Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 191 (2016)

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

Roei Tell *

May 14, 2017

Abstract

This work studies the question of *quantified derandomization*, which was introduced by Goldreich and Wigderson (STOC 2014). The generic quantified derandomization problem is the following: For a circuit class $\mathcal{C}$ and a parameter $B = B(n)$, given a circuit $C \in \mathcal{C}$ with n input bits, decide whether C rejects all of its inputs, or accepts all but $B(n)$ of its inputs. In the current work we consider three settings for this question. In each setting, we bring closer 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.

For **constant-depth circuits**, we construct an algorithm for quantified derandomization that works for a parameter $B(n)$ that is only *slightly smaller* than a "threshold" parameter, and is significantly faster than the best currently-known algorithms for standard derandomization. On the way to this result we establish a new derandomization of the switching lemma, which significantly improves on previous results when the width of the formula is small. For **constant-depth circuits with parity gates**, we lower a "threshold" 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 left as an open problem. We also consider the question of constructing hitting-set generators for multivariate **polynomials over large fields that vanish rarely**, and prove two lower bounds on the seed length of such generators.

Several of our proofs rely on an interesting technique, which we call the *randomized tests* technique. Intuitively, a standard technique to deterministically find a "good" object is to construct a simple deterministic test that decides the set of good objects, and then "fool" that test using a pseudorandom generator. We show that a similar approach works also if the simple deterministic test is replaced with a *distribution over simple tests*, and demonstrate the benefits in using a distribution instead of a single test.

*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 time 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] put forward 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 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 badness parameter $B(n)$ 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 [GW14]) is still at least as difficult as certain standard derandomization problems.

> **The "take-home" message:** Considered together, our results 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 results

Let us informally state the main results in this work, which we later outline in more detail:

- **Constant-depth circuits (see Section 1.2):** For circuits of depth D, the badness parameter $B(n) = \exp\left(n/\log^{D-O(1)}(n)\right)$ is a threshold value, since an algorithm for quantified derandomization with such a $B(n)$ implies an algorithm for *standard* derandomization of circuits of smaller depth $d \leq D - 12$ (see Theorem 1).

We show that taking $B(n)$ to be only *slightly smaller* than the threshold value allows for derandomization that is significantly faster than the best currently-known standard derandomization. Specifically, we construct a hitting-set generator for depth-D circuits with badness $B(n) = \exp\left(n/\log^{D-2}(n)\right)$ that has seed length $\tilde{O}(\log^3(n))$; in particular, the seed length *does not depend on the depth* D (see Theorem 2).

The latter is a special case of a more general result that we prove, which extends the main theorem of Goldreich and Wigderson [GW14]: We establish a trade-off between the badness parameter and the seed length of hitting-set generators for $\mathcal{AC}^0$. This is done by constructing a parametrized hitting-set generator that can work with large badness parameters, at the expense of a super-logarithmic seed (see Theorem 3). The key part in this construction is a *new derandomization of the switching lemma*, which is our main technical contribution in the context of constant-depth circuits. The seed length in the new derandomization is significantly shorter than in previous derandomizations when the width w of the formula is small (i.e., $w = o(\log(n))$).

- **Constant-depth circuits with parity gates (see Section 1.3):** We show that a threshold for derandomization of $\mathcal{AC}^0[\oplus]$ exists at depth four with the parameter 2^{n^c}, for any $c > 0$. Hence, an appealing frontier is $\mathcal{AC}^0[\oplus]$ circuits of depth three with the parameter $B(n) = 2^{n^c}$. Goldreich and Wigderson derandomized various types of such circuits, and left one last type as an open problem. We make significant progress on the last remaining type: 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 (see Theorem 6).

  We also affirm a conjecture from [GW14], by showing a reduction of the problem of hitting such $\oplus \wedge \oplus$-circuits to the problem of hitting biased $\mathbb{F}_2$-polynomials of *bounded (non-constant) degree* (see Theorem 7).

- **Polynomials that vanish rarely (see Section 1.4):** 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. We prove two lower bounds on the seed length of such hitting-set generators. The main result is that any hitting-set generator for degree-d polynomials that vanish on at most $1/\text{poly}(|\mathbb{F}|)$ of their inputs requires a seed of length similar to that of hitting-set generators for *all* degree-d polynomials (see Theorem 8).

  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 is a form of "error reduction" for polynomials that incurs only a mild increase in the degree.

Several of our results are based on a general technique that might be of independent interest, which we call the **randomized tests** technique (see Section 2.1). Intuitively, a standard approach to deterministically find an object in some predetermined set $G \subseteq \{0,1\}^n$ is to construct a simple deterministic test that decides G, and then "fool" the test using a pseudorandom generator. We show that a similar approach works if the simple deterministic test is replaced with a *distribution over simple tests*, and the pseudorandom generator is required to "fool" the residual deterministic tests. In many settings, the fact that we use randomness (i.e., use a distribution over tests) yields residual tests that are simpler than any corresponding deterministic test (see Section 2.2 for a concrete example).

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 $\Pr_{s\in\{0,1\}^\ell}[f(G(s)) \neq 0] \geq \epsilon$ (see Definition 10). The definition of hitting-set generators extends naturally to functions $\mathbb{F}^n \to \mathbb{F}$, for any field $\mathbb{F}$ (see Definition 11).

1.2 Constant-depth circuits

Let us first state the threshold values for quantified derandomization of $\mathcal{AC}^0$, and then turn to describe our algorithms for quantified derandomization. Goldreich and Wigderson showed that the value $B(n) = 2^{n/\log^{0.99\cdot D}(n)}$ is a threshold value for quantified derandomization of depth-D circuits. Specifically, they reduced the *standard* derandomization problem of depth-d circuits to the problem of quantified derandomization of circuits of depth $D \gg d$ with $B(n) = 2^{n/\log^{D-O(d)}(n)}$ (see [GW14, Thm 3.4 (full version)]). Since their work, Cheng and Li [CL16] improved the known techniques for error-reduction within $\mathcal{AC}^0$, which allows us to further decrease the threshold value, as follows:

Theorem 1 *(threshold for quantified derandomization of $\mathcal{AC}^0$). For any $d \geq 2$ and $D > d + 11$, the standard derandomization problem of depth-d circuits reduces in deterministic polynomial-time to the quantified derandomization problem of circuits of depth D that accept all but $B(n) = 2^{n/\log^{D-d-11}(n)}$ of their inputs.*

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

Theorem 2 *(quantified derandomization of $\mathcal{AC}^0$ with badness $2^{n/\log^{D-2}(n)}$). 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, "Barriers to Further Progress"]).

The badness parameters in Theorems 1 and 2 are indeed very close, yet the smaller badness parameter allows for derandomization in time $2^{\tilde{O}(\log^3(n))}$ whereas the larger badness parameter is a threshold for standard derandomization. This represents a progress towards the goal of the quantified derandomization approach, which is to *close* the gap between the two parameters: That is, to either increase the badness parameter in Theorem 2, or decrease the parameter in Theorem 1, and obtain a standard derandomization of $\mathcal{AC}^0$.

