it represents the number of bad random strings (i.e., the ones that lead the algorithm to an incorrect decision). Indeed, the standard derandomization problem is captured by the parameter $B(n) = 2^n/2$, but we are typically interested in $B(n)$'s that are much smaller. On the other hand, polynomially-bounded values (e.g., $B(n) = O(n)$) can be easily handled by an algorithm that simply evaluates C on $B(n) + 1$ fixed inputs.

Goldreich and Wigderson constructed algorithms that solve the quantified derandomization problem for various classes $\mathcal{C}$ and parameters $B = B(n)$. For example, they constructed a polynomial time hitting-set generator for $\mathcal{AC}^0$ circuits that accept all but $B(n) = 2^{n^{1-\epsilon}}$ of their inputs, for any $\epsilon > 0$. On the other hand, they showed that for some classes $\mathcal{C}$ and a sufficiently high badness parameter $B(n)$, the quantified derandomization problem is at least as difficult as the standard derandomization problem (since the latter can be reduced to the former). We call such parameter values threshold values, since a quantified derandomization with a $B(n)$ that surpasses this threshold will yield a result for a standard derandomization problem.

Our contributions in this work are of two types. On the one hand, we construct quantified derandomization algorithms that work for a broader range of parameters, compared to [GW14] (e.g., larger values of $B(n)$, or broader circuit classes). On the other hand, we show that quantified derandomization of circuit classes that are more limited (compared to what is known from [GW14]) is still at least as difficult as certain standard derandomization problems. Considered together, these result bring closer two settings of parameters: The parameter setting for which we can unconditionally construct relatively fast quantified derandomization algorithms, and the "threshold" values (for the parameters) for which any quantified derandomization algorithm implies a similar algorithm for standard derandomization.

1.1 Brief overview of our main results

Let us now informally state our main results, which we later outline in detail:

- **Constant-depth circuits:** Goldreich and Wigderson showed that for $\mathcal{AC}^0$ circuits of depth d, the badness parameter $B(n) = \exp\left(n/\log^{0.99d}(n)\right)$ is a threshold value, since an algorithm for quantified derandomization with such a $B(n)$ implies an algorithm for *standard* derandomization of depth-d circuits.

  We show that taking $B(n)$ to be only *slightly smaller* allows for derandomization of depth-d circuits that is significantly faster than the best currently-known standard derandomization. Specifically, we construct a hitting-set generator for the parameter $B(n) = \exp\left(n/\log^{d-2}(n)\right)$ that has seed length $\tilde{O}(\log^3(n))$.

- **Constant-depth circuits with parity gates:** Goldreich and Wigderson derandomized various types of layered $\mathcal{AC}^0[\oplus]$ circuits of depth 3 with $B(n) = 2^{n^c}$, for any $c < 1$; they left one last type as an open problem, and noted that a threshold exists for a similar result at depth five.

  We show that a similar threshold exists already at depth four, and make progress on the last remaining type of layered depth-3 circuit. Specifically, we construct a whitebox hitter for circuits with a top $\oplus$ gate, a middle layer of $\wedge$ gates, and a bottom layer of $\oplus$ gates, under various sub-quadratic bounds on the number of gates in the different layers, and with the parameter $B(n) = 2^{n^c}$.

- **Polynomials that vanish rarely:** We study the problem of constructing hitting-set generators for polynomials $\mathbb{F}^n \to \mathbb{F}$ that *vanish rarely*, where $\mathbb{F}$ is an arbitrary finite field (Goldreich and Wigderson studied the setting of $\mathbb{F} = \mathbb{F}_2$). We prove lower bounds on the seed length of hitting-set generators for such polynomials; in particular, we show that any hitting-set generator for degree-d polynomials that vanish on at most a $1/\text{poly}(|\mathbb{F}|)$ fraction of their inputs requires a seed of length similar to that of hitting-set generators for *all* polynomials of degree d.

  As part the proofs, we reduce the task of constructing a hitting-set generator for degree-d polynomials to the task of constructing a hitting-set generator for polynomials of degree d' that vanish rarely, where $d \leq d' \leq \text{poly}(d)$; this can be thought of as a form of "error reduction" for polynomials that causes only a mild increase in the degree.

The results for each of the three settings are detailed in Sections 1.2, 1.3, and 1.4, respectively. Towards stating the results, recall that a hitting-set generator for a class of functions $\mathcal{F}$ from $\{0,1\}^n$ to $\{0,1\}$ is an algorithm $G : \{0,1\}^\ell \to \{0,1\}^n$, for some $\ell = \ell(n)$, such that for every $f \in \mathcal{F}$ there exists some $s \in \{0,1\}^\ell$ such that $f(G(s)) \neq 0$. We say that the hitting-set generator has density $\epsilon > 0$ if for every $f \in \mathcal{F}$ it holds that $\Pr_{s \in \{0,1\}^\ell}[f(G(s)) \neq 0] > \epsilon$ (see Definition 8). The definition of hitting-set generators extends naturally to functions $\mathbb{F}^n \to \mathbb{F}$, for any field $\mathbb{F}$ (see Definition 19).

1.2 Constant-depth circuits

Our main result for $\mathcal{AC}^0$ circuits is a relatively fast quantified derandomization algorithm, with a parameter $B(n)$ that nearly matches the threshold parameter established in [GW14]. We first recall their result:

Theorem 1 *([GW13, Thm 3.4]). Assume that, for some constant $\epsilon > 0$ and for every $d \geq 2$, there exists a polynomial-time algorithm that solves the quantified derandomization problem for depth-d circuits with $B(n) = 2^{n/\log^{(1-\epsilon)\cdot d}(n)}$. Then, for any $d \geq 2$, there exists a polynomial-time algorithm that solves the* standard *derandomization problem for depth-d circuits.*

We show a derandomization of depth-d circuits with badness parameter $B(n) = 2^{n/\log^{d-2}(n)}$, which *is only slightly smaller* than the threshold value in Theorem 1, in time that is significantly faster than the current state-of-the-art for derandomizing $\mathcal{AC}^0$:

Theorem 2 *($\mathcal{AC}^0$ with badness $2^{n/\log^{d}(n)}$). For any $d \geq 2$, there exists a hitting-set generator with seed length $\tilde{O}(\log^3(n))$ for the class of depth-d circuits over n input bits that accept all but at most $B(n) = 2^{\Omega\left(n/\log^{d-2}(n)\right)}$ of their inputs.*

We stress that the power of the poly-logarithm in the seed length in Theorem 2 does not depend on the depth d. Any *standard* hitting-set generator for $\mathcal{AC}^0$ (i.e., with $B(n) = 2^n/2$) with such a seed length would be a major breakthrough, and in particular would significantly improve the lower bounds of Håstad for $\mathcal{AC}^0$ [Hås87] (see, e.g., [Vad12, Prob. 7.1] and [TX13]). Thus, while derandomizing depth-d circuits with $B(n) = \exp\left(n/\log^{d-2}(n)\right)$ is possible in (fixed) quasi-polynomial time, any derandomization of depth-d circuits with $B(n) = \exp\left(n/\log^{0.99\cdot d}(n)\right)$ will yield a standard derandomization in similar time (and, if achieved via a hitting-set generator or pseudorandom generator, will imply new circuit lower bounds).

Let us also suggest another perspective. Theorem 1 was proved in [GW13] by using randomness-efficient methods for error reduction within $\mathcal{AC}^0$.[1] Hence, Theorem 2 implies that essentially any improvement on these error reduction techniques within $\mathcal{AC}^0$ would yield a breakthrough in the derandomization of $\mathcal{AC}^0$. We comment that at least one approach for such an improvement has already been ruled-out (see [GVW15]).

1.3 Constant-depth circuits with parity gates

The next circuit class that we study is that of constant-depth circuits that also have gates computing the parity function (i.e., $\mathcal{AC}^0[\oplus]$). Specifically, we consider $\mathcal{AC}^0[\oplus]$ circuits that are *layered*, in the sense that all gates at a particular distance from the input gates are of the same gate-type. Goldreich and Wigderson derandomized almost all

[1] Specifically, Goldreich and Wigderson started from a circuit of depth d and acceptance probability $1/2$, and combined a constant-depth pseudorandom generator for $\mathcal{AC}^0$ (e.g., Nisan's [Nis91]) with randomness-efficient error reduction that uses Trevisan's extractor [Tre01], to obtain a circuit of depth d' that rejects at most $2^{n/\log^{.99d'}(n)}$ of its n-bit inputs.

types of *layered depth*-3 $\mathcal{AC}^0[\oplus]$ circuits with $B(n) = 2^{n^c}$, for any $c < 1$, with the exception of circuits of the form $\oplus \wedge \oplus$ (i.e., top $\oplus$ gate, middle layer of $\wedge$ gates, and bottom layer of $\oplus$ gates above the input gates), which they left as an open problem.

We first observe that the standard derandomization problem of CNFs can be reduced to the problem of derandomizing layered $\mathcal{AC}^0[\oplus]$ circuits of *depth four* with $B(n) = 2^{n^c}$, which yields a "threshold" at depth four with such a badness parameter. This improves on a similar result of [GW14] that refers to depth five.

Theorem 3 *(a threshold for quantified derandomization of $\mathcal{AC}^0[\oplus]$ at depth four). Assume that, for some $c > 0$, there exists a polynomial-time algorithm A such that, when A is given as input a layered depth-four $\mathcal{AC}^0[\oplus]$ circuit C over n input bits that accepts all but $B(n) = 2^{n^c}$ of its inputs, then A finds a satisfying input for C. Then, there exists a polynomial-time algorithm A' that, when given as input a polynomial-size CNF that accepts most of its inputs, then A' finds a satisfying input for the CNF.*

The improvement over [GW14] in Theorem 3 is obtained by reducing to $\mathcal{AC}^0[\oplus]$ from the problem of one-sided error derandomization of CNFs, rather than two-sided error. Thus, the main remaining challenge to handle for $\mathcal{AC}^0[\oplus]$ with $B(n) = 2^{n^c}$ before reaching the "threshold" at depth four is the last type of layered depth-3 circuit; that is, derandomizing $\oplus \wedge \oplus$ circuits with $B(n) = 2^{n^c}$. Our main result in this section is an algorithm that makes significant progress on this challenge, by derandomizing such circuits (with such a $B(n)$) under various sub-quadratic upper bounds on the circuit size, where some of these upper bounds refer to each layer separately.

Theorem 4 *(hitting biased $\oplus \wedge \oplus$ circuits). Let $\epsilon > 0$ be an arbitrary constant. Let $\mathcal{C}$ be the class of circuits of depth three with a top $\oplus$ gate, a middle layer of $\wedge$ gates, and a bottom layer of $\oplus$ gates, such that every $C \in \mathcal{C}$ over n input bits satsifies (at least) one of the following:*

1. *The size of C is $O(n)$.*
2. *The number of $\wedge$-gates is at most $n^{2-\epsilon}$, and the number of $\oplus$-gates is at most $n + n^{\epsilon/2}$.*
3. *The number of $\oplus$-gates is at most $n^{1+\epsilon}$, and the number of $\wedge$-gates is at most $\frac{1}{5} \cdot n^{1-\epsilon}$.*

Then, for some $c = c(\epsilon) > 0$, there exists a polynomial-time algorithm that, when given a circuit $C \in \mathcal{C}$ that accepts all but $B(n) = 2^{n^c}$ of its inputs, outputs a satisfying input for C.

We stress that the algorithm from Theorem 4 makes essential use of the specific circuit C that is given to the algorithm as input. For further details see Section 2.2.

1.4 Polynomials that vanish rarely

We now turn our attention to quantified derandomization of polynomials, and specifically to the problem of constructing hitting-set generators for polynomials $\mathbb{F}_2^n \to \mathbb{F}_2$ that *vanish rarely*. In this setting it is more convenient to work with a normalized badness parameter $b(n) = B(n)/2^n$: For an integer n and a degree bound $d < n$, we want to construct a hitting-set generator (with seed length $O(\log(n))$) for the class of polynomials $p : \mathbb{F}_2^n \to \mathbb{F}_2$ of total degree d that vanish on at most a $b(n)$ fraction of their inputs (i.e., $\Pr_{x \in \mathbb{F}_2^n}[p(x) = 0] \le b(n)$).

The problem is trivial when $b(n) < 2^{-d}$, since in this case p is the constant one polynomial, and Goldreich and Wigderson solved this problem when $b(n) = O\left(2^{-d}\right)$; we provide an alternative proof of their result in Appendix A. They suggested to try and extend this result to also handle $b(n) = m(n) \cdot 2^{-d}$, where $m(n) = \mathrm{poly}(n)$, and conjectured that such a result would imply a quantified derandomization of $\oplus \wedge \oplus$ circuits of size $m(n)$. [2] We affirm their conjecture, by showing that any sufficiently dense hitting-set generator for degree-d polynomials with $b(n) = m(n) \cdot 2^{-d}$ is also a hitting-set generator for $\oplus \wedge \oplus$ circuits of size $m(n)$ with $B(n) = \Omega\left(2^n\right)$.

Theorem 5 *(reducing $\oplus \wedge \oplus$ circuits to biased polynomials). Let $\mathcal{C}$ be the class of $\oplus \wedge \oplus$ circuits over n input bits with $m = m(n)$ $\wedge$-gates that accept all but $B(n) = \epsilon \cdot 2^n$ of their inputs, where $m(n) = o(2^n)$ and $\epsilon = \epsilon(n) \le 1/8$. Let $\mathcal{P}$ be the class of polynomials $\mathbb{F}_2^n \to \mathbb{F}_2$ of degree $d = \lfloor \log(m(n)) + \log(1/\epsilon) \rfloor$ that accept all but a $b(n) = (4 \cdot m(n)) \cdot 2^{-d} = 4 \cdot \epsilon$ fraction of their inputs. Then, any hitting-set generator with density $1/2 + 2 \cdot \epsilon$ for $\mathcal{P}$ is also a hitting-set generator for $\mathcal{C}$.*

Our main focus in the current section is an extension of the problem of hitting polynomials that vanish rarely to *fields larger than* $\mathbb{F}_2$. Specifically, let $\mathbb{F}$ be a finite field of size $|\mathbb{F}| = q \le \mathrm{poly}(n)$, and let $1 \le d \le (q-1) \cdot n$. We consider the problem of constructing hitting-set generators for polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree d that vanish on at most a $b(n)$ fraction of their inputs. Recall that any hitting-set generator for the class of *all* polynomial of total degree d (i.e., regardless of the fraction of inputs on which they vanish) requires a seed of $\log\left(\binom{n+d}{d}\right)$ bits, and that there exists a non-explicit pseudorandom generator for this class with a seed of $O\left(\log\left(\binom{n+d}{d}\right)\right)$ bits.[3] Moreover, for $d = O(1)$ and a sufficiently large q, *explicit* constructions of pseudorandom generators with a seed of $O(\log(n))$ bits are known (see, e.g. [Bog05, CTS13]).

Our question is whether it is possible to use a *shorter seed* if we only require that the generator will hit degree-d polynomials that vanish on $b(n)$ of their inputs. More accurately, we ask *how low* must $b(n)$ be in order for a hitting-set generator with seed length $o\left(\log\left(\binom{n+d}{d}\right)\right)$ to exist, even non-explicitly. The setting of $b(n) < q^{-d}$ is trivial, since any degree-d polynomial that has at least one root vanishes on at least q^{-d} of its inputs (this follows from Warning's second theorem; see, e.g., [Sch76, Sec. 4]). On the other hand, the setting of $b(n) = d/q$ is essentially the standard (i.e., non-quantified) problem, since any non-zero degree-d polynomial vanishes on at most a d/q fraction of its inputs. Also, recall that when d is much smaller than q (e.g., $d = q^{o(1)}$), every

[2] In [GW13, Sec. 6] it is suggested to prove this result by modifying any $\oplus \wedge \oplus$ circuit to a bounded-degree polynomial, where the modification amounts to the removal of all $\wedge$-gates with high fan-in. However, as explained in Section 2, since the top gate is a $\oplus$-gate, we cannot simply remove $\wedge$-gates with high fan-in (or remove some of the wires that feed into them).

[3] For proof of the lower bound see, e.g., the proof of Theorem 27, and for the upper bound note that a polynomial $\mathbb{F}^n \to \mathbb{F}$ of degree d can be represented by $\binom{n+d}{d} \cdot \log(q)$ bits.

degree-d polynomial vanishes on $\frac{i}{q} \pm \eta$ of its inputs, where $i \in \{0, 1, ..., d\}$ and $\eta \leq \frac{\text{poly}(d)}{q^{3/2}}$ (for a more accurate statement see, e.g., [Bog05, Sec. 2]).

Our first result for this problem is that for any degree $d \leq 0.99 \cdot q$, any hitting-set generator for degree-d polynomials with $b(n) = O(1/q)$ requires a seed of $\Omega\left(\log\left(\binom{n+d}{d}\right)\right)$ bits; that is, the value $b(n) = O(1/q)$ yields essentially no relaxation *at all* (with respect to seed length), compared to the standard problem. Our main result for this problem, however, goes much further: It turns out that even when considering the parameter $b(n) = 1/\text{poly}(q)$, any hitting-set generator for degree-d polynomials that vanish on $b(n)$ of their inputs still requires a seed of length similar to that of a hitting-set generator for *all* degree-d polynomials. Specifically, any hitting-set generator for degree-d polynomials with $b(n) = 1/\text{poly}(q)$ requires a seed of $\Omega\left(\log\left(\binom{n+d^{1/O(1)}}{d^{1/O(1)}}\right)\right)$ bits. It follows that for *any* super-constant degree $d = \omega(1)$, there does not exist a hitting-set generator with seed length $O(\log(n))$ for degree-d polynomials with $b(n) = 1/\text{poly}(q)$.