Theorem 2 is a special case of the following, more general result, which extends the main theorem of Goldreich and Wigderson [GW14]. Their algorithm works with logarithmic seed and badness parameter $B(n) = 2^{n^{1-\Omega(1)}}$. The following result is parametrized (by the parameter t), and can work with badness parameters that are larger than $2^{n^{1-\Omega(1)}}$, at the expense of a longer (i.e., super-logarithmic) seed; Theorem 2 is the special case where both the badness parameter and the seed are the largest possible in this result.

Theorem 3 *(quantified derandomization of $\mathcal{AC}^0$: a general trade-off). For any $D \geq 2$ and $t : \mathbb{N} \to \mathbb{N}$ such that $t(n) \leq O(\log(n))$, there exists a hitting-set generator that uses a seed of length $\tilde{O}(t^2 \cdot \log(n))$ for the class of depth-D circuits over n input bits that accept all but at most $B(n) = \exp\left(n^{1-1/\Omega(t)}/t^{d-2}\right)$ of their inputs.*

Indeed, the main result in [GW14] is essentially obtained (up to a $\operatorname{poly}\log\log(n)$ factor in the seed length) by setting $t = O(1)$, whereas Theorem 2 is obtained by setting $t = O(\log(n))$. Theorem 3 is based on a *new derandomization of Hastad's switching lemma*, which is our main technical contribution in this section.

Proposition 4 *(new derandomization of the switching lemma; informal). Let $n \in \mathbb{N}$ and $w \leq O(\log(n))$. Then, there exists an algorithm that on an input random seed of length $\tilde{O}(w^2 \cdot \log(n))$ outputs a restriction $\rho \in \{0,1,\star\}^n$ such that for every depth-2 formula $F : \{0,1\}^n \to \{0,1\}$ of size $\operatorname{poly}(n)$ and width w the following holds:*

- *There exist two formulas $F^{\mathtt{low}}$ and $F^{\mathtt{up}}$ such that for every $x \in \{0,1\}^n$ it holds that $F^{\mathtt{low}}(x) \leq F(x) \leq F^{\mathtt{up}}(x)$.*
- *With probability $1 - 1/\operatorname{poly}(n)$ it holds that both $F^{\mathtt{low}}{\restriction_\rho}$ and $F^{\mathtt{up}}{\restriction_\rho}$ can be computed by decision trees of depth $O(\log(n))$, and that both $F^{\mathtt{low}}{\restriction_\rho}$ and $F^{\mathtt{up}}{\restriction_\rho}$ agree with F on $1 - 1/\operatorname{poly}(n)$ of the inputs in the subcube that corresponds to the living variables under ρ.*

Note that the seed length of the algorithm from Proposition 4 depends on the width of the formula F. Previous derandomizations of the switching lemma can also be adapted to depend on the width, but when the width is $o(\log(n))$ the seed length in Proposition 4 is significantly shorter than in these adaptations; see Section 2.2 for further details.

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 or the negated 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.

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 5 *(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.*

An appealing way to approach this "threshold" at depth four (with $B(n) = 2^{n^c}$) is to derandomize $\mathcal{AC}^0[\oplus]$ *circuits of depth three* with $B(n) = 2^{n^c}$. Goldreich and Wigderson derandomized most 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 a bottom layer of $\oplus$ gates), which they left as an open problem.

Our main result in this section is an algorithm that makes significant progress on this problem, by derandomizing $\oplus \wedge \oplus$ circuits with $B(n) = 2^{n^c}$ under various sub-quadratic upper bounds on the circuit size, where some of these bounds refer to each layer separately.

Theorem 6 *(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 6 makes essential use of the specific circuit C that is given to the algorithm as input. For further details see Section 2.3.

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., $\Pr_{x \in \mathbb{F}_2^n}[p(x) = 0] \leq b(n)$).

The problem is trivial when $b(n) < 2^{-d}$, since in this case p is constant, 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 $m(n) = \text{poly}(n)$, and conjectured that such a result would imply a quantified derandomization of $\oplus \wedge \oplus$ circuits of size $m(n)$. [1] We affirm their conjecture, by showing that any sufficiently dense hitting-set generator for degree-d polynomials with $b(n) = m(n) \cdot 2^{-d}$ also hits $\oplus \wedge \oplus$ circuits of size $m(n)$ with $B(n) = \Omega\left(2^n\right)$.

Theorem 7 *(reducing hitting $\oplus \wedge \oplus$ circuits to hitting biased polynomials of bounded degree). 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 $\epsilon = \epsilon(n) \leq 1/8$. 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 $|\mathbb{F}| = q \leq \text{poly}(n)$, and let $1 \leq d \leq (q-1) \cdot n$. 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 $b(n)$

[1] In [GW14, Sec. 6 (full version)] 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 2.3, 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).

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. [2] 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 d/q of its inputs.

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. Indeed, *most* polynomials of degree d vanish on at most a $O(1/q)$ fraction of their inputs, but the fact that this is the typical case does not a-priori imply that it is not easier to handle.

Our main result for this problem, however, goes much further: It turns out that even when considering the parameter $b(n) = 1/\mathrm{poly}(q)$, 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 $b(n) = 1/\mathrm{poly}(q)$ 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 $b(n) = 1/\mathrm{poly}(q)$.

Theorem 8 *(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 $d' \leq 0.99 \cdot q^{t+1}$, 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 $d = (d')^{1/(t+1)}$.*

The proofs of both items of Theorem 8 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 $d' = \mathrm{poly}(d)$ in the proof of Item (2). See Section 2.4 for details.

[2] For proof of the lower bound see, e.g., the proof of Theorem 41, 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.

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, and in Section 4 we prove two lemmas related to the technique of randomized tests that will be used in the paper. Then, each of the subsequent sections includes proofs for a corresponding section from the introduction: In Section 5 we prove Theorems 1 and 2; in Section 6 we prove Theorems 5 and 6; and in Section 7 we prove Theorems 7 and 8. In Appendix A we provide an alternative proof of [GW14, Thm. 1.6], and in Appendices B and C we provide proofs for several claims from Sections 5 and 7, respectively.

2 Our Techniques

In this section we give overviews of the proofs of the main theorems for each of the three settings: Theorems 2 and 3 for constant-depth circuits; Theorem 6 for constant-depth circuits with parity gates; and Theorem 8 for polynomials over large fields. Since several of our proofs rely on a common technique, we will begin by describing this technique in general terms (the results that use this technique are Theorems 3 and 7, Item (1) of Theorem 8, and also Theorem 42 in Appendix A).

2.1 A general technique: Randomized tests

Let $G \subseteq \{0,1\}^n$ be a set of *good* objects, and assume that we want to efficiently and deterministically find some $x \in G$. A known technique to do so is to design a *simple deterministic test* $T : \{0,1\}^n \to \{0,1\}$ such that $T(x) = 1$ if and only if $x \in G$. The existence of such a test T is useful, since if T is sufficiently simple such that we are able to construct a hitting-set generator for T, then the generator outputs $x \in G$ with positive probability (because the output distribution of the generator contains $x \in \{0,1\}^n$ such that $T(x) = 1$). Indeed, this approach reduces the task of finding $x \in G$ to the task of designing a test T for G that is *sufficiently simple* such that we are able to construct a hitting-set generator for T.

Intuitively, the *randomized tests technique* is based on the observation that an argument similar to the one above holds also when we replace the deterministic test T by a *distribution* $\mathbf{T}$ *over simple (deterministic) tests* such that, for every fixed $x \in \{0,1\}^n$, it holds that $\mathbf{T}(x)$ computes the indicator function of G, with high probability (say, 0.9). To see this, assume that $\mathbf{T}$ is indeed such a distribution, and let $\mathbf{w}$ be a distribution over $\{0,1\}^n$ that is a hitting-set with density $1-\epsilon$ for every $T \in \mathbf{T}$. Then, on the one hand, $\Pr[\mathbf{T}(\mathbf{w}) = 1] \geq 1-\epsilon$ (because for every $T \in \mathbf{T}$ it holds that $\Pr[T(\mathbf{w}) = 1] \geq 1-\epsilon$); and on the other hand, $\Pr[\mathbf{T}(\mathbf{w}) = 0] \geq \Pr[\mathbf{w} \notin G] \cdot \max_{x \notin G}\{\Pr[\mathbf{T}(x) = 0]\}$. Combining the two statements, and recalling that for every $x \notin G$ it holds that $\Pr[\mathbf{T}(x) = 0] \geq 0.9$, it follows that $\Pr[\mathbf{w} \notin G] \leq \epsilon/0.9$, which allows us to deduce that $\mathbf{w}$ contains an object in G.

Indeed, this approach reduces the task of finding $x \in G$ to the tasks of designing a distribution $\mathbf{T}$ over simple tests as above, and of constructing a hitting-set generator with high density for the residual (deterministic) tests $T \in \mathbf{T}$. The main benefit in this approach over the previous one (in which we had a single deterministic test) is that in some cases, *the use of randomness allows us to obtain very simple residual tests, which are simpler than any deterministic test for* G; one appealing example for such a case appears in Section 2.2. We stress that when designing the distribution $\mathbf{T}$ we can be wasteful in the use of randomness, because the existence of $\mathbf{T}$ is only a part of the *analysis*: The actual algorithm for finding

$x \in G$ is merely a hitting-set generator (for the residual tests $T \in \mathbf{T}$), whereas only the proof that the generator outputs $x \in G$ relies on the existence of the distribution $\mathbf{T}$.

Two relaxations of the hypotheses for the argument above can immediately be made. First, in our argument we only used the fact that $\mathbf{T}(x) = 0$ with high probability for every $x \notin G$ (and did not explicitly rely on the hypothesis that $\mathbf{T}(x) = 1$ with high probability for *every* $x \in G$). And secondly, we do not have to assume that $\mathbf{w}$ is a hitting-set with high density for every $T \in \mathbf{T}$, but rather only need the hypothesis that $\Pr[\mathbf{T}(\mathbf{w}) = 1]$ is high.

Let us demonstrate one appealing setting in which the two relaxed hypotheses above hold, which simplifies and abstracts the setting in the proof of Theorem 3. Assume that there exists a set $E \subseteq G$ of *excellent* objects, and that almost all objects are excellent; that is, a random $x \in \{0,1\}^n$ is not only good, but also has additional useful properties. Also assume that we are able to construct a distribution $\mathbf{T}$ over simple tests that distinguishes between excellent objects and bad ones (i.e., $\mathbf{T}$ solves a promise problem with some "gap" between the "yes" instances and the "no" instances). Denoting the uniform distribution over $\{0,1\}^n$ by $\mathbf{u}_n$, in this case we have that $\Pr[\mathbf{T}(\mathbf{u}_n) = 1]$ is high, whereas $\Pr[\mathbf{T}(x) = 0]$ is high for every $x \notin G$. Indeed, in such a setting, in order to find $x \in G$ it suffices to construct a pseudorandom generator for the residual tests $T \in \mathbf{T}$ (see Lemma 15).

2.2 Constant-depth circuits: Overview of the proofs of Theorems 2 and 3

Theorem 2 is a special case of the more general Theorem 3. However, since there is a simple and more direct way to prove Theorem 2, we describe this simpler way first, and only then turn to the describe the proof of the more general theorem.

Let C be a depth-D circuit that accepts all but $B(n) = \Omega\left(2^{n/\log^{D-2}(n)}\right)$ of its inputs. The hitting-set generator first uses pseudorandom restrictions to simplify 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 an adaptation of the derandomized switching lemma of Trevisan and Xue [TX13] (either Tal's [Tal14] improvement or the adapted version in Proposition 26), which requires a seed of length $\tilde{O}(\log^3(n))$. 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 any pseudorandom generator for depth-2 circuits with seed length at most $\tilde{O}(\log^3(n))$ (e.g., 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. [3]

Turning to the more general Theorem 3, the high-level structure of its proof is similar to that of the proof of Theorem 2: We first use a derandomized switching lemma to radically simplify the circuit, while keeping more than $\log(B(n))$ variables alive, and then use a pseudorandom generator for the simplified circuit to find a satisfying input. The key difference from Theorem 2 is that the first step uses a new derandomization of the switching lemma, which we establish.

The new derandomization of the switching lemma depends on the *width* (i.e., bottom fan-in) of the depth-2 formula that we want the restriction to simplify. Previous known derandomizations of the lemma can also be adapted to depend on the width of the formula:

[3] Actually, there is one minor subtlety in this description: In the derandomizations of [TX13, Tal14], the expected number of living variables is close to $n/\log^{d-2}(n)$, but it is *not* guaranteed that approximately this many variables remain alive with high (or even constant) probability. Nevertheless, the latter does hold when instantiating their generic construction in a specific manner; see the proof of Theorem 3 for further details.

For typical settings of the parameters (e.g., polynomially-small error), the derandomization of Goldreich and Wigderson [GW14] can be adapted to yield a seed length of $\tilde{O}(2^w) \cdot \log(n)$ for formulas of width w (see Proposition 44), and the derandomization of Trevisan and Xue [TX13] can be adapted (using the pseudorandom generator of Gopalan, Meka, and Reingold [GMR13]) to yield a seed length of $\tilde{O}(w) \cdot \log^2(n)$ (see Proposition 26). We show a derandomization that requires a seed of length $\tilde{O}(w^2 \cdot \log(n))$ (see Proposition 28). Indeed, in this new result, the dependency of the seed length on w is exponentially better than in [GW14], and the seed length is shorter than in [TX13] for any $w = o(\log(n))$. The caveat, however, is that we do not show that the formula itself is simplified in the subcube corresponding to the restriction; instead, we show that the formula is *approximated* by a decision tree of bounded depth in this subcube (i.e., there exists such a decision tree that agrees with the formula on almost all inputs in the subcube). This weaker conclusion suffices for our main application (i.e., for Theorem 3) as well as for all other applications of derandomized switch lemmas that we are aware of.