Theorem 6 *(hitting polynomials that vanish rarely over large fields; informal). For a constant $k \in \mathbb{N}$, let $n \in \mathbb{N}$, and let $\mathbb{F}$ be a field of size $|\mathbb{F}| = q \leq n^k$. Then:*

1. *For any degree $d \leq 0.99 \cdot q$, any hitting-set generator with constant density for the class of polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree d that vanish on at most $b(n) = O(1/q)$ their inputs requires a seed of $\Omega\left(\log\left(\binom{n+d}{d}\right)\right)$ bits.*

2. *For any even constant $t \geq 2$ and degree d' such that $(2k)^{t(t+1)} \leq d' \leq 0.99 \cdot q^{t+1}$, any hitting-set generator for the class of polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree d' that vanish on at most $b(n) = O\left(q^{-t^2/4}\right)$ of their inputs requires a seed of $\Omega\left(\log\left(\binom{n+d}{d}\right)\right)$ bits, where $d = (d')^{1/(t+1)}$.*

Regarding Item (1) of Theorem 6, recall that *most* polynomials of degree d vanish on at most a $O(1/q)$ fraction of their inputs. However, the fact that the case of $b(n) = O(1/q)$ is the typical case does not a-priori imply that this is not a relatively easier case to handle. The proofs of both items of Theorem 6 consist of reducing the problem of constructing a hitting-set generator for *all* polynomials of degree $d \in \mathbb{N}$ to the problem of constructing a hitting-set generator for polynomials that *vanish rarely* and are of degree d', where $d' = d$ in the proof of Item (1) and $d' = \text{poly}(d)$ in the proof of Item (2). For further details see Section 2.3.2.

1.5 Organization of the paper

In Section 2 we explain, in high level, the techniques used to obtain our results. Section 3 contains preliminary definitions and statements of some well-known facts. Then, each of the subsequent sections includes proofs for a corresponding section from the introduction: In Section 4 we prove Theorem 2; in Section 5 we prove Theorem 3 and Theorem 4; and in Section 6 we prove Theorem 5 and Theorem 6. In Appendix A we provide an alternative proof of [GW14, Thm. 1.6], and in Appendices B and C we provide proofs for several technical claims from Sections 4 and 6, respectively.

2 Our Techniques

2.1 Constant-depth circuits

Let us first describe, in high-level, the proof of Theorem 2. Considering any depth-d circuit C that accepts all but $B(n) = \Omega\left(2^{n/\log^{d-2}(n)}\right)$ of its inputs, the generator first uses pseudorandom restrictions to simplify the circuit C to a depth-2 circuit, by fixing values for all but $n' = \Omega(n/\log^{d-2}(n))$ of the variables. These pseudorandom restrictions are chosen using a suitable derandomized switching lemma (specifically, Tal's [Tal14] improvement of the lemma of Trevisan and Xue [TX13]), whose seed length is $\tilde{O}(\log^3(n))$. At this point, there are $n' \geq \log(B(n)) + 1$ living variables, and therefore the simplified circuit (over n' input bits) has acceptance probability at least $1/2$ (since C has at most $B(n)$ unsatisfying inputs). Hence, we can use a pseudorandom generator for depth-2 circuits with seed length at most $O(\log^3(n))$ (e.g., that of Bazzi [Baz09] or that of De *et al.* [DETT10]) in order to fix values for the remaining n' variables, thus finding a satisfying input for C, with high probability.

One subtlety in the above is the following. In the derandomized switching lemma of [TX13, Tal14], the expected number of living variables is very close to $n/\log^{d-2}(n)$, but the lemma does *not* guarantee that approximately this many variables remain alive with high (or even constant) probability. Nevertheless, we show that the latter does indeed hold, when instantiating one generic component in the lemma (i.e., a pseudorandom generator for depth-2 circuits) in a specific manner (i.e., using the pseudorandom generator of De *et al.* [DETT10]).

2.2 Constant-depth circuits with parity gates

Let us now describe the high-level strategy of the algorithms of Theorem 4. First observe that any $\oplus\wedge\oplus$ circuit C computes an n-variate polynomial over $\mathbb{F}_2$, and that the total degree of this polynomial equals the maximal fan-in of $\wedge$-gates in the circuit. Our approach will be to find an *affine subspace* W of dimension more than $\log(B(n))$ such that when C is restricted to the affine subspace, the fan-in of all $\wedge$-gates becomes constant. Thus, when restricted to W, the circuit C becomes a non-zero polynomial of constant degree, which means that we can then hit it using a pseudorandom generator for polynomials of constant degree (i.e., Viola's [Vio09]).

In order to find the affine subspace W, the algorithm considers *affine restrictions*, which are obtained by fixing values to some of the bottom $\oplus$-gates. These are analogous to standard "bit-fixing" restrictions; however, in contrast to the latter, we cannot consider *any* sequence of fixed values to the bottom $\oplus$-gates, because in our setting the bottom $\oplus$-gates might not be linearly independent (and thus the values of some $\oplus$-gates might depend on the values of other $\oplus$-gates). In particular, this means that we cannot use random (or pseudorandom) restrictions in which the value of each $\oplus$-gate is chosen obliviously of the $\oplus$-gates of the circuit.

Our algorithm circumvents this problem by constructing a restriction that corresponds to the *specific* $\oplus \wedge \oplus$ circuit that is given to the algorithm as input. Each of the three items of Theorem 4 uses a different construction. For concreteness, let us now describe the construction of Item (2) of Theorem 4, and let us also fix specific parameter values to work with: We assume, for simplicity, that the number of bottom $\oplus$-gates is *exactly* n; and we assume that the number of $\wedge$-gates is $n^{1.1}$, and that the circuit accepts all but $\Omega\left(2^{n^{1/3}}\right)$ of its inputs.

First assume, for a moment, that the fan-in of each $\wedge$-gate in the middle layer of the circuit is upper bounded by $\sqrt{n}$. In this case we can restrict the bottom $\oplus$-gates as follows. Consider a random restriction process in which each bottom $\oplus$-gate is fixed independently with probability $1-p = 1-n^{-2/3}$, and the *values* for the fixed gates are chosen afterwards, in an *arbitrary consistent manner*. With high probability, the restriction will yield a subspace of dimension approximately $p \cdot n = n^{1/3} > \log(B(n))$. Also, since each $\wedge$-gate g has fan-in at most $w = \sqrt{n}$, and $p = 1/w^{1+\Omega(1)}$, with very high probability, all but $O(1)$ of the gates that feed into g are fixed by this process.[4] In fact, the above two statements hold even if we choose the restriction according to an $O(1)$-independent distribution, rather than uniformly.

Needless to say, we cannot actually assume that the fan-in of $\wedge$-gates is bounded by $\sqrt{n}$. Thus, our strategy will be to first *mildy* reduce the fan-in of $\wedge$-gates (from n to $\sqrt{n}$), and then invoke the pseudorandom restriction process described above. A standard approach to mildly reduce the fan-in of $\wedge$-gates is to simply remove some of the incoming wires to each $\wedge$-gate. However, this approach *does not* work in our setting, since the top gate is a $\oplus$-gate, which means that such a modification might turn unsatisfying inputs into satisfying ones (and thus hitting the modified circuit might not yield a satisfying input to the original circuit).

To reduce the fan-in of $\wedge$-gates to $\sqrt{n}$, we follow Kopparty and Srinivasan [KS12] in adapting the approach of Chaudhuri and Radhakrishnan [CR96] to the setting of $\oplus \wedge \oplus$ circuits.[5] Specifically, we first iteratively fix each $\oplus$-gate that has *fan-out* more than $n^{1/4}$ to a *non-accepting* value; note that such an action also fixes $n^{1/4}$ $\wedge$-gates in the middle layer, and hence in this step we fix values for at most $n^{1.1}/n^{1/4} = o(n)$ bottom $\oplus$-gates (because afterwards, there are no more living $\wedge$-gates, so the entire circuit is trivial). Note that at this point, the number of wires feeding the middle layer is at most $n \cdot n^{1/4} = n^{1.25}$. Now, for each $\wedge$-gate g with *fan-in* more than $\sqrt{n}$, we fix a $\oplus$-gate that feeds into g to a *non-accepting* value, thereby also fixing g; note that each such action eliminates $\sqrt{n}$ wires that feed into the middle layer, and therefore in this

[4] To see this, note that the probability that there exists a subset of size c of the $\oplus$-gates that feed into g in which all the $\oplus$-gates are unfixed is at most $\binom{w}{c} \cdot p^c = 1/\text{poly}(n)$, for a sufficiently large $c = O(1)$.

[5] Originally, [CR96] applied their approach to $\mathcal{AC}^0$ circuits, and [KS12] later adapted this approach to $\mathcal{AC}^0[\oplus]$ circuits. Our adaptation is slightly different technically than in [KS12], to suit the specific circuit structure $\oplus \wedge \oplus$; but more importantly, while both [CR96, KS12] use the approach as part of the analysis (to prove lower bounds), we use this approach as a (non-black-box) algorithm for derandomization.

step we fix at most $n^{1.25}/\sqrt{n} = o(n)$ bottom $\oplus$-gates. Overall, the fan-in of each $\wedge$-gate has been reduced to $\sqrt{n}$, and we imposed at most $o(n)$ affine conditions.

To see that the final subspace W is of dimension more than $\log(B(n))$, note that the dimension of W equals the number of living $\oplus$-gates (because we assumed that the initial number of $\oplus$-gates is exactly n). After the first step of the algorithm (i.e., reducing the fan-in of $\wedge$-gates to $\sqrt{n}$), we are left with $(1 - o(1)) \cdot n$ living $\oplus$-gates, and the second step (i.e., the pseudorandom restriction) leaves a fraction of $p = n^{-2/3}$ of them alive. Thus, the expected dimension of W is $\Omega(p \cdot n) = \Omega\left(n^{1/3}\right) > \log(B(n))$.

The approach above actually works for a broader range of parameters, and in particular when the number of $\wedge$-gates is $n^{2-\epsilon}$, for any constant $\epsilon > 0$, and when the number of $\oplus$-gates is $n + n^c$, for any $c < \epsilon$ (see details in Section 5.2.3). In Items (1) and (3), we consider circuits in which the number of $\oplus$-gates is significantly larger than n, namely $O(n)$ and $O\left(n^{1+\epsilon}\right)$, respectively. The proofs of both these items use algorithms that are variations of the first step of the algorithm described above, and these proofs are detailed in Sections 5.2.2 and 5.2.4, respectively.

2.3 Polynomials that vanish rarely

Both proofs of Theorem 5 and of Theorem 6 rely on a claim that is implicit in the work of Bogdanov and Viola [BV10]. To state the claim, we first need to formally define a notion that is implicit in many previous works. Specifically, consider the following question: Given a "complicated" function $p : \mathbb{F}^n \to \mathbb{F}$, can we compute the value of p at any $x \in \mathbb{F}^n$, with high probability, by randomly choosing a function $h : \mathbb{F}^n \to \mathbb{F}$ from a class $\mathcal{H}$ of "simpler" functions, and outputing $h(x)$? Note that we want the distribution over $\mathcal{H}$ to be a single, fixed distribution, such that h is chosen obliviously of x. This is trivial if $\mathcal{H} \supseteq \{p\}$ (i.e., if $\mathcal{H}$ is not "simpler" than p), or if we want to be correct only with probability $1/|\mathbb{F}|$ (since we can just guess a random value). But the point is that we want both that $\mathcal{H}$ will be simpler than p, and that the computation will be correct with probability (significantly) larger than $1/|\mathbb{F}|$.

Actually, we are also willing to tolerate a more relaxed version of the problem above, in which the distribution is supported on functions $\{\mathbb{F}^n \to \mathbb{F}\}$, and is allowed to only *typically* be in $\mathcal{H}$, rather than always be in $\mathcal{H}$. When there exists a distribution $\mathbf{h}$ that satisfies both conditions above (i.e., $\mathbf{h}$ is typically in $\mathcal{H}$, and for every $x \in \mathbb{F}^n$ it holds that $\mathbf{h}(x) = p(x)$, with high probability), we say that we can randomly compute p by a distribution that is typically in $\mathcal{H}$.

The key claim that we will use in this context, which generalizes and extends a claim that is implicit in [BV10, Lemma 23], is the following: If we can randomly compute a function p by a distribution $\mathbf{h}$ that is typically in $\mathcal{H}$, then any distribution that "fools" $\mathcal{H}$ also "fools" p (see Section 6.1 for a precise statement). In particular, any hitting-set generator with sufficient density for $\mathcal{H}$ is also a hitting-set generator (with smaller density) for p. Note that when using the claim, we are not interested in the complexity of computing the distribution $\mathbf{h}$, but rather only in its *existence*. Thus, when trying to construct a hitting-set generator for p, we can construct the distribution $\mathbf{h}$ while being very wasteful in the use of randomness and other resources (because this distribution is only used in the analysis).

2.3.1 The proof of Theorem 5

Theorem 5 asserts that the problem of hitting $\oplus \wedge \oplus$ circuits reduces to the problem of constructing a hitting-set generator for polynomials that vanish rarely. This claim is proved by showing how to randomly compute any $\oplus \wedge \oplus$ circuit C with m $\wedge$-gates that accepts all but an $\epsilon = \epsilon(n)$ fraction of its inputs by a distribution over polynomials $\mathbb{F}_2^n \to \mathbb{F}_2$ that is typically in the class $\mathcal{P}_d$, where $\mathcal{P}_d$ consists of polynomials of degree $d = \log(m) + \log(1/\epsilon)$ that accept all but an $4 \cdot \left(m \cdot 2^{-d}\right) = 4 \cdot \epsilon$ fraction of their inputs.

To construct the distribution over polynomials, we use the classical approximating polynomials of Razborov [Raz87], to randomly modify the circuit C into a circuit in which the fan-in of $\wedge$-gates in the middle layer is at most $d = \log(m) + \log(1/\epsilon)$. The latter circuit is indeed a polynomial of degree d, and in expectation, it vanishes on at most $2 \cdot \epsilon$ of its inputs (because with probability at least $1 - \epsilon$ it agrees with the original circuit, which rejects at most ϵ of its inputs). Thus, with probability at least $1/2$, the random degree-d polynomial vanishes on at most $4 \cdot \epsilon$ of the inputs.

2.3.2 The proof of Theorem 6

The main component in the proof of Theorem 6 is a reduction of the task of constructing a hitting-set generator for polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree $d \leq 0.99 \cdot |\mathbb{F}|$ to the task of constructing a hitting-set generator for polynomials $\mathbb{F}^{O(n)} \to \mathbb{F}$ of degree $d' \geq d$ that vanish rarely. Since any hitting-set generator for all polynomials of degree d requires a seed of $\Omega\left(\log\left(\binom{n+d}{d}\right)\right)$ bits, we obtain the lower bound on hitting-set generators for polynomials $\mathbb{F}^{O(n)} \to \mathbb{F}$ of degree d' that vanish rarely. The aforementioned reduction can be thought of as a form of "randomness-efficient error reduction" for polynomials such that the increase in degree from d to d' is mild (or even $d' = d$).

Let $p : \mathbb{F}^n \to \mathbb{F}$ be of degree d. The first observation is that since $d \leq 0.99 \cdot |\mathbb{F}|$, it holds that $\Pr_{x \in \mathbb{F}^n}[p(x) = 0] \leq 0.99$, which implies that the probability over a random subspace $W \subseteq \mathbb{F}^n$ of constant dimension that $p\restriction_W \equiv 0$ is very small (because such a subspace consists of $\mathrm{poly}(|\mathbb{F}|)$ points that are $O(1)$-wise independent). Our strategy is therefore to try and construct a polynomial $p' : \mathbb{F}^{O(n)} \to \mathbb{F}$ that satisfies the following: The polynomial p' gets as input a tuple $\vec{u} \in \mathbb{F}^{O(n)}$ that defines a subspace $W = W_{\vec{u}}$, and outputs zero if and only if $p\restriction_W \equiv 0$. Note that any polynomial p' that satisfies this condition vanishes rarely, because $p\restriction_W \not\equiv 0$ for almost all subspaces W. And indeed, hitting p' yields a subspace W such that $p\restriction_W \not\equiv 0$, which allows us to hit p, by using additional $O(\log(|\mathbb{F}|)) \leq O(\log(n))$ random bits to choose an input $w \in W$. (This approach is reminiscent of Bogdanov's [Bog05] reduction of the construction of pseudorandom generators to the construction of hitting-set generators.)

The main challenge in constructing such a polynomial p' is the following: Given a tuple $\vec{u} \in \mathbb{F}^{O(n)}$ that defines a subspace $W = W_{\vec{u}} \subseteq \mathbb{F}^n$, how can we test efficiently (i.e., with degree d' that is not much larger than d) whether or not $p\restriction_W \equiv 0$? Indeed, a naive solution is to compute the OR function of the values $\{p(w) : w \in W\}$ (i.e., compute the polynomial that outputs 1 if and only if there exists $w \in W$ such that $p(w) \neq 0$), but this solution requires a very high degree $d' \geq \text{poly}(|\mathbb{F}|)$. We present two solutions for this problem: The first yields $d' = \text{poly}(d)$, and corresponds to Item (2) of Theorem 6, and the second yields $d' = d$, and corresponds to Item (1) of Theorem 6.

The first solution relies on the observation that instead of testing whether or not there exists $w \in W$ such that $p(w) \neq 0$, we can test whether or not there exists a non-zero coefficient in the representation of $p\restriction_W$ as a polynomial $\mathbb{F}^{O(1)} \to \mathbb{F}$. Since $p\restriction_W$ is of degree d, the number of coefficients of $p\restriction_W$ is $\text{poly}(d)$. Moreover, each of the coefficients of $p\restriction_W$ is a actually a polynomial of degree d in $\vec{u}$ (see Claim 25.1 for proof of this fact). Thus, instead of taking an OR of $\text{poly}(|\mathbb{F}|)$ values (i.e., of the values in $\{p(w) : w \in W\}$), we can take an OR of $\text{poly}(d)$ values, where each of these values can be computed by a polynomials of degree d in $\vec{u}$.

The first solution is not complete yet, since computing the OR function of $k = \text{poly}(d)$ values requires degree $(|\mathbb{F}| - 1) \cdot k$. To solve this problem, observe that we do not actually need to output 1 on every non-zero input; in fact, it suffices that on every non-zero input, we output *some* non-zero value in $\mathbb{F}$. We call such functions multivalued OR functions, and show that there exists a polynomial $\mathbb{F}^k \to \mathbb{F}$ of degree less than $2 \cdot k$ that computes a multivalued OR function of its inputs (see Proposition 24). It follows that there exists a polynomial $p' : \mathbb{F}^{O(n)} \to \mathbb{F}$ of degree $d' = \text{poly}(d)$ that vanishes on at most $1/\text{poly}(|\mathbb{F}|)$ of its inputs (corresponding to the probability that $p\restriction_W \equiv 0$) such that every non-zero input $\vec{u}$ to p' yields a subspace $W = W_{\vec{u}}$ such that $p\restriction_W \not\equiv 0$.

The solution described above yields the lower bound in Item (2) of Theorem 6, which refers to the badness parameter $b(n) = 1/\text{poly}(|\mathbb{F}|)$. To obtain the lower bound in Item (1), we will again reduce the task of hitting $p : \mathbb{F}^n \to \mathbb{F}$ to the task of hitting $p' : \mathbb{F}^{O(n)} \to \mathbb{F}$ as above, but we will then further reduce the task of hitting p' to the task of hitting polynomials of degree d that vanish rarely (i.e., vanish on at most $O(1/|\mathbb{F}|)$ of their inputs), obtaining a lower bound on the latter. To do so, we show how to randomly compute p' by a distribution that is typically in the class $\mathcal{P}$ of polynomials of degree d that vanish on at most $O(1/|\mathbb{F}|)$ of their inputs (see Proposition 26), and then rely on the claim described in the beginning of Section 2.3 to deduce that any hitting-set generator for $\mathcal{P}$ is also a hitting-set generator for p'.

Recall that p' gets an input $\vec{u}$, and computes a multivalued OR function of $k = \text{poly}(d)$ degree-d polynomials in $\vec{u}$ (corresponding to the coefficients of $p\restriction_{W_{\vec{u}}}$). The distribution that randomly computes p', denoted by $\mathbf{h}$, is simply a random $\mathbb{F}$-linear combination of these k degree-d polynomials. Note that $\mathbf{h}$ is supported on polynomials of degree d, and randomly computes p' with error $1/|\mathbb{F}|$. Moreover, since p' vanishes very rarely (i.e., on at most $1/\text{poly}(|\mathbb{F}|)$ of its inputs), and the error in randomly computing p' is $1/|\mathbb{F}|$, the expected fraction of inputs on which a polynomial in $\mathbf{h}$ vanishes is at most $O(1/|\mathbb{F}|)$. Thus, $\mathbf{h}$ is typically in the class $\mathcal{P}$ of degree-d polynomials that vanish on at most $O(1/|\mathbb{F}|)$ of their inputs. Invoking the claim from the beginning of Section 2.3, any sufficiently dense hitting-set generator for $\mathcal{P}$ also hits p', which allows us to hit p using additional $O(\log(|\mathbb{F}|)) = O(\log(n))$ bits.

3 Preliminaries

Throughout the paper, the letter n will always denote the number of input variables to a function or a circuit. We denote by $\{\mathcal{D} \to \mathcal{R}\}$ the set of functions from domain $\mathcal{D}$ to range $\mathcal{R}$. Distributions and random variables will always be denoted by boldface letters. Given a domain Σ, which will typically be clear from the context, we denote by $\mathbf{u}_k$ the uniform distribution over Σ^k. Given a distribution $\mathbf{d}$, we write $x \sim \mathbf{d}$ to denote a value x that is sampled according to $\mathbf{d}$; when we write $x \in \Sigma^k$ in probabilistic expressions, we mean the uniform distribution over Σ^k.

3.1 Circuit classes and restrictions

We will consider Boolean circuit families $\{C_n\}_{n\in\mathbb{N}}$ such that C_n gets n input bits and outputs a single bit. The circuit class $\mathcal{AC}^0$ consists of all circuit families over the De-Morgan basis (i.e., the gates of the circuit can compute the $\wedge, \vee$, and $\neg$ functions) such that the circuit gates have unbounded fan-in and fan-out, and for every $n \in \mathbb{N}$, the size of C_n (i.e., number of gates) is at most $\text{poly}(n)$, and the depth of C_n (i.e., longest path from an input gate to the output gate) is upper bounded by a constant.

The circuit class $\mathcal{AC}^0[\oplus]$ is defined similarly to $\mathcal{AC}^0$, the only difference being that the basis is extended: The gates can compute the $\wedge, \vee, \neg$, and $\oplus$ functions (rather than only $\wedge, \vee$, and $\neg$). We stress that a $\oplus$-gate can compute either the parity of its input gates, or the negated parity of its input gates. We also assume that all $\mathcal{AC}^0[\oplus]$ circuits are *layered*, in the sense that in a fixed circuit, for every integer d, all gates at distance d from the input gates are of the same gate-type (i.e., either $\wedge$, or $\vee$, or $\oplus$).

Given a function $f : \{0,1\}^n \to \{0,1\}$, a restriction of f is a subset $W \subseteq \{0,1\}^n$. We say that a function f simplifies under a restriction W to a function from a class $\mathcal{H}$ if there exists $h \in \mathcal{H}$ such that for every $w \in W$ it holds that $h(w) = f(w)$. A restriction to a subcube is represented by a string $\rho \in \{0,1,\star\}^n$, where the subcube consists of all $x \in \{0,1\}^n$ such that for every $i \in [n]$ for which $\rho_i \neq \star$ it holds that $x_i = \rho_i$. The living variables under ρ are the input bits indexed by the set $\{i \in [n] : \rho_i = \star\}$. The restricted function $f\restriction_\rho : \{0,1\}^n \to \{0,1\}$ is defined by $f\restriction_\rho(x) = f(y)$, where for every $i \in [n]$ it holds that $y_i = x_i$ if $\rho_i = \star$ and $y_i = \rho_i$ otherwise. We will also consider the composition of restrictions, where a composition $\rho = \rho_1 \circ \rho_2$ yields the restricted function $f\restriction_\rho = \left(f\restriction_{\rho_2}\right)\restriction_{\rho_1}$.

3.2 Pseudorandom generators and hitting-set generators

We will use the following two standard definitions of pseudorandom generators and of hitting-set generators.

Definition 7 *(pseudorandom generators). Let $\mathcal{F} = \bigcup_{n\in\mathbb{N}} \mathcal{F}_n$, where for every $n \in \mathbb{N}$ it holds that $\mathcal{F}_n$ is a set of functions $\{0,1\}^n \to \{0,1\}$, and let $\epsilon : \mathbb{N} \to [0,1]$ and $\ell : \mathbb{N} \to \mathbb{N}$. An algorithm G is a* pseudorandom generator for $\mathcal{F}$ with error parameter ϵ and seed length ℓ *if for every $n \in \mathbb{N}$, when G is given as input 1^n and a random seed of length $\ell(n)$, it* *outputs a string in* $\{0,1\}^n$ *such that for every* $f \in \mathcal{F}_n$ *it holds that* $\left| \Pr_{x\in\{0,1\}^n}[f(x) = 1] - \Pr_{y\in\{0,1\}^{\ell(n)}}[f(G(1^n,y)) = 1] \right| < \epsilon$.

Definition 8 *(hitting-set generators). Let* $\mathcal{F} = \bigcup_{n\in\mathbb{N}} \mathcal{F}_n$*, where for every* $n \in \mathbb{N}$ *it holds that* $\mathcal{F}_n$ *is a set of functions* $\{0,1\}^n \to \{0,1\}$*, and let* $\ell : \mathbb{N} \to \mathbb{N}$*. An algorithm* G *is a* hitting-set generator for $\mathcal{F}$ with seed length ℓ *if for every* $n \in \mathbb{N}$*, when* G *is given as input* 1^n *and a random seed of length* $\ell(n)$*, it outputs a string in* $\{0,1\}^n$ *such that for every* $f \in \mathcal{F}_n$ *it holds that* $\Pr_{y\in\{0,1\}^{\ell(n)}}[f(G(1^n,y)) \neq 0] > 0$*. For* $\epsilon : \mathbb{N} \to (0,1]$*, we say that* G *has* density ϵ *if for every* $n \in \mathbb{N}$ *and* $f \in \mathcal{F}_n$ *it holds that* $\Pr_{y\in\{0,1\}^{\ell(n)}}[f(G(1^n,y)) \neq 0] \geq \epsilon(n)$*.*

In Section 6 we extend Definition 8 by defining hitting-set generators for functions over fields larger than $\mathbb{F}_2$.

3.3 Tail bounds for distributions with limited independence

We will use the following well-known tail bound for t-wise independent distributions (for a proof see [BR94, Lemma 2.3]):

Fact 9 *(tail bound for t-wise independent distributions). Let* $t \geq 4$ *be an even number. Let* $X_1, ..., X_N$ *be variables in* $\{0,1\}$ *that are t-wise independent, and denote* $\mu = \mathbb{E}\left[\frac{1}{N} \cdot \sum_{i\in[N]} X_i\right]$*. Then, for any* $\zeta > 0$ *it holds that* $\Pr\left[\left|\frac{1}{N} \cdot \sum_{i\in[N]} X_i - \mu\right| \geq \zeta\right] \leq 8 \cdot \left(\frac{t\cdot\mu\cdot N + t^2}{\zeta^2 \cdot N^2}\right)^{t/2}$*.*

When the variables are not t-wise independent, but rather "almost" t-wise independent, a weaker tail bound still holds. Specifically, we say that $X_1, ..., X_N \in \{0,1\}^N$ are δ-almost t-wise independent if for every set $S \subseteq [N]$ of size $|S| = t$, the statistical distance between $(X_i)_{i\in S}$ and the uniform distribution over $\{0,1\}^t$ is at most δ. Then, the following well-known tail bound holds:

Fact 10 *(tail bound for almost t-wise independent distributions). Let* $t \geq 4$ *be an even number, and let* $\delta : \mathbb{N} \to [0,1]$*. Let* $X_1, ..., X_N$ *be variables in* $\{0,1\}$ *that are* $\delta(N)$*-almost t-wise independent, and denote* $\mu = \mathbb{E}\left[\frac{1}{N} \cdot \sum_{i\in[N]} X_i\right]$*. Then, for any* $\zeta > 0$ *it holds that* $\Pr\left[\left|\frac{1}{N} \cdot \sum_{i\in[N]} X_i - \mu\right| \geq \zeta\right] < 8 \cdot \left(\frac{t\cdot\mu\cdot N + t^2}{\zeta^2 \cdot N^2}\right)^{t/2} + (2 \cdot N)^t \cdot \delta(N)$*.*

For a proof of Fact 10 see, e.g., [LRTV09, Lemma 18].

4 Proof of Theorem 2

Towards proving Theorem 2, we first need to slightly adapt the derandomized switching lemma in [TX13, Tal14], in order to deduce that when keeping each variable alive with probability p, the total number of variables that the restriction keeps alive is $\Omega(p \cdot n)$, with high probability.[6]

[6] Indeed, this is not proved in [TX13], since they only need the fact that each variable remains alive with probability about p (for a proof of (a generalization of) the latter statement, see [Tal14, Thm F.4]).

Lemma 11 *(an adaptation of the derandomized switching lemma of [TX13, Tal14]). Let $d \geq 3$ be an integer, let $M : \mathbb{N} \to \mathbb{N}$ be a polynomially-bounded function, and let $\epsilon : \mathbb{N} \to [0,1]$ be a function such that $\epsilon(n) \geq 2^{-o(n^{1/d})}$. Then, there exists $k \in \mathbb{N}$ and and $c' > 0$ such that the following holds: There exists a polynomial-time algorithm G such that for every $n \in \mathbb{N}$, when the algorithm G gets as input a random seed of length $\tilde{O}(\log^3(n/\epsilon))$, it outputs a restriction $\rho \in \{0,1,\star\}^n$ such that for every circuit C over n input bits of size at most $M(n)$ and depth d, with probability at least $1-\epsilon$ it holds that:*

1. *The circuit $C\restriction_\rho$ can be computed by a depth-2 circuit of size at most $(n/\epsilon)^k$.*

2. *The restriction ρ leaves at least $c' \cdot \left(n/\log^{d-2}(n/\epsilon)\right)$ variables alive.*

Proof. Loosely speaking, the original result of Trevisan and Xue [TX13] was that "any distribution that fools CNFs also fools the switching lemma". To choose our restrictions, we will use the specific pseudorandom generator for depth-2 circuits of Tal [Tal14], which improves on the generator of De *et al.* [DETT10]. This generator is essentially a δ-biased distribution, and for our purposes we will use $\delta \approx 2^{-O(\log^3(n/\epsilon))}$. Such a distribution admits strong tail bounds (see Fact 10), and this fact will allow us deduce that when keeping each variable alive with probability p, the total number of variables that the restriction keeps alive is $\Omega(p \cdot n)$, with high probability.

Let us now detail the argument in full. Consider a circuit C over n inputs of depth d and size M. For the purpose of the analysis, assume that the circuit C is layered, and add a dummy layer of gates above the inputs, such that the fan-in of the bottom gates of C is one. We apply $d-1$ restrictions that will allow us convert the formulas in the next-to-bottom layer of the circuit from CNFs to DNFs, or vice versa, and thus reduce the depth of the circuit, until the circuit is of depth two. To choose each restriction we will rely on the following lemma of Trevisan and Xue:

Lemma 12 *(the derandomized switching lemma of [TX13, Lemma 7]). Let F be a CNF over n inputs with M' clauses, each clause of width at most t. For a positive parameter $p = 2^{-q}$, where $q \in \mathbb{N}$, let $\rho \in \{0,1,\star\}^n$ be a restriction that is chosen according to distribution over $\{0,1\}^{(q+1)\cdot n}$ that δ_0-fools all CNFs with at most $m = M' \cdot 2^{t\cdot(q+1)}$ clauses.[7] Then, the probability that $F\restriction_\rho$ cannot be computed by a decision tree of depth s is at most*

$$2^{s+t+1} \cdot (5pt)^s + \delta_0 \cdot 2^{(s+1)\cdot(2\cdot t+\log(M'))} . \tag{4.1}$$

The parameters that we use for the restrictions are standard: In the first restriction we use the values $t = 1$ and $p = 1/O(1)$ and $s = O(\log(n/\epsilon))$, which implies that after the restriction, the bottom layer has at most $M' = M \cdot 2^s = \text{poly}(n/\epsilon)$ gates; and in the next $d-2$ restrictions we use the values $t = s = O(\log(n/\epsilon))$ and $p = 1/O(\log(n/\epsilon))$. We want that for each restriction, the expression in Eq. (4.1) will be

[7] We identify strings in $\{0,1\}^{(q+1)\cdot n}$ with restrictions as follows: Each variable is assigned a block of $q+1$ bits in the string; the variable remains alive if the first q bits in the block are all zeroes, and otherwise takes the value of the $(q+1)^{th}$ bit.

at most $\frac{\epsilon}{(d-1)\cdot M}$; thus, we will choose each restriction according to a distribution that fools CNFs of size $m = 2^{\tilde{O}(\log(n/\epsilon))}$ with error $\delta_0 = 2^{-O(\log^2(n/\epsilon))}$. According to [Tal14, Thm E.6], any δ_1-biased distribution fools all CNFs of size m with error δ_0, where $\delta_1 = 2^{-O(\log(m)\cdot\log(m/\delta_0)\cdot\log\log(m))} = 2^{-\tilde{O}(\log^3(n/\epsilon))}$. Thus, we choose each of the $d-1$ restrictions according to a δ_1-biased distribution over $\{0,1\}^{(q+1)\cdot n}$, where $q = O(1)$ in the first restriction, and $q = \log\log(n) + O(1)$ in the subsequent $d-2$ restrictions. The required seed length is $(d-1)\cdot O(\log(n\cdot\log\log(n)/\delta_1)) = \tilde{O}\left(\log^3(n/\epsilon)\right)$.