Our starting point in the proof of this lemma is a result of Gopalan, Meka, and Reingold [GMR13], which asserts that for any depth-2 formula F of width w and any $\beta > 0$, there exists a formula $F^{\mathtt{low}}$ of width at most w and size at most $m' = 2^{\tilde{O}(w) \cdot \log\log(1/\beta)}$ such that $F^{\mathtt{low}}$ is "lower-sandwiching" for F (i.e., $F^{\mathtt{low}}(x) \leq F(x)$ for all $x \in \{0,1\}^n$) and $\Pr_{x \in \{0,1\}^n}[F(x) \neq F^{\mathtt{low}}(x)] \leq \beta$. Now, since $F^{\mathtt{low}}$ is both small (i.e., m' is upper bounded) and of bounded width, we can find a restriction that simplifies it using a relatively short seed; specifically, we can use an adapted version of the lemma of [TX13] (see Proposition 26), and the required seed length (when we want the probability of error to be $1/\mathrm{poly}(n)$) is only $\tilde{O}(w) \cdot \log(m') \cdot \log(n) = \tilde{O}(w^2) \cdot \log(n) \cdot \log\log(1/\beta)$.

The main challenge that underlies this approach is that, while $F^{\mathtt{low}}$ agrees with F on most inputs $x \in \{0,1\}^n$, it is not clear that $F^{\mathtt{low}}$ also agrees with F on most inputs *in the subcube that corresponds to* ρ; that is, it is not guaranteed that $F^{\mathtt{low}}{\restriction_\rho}$ will agree with $F{\restriction_\rho}$ on most of *their* inputs. To make sure that $F^{\mathtt{low}}{\restriction_\rho}$ will agree with $F{\restriction_\rho}$ on most of their inputs, we will choose ρ such that it "fools" additional tests that check whether or not $F^{\mathtt{low}}{\restriction_\rho}$ and $F{\restriction_\rho}$ indeed typically agree. To design these tests we use the randomized tests technique: Specifically, a natural randomized test to decide whether or not $F^{\mathtt{low}}{\restriction_\rho}$ and $F{\restriction_\rho}$ typically agree is to sample random inputs inside the subcube that corresponds to ρ, and accept if and only if $F^{\mathtt{low}}{\restriction_\rho}$ and $F{\restriction_\rho}$ agree on the sampled inputs.

Indeed, *the residual tests under this distribution are simpler (in any reasonable sense) than any deterministic test* that decides whether or not $F^{\mathtt{low}}{\restriction_\rho}$ and $F{\restriction_\rho}$ agree on most of their inputs. The remaining task is thus to construct a hitting-set generator with high density for these residual tests. We will now describe how to do so, relying both on the specific details of the construction of $F^{\mathtt{low}}$ from [GMR13], in order to construct circuits with a specific structure that will be convenient for us for each residual test, and on relaxations of the randomized tests technique that follow the ones suggested in the end of Section 2.1.

We want to use the lemma to simplify polynomially-many depth-2 formulas (i.e., simplify an entire "layer" of a constant-depth circuit). Thus, we want that for every fixed formula F it will hold that $F^{\mathtt{low}}{\restriction_\rho}$ and $F{\restriction_\rho}$ agree on an all but an α-fraction of their inputs, where $\alpha = 1/\mathrm{poly}(n)$. We say that a restriction ρ is *good* if $F^{\mathtt{low}}{\restriction_\rho}$ and $F{\restriction_\rho}$ agree with probability at least $1 - \alpha$. If we start from a formula $F^{\mathtt{low}}$ with the approximation parameter $\beta = \mathrm{poly}(\alpha)$, then almost all restrictions ρ' are *excellent*, in the sense that $F^{\mathtt{low}}{\restriction_{\rho'}}$ and $F{\restriction_{\rho'}}$ agree with probability $1 - \sqrt{\beta} \gg 1 - \alpha$. For each fixed F and $F^{\mathtt{low}}$, to distinguish between excellent restrictions and restrictions that are not good, the distribution $\mathbf{T}$ of tests uniformly

samples $\text{poly}(\alpha)$ inputs inside the subcube that corresponds to its input restriction ρ, and accepts ρ if and only if F and $F^{\mathtt{low}}$ agree on the sampled inputs.

The next step is to show that each residual test $T \in \mathbf{T}$ can be computed by a circuit with a convenient structure. To do so, we observe that the construction of $F^{\mathtt{low}}$ in [GMR13] is based on a sequence of *specific syntactic modifications* to F: Each syntactic modification is a simplification of a quasi-sunflower, a notion introduced by Rossman [Ros14] (for more specific details see Section 5.2.1). We define the tests $T \in \mathbf{T}$ to accept if and only if the *specific syntactic modifications* used to transform F into $F^{\mathtt{low}}$ did not affect the formula at the relevant inputs. Then, we show that each such test T can be decided by a depth-3 circuit with a top AND gate and bottom fan-in w (relying on the hypothesis that the original formula F has width w; see Claim 29.3).

Now, since almost all restrictions are excellent, and each excellent restriction is accepted with high probability by $\mathbf{T}$, it follows that almost all tests in $\mathbf{T}$ belong to the subset $\mathbf{T}' \subseteq \mathbf{T}$ of tests that accept almost all of their input restrictions. We will in fact construct a hitting-set generator for the residual tests $T \in \mathbf{T}'$. This can be done relying both on the fact that $T \in \mathbf{T}'$ has very high acceptance probability and on the fact that it can be computed by a depth-3 circuit with a top AND gate and bottom fan-in w (the latter allows us to use the pseudorandom generator of [GMR13] for formulas of small width; see Claim 29.4).

To prove Theorem 3, we will repeat the following step: First reduce the width of the formulas in the next-to-bottom layer by a pseudorandom restriction (see Claim 30.1), and then use the new switching lemma to approximate the circuit by a circuit in which all the formulas in the next-to-bottom layer are simplified (and thus the latter circuit has smaller depth). Since all our approximations are "lower-sandwiching", any satisfying input for the latter circuit is also satisfying for the former circuit.

2.3 Constant-depth circuits with parity gates: Overview of the proof of Theorem 6

Let us now describe the high-level strategy of the algorithms of Theorem 6. 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, we use *affine restrictions*, which are obtained by fixing values to some of the bottom $\oplus$-gates. These are analogous to standard "bit-fixing" restrictions, but in contrast to the latter, we cannot consider *any* sequence of fixed values to the bottom $\oplus$-gates: This is the case because 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. For concreteness, let us now describe the construction of Item (2) of Theorem 6, 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 $\oplus$-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 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 *mildly* reduce the fan-in of $\wedge$-gates (from n to $\sqrt{n}$), and then invoke the 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). 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; each such action eliminates $\sqrt{n}$ wires that feed into the middle layer, and therefore in this 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 6.2.3). 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 6.2.2 and 6.2.4, respectively.

[4] For any $\wedge$-gate g with initial fan-in $d_\wedge$, the probability that there exists a set of size c of $\oplus$-gates that feed into g that are all unfixed is at most $\binom{d_\wedge}{c} \cdot p^c = 1/\mathrm{poly}(n)$, 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.

2.4 Polynomials that vanish rarely: Overview of the proof of Theorem 8

The main component in the proof of Theorem 8 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 $\text{poly}(|\mathbb{F}|)$ points that are $O(1)$-wise independent). Our strategy will be 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 $w \in W$. (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 $p\restriction_W \not\equiv 0$? 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 8, and the second yields $d' = d$, and corresponds to Item (1) of Theorem 8.