Now, observe that in each δ_1-biased distribution over $\{0,1\}^{(q+1)\cdot n}$, the blocks of $q \leq \log\log(n/\epsilon) + O(1)$ bits that are used to decide if each variable is kept alive are δ-almost $O(\log(1/\epsilon))$-wise independent, where $\delta = 2^{-\tilde{O}(\log^3(n/\epsilon))}$. Relying on Fact 10 and on the hypothesis that $\epsilon(n) \geq 2^{-o(n^{1/d})}$, with probability at least $1-\epsilon$ it holds that the number of living variables in the end of this process is $\Omega\left(n/\log^{d-2}(n/\epsilon)\right)$ (the exact calculation needed to verify the latter claim is straightforward but slightly tedious, so we defer its presentation to Appendix B). ■

Let us now re-state Theorem 2 more formally, and then prove it:

Theorem 13 *(Theorem 2, restated). Let $d \geq 2$, let $M : \mathbb{N} \to \mathbb{N}$ be a polynomially-bounded function, and let $\epsilon : \mathbb{N} \to [0,1]$ such that $\epsilon(n) \geq 2^{-o(n^{1/d})}$. Let $c > 0$ be any constant that is smaller than the constant c' from Lemma 11. For any $n \in \mathbb{N}$, let $\mathcal{C}_n$ be the class of depth-d circuits over n input bits of size M that accept all but at most $B(n)$ of their inputs, where $\log(B(n)) = c\cdot\left(\frac{n}{\log^{d-2}(n/\epsilon(n))}\right)$, and let $\mathcal{C} = \bigcup_{n\in\mathbb{N}}\mathcal{C}_n$. Then, there exists a polynomial-time hitting-set generator for $\mathcal{C}$ with seed length $\tilde{O}(\log^3(n/\epsilon))$ and density $1-\epsilon$.*

Proof. For any $n \in \mathbb{N}$, given a seed of length $\ell = \tilde{O}(\log^3(n/\epsilon))$, the hitting-set generator first invokes the algorithm from Lemma 11 with the error parameter $\epsilon(n)/3$, to obtain a restriction ρ. [8] Then, the hitting-set generator sets values for the variables that are not fixed by ρ, using the pseudorandom generator in [Tal14, Thm E.6] for depth-2 circuits of size $M'(n) = (n/\epsilon)^k$ (where k is the constant from Lemma 11) with error parameter $\epsilon(n)/3$. (The pseudorandom generator for depth-2 circuits requires a seed of length $\tilde{O}(\log(n) + \log(M'(n)/\epsilon(n))\cdot\log(M'(n))) = o(\ell)$.)

Turning to the analysis, let C be a circuit over n input bits of depth d and size $M(n)$. By Lemma 11, with probability at least $1-\epsilon(n)/3$ the restriction ρ is such that the circuit $C\restriction_\rho$ is a depth-2 circuit of size at most $M'(n) = (n/\epsilon(n))^k$, and ρ leaves at least $c'\cdot n/\log^{d-2}(n/\epsilon(n))$ variables alive. Now, fix a restriction ρ that satisfies both these conditions. Since $C\restriction_\rho$ cannot have more than $B(n)$ unsatisfying inputs, and the number of living variables under ρ is $(c'/c)\cdot\log(B(n)) > \log(B(n))$, it follows that the acceptance probability of $C\restriction_\rho$ is at least $1-\frac{B(n)}{2^{(c'/c)\cdot\log(B(n))}} = 1-\frac{1}{B(n)^{c'/c-1}} > 1-\epsilon(n)/3$. Thus, for any fixed restriction ρ that satisfies both conditions above, the pseudorandom

[8] For circuits of depth $d = 2$, this preliminary step is not needed. Actually, in this case (i.e., $d = 2$), a quantified derandomization algorithm for $B(n) = 2^n/\text{poly}(n)$ was proved in [GW14, Prop 3.1].

generator hits a satisfying input for C with probability at least $1 - \frac{2}{3} \cdot \epsilon(n)$. It follows that the density of the hitting-set generator is at least $1 - \epsilon(n)$. ■

5 Constant-depth circuits with parity gates

In this section we prove the claims made in Section 1.3: In Section 5.1 we prove Theorem 3, and in Section 5.2 we prove Theorem 4.

5.1 Proof of Theorem 3

The proof is a variation on [GW14, Thm 4.2 and Remark 4.4]. Starting from a CNF C, we will employ error-reduction within $\mathcal{AC}^0[\oplus]$, by first sampling inputs for C using Trevisan's extractor [Tre01], and then taking the disjunction of the evaluation of C on these inputs (rather than an approximate majority, as in [GW14]). This will yield a layered circuit of the form $\vee \wedge \vee \oplus$ that accepts all but 2^{n^c} of its inputs, for any desired $c > 0$. Details follow.

Let $C : \{0,1\}^n \rightarrow \{0,1\}$ be a CNF that accepts most of its inputs. For $n' = n^{(1/c)+1}$ and $s = O(\log(n))$, let $E : \{0,1\}^{n'} \times \{0,1\}^s \rightarrow \{0,1\}^n$ be Trevisan's extractor instantiated for min-entropy $(n')^c = n^{1+\Omega(1)}$ and error parameter $1/4$. We construct a circuit $C' : \{0,1\}^{n'} \rightarrow \{0,1\}$ that first computes the values $E(x,z)$, for each possible seed $z \in \{0,1\}^s$, then evaluates C on each value $E(x,z)$, and finally takes an OR of these evaluations; that is, $C'(x) = \vee_{z \in \{0,1\}^s} C\left(E(x,z)\right)$.

Note that C' is a layered depth-4 circuit of the form $\vee \wedge \vee \oplus$, since for each seed $z \in \{0,1\}^s$, the residual function $E_z(x) = E(x,z)$ is just a linear transformation of x. Also note that the number of inputs $x \in \{0,1\}^{n'}$ for which $\Pr_z[C(E(x,z))] < 1/4$ is at most $2^{(n')^c}$. [9] In particular, C' accepts all but at most $2^{(n')^c}$ of its inputs, and for each satisfying input x for C', we can find a corresponding satisfying input for C among $\{E(x,z)\}_{z \in \{0,1\}^s}$.

5.2 Proof of Theorem 4

The current section is organized as follows. In Section 5.2.1 we present two algorithmic tools that will be used in the proof: An adaptation of the approach of Chaudhuri and Radhakrishnan [CR96] to our setting, and an adaptation of Viola's pseudorandom generator [Vio09] to polynomials that are defined over an affine subspace. Then, in the next three sections, we prove the corresponding three items of Theorem 4.

We rely on the notion of *affine restrictions*. A restriction of a circuit $C : \{0,1\}^n \rightarrow \{0,1\}$ to an affine subspace $W \subseteq \{0,1\}^n$ will be constructed by accumulating a list of (independent) affine conditions that defines W. That is, each of the various algorithms will construct a full-rank matrix A and a vector b such that $W = \{x : Ax = b\}$. For an affine function g, when we say that an algorithm "adds $g = 0$ to the list of affine

[9] Otherwise, the uniform distribution on such inputs yields a source X of min-entropy $(n')^c$ such that C distinguishes $E(X)$ from the uniform distribution over $\{0,1\}^n$ with probability $1/4$.

conditions", we mean that it extends A by adding the linear part of g as an additional row to A, and extends b by adding the constant term of g as an additional bit to b (i.e., if $g(x) = \sum_{i=1}^{n} c_i x_i + c_0$ then the row $c = (c_1, ..., c_n)$ is added to A and c_0 is added to b). After each addition of a condition, we will say that the algorithm "simplifies the circuit accordingly"; by this we mean that for any $\oplus$-gate g' in the bottom layer whose linear function is dependent on the rows of A, the algorithm fixes g' to the appropriate value determined by A and b, and, if g' was fixed to zero, then the algorithm removes all the $\wedge$-gates that g' feeds into.

5.2.1 Two algorithmic tools

Let us first adapt the approach of Chaudhuri and Radhakrishnan [CR96], which was originally used to construct "bit-fixing" restrictions for $\mathcal{AC}^0$ circuits, to the setting of $\oplus \wedge \oplus$ circuits and affine restrictions.

Proposition 14 *(whitebox affine restrictions for $\oplus \wedge \oplus$ circuits). For two integers $m_\wedge$ and $m_\oplus$, let $\mathcal{C}$ be the class of $\oplus \wedge \oplus$ circuits over n input bits with $m_\wedge$ gates in the middle layer and $m_\oplus$ gates in the bottom layer. Then, for any two integers $d_\oplus$ and $d_\wedge$, there exists a polynomial-time algorithm that, when given as input a circuit $C \in \mathcal{C}$, outputs an affine subspace $W \subseteq \{0,1\}^n$ such that:*

1. *In the restriction of C to W, each $\wedge$-gate in the middle layer has fan-in at most $d_\wedge$.*
2. *The subspace W is of co-dimension at most $\frac{m_\wedge}{d_\oplus} + \frac{d_\oplus \cdot m_\oplus}{d_\wedge}$.*

Proof. The algorithm operates in two steps. In the first step, as long as there exists a $\oplus$-gate g in the bottom layer with *fan-out* at least $d_\oplus$, the algorithm adds the condition $g = 0$ to the list of affine conditions, and simplifies the circuit accordingly. Note that each addition of a condition as above fixes at least $d_\oplus$ of the $\wedge$-gates in the middle layer, and thus at most $m_\wedge / d_\oplus$ conditions are added (or else the entire circuit simplifies to a constant). Hence, after the first step concludes, the fan-out of each $\oplus$-gate in the bottom layer is $d_\oplus$, and at most $m_\wedge / d_\oplus$ affine conditions have been accumulated.

In the second step, as long as there exists an $\wedge$-gate g in the middle layer with *fan-in* at least $d_\wedge$, the algorithm (arbitrarily) chooses one $\oplus$-gate g' that feeds into g, adds the condition $g' = 0$ to the list of affine conditions, and simplifies the circuit accordingly. Note that, in the beginning of the second step, the number of wires feeding the middle layer is at most $d_\oplus \cdot m_\oplus$ (since there are at most $m_\oplus$ gates in the bottom layer, each of them with fan-out at most $d_\oplus$). Now, note that each addition of an affine condition in the second step eliminates at least $d_\wedge$ wires; thus, the algorithm adds at most $\frac{d_\oplus}{d_\wedge} \cdot m_\oplus$ conditions in the second step. After the second step is complete, each $\wedge$-gate in the middle layer has fan-in at most $d_\wedge$, and the list of affine conditions contains at most $m_\wedge / d_\oplus + \frac{d_\oplus}{d_\wedge} \cdot m_\oplus$ conditions. ■

We now verify that we can use Viola's pseudorandom generator [Vio09] in order to "fool" $\oplus \wedge \oplus$ circuits that, when restricted to an affine subspace, have a constant maximal fan-in of the $\wedge$-gates.

Proposition 15 *(invoking Viola's PRG in an affine subspace). There exists an algorithm G that, for every $n \in \mathbb{N}$, when G is given as input an integer D, a seed of $\ell = O(\log(n))$ bits, and a basis for an affine subspace $W \subseteq \{0,1\}^n$, then G runs in time $\mathrm{poly}(n)$ and satisfies the following: For every $\oplus \wedge \oplus$ circuit C over n input bits such that C simplifies under the restriction W to a $\oplus \wedge \oplus$ circuit in which the maximal fan-in of $\wedge$-gates is D and such that $C\restriction_W \not\equiv 0$, it holds that $\Pr[C(G(\mathbf{u}_\ell)) = 1] > 0$.*

Proof. Denote the dimension of W by $m = \dim(W)$. The algorithm G first finds a full-rank $n \times m$ matrix B and $s \in \{0,1\}^n$ such that $x \mapsto Bx + s$ maps $\{0,1\}^m$ to W. Then, the algorithm G uses its random seed to invoke Viola's pseudorandom generator for polynomials $\mathbb{F}_2^m \to \mathbb{F}_2$ of degree D, with error parameter $2^{-(D+1)}$, thus obtaining a string $x \in \{0,1\}^m$. Finally, the algorithm G outputs the string $Bx + s$.

Now, let C be $\oplus \wedge \oplus$ circuit as in the hypothesis, and consider the polynomial $p : \mathbb{F}_2^m \to \mathbb{F}_2$ such that $p(x) = C(Bx + s)$. Note that p is of degree D, because C computes an sum of monomials of degree D over $\mathbb{F}_2$, and the affine transformation does not increase the degree. Also, using our hypothesis that p is non-zero, it follows that the acceptance probability of p is at least 2^{-D}. Thus, the probability that Viola's generator will output x such that $p(x) = 1$ is at least $2^{-(D+1)} > 0$, and each such x yields a string $y = Bx + s$ such that $C(y) = 1$. ■

5.2.2 Linear-sized circuits with $B(n) = 2^{-\Omega(n)}$

We prove the first item of Theorem 4 by invoking the whitebox algorithm from Proposition 14 with appropriate parameters $d_\wedge, d_\oplus = O(1)$, and then using the generator from Proposition 15.

Proposition 16 *(Theorem 4, Item (1): hitting biased linear-sized $\oplus \wedge \oplus$ circuits). Let $\epsilon > 0$ be an arbitrarily small constant, and let $c > 0$ be an arbitrarily large constant. Let $\mathcal{C}$ be the class of $\oplus \wedge \oplus$ circuits such that any circuit $C \in \mathcal{C}$ over n input bits has at most $c \cdot n$ gates and accepts all but at most $2^{(1-\epsilon)\cdot n}$ of its inputs. Then, there exists a polynomial-time algorithm that, when given any circuit $C \in \mathcal{C}$, finds a satisfying input for C.*

Proof. The algorithm first invokes the algorithm from Proposition 14 with parameters $d_\oplus = \frac{4 \cdot c}{\epsilon}$ and $d_\wedge = d_\oplus^2$, to obtain an affine subspace W of co-dimension at most

$$\frac{m_\wedge}{d_\oplus} + \frac{d_\oplus \cdot m_\oplus}{d_\wedge} < 2 \cdot \frac{c \cdot n}{(4 \cdot c)/\epsilon} = \frac{\epsilon}{2} \cdot n$$

such that in the restriction of C to W, every $\wedge$-gate in the middle layer has fan-in at most $d_\wedge = O(1)$. Since the circuit C has at most $2^{(1-\epsilon)\cdot n}$ unsatisfying inputs, it follows that $\Pr_{w \in W}[C(w) = 1] \geq 1 - 2^{-(\epsilon/2)\cdot n}$. Thus, the algorithm concludes by invoking the algorithm from Proposition 15. ■

5.2.3 Sub-quadratic circuits with $(1 + o(1)) \cdot n$ bottom $\oplus$-gates and $B(n) = 2^{n^c}$

We now prove the second item of Theorem 4.

Proposition 17 *(Theorem 4, Item (2): hitting biased sub-quadratic $\oplus \wedge \oplus$ circuits). Let $\epsilon > 0$ and let $0 < c < \epsilon$. Let $\mathcal{C}$ be the class of $\oplus \wedge \oplus$ circuits such that any $C \in \mathcal{C}$ over n input bits has at most $n + n^c$ bottom $\oplus$-gates, and at most $n^{2-\epsilon}$ middle $\wedge$-gates, and accepts all but $B(n) = 2^{n^c}$ of its inputs. Then, there exists a polynomial-time algorithm that, when given any circuit $C \in \mathcal{C}$, finds a satisfying input for C.*

Proof. Recall that a high-level overview of the proof, which used the parameter values $m_\wedge = n^{1.1}$ and $m_\oplus = n$, appeared in Section 2.2. Let us first explain, in high-level, how to handle the setting of $m_\wedge \leq n^{2-\epsilon}$; for the moment, we are still assuming that $m_\oplus = n$. As in the overview in Section 2.2, the algorithm works in two steps. In the first step, we use Proposition 14 to fix $o(m_\oplus)$ of the $\oplus$-gates such that after the restriction, the fan-in of the $\wedge$-gates is bounded by $w = n^{1-\alpha\cdot\epsilon}$, where $\alpha < 1$ is a constant slightly smaller than 1; this is possible because $m_\wedge \leq n^{2-\epsilon}$ (see the proof details below). In the second step, we restrict the $\oplus$-gates using an $O(1)$-independent distribution, keeping each $\oplus$-gate alive with probability $p = n^{-(1-\beta\cdot\epsilon)}$, where $\beta < \alpha$ (and recall that we choose arbitrary consistent values for the gates that are fixed). The crucial point is the following: On the one hand, since $p \leq 1/w^{1+\Omega(1)}$, after the second step the fan-in of the $\wedge$-gates is upper-bounded by a constant (as explained in Section 2.2); and on the other hand, the number of living $\oplus$-gates after the second step is approximately $p \cdot (1 - o(1)) \cdot n = \Omega\left(n^{\beta\cdot\epsilon}\right) > n^c = \log(B(n))$, where the inequality holds if we choose $\beta > c/\epsilon$ (which is possible if we initially choose $\alpha \in (c/\epsilon, 1)$).

To see how we handle the setting of $m_\oplus \leq n + n^c$ (rather than $m_\oplus = n$), note that the overall number of affine conditions that the algorithm imposes is $m_\oplus - \Omega(p \cdot m_\oplus)$. Since $m_\oplus \leq n + o(p \cdot n)$, the number of affine conditions is at most $n - \Omega(p \cdot n)$, which means that the affine subspace W is of dimension $\Omega(p \cdot n) > \log(B(n))$.

Let us now provide the full details for the proof. Assume, without loss of generality, that $m_\oplus \geq n$ (we can add dummy gates if necessary). We first invoke the algorithm from Proposition 14 with parameters $d_\wedge = n^{1-\alpha\cdot\epsilon}$, where $\alpha = \frac{(c/\epsilon)+1}{2}$, and $d_\oplus = n^{1-\alpha'\cdot\epsilon}$, where $\alpha' = (c/\epsilon) + (2/3) \cdot (1 - c/\epsilon) > \alpha$. The algorithm outputs an affine subspace of co-dimension at most

$$\begin{aligned}\frac{m_\wedge}{d_\oplus} + \frac{d_\oplus \cdot m_\oplus}{d_\wedge} &\leq n^{2-\epsilon-(1-\alpha'\cdot\epsilon)} + n^{1-\alpha'\cdot\epsilon-(1-\alpha\cdot\epsilon)} \cdot m_\oplus \\ &= n^{1-(1-\alpha')\cdot\epsilon} + n^{-(\alpha'-\alpha)\cdot\epsilon} \cdot m_\oplus \ ,\end{aligned}$$

which is $o(m_\oplus)$, such that in the restriction of C to the subspace, every $\wedge$-gate in the middle layer has fan-in at most $d_\wedge = n^{1-\alpha\cdot\epsilon}$.

Denote the number of $\oplus$-gates that were not fixed in the previous step by m', and consider the following pseudorandom restriction process. For a sufficiently large constant $\gamma > 1$ (which will be determined later), we use a γ-wise independent distribution over $[1/p]^{n'}$, where $p = n^{-(1-\beta\cdot\epsilon)}$ and $\beta = (c/\epsilon) + (1/3) \cdot (1 - c/\epsilon) < \alpha$. [10] Denote the random variable that is the output string of this distribution by $\rho \in [1/p]^{n'}$. For

[10] We will actually use the value $p = 2^{-\lceil(1-\beta\cdot\epsilon)\cdot\log(n)\rceil}$, such that $1/p$ is a power of 2, but the difference between this value and $n^{-(1-\beta\cdot\epsilon)}$ is insignificant in what follows.

every $\oplus$-gate that has not been restricted by the algorithm from Proposition 14, the algorithm now marks the gate as "alive" if and only if the corresponding element in the string ρ equals zero; otherwise, it marks the gate as "fixed".

For any $\wedge$-gate g in the middle-layer, the probability that at least γ gates that feed into g are marked "alive" is at most

$$\binom{d_\wedge}{\gamma} \cdot p^\gamma < n^{(1-\alpha\cdot\epsilon)\cdot\gamma} \cdot n^{-(1-\beta\cdot\epsilon)\cdot\gamma} = n^{-(\alpha-\beta)\cdot\epsilon\cdot\gamma} ,$$

which can be made less than $1/m_\wedge = n^{-(2-\epsilon)}$ by an appropriate choice of γ (i.e., $\gamma > \frac{2-\epsilon}{(\alpha-\beta)\cdot\epsilon}$). After union-bounding over all $\wedge$-gates, we have that with probability at least 0.99, each $\wedge$-gate is fed by less than γ of the "alive" $\oplus$-gates. Also note that with probability at least 0.99, the number of $\oplus$-gates that were marked as "alive" is at least $(p \cdot m')/2$; this is because the distribution is γ-wise independent (so we can use Fact 9). The algorithm and finds a choice of ρ, denoted by ρ_0, that meets both these conditions (by enumerating the outputs of the γ-wise independent distribution). Then, the algorithm iteratively fixes values for the $\oplus$-gates that are marked as "fixed" by ρ_0. Specifically, as long as there is a $\oplus$-gate g that is marked as "fixed" by ρ_0, the algorithm adds the condition $g = 0$ to the list of affine conditions that defines W, and simplifies the circuit accordingly.

Let us now count the number of affine conditions that the algorithm imposed (i.e., the co-dimension of W). After all the restrictions, the number of living variables is at least $(p/2) \cdot m' \geq (p/2) \cdot (1 - o(1)) \cdot m_\oplus \geq (p/3) \cdot m_\oplus$, which implies that the number of affine conditions is at most $m_\oplus - (p/3) \cdot m_\oplus$. Since $m_\oplus \leq n + n^c$, we have that

$$\begin{aligned} m_\oplus - (p/3) \cdot m_\oplus &< n + n^c - (p/3) \cdot n \\ &= n + n^c - \frac{1}{3} \cdot n^{\beta\cdot\epsilon} , \end{aligned}$$

which is less than $n - n^c$, because $n^c = o(n^{\beta\cdot\epsilon})$ (since $\beta \cdot \epsilon = c + \Omega(1)$).

Thus, the algorithm is left with a subspace W of dimension more than $n^c = \log(B(n))$ such that when the circuit C is restricted to the subspace W, the fan-in of every $\wedge$-gate in the middle layer is at most $\gamma = O(1)$. Hence, at this point the algorithm can invoke the algorithm from Proposition 15, and find a satisfying input for C in W. ■

5.2.4 Circuits with a slightly super-linear number of bottom $\oplus$-gates and slightly sub-linear number of $\wedge$-gates

We now prove the third item of Theorem 4. The crucial observation here is that after invoking the algorithm from Proposition 14, the number of $\oplus$-gates is at most $m_\wedge \cdot d_\wedge$, since this is the number of wires that feed into the middle layer.

Proposition 18 *(Theorem 4, Item (3): hitting biased $\oplus\wedge\oplus$ circuits with a super-linear number of $\oplus$-gates). For any constant $\epsilon > 0$, let $\mathcal{C}$ be the class of $\oplus\wedge\oplus$ circuits such that* *any circuit $C \in \mathcal{C}$ over n input bits has at most $n^{1+\epsilon}$ gates in the bottom layer and at most $(1/5) \cdot n^{1-\epsilon}$ gates in the middle layer, and accepts all but at most $B(n) = 2^{n/15}$ of its inputs. Then, there exists a polynomial-time algorithm that, when given any circuit $C \in \mathcal{C}$, finds a satisfying input for C.*

Proof. We first invoke the algorithm from Proposition 14 with parameters $d_\oplus = 1$ and $d_\wedge = (5/2) \cdot n^\epsilon$. The algorithm outputs an affine subspace W' of co-dimension at most

$$\frac{m_\wedge}{d_\oplus} + \frac{d_\oplus \cdot m_\oplus}{d_\wedge} \leq (1/5) \cdot n^{1-\epsilon} + (2/5) \cdot n$$

such that in the restriction of C to W', every $\wedge$-gate in the middle layer has fan-in at most $d_\wedge = (5/2) \cdot n^\epsilon$. Since there are at most $m_\wedge = (1/5) \cdot n^{1-\epsilon}$ gates in the middle layer, it follows that there are at most $m_\wedge \cdot d_\wedge = n/2$ bottom $\oplus$-gates that influence the output of $C\restriction_{W'}$. By fixing values for these gates, we obtain a subspace W of dimension at least $(1/2 - (2/5) - o(1)) \cdot n > n/15$ such that $C\restriction_W$ is constant. Since $B(n) = 2^{n/15}$, it follows that $C\restriction_W \equiv 1$, and thus we can output any $w \in W$. ■

6 Polynomials that vanish rarely

In this section we prove Theorem 5 and Theorem 6. In Section 6.1 we will state and prove a lemma that will be used in both proofs and may also be of independent interest. The proofs of Theorem 5 and Theorem 6 appear in the subsequent sections.

We will use a standard definition for hitting-set generators over large fields, which extends Definition 8. The following definition requires that the generator G will output a value x such that the relevant function evaluates to any non-zero value on x.

Definition 19 *(hitting-set generators over large fields). For every $n \in \mathbb{N}$, let $\mathbb{F}$ be a finite field of size that may depend on n, and let $\mathcal{F}_n$ be a set of functions $\mathbb{F}^n \to \mathbb{F}$. Let $\mathcal{F} = \bigcup_{n \in \mathbb{N}} \mathcal{F}_n$. For a function $\ell : \mathbb{N} \to \mathbb{N}$, an algorithm G is a* hitting-set generator for $\mathcal{F}$ with seed length *ℓ if for every $n \in \mathbb{N}$, when G is given as input 1^n and a random seed of $\ell(n)$ bits (i.e., a random string in $\{0,1\}^{\ell(n)}$), it outputs n elements of $\mathbb{F}$ such that for every $f \in \mathcal{F}_n$ it holds that $\Pr_{y \in \{0,1\}^{\ell(n)}}[f(G(1^n, y)) \neq 0] > 0$. For $\epsilon : \mathbb{N} \to (0,1]$, we say that G has density ϵ if for every $n \in \mathbb{N}$ and $f \in \mathcal{F}_n$ it holds that $\Pr_y[f(G(1^n, y)) \neq 0] \geq \epsilon(n)$.*

In Definition 19, the generator G gets a seed from $\{0,1\}^\ell$, rather than from $\mathbb{F}^\ell$ (as is also common in some texts); indeed, the seed length $\ell(n)$ of the generator G might depend on the size of $\mathbb{F}$. This choice was made because it is more general, and because we want to measure the seed length in bits.

6.1 Randomly computing a function by a distribution that is typically over simpler functions

Detailing the discussion in Section 2.3, let us now formally define what it means to "randomly compute a function by a distribution of functions that is typically over simpler functions".

Definition 20 *(randomly computing a function). Let $\mathbb{F}$ be a finite field, let $p : \mathbb{F}^n \to \mathbb{F}$, and let $\mathcal{H}$ be a class of functions $\mathbb{F}^n \to \mathbb{F}$. For $\rho, \rho' > 0$, we say that p can be* randomly computed with error ρ by a distribution $\mathbf{h}$ that is $(1 - \rho')$-typically in $\mathcal{H}$, *if:*

1. *For every $x \in \mathbb{F}^n$ it holds that $\Pr[p(x) = \mathbf{h}(x)] \geq 1 - \rho$.*

2. *The probability that $\mathbf{h} \in \mathcal{H}$ is at least $1 - \rho'$.*

If $\rho' = 0$, then we say that $\mathbf{h}$ is always in $\mathcal{H}$.

Recall that the bias of a function $p : \mathbb{F}^n \to \mathbb{F}$ under distribution $\mathbf{w}$ is defined as $\mathbb{E}[\mathrm{e}(p(\mathbf{w}))]$, where $\mathrm{e} : \mathbb{F} \to \mathbb{C}$ is some (fixed) non-trivial character. The following claim is implicit in Bogdanov and Viola [BV10, Proof of Lemma 23]: If p can be computed with error ρ by a distribution $\mathbf{h}$ that is always in $\mathcal{H}$, then any distribution $\mathbf{w}$ over $\mathbb{F}^n$ that "fools" every $h \in \mathcal{H}$ also "fools" p, where "fooling" a function f means that $\left|\mathbb{E}[\mathrm{e}(f(\mathbf{w}))] - \mathbb{E}[\mathrm{e}(f(\mathbf{u}_n))]\right|$ is small.

We explicitly state and prove the claim above, while also extending it in three ways. First, we relax the requirement that $\mathbf{h}$ is supported only on $\mathcal{H}$, by allowing that $\Pr[\mathbf{h} \notin \mathcal{H}] = \rho' > 0$. Secondly, instead of only considering characters $\mathrm{e} : \mathbb{F} \to \mathbb{C}$, we consider any arbitrary mapping $\xi : \mathbb{F} \to \mathbb{C}$. And lastly, we also consider a "hitting" version, which asserts that if for every $h \in \mathcal{H}$ it holds that $h(\mathbf{w}) \neq 0$, with high probability, then $p(\mathbf{w})$ is also non-zero, with high probability.

Lemma 21 *(an extension of a claim that is implicit in [BV10]). Let $n \in \mathbb{N}$, and let $\mathbb{F}$ be any finite field. Let $\rho, \rho', \rho'' > 0$ be three parameters. Let $p : \mathbb{F}^n \to \mathbb{F}$, let $\mathcal{H} \subseteq \{\mathbb{F}^n \to \mathbb{F}\}$, and assume that p can be randomly computed with error ρ by a distribution $\mathbf{h}$ over $\{\mathbb{F}^n \to \mathbb{F}\}$ that is $(1 - \rho')$-typically in $\mathcal{H}$. Then,*

1. *Let $\xi : \mathbb{F} \to \mathbb{C}$ be any mapping, and let $\delta = \max_{v,w \in \mathbb{F}} \{|\xi(v) - \xi(w)|\}$. Let $\mathbf{w}$ be a distribution over $\mathbb{F}^n$ such that for every $h \in \mathcal{H}$ it holds that $\left|\mathbb{E}[\xi(h(\mathbf{u}_n))] - \mathbb{E}[\xi(h(\mathbf{w}))]\right| < \rho''$. Then, $\left|\mathbb{E}[\xi(p(\mathbf{u}_n))] - \mathbb{E}[\xi(p(\mathbf{w}))]\right| < 2\delta \cdot \rho + \delta \cdot \rho' + \rho''$.*

2. *Let $S \subseteq \mathbb{F}$. Let $\mathbf{w}'$ be a distribution over $\mathbb{F}^n$ such that for every $h \in \mathcal{H}$ it holds that $\Pr[h(\mathbf{u}_n) \in S] \geq 1 - \rho''$. Then, $\Pr[p(\mathbf{w}') \in S] \geq 1 - \rho - \rho' - \rho''$.*

Note that Item (2) of Lemma 21 can be used with the set $S = \mathbb{F}^* = \mathbb{F} \setminus \{0\}$ to deduce that $p(\mathbf{w}') \neq 0$, with high probability.

Proof of Lemma 21. We first prove Item (1). For simplicity of notation, define $p' = \xi \circ p : \mathbb{F}^n \to \mathbb{C}$ and $h' = \xi \circ h : \mathbb{F}^n \to \mathbb{C}$. By the triangle inequality, we have that

$$\begin{aligned}
\left|\mathbb{E}[p'(\mathbf{u}_n)] - \mathbb{E}[p'(\mathbf{w})]\right| \leq & \left|\mathbb{E}[p'(\mathbf{u}_n)] - \mathbb{E}_{h \sim \mathbf{h}}\left[h'(\mathbf{u}_n)\right]\right| + \\
& \left|\mathbb{E}_{h \sim \mathbf{h}}\left[h'(\mathbf{u}_n)\right] - \mathbb{E}_{h \sim \mathbf{h}}\left[h'(\mathbf{w})\right]\right| + \\
& \left|\mathbb{E}_{h \sim \mathbf{h}}\left[h'(\mathbf{w})\right] - \mathbb{E}[p'(\mathbf{w})]\right| . \qquad (6.1)
\end{aligned}$$

To upper bound the first term in Eq. (6.1), note that

$$\begin{aligned}\Big|\mathbb{E}[p'(\mathbf{u}_n)] - \mathbb{E}_{h\sim\mathbf{h}}\left[h'(\mathbf{u}_n)\right]\Big| &\le \mathbb{E}_{u\in\mathbb{F}^n, h\sim\mathbf{h}}\left[\Big|p'(u) - h'(u)\Big|\right] \\ &\le \mathbb{E}_{u\in\mathbb{F}^n}\left[\Pr_{h\sim\mathbf{h}}[h(u) \neq p(u)] \cdot \max_{v,w\in\mathbb{F}}\{|\xi(v) - \xi(w)|\}\right] \\ &\le \delta \cdot \rho\,,\end{aligned}$$

where the last inequality holds because for every fixed $u \in \mathbb{F}^n$ it holds that $\Pr_{h\sim\mathbf{h}}[h(u) \neq p(u)] \le \rho$. The third item is similarly upper bounded by $\delta \cdot \rho$, by replacing the uniform choice of $u \in \mathbb{F}^n$ with a choice of u according to the distribution $\mathbf{w}$.

To upper bound the second term in Eq. (6.1), note that

$$\begin{aligned}\Big|\mathbb{E}_{h\sim\mathbf{h}}[h'(\mathbf{u}_n)] - \mathbb{E}_{h\sim\mathbf{h}}[h'(\mathbf{w})]\Big| &\le \mathbb{E}_{h\sim\mathbf{h}}\left[\Big|\mathbb{E}[h'(\mathbf{u}_n)] - \mathbb{E}[h'(\mathbf{w})]\Big|\right] \\ &\le \Pr_{h\sim\mathbf{h}}[h \notin \mathcal{H}] \cdot \max_{v,w\in\mathbb{F}}\{|\xi(v) - \xi(w)|\} \\ &\quad + \mathbb{E}_{h\sim\mathbf{h}}\left[|\mathbb{E}[h'(\mathbf{u}_n)] - \mathbb{E}[h'(\mathbf{w})]|\Big| h \in \mathcal{H}\right]\,,\end{aligned}$$

which is upper bounded by $\delta \cdot \rho' + \rho''$. (Specifically, the first term is upper bounded by $\delta \cdot \rho'$, whereas to bound the second term by ρ'' we use the hypothesis that for every $h \in \mathcal{H}$ it holds that $\Big|\mathbb{E}[h'(\mathbf{u}_n)] - \mathbb{E}[h'(\mathbf{w})]\Big| < \rho''$.)