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 actually a polynomial of degree d in $\vec{u}$ (see Claim 39.1). 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 polynomial 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 38). 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 8, 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 finding a subspace W such that $p|_W \not\equiv 0$, but we will then further reduce the latter task to the task of hitting polynomials of degree d that vanish on at most $O(1/|\mathbb{F}|)$ of their inputs. To do so, we use a variation on the technique of randomized tests. Specifically, we construct a distribution $\mathbf{h}$ over polynomials $\mathbb{F}^{O(n)} \to \mathbb{F}$ that satisfies: (1) For every $\vec{u} \in \mathbb{F}^{O(n)}$ such that $p|_{W_{\vec{u}}} \equiv 0$ it holds that $\mathbf{h}(\vec{u}) = 0$, with probability one; (2) The distribution $\mathbf{h}$ is typically in the class $\mathcal{P}$ of degree-d polynomials that vanish on at most $O(1/|\mathbb{F}|)$ of their inputs. We will then rely on arguments similar to those in Section 2.1, to deduce that any sufficiently dense hitting-set generator for $\mathcal{P}$ outputs $\vec{u}$ such that $p|_{W_{\vec{u}}} \not\equiv 0$ (see Lemma 16).

Recall that the coefficients of $p|_{W_{\vec{u}}}$ are degree-d polynomials in $\vec{u}$. The aforementioned distribution, denoted by $\mathbf{h}$, is simply a random $\mathbb{F}$-linear combination of these degree-d polynomials. Note that $\mathbf{h}$ is supported on polynomials of degree d, and indeed for every $\vec{u}$ such that $p|_{W_{\vec{u}}} \equiv 0$ it holds that $\mathbf{h}(\vec{u}) = 0$, with probability one. Moreover, since almost all $\vec{u}$'s are such that $p|_{W_{\vec{u}}} \not\equiv 0$, and for each such $\vec{u}$ it holds that $\Pr[\mathbf{h}(\vec{u}) \neq 0] = 1 - 1/|\mathbb{F}|$, the expected fraction of inputs on which a polynomial in $\mathbf{h}$ vanishes is at most $O(1/|\mathbb{F}|)$. Thus, most of the polynomials in the support of $\mathbf{h}$ are in $\mathcal{P}$. We can therefore deduce that any sufficiently dense hitting-set generator for $\mathcal{P}$ also outputs $\vec{u}$ such that $p|_{W_{\vec{u}}} \not\equiv 0$, 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 $\{\mathfrak{D} \to \mathfrak{R}\}$ the set of functions from domain $\mathfrak{D}$ to range $\mathfrak{R}$. Distributions and random variables will always be denoted by boldface letters. Given a set 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. We also assume that for every $n \in \mathbb{N}$ it holds that C_n has $2 \cdot n$ input gates that correspond to the input literals (i.e., the input bits $x_1, ..., x_n$ and their negations $\neg x_1, ..., \neg x_n$); and that C_n is *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 $\vee$).

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 $f|_\rho : \{0,1\}^n \to \{0,1\}$ is defined by $f|_\rho(x) = f(y)$, 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 $f|_\rho = \left(f|_{\rho_2}\right)|_{\rho_1}$.

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 9 *(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 $\left|\Pr_{x\in\{0,1\}^n}[f(x) = 1] - \Pr_{y\in\{0,1\}^{\ell(n)}}[f(G(1^n,y)) = 1]\right| < \epsilon$.*

If G is a pseudorandom generator with error parameter ϵ for a class of functions $\mathcal{F}$, then we say that functions from $\mathcal{F}$ are ϵ-fooled *by G.*

Definition 10 *(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 $\Pr_{y\in\{0,1\}^{\ell(n)}}[f(G(1^n,y)) \neq 0] \geq \epsilon(n)$.*

We now extend Definition 10 by defining hitting-set generators for functions over fields larger than $\mathbb{F}_2$. 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 11 *(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 11, 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.

3.3 Distributions with limited independence

We say that random variables $\mathbf{x}_1, ..., \mathbf{x}_n \in \{0,1\}^n$ are t-wise independent if for every set $S \subseteq [n]$ of size $|S| = t$, the marginal distribution $(\mathbf{x}_i)_{i \in S}$ is uniform over $\{0,1\}^t$. We will use the following well-known tail bound (for a proof see [BR94, Lemma 2.3]):

Fact 12 *(tail bound for t-wise independent distributions). Let $n \in \mathbb{N}$, and let $t \geq 4$ be an even number. Let $\mathbf{x}_1, ..., \mathbf{x}_n$ be random variables in $\{0,1\}$ that are t-wise independent, and denote $\mu = \mathbb{E}\left[\frac{1}{n} \cdot \sum_{i \in [n]} \mathbf{x}_i\right]$. Then, for any $\zeta > 0$ it holds that $\Pr\left[\left|\frac{1}{n} \cdot \sum_{i \in [n]} \mathbf{x}_i - \mu\right| \geq \zeta\right] \leq 8 \cdot \left(\frac{t \cdot \mu \cdot n + t^2}{\zeta^2 \cdot n^2}\right)^{t/2}$.*

We say that $\mathbf{x}_1, ..., \mathbf{x}_n \in \{0,1\}^n$ are δ-almost t-wise independent if for every set $S \subseteq [n]$ of size $|S| = t$, the statistical distance between $(\mathbf{x}_i)_{i \in S}$ and the uniform distribution over $\{0,1\}^t$ is at most δ. Then, the following well-known tail bound holds:

Fact 13 *(tail bound for almost t-wise independent distributions). Let $n \in \mathbb{N}$, let $t \geq 4$ be an even number, and let $\delta > 0$. Let $\mathbf{x}_1, ..., \mathbf{x}_n$ be random variables in $\{0,1\}$ that are δ-almost t-wise independent, and denote $\mu = \mathbb{E}\left[\frac{1}{n} \cdot \sum_{i \in [n]} \mathbf{x}_i\right]$. Then, for any $\zeta > 0$ it holds that $\Pr\left[\left|\frac{1}{n} \cdot \sum_{i \in [n]} \mathbf{x}_i - \mu\right| \geq \zeta\right] < 8 \cdot \left(\frac{t \cdot \mu \cdot n + t^2}{\zeta^2 \cdot n^2}\right)^{t/2} + (2 \cdot n)^t \cdot \delta$.*

For a proof of Fact 13 see, e.g., [LRTV09, Lemma 18]. We will frequently use Fact 13 with the parameters $t = O(1)$, and $\zeta = \mu/2$, and $\delta = 1/p(n)$ where p is a sufficiently large polynomial; in this case, we have that $\Pr\left[\frac{1}{n} \cdot \sum_{i \in [n]} \mathbf{x}_i \notin \mu \pm (\mu/2)\right] = O\left(1/\left(\mu \cdot n\right)^{t/2}\right)$.

We will also need the following fact, which, loosely speaking, asserts that concatenating two independently-chosen distributions that are almost t-wise independent yields a distribution that is still almost t-wise independent.