To prove Item (2), first observe that $\Pr[p(\mathbf{w}') \in S] \ge \mathbb{E}_{h\sim\mathbf{h}}[\Pr[h(\mathbf{w}') \in S]] - \rho$. This is the case because

$$\begin{aligned}\mathbb{E}_{h\sim\mathbf{h}}\left[\Pr[h(\mathbf{w}') \in S]\right] &= \mathbb{E}_{x\sim\mathbf{w}'}\left[\Pr_{h\sim\mathbf{h}}[h(x) \in S]\right] \\ &\le \Pr_{x\sim\mathbf{w}'}[p(x) \in S] + \Pr_{x\sim\mathbf{w}'}[p(x) \notin S] \cdot \max_{x:p(x)\notin S}\left\{\Pr_{h\sim\mathbf{h}}[h(x) \in S]\right\} \\ &\le \Pr[p(\mathbf{w}') \in S] + \rho\,.\end{aligned}$$

By our hypothesis, the probability that $\mathbf{h} \notin \mathcal{H}$ is at most ρ', and for every $h \in \mathcal{H}$ it holds that $\Pr[h(\mathbf{w}') \notin S] \le \rho''$. Therefore, $\mathbb{E}_{h\sim\mathbf{h}}[\Pr[h(\mathbf{w}') \notin S]] \le \rho' + (1-\rho') \cdot \rho'' \le \rho' + \rho''$, which implies that $\Pr[p(\mathbf{w}') \in S] \ge 1 - (\rho' + \rho'') - \rho$. ■

We note that further generalization of Lemma 21 is possible. In particular, the proof above does not use the fact that the domain is $\mathbb{F}^n$ and the range is $\mathbb{F}$, and essentially works without change for an arbitrary domain $\mathcal{D}$ and range $\mathcal{R}$.

6.2 Proof of Theorem 5

In this section we prove Theorem 5. Recall that in the current section (and throughout Section 6) we consider a normalized "badness" parameter $b(n) = B(n)/2^n$. We will in fact prove a more general version of Theorem 5, which depends on additional parameters; after stating this general version, we will spell out the parameter choices that yield Theorem 5.

Proposition 22 *(Theorem 5, parametrized version). For $m : \mathbb{N} \to \mathbb{N}$ and $b : \mathbb{N} \to [0, \frac{1}{2}]$, let $\mathcal{C}$ be the class of $\oplus \wedge \oplus$ circuits over n input bits with $m = m(n)$ $\wedge$-gates that accept all but a $b(n)$ fraction of their inputs. For any $d \geq 2$ and $c' \leq 2^d/m$, let $\mathcal{P}_d^{c'}$ be the class of polynomials $\mathbb{F}_2^n \to \mathbb{F}_2$ of degree d that accept all but a $c' \cdot \left(m \cdot 2^{-d}\right)$ fraction of their inputs.*

Let d be an integer such that $\log(m) < d \leq \min\{\log(m) + \log(1/b(n)), n\}$, and let $2 < c' \leq 2^d/m$ be a real number. Assume that there exists a hitting-set generator G with density more than $(2/c') + m \cdot 2^{-d}$ for $\mathcal{P}_d^{c'}$. Then, G is a hitting-set generator for $\mathcal{C}$.

To obtain parameters as in Theorem 5, let $\epsilon = \epsilon(n)$ such that $2^{-n/2} \leq \epsilon \leq 1/8$, and let $m = m(n) \leq 2^{n/2}$. For $d = \lfloor \log(m) + \log(1/\epsilon) \rfloor \leq n$ and $c' = 4 \leq 2^d/m$, assume that there exists a hitting-set generator G for the class $\mathcal{P}_d^{c'}$ with density $1/2 + 2 \cdot \epsilon \geq (2/c') + m \cdot 2^{-d}$. Then, Proposition 22 asserts that G is a hitting-set generator for the class of $\oplus \wedge \oplus$ circuits with m $\wedge$-gates that accept all but $\epsilon \cdot 2^n$ of their inputs.

Proof. Let $C : \{0,1\}^n \to \{0,1\}$ be a $\oplus \wedge \oplus$ circuit with m $\wedge$-gates that accepts all but a $b(n)$ fraction of its inputs. We will show how to randomly compute C by a distribution that is typically in the class $\mathcal{P}_d^{c'}$, and then rely on Lemma 21 to deduce that any sufficiently dense hitting-set generator for $\mathcal{P}_d^{c'}$ also hits C.

The distribution over polynomials is obtained using Razborov's approximating polynomials method [Raz87]. Our goal is to randomly replace each $\wedge$-gate g that has fan-in more than d with a polynomial $g' : \{0,1\}^n \to \{0,1\}$ of degree d such that for every fixed input $x \in \{0,1\}^n$ it holds that $g(x) = g'(x)$ with probability at least $1 - 2^{-d}$. To this purpose, given $g(x) = \wedge_{j=1}^k L_j(x)$, where $k > d$ and the L_j's are linear functions, we randomly choose d subsets $S_1, ..., S_d \subseteq [k]$, and replace g with the $\mathbb{F}_2$-polynomial $g'(x) = \Pi_{i=1}^d \left(1 + \sum_{j \in S_i} (L_i(x) + 1)\right)$. [11]

The above yields a random polynomial $p : \mathbb{F}_2^n \to \mathbb{F}_2$ of degree at most d such that for every fixed $x \in \{0,1\}^n$ it holds that $\Pr[p(x) = C(x)] \geq 1 - m \cdot 2^{-d}$. The expected fraction of unsatisfying inputs for p is at most $2m \cdot 2^{-d}$; this is because

$$\begin{aligned}\mathbb{E}_p\left[\Pr_x[p(x) = 0]\right] &= \mathbb{E}_x\left[\Pr_p[p(x) = 0]\right] \\ &\leq \Pr_x[C(x) = 0] + \Pr_x[C(x) = 1] \cdot \max_x\left\{\Pr_p[p(x) \neq C(x)]\right\} \\ &\leq b(n) + m \cdot 2^{-d},\end{aligned}$$

and since $d \leq \log(m) + \log(1/b(n))$ we have that $m \cdot 2^{-d} \geq b(n)$. Thus, the probability that the fraction of unsatisfying inputs for p is more than $c' \cdot \left(m \cdot 2^{-d}\right)$ is at most $2/c'$.

The above shows that C be be randomly computed with error $m \cdot 2^{-d}$ by a distribution that is $(1 - 2/c')$-typically in $\mathcal{P}_d^{c'}$. Now, let $G : \{0,1\}^\ell \to \{0,1\}^n$ be a hitting-set generator with density $1 - c > (2/c') + m \cdot 2^{-d}$ for $\mathcal{P}_d^{c'}$. Relying on Item (2) of

[11] Using the standard analysis, if $g(x) = 1$, then $L_j(x) = 1$ for all $j \in [k]$, which implies that $g'(x) = 1$ with probability one; and if $g(x) = 0$, then for every $i \in [d]$, with probability $1/2$ over choice of S_i it holds that $\sum_{j \in S_i} (L_i(x) + 1) = 1$, which implies that $g'(x) = 0$ with probability $1 - 2^{-d}$.

Lemma 21, we have that

$$\Pr[C(G(\mathbf{u}_\ell)) = 1] \geq 1 - m \cdot 2^{-d} - (2/c') - c > 0\,,$$

which concludes the proof. ■

6.3 Proof of Theorem 6

For this section, we first define and construct *multivalued OR functions*. We say that a function $f : \mathbb{F}^k \to \mathbb{F}$ is a multivalued OR function if $f(0, ..., 0) = 0$, and for every $x \neq (0, ..., 0)$ it holds that $f(x) \neq 0$. Indeed, for any non-zero input $x \neq (0, ..., 0)$, we require that f outputs *some* non-zero value.

Definition 23 *(multivalued OR functions). Let $\mathbb{F}$ be a finite field, and let k be an integer. We say that $f : \mathbb{F}^k \to \mathbb{F}$ is a* multivalued OR function *if for every $x \in \mathbb{F}^k$ such that $x \neq (0, 0, ..., 0)$ it holds that $f(x) \neq 0$.*

Note that the function that outputs 1 on all non-zero inputs (and vanishes at $(0, ..., 0)$) satisfies Definition 23, but this function has a very high degree as a polynomial (i.e., it has degree $k \cdot |\mathbb{F} - 1|$, which is in fact the maximal degree). In contrast, we are interested in computing multivalued OR functions by polynomials of much lower degree. We now show that for any k, there exists a polynomial $\mathbb{F}^k \to \mathbb{F}$ of degree at most $2 \cdot k$ that computes a multivalued OR function of its k variables.

Proposition 24 *(construction of a multivalued OR function). Let $\mathbb{F}$ be a finite field, and let k be an integer. Then, there exists a polynomial $p : \mathbb{F}^k \to \mathbb{F}$ of degree $2^{\lceil \log(k) \rceil}$ that computes a multivalued OR function of its k variables.*

Proof. Let us first assume that k is a power of two. We want to construct a k-variate polynomial of degree k that vanishes only at $(0, ..., 0)$. We will first construct a bivariate polynomial that vanishes only at $(0, 0)$, and then recurse the construction, to repeatedly double the number of variables as well as the degree, while maintaining the invariant that the polynomial vanishes if and only if all of its inputs are zero.

Let $\alpha \in \mathbb{F}$ be a quadratic non-residue (i.e., for every $c \in \mathbb{F}$ it holds that $c^2 \neq \alpha$). The initial bivariate polynomial is defined by $f^{(2)}(x_1, x_2) = x_1^2 + \alpha \cdot x_2^2$. Observe that there does not exist a solution other than $(0, 0)$ to the equation $f^{(2)}(x_1, x_2) = 0$, since α is not a quadratic residue. Now, for every $k \geq 4$ that is a power of two, let $f^{(k)}(x_1, ..., x_k) = \left(f^{(k/2)}(x_1, ..., x_{k/2})\right)^2 + \alpha \cdot \left(f^{(k/2)}(x_{k/2+1}, ..., x_k)\right)^2$. Observe that $f^{(k)}(x_1, ..., x_k) = 0$ if and only if $x_i = 0$ for every $i \in [k]$, whereas $\deg(f^{(k)}) = k$. Finally, for any k that is not a power of two, we can use a straightforward padding argument to obtain a polynomial of degree $2^{\lceil \log(k) \rceil}$. ■

We are now ready to prove the main claim that will be used in the proof of Theorem 6. The following proposition reduces the task of hitting any polynomial $p : \mathbb{F}^n \to \mathbb{F}$ of degree d to the task of hitting a polynomial $p' : \mathbb{F}^{t \cdot n} \to \mathbb{F}$ of degree $d' = \text{poly}(d)$ that vanishes very rarely.

Proposition 25 *(reducing hitting polynomials to hitting polynomials that vanish rarely). Let $t \geq 2$ be an even integer, and let $\epsilon > 0$ be a real number. Let $n \in \mathbb{N}$, let $\mathbb{F}$ be a finite field of cardinality $|\mathbb{F}| = q$, and let $1 \leq d \leq (1-\epsilon) \cdot q$. Assume that there exists a hitting-set generator with seed length s for the class of polynomials $\mathbb{F}^{t \cdot n} \to \mathbb{F}$ of degree $d' = (2 \cdot d)^t$ that vanish on at most a $b(n) = O\left(q^{-t^2/4}\right)$ fraction of their inputs, where the O-notation hides a constant that depends on t and on ϵ. Then, there exists a hitting-set generator with seed length $s' = s + (t-1) \cdot \lceil \log(q) \rceil$ for the class of all polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree d.*

A high-level overview of the proof of Proposition 25 appeared in Section 2.3. We stress that the field size $|\mathbb{F}| = q$ is the same both for the polynomials $\mathbb{F}^n \to \mathbb{F}$ and for the polynomials $\mathbb{F}^{t \cdot n} \to \mathbb{F}$.

Proof. For any tuple of t elements $\vec{u} = \left(u^{(0)}, u^{(1)}, ..., u^{(t-1)}\right) \in \mathbb{F}^{t \cdot n}$, denote by $W_{\vec{u}} \subseteq \mathbb{F}^n$ the affine subspace $W_{\vec{u}} = \{u^{(0)} + \alpha_1 \cdot u^{(1)} + ... + \alpha_{t-1} \cdot u^{(t-1)} : \alpha_1, ..., \alpha_{t-1} \in \mathbb{F}\}$. Also, denote by $\mathcal{P}_{d'}$ the class of polynomials $\mathbb{F}^{t \cdot n} \to \mathbb{F}$ of degree d' that vanish on at most $b(n)$ of their inputs.

Our proof strategy is as follows. For any polynomial $p : \mathbb{F}^n \to \mathbb{F}$ of degree d, we will construct a corresponding polynomial $p' : \mathbb{F}^{t \cdot n} \to \mathbb{F}$ of degree at most $d' = (2 \cdot d)^t$ such that $p'(\vec{u}) = 0$ if and only if $p\restriction_{W_{\vec{u}}} \equiv 0$. We will show that with high probability over choice of $\vec{u}$ it holds that $p\restriction_{W_{\vec{u}}} \not\equiv 0$, which implies that the polynomial p' vanishes rarely; that is, we will show that $p' \in \mathcal{P}_{d'}$. Thus, for every $p : \mathbb{F}^n \to \mathbb{F}$ of degree d, a hitting-set generator G for $\mathcal{P}_{d'}$ also hits p', which means that the generator finds a subspace $W_{\vec{u}}$ such that $p\restriction_{W_{\vec{u}}} \not\equiv 0$. This allows us to find a satisfying input for p by invoking G and then choosing a random input in $W_{\vec{u}}$. Details follow.

Let us first fix an arbitrary $p : \mathbb{F}^n \to \mathbb{F}$, and construct the corresponding polynomial $p' : \mathbb{F}^{t \cdot n} \to \mathbb{F}$. For an input $\vec{u} \in \mathbb{F}^{t \cdot n}$ and $i \in [t]$, denote $u^{(i)} = (u_1^{(i)}, ..., u_n^{(i)}) \in \mathbb{F}^n$, and observe that the polynomial $p\restriction_{W_{\vec{u}}}(\alpha_1, ..., \alpha_{t-1})$ is of the form

$$\begin{aligned} p\restriction_{W_{\vec{u}}}(\alpha_1, ..., \alpha_{t-1}) &= p\left(u^{(0)} + \alpha_1 \cdot u^{(1)} + ... + \alpha_{t-1} \cdot u^{(t-1)}\right) \\ &= p\left(u_1^{(0)} + \alpha_1 \cdot u_1^{(1)} + ... + \alpha_{t-1} \cdot u_1^{(t-1)}, ..., u_n^{(0)} + \alpha_1 \cdot u_n^{(1)} + ... + \alpha_{t-1} \cdot u_n^{(t-1)}\right) \\ &= \sum_{i_1 + i_2 + ... + i_{t-1} \leq d} c_{i_1, ..., i_{t-1}}(\vec{u}) \cdot \alpha_1^{i_1} \cdot ... \cdot \alpha_{t-1}^{i_{t-1}} , \end{aligned} \tag{6.2}$$

where for every $i_1 + i_2 + ... + i_{t-1} \leq d$ it holds that $c_{i_1,...,i_{t-1}}(\vec{u})$ is the coefficient of the monomial $\alpha_1^{i_1} \cdot ... \cdot \alpha_{t-1}^{i_{t-1}}$ in $p\restriction_{W_{\vec{u}}}$.

Note that $p\restriction_{W_{\vec{u}}} \equiv 0$ if and only if for every tuple $(i_1, ..., i_{t-1})$ such that $i_1 + ... + i_{t-1} \leq d$ it holds that $c_{i_1,...,i_{t-1}}(\vec{u}) = 0$. Thus, we wish to construct a polynomial p' such that $p'(\vec{u}) \neq 0$ if and only if there exists $(i_1, ..., i_{t-1})$ such that $i_1 + ... + i_{t-1} \leq d$ and $c_{i_1,...,i_{t-1}}(\vec{u}) \neq 0$. Note that the number of coefficients of $p\restriction_{W_{\vec{u}}}$ is $k = \binom{d+t-1}{t-1}$. The polynomial $p' : \mathbb{F}^{t \cdot n} \to \mathbb{F}$ is a multivalued OR function of these k coefficients $c_{i_1,...,i_{t-1}}(\vec{u})$, which we construct using Proposition 24. To upper-bound the degree of p' (by d'), note that each $c_{i_1,...,i_{t-1}}$ is a polynomial of degree at most d in $\vec{u}$.