Fact 14 *(concatenating almost t-wise independent distributions). Let $n, n' \in \mathbb{N}$, let $\delta, \delta' < \frac{1}{2}$, and let $t \in \mathbb{N}$. Let $\mathbf{y}$ be a distribution over $\{0,1\}^n$ that is δ-almost t-wise independent, and let $\mathbf{z}$ be a distribution over $\{0,1\}^{n'}$ that is δ'-almost t-wise independent. Let $\mathbf{r} = \mathbf{y} \circ \mathbf{z}$ be a distribution that is obtained by concatenating a sample from $\mathbf{y}$ and an independent sample from $\mathbf{z}$. Then, the distribution $\mathbf{r}$ is $(\delta + \delta')$-almost t-wise independent.*

Proof. Fix a set $S \subseteq [n + n']$ of size $|S| = t$, and let us prove that the ℓ_1-distance between $\mathbf{r}_S$ and the uniform distribution is at most $2 \cdot (\delta + \delta')$ (which implies that the statistical distance between them is at most $\delta + \delta'$). Partition S into $W = S \cap [n]$ and $W' = S \setminus [n]$, and denote $w = |W|$ and $w' = |W'|$. Then, we have that:

$$\begin{aligned} \left\|\mathbf{r}_S - \mathbf{u}_t\right\|_1 &= \left\|\mathbf{y}_W \circ \mathbf{z}_{W'} - \mathbf{u}_w \circ \mathbf{u}_{w'}\right\|_1 \\ &\leq \left\|\mathbf{y}_W \circ \mathbf{z}_{W'} - \mathbf{y}_W \circ \mathbf{u}_{w'}\right\|_1 + \left\|\mathbf{y}_W \circ \mathbf{u}_{w'} - \mathbf{u}_w \circ \mathbf{u}_{w'}\right\|_1 \\ &= \left\|\mathbf{z}_{W'} - \mathbf{u}_{w'}\right\|_1 + \left\|\mathbf{y}_W - \mathbf{u}_w\right\|_1 \ , \end{aligned}$$

which is upper-bounded by $2 \cdot \delta' + 2 \cdot \delta$. ■

4 Randomized tests

In this section we state and prove three lemmas that are related to the technique of randomized tests. The first lemma (i.e., Lemma 15) corresponds to the high-level description

in Sections 2.1 and 2.2, and will be useful for us in Section 5. The next two lemmas (i.e., Lemmas 16 and 18) are variations that will be useful for us in Section 7.

Towards stating Lemma 15, let us recall the setting that was described in Sections 2.1 and 2.2: For a set $G \subseteq \{0,1\}^n$ of good objects, our goal is to find some $x \in G$; almost all objects are excellent, i.e. not only good but also in a subset $E \subseteq G$ with additional useful properties; there exists a distribution $\mathbf{T}$ over simple tests that distinguishes between excellent objects and objects that are not good; and the distribution $\mathbf{w}$ "fools" almost all tests $T \in \mathbf{T}$. In this case, $\mathbf{w}$ contains an object in G.

Lemma 15 *(randomized tests). Let $n \in \mathbb{N}$, and let $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5 > 0$ be error parameters.*

- *Let $G \subseteq \{0,1\}^n$, and let $E \subseteq G$ such that $\Pr_{x \in \{0,1\}^n}[x \in E] \geq 1 - \epsilon_1$.*
- *Let $\mathbf{T}$ be a distribution over functions $T : \{0,1\}^n \to \{0,1\}$ such that for every $x \in E$ it holds that $\Pr_{T \sim \mathbf{T}}[T(x) = 1] \geq 1 - \epsilon_2$ and for every $x \notin G$ it holds that $\Pr_{T \sim \mathbf{T}}[T(x) = 0] \geq 1 - \epsilon_3$.*
- *Let $\mathbf{w}$ be a distribution that ϵ_5-fools all but an ϵ_4-fraction of the tests in $\mathbf{T}$; that is, the probability over $T \sim \mathbf{T}$ that $\left| \Pr[T(\mathbf{u}_n) = 1] - \Pr[T(\mathbf{w}) = 1] \right| > \epsilon_5$ is at most ϵ_4.*

Then, the probability that $\mathbf{w} \in G$ is at least $1 - (\epsilon_1 + \epsilon_2 + \epsilon_3 + 2\epsilon_4 + \epsilon_5)$.

Recall that in the proof of Theorem 3, the set of tests that are "fooled" by $\mathbf{w}$ is the set of tests that accept almost all of their inputs.

Proof of Lemma 15. Let $\mathcal{T}$ be the set of tests in the support of $\mathbf{T}$ that are ϵ_5-fooled by $\mathbf{w}$; that is, $\mathcal{T} = \left\{ T \in \mathrm{supp}(\mathbf{T}) : \left| \Pr[T(\mathbf{u}_n) = 1] - \Pr[T(\mathbf{w}) = 1] \right| \leq \epsilon_5 \right\}$. To upper-bound the probability that $\mathbf{w} \notin G$, first note that a random test $T \sim \mathbf{T}$ accepts a random input $x \in \{0,1\}^n$ with high probability; this is the case because

$$\Pr_{T \sim \mathbf{T}}[T(\mathbf{u}_n) = 1] \geq \Pr[\mathbf{u}_n \in E] \cdot \min_{x \in E} \left\{ \Pr_{T \sim \mathbf{T}}[T(x) = 1] \right\} \geq 1 - (\epsilon_1 + \epsilon_2) \,. \tag{4.1}$$

It follows that a random test $T \sim \mathbf{T}$ also accepts a *pseudorandom input* from the distribution $\mathbf{w}$ with high probability, since

$$\begin{aligned} \Pr_{T \sim \mathbf{T}}[T(\mathbf{w}) = 1] &\geq \Pr_{T \sim \mathbf{T}}[T \in \mathcal{T}] \cdot \Pr_{T \sim \mathbf{T}}[T(\mathbf{w}) = 1 | T \in \mathcal{T}] \\ &\geq (1 - \epsilon_4) \cdot \left(\Pr_{T \sim \mathbf{T}}[T(\mathbf{u}_n) = 1 | T \in \mathcal{T}] - \epsilon_5 \right) \\ &\geq (1 - \epsilon_4) \cdot \left(\Pr_{T \sim \mathbf{T}}[T(\mathbf{u}_n) = 1] - \epsilon_4 - \epsilon_5 \right) \,, \end{aligned}$$

which, relying on Eq. (4.1), is lower-bounded by $1 - \epsilon_1 - \epsilon_2 - 2\epsilon_4 - \epsilon_5$.