Claim 25.1. *For every $(i_1, ..., i_{t-1})$ such that $i_1 + ... + i_{t-1} \leq d$ it holds that $c_{i_1,...,i_{t-1}}$, as defined in Eq. (6.2), is a polynomial of degree at most d in $\vec{u} = (u^{(0)}, ..., u^{(t-1)}) \in \mathbb{F}^{t \cdot n}$.*

Proof. Consider the polynomial $p\restriction_{W_{\vec{u}}}[\alpha_1, ..., \alpha_{t-1}]$ as a function of $\vec{u}$. By the definition of $p\restriction_{W_{\vec{u}}}$, it holds that $p\restriction_{W_{\vec{u}}}[\alpha_1, ..., \alpha_{t-1}] = p[\beta_1, ..., \beta_n]$, where for every $i \in [n]$ it holds that $\beta_i = u_i^{(0)} + \alpha_i \cdot u_i^{(1)} + ... + \alpha_{t-1} \cdot u_i^{(t-1)}$. Note that for every $i \in [n]$ it holds that β_i is a linear function of $\vec{u}$. Since p is of total degree d, the polynomial $p[\beta_1, ...\beta_n]$ is a sum of monomials of degree at most d in $\beta_1, ..., \beta_n$, and because each β_i is linear in $\vec{u}$, each such monomial is a polynomial of degree at most d in $\vec{u}$. □

Therefore, the degree of p' is less than $2 \cdot \binom{d+t-1}{t-1} \cdot d < (2 \cdot d)^t = d'$. Finally, let us upper-bound the probability that p' vanishes, in order to show that $p' \in \mathcal{P}_{d'}$. To do so, note that $\Pr_{x \in \mathbb{F}^n}[p(x) = 0] \leq d/q \leq 1 - \epsilon$ (where the first inequality is by the Schwartz-Zippel lemma, and the second inequality is by the hypothesis that $d \leq (1 - \epsilon) \cdot q$). Also recall that when uniformly choosing $\vec{u} \in \mathbb{F}^{t \cdot n}$, the points in $W_{\vec{u}}$ are t-wise independent. Relying on Fact 9, we deduce that:

Claim 25.2. *The probability over choice of $\vec{u}$ that $p\restriction_{W_{\vec{u}}} \equiv 0$ is at most $O\left(d^{t/2} \cdot q^{-t^2/2}\right)$, where the O-notation hides a constant that depends on t and on ϵ.*

The proof of Claim 25.2 amounts to a straightforward calculation, so we defer it to Appendix C. Relying on Claim 25.2 and on the hypothesis that $d \leq (1 - \epsilon) \cdot q$, we deduce that $\Pr_{\vec{u}}[p'(\vec{u}) = 0] = \Pr_{\vec{u}}\left[p\restriction_{W_{\vec{u}}} \equiv 0\right] < O\left(q^{-t^2/2 + t/2}\right) \leq O\left(q^{-t^2/4}\right) = b(n)$.

Now, assuming that we have a hitting-set generator G with density ρ for $\mathcal{P}_{d'}$, we construct a hitting-set generator for degree-d polynomials as follows. We invoke G to obtain a tuple $\vec{u} \in \mathbb{F}^{t \cdot n}$, and then use additional $(t-1) \cdot \lceil \log(q) \rceil$ bits of randomness to choose an element in the affine subspace $W_{\vec{u}}$. Since G finds $\vec{u}$ such that $p\restriction_{W_{\vec{u}}} \not\equiv 0$, with positive probability, our hitting-set generator hits p, with positive probability. ■

Proposition 25 reduces the task of hitting a polynomial $\mathbb{F}^n \to \mathbb{F}$ of degree d to the task of hitting of a polynomial $p' : \mathbb{F}^{t \cdot n} \to \mathbb{F}$ of higher degree $d' = \text{poly}(d)$ that vanishes very rarely. The following proposition shows how to reduce the task of hitting p to the task of hitting polynomials of the *same degree* as p that vanish with probability at most $O(1/|\mathbb{F}|)$.

Proposition 26 *(reducing hitting polynomials to hitting polynomials of the same degree that vanish infrequently). Let $n \in \mathbb{N}$, and let $\mathbb{F}$ be a finite field of cardinality $|\mathbb{F}| = q$. For any $c' > 0$ and $d \geq 1$, let $\mathcal{P}_{d,c'}$ be the class of polynomials $\mathbb{F}^{2 \cdot n} \to \mathbb{F}$ of degree d that vanish on at most a $b(n) = c'/q$ fraction of their inputs. Then, for any integer d such that $d + 2\sqrt{d} \leq q$ and any $2 < c' \leq d$, the following holds:*

If there exists a hitting-set generator for the class $\mathcal{P}_{d,c'}$ with seed length $s = s(n, q, d, c')$ and density more than $1/q + 2/c'$, then there exists a hitting-set generator for polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree d with seed length $s' = s + \lceil \log(q) \rceil$.

Proof. The starting point of the current proof is the proof of Proposition 25, with the fixed parameter $t = 2$. [12] We first show how to randomly compute the polynomial $p' : \mathbb{F}^{2\cdot n} \to \mathbb{F}$ by polynomials of degree d that typically vanish with probability c'/q, and then rely on Lemma 21, to show that any sufficiently dense hitting-set generators for degree-d polynomials that vanish with probability c'/q also hits p', which allows us to hit p with additional $\lceil \log(q) \rceil$ random bits.

Recall that $p'(\vec{u})$ computes a multivalued OR of the $d+1$ coefficients of $p\restriction_{W_{\vec{u}}}$, which are degree-$d$ polynomials in $\vec{u}$, denoted by $c_1, ..., c_{d+1}$. We randomly compute p' by taking a random $\mathbb{F}$-linear combination of the c_i's. That is, for a random tuple $\vec{\beta} = (\beta_0, \beta_1, ..., \beta_d) \in \mathbb{F}^{(d+1)\cdot n}$, we define $h_{\vec{\beta}}(\vec{u}) = \sum_{i=0}^{d} \beta_i \cdot c_i(\vec{u})$. Note that for every $\vec{\beta} \in \mathbb{F}^{(d+1)\cdot n}$ it holds that $h_{\vec{\beta}}$ is of degree d. Also, if $p'(\vec{u}) = 0$ (i.e., all the $c_i(\vec{u})$'s equal zero), then $h_{\vec{\beta}}(\vec{u}) = 0$ with probability one, and otherwise, $h_{\vec{\beta}}(\vec{u}) \neq 0$ with probability $1 - 1/q$. Therefore, this distribution computes p' with error at most $1/q$.

We now show that at least a $(1 - 2/c')$ fraction of the $h_{\vec{\beta}}$'s vanish on at most c'/q of their inputs. Since the points in W are pairwise-independent, we have that:

Claim 26.1. *For any $\epsilon > 0$, if $d \leq (1-\epsilon) \cdot q$, then the probability over choice of $\vec{u}$ that $p\restriction_{W_{\vec{u}}} \equiv 0$ is at most $4 \cdot \left(\frac{d}{\epsilon^2 \cdot q^2}\right)$.*

The proof of Claim 26.1 appears in Appendix C. In our case, we have that $d \leq (1-\epsilon) \cdot q$, where $\epsilon = \frac{2\sqrt{d}}{q}$ (because $d + 2\sqrt{d} \leq q$); therefore, Claim 26.1 implies that $\Pr_{\vec{u}}[p'(\vec{u}) = 0] \leq 1/q$. Hence, over a random choice of $\vec{\beta}$, the expected fraction of inputs on which $h_{\vec{\beta}}$ vanishes is

$$\begin{aligned} \mathbb{E}_{\vec{\beta}}\left[\Pr_{\vec{u}}\left[h_{\vec{\beta}}(\vec{u}) = 0\right]\right] &= \mathbb{E}_{\vec{u}}\left[\Pr_{\vec{\beta}}\left[h_{\vec{\beta}}(\vec{u}) = 0\right]\right] \\ &\leq \Pr_{\vec{u}}[p'(\vec{u}) = 0] + \Pr_{\vec{u}}[p'(\vec{u}) \neq 0] \cdot \max_{\vec{u}}\left\{\Pr_{\vec{\beta}}[h_{\vec{\beta}}(\vec{u}) \neq p'(\vec{u})]\right\}, \end{aligned}$$

which is upper bounded by $2/q$. It follows that the probability that $h_{\vec{\beta}}$ vanishes on more than c'/q fraction of its inputs is at most $2/c'$.

The above shows that p' can be randomly computed with error $1/q$ by a distribution that is $(1-2/c')$-typically in $\mathcal{P}_{d,c'}$. Now, assume that there exists a hitting-set generator G for $\mathcal{P}_{d,c'}$ with density $1 - c > 1/q + 2/c'$; then, Item (2) of Lemma 21 implies that

$$\Pr[h(G(\mathbf{u}_\ell)) = 1] > 1 - 1/q - 2/c' - c > 0\,.$$

[12] Larger values of t will not help to reduce the vanishing probability of the polynomials in the target of the reduction, due to the error of $1/q$ in the randomized computation of p'. However, larger values of t can help us relax the requirement that $d + 2\sqrt{d} \leq q$, and allow for slightly larger values of d (that are still below q). We do not pursue this direction in the current text.

Finally, similarly to the proof of Proposition 25, we can invoke G to obtain $\vec{u} \in \mathbb{F}^{2 \cdot n}$, and then use another $\log(q)$ bits to uniformly choose an element in the affine line $W_{\vec{u}}$, thus hitting p with positive probability. ■

Let us now formally state Theorem 6, and prove it as a corollary of Propositions 25 and 26.

Theorem 27 *(Theorem 6, restated). Let $k \in \mathbb{N}$, let $t \geq 2$ be an even integer, and let $\epsilon > 0$ be a real number. Let $n \in \mathbb{N}$ be sufficiently large, and let $\mathbb{F}$ be a field of size $|\mathbb{F}| = q \leq n^k$. Then, the following holds:*

1. *Let d be an integer such that $d \geq k+1$ and $d + 2 \cdot \sqrt{d} \leq q$, and let $c' \in (2, d]$. Then, any hitting-set generator with density more than $1/q + 2/c'$ for the class of polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree d that vanish on at most a $b(n) = c'/q$ fraction of their inputs requires seed of $\Omega\left(\log\left(\binom{n+d}{d}\right)\right)$ bits.*
2. *Let d' be an integer such that $(2k)^{t(t+1)} \leq d' \leq (1-\epsilon) \cdot q^{t+1}$. Then, any hitting-set generator for the class of polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree d' that vanish on at most a $b(n) = O\left(q^{-t^2/4}\right)$ fraction of their inputs requires seed of $\Omega\left(\log\left(\binom{n+d}{d}\right)\right)$ bits, where $d = (d')^{1/(t+1)}$.*

In the two items above, the constants hidden in the Ω-notation of the lower bound may depend on k, on ϵ, and (in the first item) on t.

Proof. Recall that any hitting-set generator for the class of all polynomials $\mathbb{F}^n \to \mathbb{F}$ of degree d (i.e., without any assumption about their vanishing probability) must use a seed of at least $s' \geq \log\left(\binom{n+d}{d}\right)$ bits. This is the case because otherwise we can interpolate the $2^{s'} < \binom{n+d}{d}$ points in the image of the hitting-set generator by a non-zero degree-d polynomial. Also note that it suffices to prove the lower bounds for n that is a multiple of $t = O(1)$, due to a padding argument (i.e., because any hitting-set generator for polynomials $\mathbb{F}^n \to \mathbb{F}$ that vanish on at most $O\left(q^{-t^2/4}\right)$ of their inputs can be used as a hitting-set generator for polynomials $\mathbb{F}^{n-O(1)} \to \mathbb{F}$ that vanish on the same fraction of inputs, by adding dummy variables; and ditto for $O(1/q)$).

To prove Item (1), assume that there exists a hitting-set generator with seed length s and density more than $1/q + 2/c'$ for polynomials of degree d that vanish on c'/q of their inputs. Relying on Proposition 26, there exists a hitting-set generator for all polynomials $\mathbb{F}^{n/2} \to \mathbb{F}$ of degree d with seed length $s' = s + \lceil \log q \rceil$. Since $s' \geq \log\left(\binom{n/2+d}{d}\right)$, we deduce that $s \geq \log\left(\binom{n/2+d}{d}\right) - \lceil \log(q) \rceil = \Omega\left(\log\left(\binom{n/2+d}{d}\right)\right)$, where the equality holds because $q \leq n^k$ and $d \geq k+1$. Finally, we rely on the following elementary fact:

Fact 27.1. *Let t be a constant integer. Let n and d be two integers such that the sum $n+d$ is sufficiently large. Then, we have that $\log\left(\binom{n/t+d}{d}\right) = \Omega\left(\log\left(\binom{n+d}{d}\right)\right)$, where the constant hidden inside the Ω-notation depends on t.*

The proof of Fact 27.1 appears in Appendix C. It follows from Fact 27.1 that $s \geq \Omega\left(\log\left(\binom{n+d}{d}\right)\right)$, which concludes the proof of Item (1).

The proof of Item (2) is similar to that of Item (1). Assume that there exists a hitting-set generator with seed length s for the class of degree-d' polynomials $\mathbb{F}^n \to \mathbb{F}$ that vanish on at most a $O\left(q^{-t^2/4}\right)$ fraction of their inputs. Let $d = \lfloor (d')^{1/t}/2 \rfloor$ (such that $d' \geq (2 \cdot d)^t$). According to Proposition 25, there exists a hitting-set generator for all polynomials $\mathbb{F}^{n/t} \to \mathbb{F}$ of degree d with seed length $s' = s + (t-1) \cdot \lceil \log(q) \rceil$. Since we know that $s' \geq \log\left(\binom{n/t+d}{d}\right)$, it holds that s is lower bounded by

$$\begin{aligned}\log\left(\binom{n/t+d}{d}\right) - (t-1) \cdot \lceil \log(q) \rceil &= \Omega\left(\log\left(\binom{n/t+d}{d}\right)\right) \\ &= \Omega\left(\log\left(\binom{n+d}{d}\right)\right) \\ &= \Omega\left(\log\left(\binom{n+(d')^{1/(t+1)}}{(d')^{1/(t+1)}}\right)\right),\end{aligned}$$

where the first equality is because $q \leq n^k$ and $d \geq \frac{(2k)^{t+1}}{2} \geq (t+1) \cdot k$, the second equality is due to Fact 27.1, and the last equality is because $d \geq (d')^{1/(t+1)}$. ■

Acknowledgements

The author thanks his advisor, Oded Goldreich, for many helpful discussions, and for his guidance and support during the research and writing process. The author thanks Inbal Livni for very useful discussions about polynomials that vanish rarely, and Avishay Tal for very useful discussions about constant-depth circuits. Part of this research was conducted during the workshop on small-depth circuits in St. Petersburg (May 2016), and the author is grateful to the organizers of the workshop. This research was partially supported by the Minerva Foundation with funds from the Federal German Ministry for Education and Research. The research was also partially supported by Irit Dinur's ERC grant number 239986.

References

[Baz09] Louay M. J. Bazzi. Polylogarithmic independence can fool DNF formulas. *SIAM Journal of Computing*, 38(6):2220–2272, 2009.

[Bog05] Andrej Bogdanov. Pseudorandom generators for low degree polynomials. In *Proc. 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 21–30. 2005.

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *Proc. 35th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 276–287, 1994.

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

[CR96] Shiva Chaudhuri and Jaikumar Radhakrishnan. Deterministic restrictions in circuit complexity. In *Proc. 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 30–36, 1996.

[CTS13] Gil Cohen and Amnon Ta-Shma. Pseudorandom generators for low degree polynomials from algebraic geometry codes. *Electronic Colloquium on Computational Complexity: ECCC*, 20:155, 2013.

[DETT10] Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. In *Proc. 14th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 504–517, 2010.

[GVW15] Oded Goldreich, Emanuele Viola, and Avi Wigderson. On randomness extraction in AC0. In *Proc. 30th Annual IEEE Conference on Computational Complexity (CCC)*, pages 601–668, 2015.

[GW13] Oded Goldreich and Avi Widgerson. On derandomizing algorithms that err extremely rarely. *Electronic Colloquium on Computational Complexity: ECCC*, 20:152, 2013.

[GW14] Oded Goldreich and Avi Widgerson. On derandomizing algorithms that err extremely rarely. In *Proc. 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 109–118. 2014.

[Hås87] Johan Håstad. *Computational Limitations of Small-depth Circuits*. MIT Press, 1987.

[IW99] Russell Impagliazzo and Avi Wigderson. P $=$ BPP if E requires exponential circuits: derandomizing the XOR lemma. In *Proc. 29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 220–229. 1999.

[KS12] Swastik Kopparty and Srikanth Srinivasan. Certifying polynomials for $\mathrm{AC}^0[\oplus]$ circuits, with applications. In *Proc. 32nd Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 36–47. 2012.

[LRTV09] Shachar Lovett, Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom bit generators that fool modular sums. In *Proc. 13th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 615–630. 2009.

[Nis91] Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[Raz87] Alexander A. Razborov. Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematical Notes of the Academy of Science of the USSR*, 41(4):333–338, 1987.

[Sch76] Wolfgang M. Schmidt. *Equations over Finite Fields: An Elementary Approach*. Springer-Verlag Berlin, 1976.

[Tal14] Avishay Tal. Tight bounds on the fourier spectrum of $\mathcal{AC}^0$. *Electronic Colloquium on Computational Complexity: ECCC*, 21:174, 2014.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001.