However, if $\Pr[\mathbf{w} \notin G]$ is high, then there is significant probability that a random test from $\mathbf{T}$ will reject a pseudorandom input from $\mathbf{w}$. Specifically,

$$\Pr_{T \sim \mathbf{T}}[T(\mathbf{w}) = 0] \geq \Pr[\mathbf{w} \notin G] \cdot \min_{x \notin G} \left\{ \Pr_{T \sim \mathbf{T}}[T(x) = 0] \right\} \geq \Pr[\mathbf{w} \notin G] - \epsilon_3 \,,$$

and it follows that $\Pr[\mathbf{w} \notin G] \leq \epsilon_1 + \epsilon_2 + \epsilon_3 + 2\epsilon_4 + \epsilon_5$. ■

We now present two variations on the argument above that are applicable in the setting of polynomials over finite fields. For a finite field $\mathbb{F}$, let $G \subseteq \mathbb{F}^n$, and assume that there exists a distribution $\mathbf{h}$ over polynomials $\mathbb{F}^n \to \mathbb{F}$ such that for every $x \notin G$ it holds that $\mathbf{h}(x) = 0$, with high probability. Further assume that there exists a hitting-set generator with high density for the polynomials h in the support of $\mathbf{h}$. Then, using an argument similar to the one in the beginning of Section 2.1, the hitting-set generator contains $x \in G$. [6]

Lemma 16 *(randomized tests over finite fields). Let $n \in \mathbb{N}$, let $\mathbb{F}$ be any finite field, and let $\epsilon_1, \epsilon_2, \epsilon_3 > 0$ be three parameters. Assume that, for some $G \subseteq \mathbb{F}^n$, it holds that:*

1. *There exists a distribution $\mathbf{h}$ over $\{\mathbb{F}^n \to \mathbb{F}\}$ such that for every $x \notin G$ it holds that $\Pr_{h \sim \mathbf{h}}[h(x) = 0] \geq 1 - \epsilon_1$.*
2. *There exists a set $\mathcal{H} \subseteq \{\mathbb{F}^n \to \mathbb{F}\}$ such that $\Pr_{h \sim \mathbf{h}}[h \in \mathcal{H}] \geq 1 - \epsilon_2$.*
3. *There exists a distribution $\mathbf{w}$ over $\mathbb{F}^n$ such that for every $h \in \mathcal{H}$ it holds that $\Pr[h(\mathbf{w}) \neq 0] \geq 1 - \epsilon_3$.*

Then, $\Pr[\mathbf{w} \in G] \geq 1 - \epsilon_1 - \epsilon_2 - \epsilon_3$.

Proof. We first show that $\Pr[\mathbf{w} \in G] \geq \mathbb{E}_{h \sim \mathbf{h}}[\Pr[h(\mathbf{w}) \neq 0]] - \epsilon_1$. This is the case because

$$\begin{aligned}
\mathbb{E}_{h \sim \mathbf{h}}\left[\Pr[h(\mathbf{w}) \neq 0]\right] &= \mathbb{E}_{x \sim \mathbf{w}}\left[\Pr_{h \sim \mathbf{h}}[h(x) \neq 0]\right] \\
&\leq \Pr_{x \sim \mathbf{w}}[x \in G] + \Pr_{x \sim \mathbf{w}}[x \notin G] \cdot \max_{x \notin G}\left\{\Pr_{h \sim \mathbf{h}}[h(x) \neq 0]\right\} \\
&\leq \Pr[\mathbf{w} \in G] + \epsilon_1 \,.
\end{aligned}$$

Now, by our hypothesis, the probability that $\mathbf{h} \in \mathcal{H}$ is at least $1 - \epsilon_2$, and for every $h \in \mathcal{H}$ it holds that $\Pr[h(\mathbf{w}) \neq 0] \geq 1 - \epsilon_3$. Therefore,

$$\mathbb{E}_{h \sim \mathbf{h}}[\Pr[h(\mathbf{w}) \neq 0]] \geq \Pr_{h \sim \mathbf{h}}[h \in \mathcal{H}] \cdot \Pr_{h \sim \mathbf{h}}[h(\mathbf{w}) \neq 0 | h \in \mathcal{H}] \geq 1 - \epsilon_2 - \epsilon_3 \,,$$

which implies that $\Pr[\mathbf{w} \in G] \geq 1 - \epsilon_1 - \epsilon_2 - \epsilon_3$. ■

In the next argument, instead of trying to hit a fixed set $G \subseteq \mathbb{F}^n$, we will fix a polynomial $p : \mathbb{F}^n \to \mathbb{F}$, and try to "fool" p (i.e., we want to construct a pseudorandom generator for p). Indeed, we will need to explain exactly what we mean by "fooling" in the context of functions over finite fields. Towards presenting the argument, let us first define the notion of *randomly computing p by a distribution of functions that is typically over simpler functions*.

Definition 17 *(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 $1 - \rho'$.*

[6] Recall that this argument is different than the argument in Lemma 15: On the one hand, we do not assume that G is dense, or that for every $x \in G$ it holds that $\mathbf{h}(x) \neq 0$, with high probability; but on the other hand, we require a hitting-set generator with high density for $h \in \mathrm{supp}(\mathbf{h})$ (rather than a pseudorandom generator).

The following claim extends an argument that is implicit in the work of Bogdanov and Viola [BV10, Proof of Lemma 23]. Loosely speaking, our claim is the following: If p can be computed with small error by a distribution $\mathbf{h}$ that is typically 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 for some (fixed) mapping $\xi : \mathbb{F} \to \mathbb{C}$ it holds that $\left|\mathbb{E}[\xi(f(\mathbf{w}))] - \mathbb{E}[\xi(f(\mathbf{u}_n))]\right|$ is small. [7]

Lemma 18 *(an extension of a claim that is implicit in [BV10]). Let $n \in \mathbb{N}$, and let $\mathbb{F}$ be any finite field. Let $\epsilon_1, \epsilon_2, \epsilon_3 > 0$ 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 ϵ_1 by a distribution $\mathbf{h}$ over $\{\mathbb{F}^n \to \mathbb{F}\}$ that is $(1-\epsilon_2)$-typically in $\mathcal{H}$.*

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 $\left|\mathbb{E}[\xi(h(\mathbf{u}_n))] - \mathbb{E}[\xi(h(\mathbf{w}))]\right| < \epsilon_3$. Then, $\left|\mathbb{E}[\xi(p(\mathbf{u}_n))] - \mathbb{E}[\xi(p(\mathbf{w}))]\right| < 2\delta \cdot \epsilon_1 + \delta \cdot \epsilon_2 + \epsilon_3$.

Proof. 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

$$\begin{aligned}\left|\mathbb{E}[p'(\mathbf{u}_n)] - \mathbb{E}[p'(\mathbf{w})]\right| \leq &\left|\mathbb{E}[p'(\mathbf{u}_n)] - \mathbb{E}_{h\sim\mathbf{h}}\left[h'(\mathbf{u}_n)\right]\right| + \\ &\left|\mathbb{E}_{h\sim\mathbf{h}}\left[h'(\mathbf{u}_n)\right] - \mathbb{E}_{h\sim\mathbf{h}}\left[h'(\mathbf{w})\right]\right| + \\ &\left|\mathbb{E}_{h\sim\mathbf{h}}\left[h'(\mathbf{w})\right] - \mathbb{E}[p'(\mathbf{w})]\right| . \end{aligned} \tag{4.2}$$