[TX13] Luca Trevisan and TongKe Xue. A derandomized switching lemma and an improved derandomization of AC0. In *Proc. 28th Annual IEEE Conference on Computational Complexity (CCC)*, pages 242–247. 2013.

[Vad12] Salil P. Vadhan. *Pseudorandomness*. Foundations and Trends in Theoretical Computer Science. Now Publishers, 2012.

[Vio09] Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. *Computational Complexity*, 18(2):209–217, 2009.

Appendix A Alternative proof for Theorem 1.6 in [GW14]

Goldreich and Wigderson [GW14, Thm 1.6] proved that for any $d < n$, there exists a pseudorandom generator with seed length $O(\log(n))$ for the class of polynomials $p : \mathbb{F}_2^n \to \mathbb{F}_2$ of degree d that vanish at most a $b(n) = O\left(2^{-d}\right)$ fraction of their inputs (the theorem statement in [GW14] asserts the existence of a hitting-set generator, but in their proof they actually construct a pseudorandom generator). Their proof is based on a refinement of a lemma of Viola [Vio09, Lemma 4]. We present an alternative proof of their result, which relies on Lemma 21.

High-level outline. Let $p : \mathbb{F}_2^n \to \mathbb{F}_2$ be a polynomial of degree d that vanishes on at most $b(n) = O\left(2^{-d}\right)$ of its inputs. We will randomly compute p by a distribution over polynomials of constant degree, and rely on Lemma 21 to deduce that any pseudorandom generator for polynomials of constant degree also "fools" p.

The family of polynomials of constant degree that we will use to randomly compute p is defined as follows. For $d' = d - O(1)$ and a tuple $\vec{r} = (r_1, ..., r_{d'}) \in \mathbb{F}_2^{d' \cdot n}$, let $h_{\vec{r}} : \mathbb{F}_2^n \to \mathbb{F}_2$ be defined by

$$h_{\vec{r}}(x) = 1 + \Delta_{\vec{r}} p(x) = 1 + \sum_{S \subseteq [d']} p\left(x + \sum_{i \in S} r_i\right) , \tag{A.1}$$

where $\Delta_{\vec{r}} p(x)$ is the iterated directional derivative of p in directions $r_1, ..., r_{d'}$ (for a definition see, e.g., [O'D14, Def. 6.48]). Note that $h_{\vec{r}}$ is a polynomial of degree at most $d - d' = O(1)$. The family $\mathcal{H}$ of polynomials that we will use to randomly compute p is induced by all possible choices of $\vec{r} \in \mathbb{F}_2^{d' \cdot n}$; that is, $\mathcal{H} = \left\{ h_{\vec{r}} : \vec{r} \in \mathbb{F}_2^{d' \cdot n} \right\}$.

The key argument is that for every fixed input $x \in \mathbb{F}_2^n$, when uniformly choosing $h_{\vec{r}} \in \mathcal{H}$, with sufficiently good probability it holds that $p(x) = h_{\vec{r}}(x)$. To see this, note that if for every non-empty $S \subseteq [d']$ it holds that $p\left(x + \sum_{i \in S} r_i\right) = 1$, then $\Delta_{\vec{r}} p(x) = p(x) + (2^{d'} - 1) = p(x) + 1$, which implies that $h_{\vec{r}}(x) = p(x)$. Since p vanishes on at most $b(n)$ of its inputs, the latter event happens with probability at least $1 - 2^{d'} \cdot b(n) = \Omega(1)$. Thus, relying on Lemma 21, any pseudorandom generator for $\mathcal{H}$ also "fools" p. Let us now formalize and parametrize this argument.

Theorem 28 *($\mathbb{F}_2$-polynomials with $b(n) = O(2^{-d})$). Let $c > 0$ be an arbitrarily large constant. Let $n \in \mathbb{N}$, let $d < n$, and let $p : \mathbb{F}_2^n \to \mathbb{F}_2$ be a polynomial of degree d that vanishes on at most $b(n) = c \cdot \left(2^{-d}\right)$ of its inputs. Then, for every $\epsilon > 0$, any pseudorandom generator with error $\epsilon/4$ for polynomials of degree $\lceil \log(c/\epsilon) \rceil$ is also a pseudorandom generator with error ϵ for p, where pseudorandom generators for $\mathbb{F}_2$-polynomials are defined in Definition 7.*

Proof. Let $d' = d - \lfloor \log(c/\epsilon) \rfloor$, let $\mathcal{H} = \left\{ h_{\vec{r}} : \vec{r} \in \mathbb{F}_2^{d' \cdot n} \right\}$ such that for every $\vec{r} \in \mathbb{F}_2^{d' \cdot n}$ the function $h_{\vec{r}}$ is defined as in Eq. (A.1), and let $\mathbf{h}$ be the uniform distribution over $\mathcal{H}$. Note that for every fixed $x \in \mathbb{F}_2^n$ it holds that $\Pr[\mathbf{h}(x) = p(x)] > 1 - \epsilon$; this is the case because for every non-empty $S \subseteq [d']$, the probability that $p(x + \sum_{i \in S} r_i) = 0$ is at most $b(n)$, which implies that with probability at least $1 - b(n) \cdot (2^{d'} - 1) > 1 - \epsilon$ we have that $\mathbf{h}(x) = 1 + p(x) + \left(2^{d'} - 1\right) = p(x)$.

Now, let $\xi : \mathbb{F}_2 \to \mathbb{C}$ be the character $\xi(x) = (-1)^x$. Note that $\delta = \max_{x \in \mathbb{F}_2}\{|\xi(x)|\} = 1$, and that for any function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and any distribution $\mathbf{w}$ over $\mathbb{F}_2^n$ it holds that $\left| \mathbb{E}[\xi(f(\mathbf{w}))] - \mathbb{E}[\xi(f(\mathbf{u}_n))] \right| = 2 \cdot \left| \Pr[f(\mathbf{w}) = 1] - \Pr[f(\mathbf{u}_n) = 1] \right|$. Let G be a pseudorandom generator for polynomials of degree $\lceil \log(c/\epsilon) \rceil$ with seed length $\ell : \mathbb{N} \to \mathbb{N}$ and error $\epsilon/4$. According to Item (1) of Lemma 21, it holds that $\left| \mathbb{E}[\xi(p(G(\mathbf{u}_{\ell(n)})))] - \mathbb{E}[\xi(p(\mathbf{u}_n))] \right| \leq 2 \cdot (\epsilon/2) + \epsilon = 2 \cdot \epsilon$, which implies that G is a pseudorandom generator with error ϵ for p. ■

Appendix B Proof of a claim from Section 4

In this appendix we prove a claim that was made in the proof of Lemma 11. Specifically, referring to the proof of Lemma 11, we show that with high probability, after applying each restriction to the circuit, the number of variables that remain alive is $\Omega(p_i \cdot n_i)$, where p_i is the relevant parameter (i.e., $p_1 = 1/O(1)$ and $p_i = 1/O(\log(n/\epsilon))$ for $i \in \{2, ..., d-2\}$) and n_i is the number of living variables after the $(i-1)^{th}$ iteration.

Claim 29. *Let $\rho_1, ..., \rho_{d-1}$ be the restrictions applied in Lemma 11. Then, for every $i \in [d-1]$, with probability at least $1 - \frac{i \cdot \epsilon}{2(d-1)}$, the number of variables left alive after the restriction ρ_i is is at least $\Omega(n/\log^{i-1}(n/\epsilon))$.*

Proof. Recall that each restriction is chosen according to a δ_1-biased distribution over $\{0,1\}^{(q+1)\cdot n}$, where $\delta_1 = 2^{\tilde{O}(\log^3(n/\epsilon))}$ and $q \leq \log\log(n) + O(1)$. Therefore, for every restriction, the indicator variables that indicate whether or not each variable is kept alive by the restriction are δ-almost t-wise independent, for $\delta = 2^{\tilde{O}(\log^3(n/\epsilon))}$ and $t = O(\log(1/\epsilon))$ (because each such indicator variable corresponds to a block of $q \leq \log\log(n) + O(1)$ bits in the distribution over $\{0,1\}^{(q+1)\cdot n}$) .

We prove Claim 29 by induction on i. For $i = 1$, we use Fact 10 with parameters $N = n$ and $\mu = 1/O(1)$ and $\zeta = 1/(2 \cdot \mu)$, to deduce that the probability that ρ_1 keeps $\Omega(n)$ variables alive is at least $1 - \epsilon/2(d-1)$. Note that we used the hypothesis that $t = O(\log(1/\epsilon)) = o(n^{1/d})$ to upper-bound the first term in the bound of Fact 10; that is, to deduce that $8 \cdot \left(\frac{t \cdot \mu \cdot N + t^2}{\zeta^2 \cdot N^2}\right)^{t/2} \leq 8 \cdot \left(\frac{O(\log(1/\epsilon) \cdot n + \log^2(1/\epsilon))}{n^2}\right)^{O(\log(1/\epsilon))} \leq \frac{\epsilon}{2(d-1)}$.

For $i \geq 2$, let us condition on the event that after applying the $(i-1)^{th}$ restriction, the number of live variables is at least $n_i = \Omega(n/\log^{i-2}(n/\epsilon))$; by the induction hypothesis, this event happens with probability at least $1 - \frac{(i-1)\cdot\epsilon}{2(d-1)}$. We use Fact 10 with parameters $N = n_i$ and $\mu = p$ and $\zeta = p/2$, to deduce that the probability that ρ_i keeps $\Omega(p \cdot n_i)$ variables alive is at least $1 - \epsilon/2(d-1)$. Similarly to the case of $i = 1$, for the latter statement we used the fact that $t = O(\log(1/\epsilon)) = o(n_i)$ (which holds because $n_i = n/\log^{d-2}(n/\epsilon)$ and $\log(1/\epsilon) \cdot \log^{d-2}(n/\epsilon) \leq \log^{d-1}(n/\epsilon) = o(n)$, where the last equality is due to our hypothesis that $\epsilon(n) \geq 2^{-o(n^{1/d})}$) to upper-bound the first term in the bound of Fact 10. □

Appendix C Proofs of technical claims from Section 6

In this appendix we prove several technical claims that were made in the proofs of Proposition 25, Proposition 26, and Theorem 27.

Let us first prove a claim that generalizes Claims 25.2 and 26.1, which were made in the proofs of Proposition 25 and Proposition 26, respectively. Recall that for any tuple of t elements $\vec{u} = (u^{(0)}, ..., u^{(t-1)}) \in \mathbb{F}^{t \cdot n}$, we denote by $W_{\vec{u}} \subseteq \mathbb{F}^n$ the affine subspace $W_{\vec{u}} = \{u^{(0)} + \alpha_1 \cdot u^{(1)} + ... + \alpha_{t-1} \cdot u^{(t-1)} : \alpha_1, ..., \alpha_{t-1} \in \mathbb{F}\}$. Then, the following holds:

Claim 30 *(Claims 25.2 and 26.1, generalized). Let $t \geq 2$ be an even integer, and let $\epsilon \in (0,1)$. Let $n \in \mathbb{N}$, let $\mathbb{F}$ be a field of size $|\mathbb{F}| = q$, and let $p : \mathbb{F}^n \to \mathbb{F}$ be a polynomial of degree $d \leq (1-\epsilon) \cdot q$. Uniformly choose $\vec{u} = (u^{(0)}, ..., u^{(t-1)}) \in \mathbb{F}^{t \cdot n}$, and let $W = W_{\vec{u}}$. Then, the probability that $p\restriction_W \equiv 0$ is at most $O\left(d^{t/2} \cdot q^{-t^2/2} \cdot \epsilon^{-t}\right)$, where the O-notation hides a constant that depends on t; in particular, when $t = 2$, the hidden constant is just 4.*

Proof. For $i = 1, ..., q^{t-1}$, let $\mu_W^{(i)}$ be the indicator variable of whether p vanishes on the i^{th} point in W (according to some canonical ordering of points in $\mathbb{F}^n$), and let $\mu_W = \mathbb{E}_{i \in [q^{t-1}]}\left[\mu_W^{(i)}\right] = \Pr_{\vec{x} \in W}[p(\vec{x}) = 0]$. Denote by $b = \Pr_{x \in \mathbb{F}^n}[p(x) = 0]$, and note that $b \leq d/q \leq 1 - \epsilon$, where the first inequality is by the Schwartz-Zippel lemma, and the second inequality is by the hypothesis that $d \leq (1-\epsilon) \cdot q$.

We handle the case of $t = 2$ and the case of $t \geq 4$ separately. Starting with the former, note that for every $i \neq j \in [q]$ it holds that $\mu_W^{(i)}$ and $\mu_W^{(j)}$ are independent, and that $Var\left(\mu_W^{(i)}\right) \leq b$. Relying on Chebyshev's inequality, we have that

$$\Pr_W[|\mu_W - b| > \epsilon/2] \leq \frac{b}{(\epsilon/2)^2 \cdot q} \leq 4 \cdot \left(\frac{d}{\epsilon^2 \cdot q^2}\right) .$$

For the case of $t \geq 4$, we rely on Fact 9. In our case, the t-wise independent variables are $\mu_W^{(1)}, ..., \mu_W^{(q^{t-1})}$, their average is $\frac{1}{q^{t-1}} \cdot \sum_{i \in [q^{t-1}]} \mu_W^{(i)} = \mu_W$, and their expected average is $b \leq 1 - \epsilon$. Using Fact 9 with $\zeta = \epsilon/2$, we have that

$$\begin{aligned} \Pr_W[|\mu_W - b| \geq \epsilon/2] &\leq 8 \cdot \left(\frac{t \cdot b \cdot q^{t-1} + t^2}{(\epsilon/2)^2 \cdot (q^{t-1})^2}\right)^{t/2} \\ &\leq 8 \cdot \left(\frac{2 \cdot t^2 \cdot \max\left\{b, q^{-(t-1)}\right\}}{(\epsilon/2)^2 \cdot q^{t-1}}\right)^{t/2} \\ &\leq \left(8 \cdot 2^{t/2} \cdot (2t)^t\right) \cdot \left(\frac{d/q}{\epsilon^2 \cdot q^{t-1}}\right)^{t/2} , \end{aligned}$$

which is $O\left(d^{t/2} \cdot q^{-t^2/2} \cdot \epsilon^{-t}\right)$. □

We now prove Fact 27.1, which was stated in the proof of of Theorem 27:

Fact 31. *Let t be a constant integer. Let n and d be two integers such that the sum $n + d$ is sufficiently large. Then, we have that $\log\left(\binom{n/t+d}{d}\right) = \Omega\left(\log\left(\binom{n+d}{d}\right)\right)$, where the constant hidden inside the Ω-notation depends on t.*

Proof. Let $c = \frac{1}{t \cdot e}$, where $e = 2.71...$. If $d \leq c \cdot (n/t + d)$, then the assertion follows from the standard bound $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{n \cdot e}{k}\right)^k$.[13] Similarly, if $(n/t) \leq c' \cdot (n/t + d)$, where

[13] Specifically, $\log\left(\binom{n+d}{d}\right) \leq d \cdot \left(\log\left(\frac{n+d}{d}\right) + \log(e)\right) < d \cdot \left(\log\left(\frac{(n/t)+d}{d}\right) + \log(t \cdot e)\right) \leq 2 \cdot d \cdot \log\left(\frac{(n/t)+d}{d}\right) \leq 2 \cdot \log\left(\binom{n/t+d}{d}\right)$, where the penultimate inequality relies on the fact that $\frac{(n/t)+d}{d} \geq t \cdot e$.

$c' = 1/e$, then the assertion follows by showing that $\log\left(\binom{n/t+d}{n/t}\right) = \Omega\left(\log\left(\binom{n+d}{n}\right)\right)$, relying on the same standard bound. [14]

Otherwise, we have that $d > c \cdot (n/t + d)$ and $n/t > c' \cdot (n/t + d)$. In this case we use Stirling's approximation: Let $H_2(\cdot)$ be the binary entropy function, and denote $\alpha = \frac{d}{d+n}$ and $\alpha' = \frac{d}{d+(n/t)}$. Note that $\frac{c}{t} < \alpha < 1 - c'$, and that $c < \alpha' < 1 - c'$, which implies that $H_2(\alpha) = \Omega(1)$ and $H_2(\alpha') = \Omega(1)$. Hence, we deduce that $\log\left(\binom{n+d}{d}\right) \le H_2(\alpha) \cdot (n + d)$, whereas $\log\left(\binom{n/t+d}{d}\right) \ge (H_2(\alpha') - o(1)) \cdot (n/t + d) = \Omega\left(H_2(\alpha) \cdot (n + d)\right)$. □

[14] Specifically, $\log\left(\binom{n+d}{n}\right) \le n \cdot \left(\log\left(\frac{n+d}{n}\right) + \log(e)\right) < n \cdot \left(\log\left(\frac{(n/t)+d}{(n/t)}\right) + \log(e)\right) \le 2 \cdot n \cdot \log\left(\frac{(n/t)+d}{(n/t)}\right) \le (2 \cdot t) \cdot \log\left(\binom{n/t+d}{n/t}\right)$, where the penultimate inequality relies on the fact that $\frac{(n/t)+d}{n/t} \ge e$.