To upper bound the first term in Eq. (4.2), note that

$$\begin{aligned}\left|\mathbb{E}[p'(\mathbf{u}_n)] - \mathbb{E}_{h\sim\mathbf{h}}\left[h'(\mathbf{u}_n)\right]\right| &\leq \mathbb{E}_{u\in\mathbb{F}^n, h\sim\mathbf{h}}\left[\left|p'(u) - h'(u)\right|\right] \\ &\leq \mathbb{E}_{u\in\mathbb{F}^n}\left[\Pr_{h\sim\mathbf{h}}[h(u) \neq p(u)] \cdot \max_{v,w\in\mathbb{F}}\{|\xi(v) - \xi(w)|\}\right] \\ &\leq \delta \cdot \epsilon_1 ,\end{aligned}$$

where the last inequality holds because for every fixed $u \in \mathbb{F}^n$ it holds that $\Pr_{h\sim\mathbf{h}}[h(u) \neq p(u)] \leq \epsilon_1$. The third item is similarly upper bounded by $\delta \cdot \epsilon_1$, 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. (4.2), note that

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

which is upper bounded by $\delta \cdot \epsilon_2 + \epsilon_3$. (Specifically, the first term is upper bounded by $\delta \cdot \epsilon_2$, whereas to bound the second term by ϵ_3 we use the hypothesis that for every $h \in \mathcal{H}$ it holds that $\left|\mathbb{E}[h'(\mathbf{u}_n)] - \mathbb{E}[h'(\mathbf{w})]\right| < \epsilon_3$.) ■

[7] A standard choice for ξ is any fixed non-trivial character $\mathrm{e} : \mathbb{F} \to \mathbb{C}$.

5 Constant-depth circuits

5.1 Proof of Theorem 1

Let $c = D - d - 11$. Starting from a depth-d circuit $C : \{0,1\}^n \to \{0,1\}$, we will employ error-reduction within $\mathcal{AC}^0$, by first sampling inputs for C using the seeded extractor of Cheng and Li [CL16], and then taking the disjunction of the evaluation of C on these inputs. The extractor will be of depth $c + 10$, and will work for min-entropy $n' / \log^c(n')$, where n' is the number of random bits that it uses. Thus, this construction will yield a circuit $C' : \{0,1\}^{n'} \to \{0,1\}$ of depth $D = d + (c + 10) + 1$ that accepts all but $2^{n'/\log^c(n')} = 2^{n'/\log^{D-d-11}(n')}$ of its inputs. Details follow.

Let $C : \{0,1\}^n \to \{0,1\}$ be a circuit of depth d. We will rely on the following theorem from [CL16], which we cite with minor changes of notation:

Theorem 19 *(an $\mathcal{AC}^0$-computable seeded extractor [CL16, Thm 1.5]). For any constant $c \in \mathbb{N}$, and $k = \Omega\left(n' / \log^c(n')\right)$ and any $\epsilon = 1/\mathrm{poly}(n')$, there exists an explicit construction of a strong (k, ϵ)-extractor $Ext : \{0,1\}^{n'} \times \{0,1\}^d \to \{0,1\}^n$ that can be computed by an $\mathcal{AC}^0$ circuit of depth $c + 10$, where $d = O(\log(n))$, $n = k^{\Omega(1)}$ and the extractor family has locality $O(\log^{c+5}(n))$.*

We will not need the strongness property or the locality property in the current proof. Let $n' = \mathrm{poly}(n)$ such that for $k = \Omega\left(n' / \log^c(n')\right)$ it holds that $n = k^{\Omega(1)}$, and let $Ext : \{0,1\}^{n'} \times \{0,1\}^d \to \{0,1\}^n$ be the seeded extractor from Theorem 19, instantiated with error parameter $\epsilon = 1/4$. We construct a circuit $C' : \{0,1\}^{n'} \to \{0,1\}$ that first computes the values $Ext(x,z)$, for each possible seed $z \in \{0,1\}^d$, then evaluates C on each value $E(x,z)$, and finally takes an OR of these evaluations; that is, $C'(x) = \vee_{z \in \{0,1\}^s} C\left(Ext(x,z)\right)$.

Note that C' has depth D and size $\mathrm{poly}(n)$. Also note that the number of inputs $x \in \{0,1\}^{n'}$ for which $\Pr_z[C(Ext(x,z))] < 1/4$ is at most $2^{n'/\log^c(n')}$. [8] In particular, C' accepts all but at most $2^{n'/\log^c(n')}$ of its inputs, and for each satisfying input x for C', we can find a corresponding satisfying input for C among $\{Ext(x,z)\}_{z \in \{0,1\}^s}$.

5.2 Proofs of Theorems 2 and 3

The first step towards proving Theorems 2 and 3 is to establish a derandomized switching lemma that simplifies depth-2 formulas of *bounded-width*; after presenting several required definitions in Section 5.2.1, we prove the lemma in Section 5.2.2. Then, in Section 5.2.3, we use the lemma to prove Theorems 2 and 3.

5.2.1 Preliminary definitions, and results from [GMR13]

For any restriction $\rho \in \{0,1,\star\}^n$, denote by $\mathfrak{C}(\rho)$ the subcube that corresponds to the living variables under ρ; that is, $\mathfrak{C}(\rho) = \{x \in \{0,1\}^n : \forall i \in [n] \text{ s.t. } \rho_i \neq \star \text{ it holds that } x_i = \rho_i\}$. We identify strings $r \in \{0,1\}^{(q+1)\cdot n}$, where $n, q \in \mathbb{N}$, with restrictions $\rho = \rho_r \in \{0,1,\star\}^n$, 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. When we refer to a "block" in the string that corresponds to a restriction, we mean a block of $q+1$ bits that corresponds to some variable. When we say that a restriction is chosen from a distribution $\mathbf{r}$ over $\{0,1\}^{(q+1)\cdot n}$, we mean that a string is chosen according

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

to $\mathbf{r}$, and interpreted as a restriction. Moreover, when we say that an algorithm "reads bits" in the restriction, we mean that it reads bits in the corresponding string.

In addition, we will sometimes identify a *pair* of strings $y \in \{0,1\}^{q \cdot n}$ and $z \in \{0,1\}^n$ with a restriction $\rho = \rho_{y,z}$. In this case, the restriction $\rho = \rho_{y,z}$ is the restriction ρ_r that is obtained by combining y and z to a string r in the natural way (i.e., appending a bit from z to each block of q bits in y). Note that the string y determines which variables ρ keeps alive, and the string z determines the values that ρ assigns to the fixed variables.

Throughout the section, whenever we consider a depth-2 formula for a function $F : \{0,1\}^n \to \{0,1\}$, we allow the formula to be a redundant representation of F (i.e., not necessarily the most concise representation of F as a formula), and in particular we allow formulas in which some clauses are simply constants. We will identify any clause of a depth-2 formula with the corresponding subset of the literals; the clause is a conjunction of the literals if the formula is a DNF, and otherwise it is a disjunction of the literals. We say that a function $F^{\mathtt{low}} : \{0,1\}^n \to \{0,1\}$ is lower-sandwiching for F if for every $x \in \{0,1\}^n$ it holds that $F^{\mathtt{low}}(x) \leq F(x)$. Similarly, we say that $F^{\mathtt{up}} : \{0,1\}^n \to \{0,1\}$ is upper-sandwiching for F if for every $x \in \{0,1\}^n$ it holds that $F(x) \leq F^{\mathtt{up}}(x)$.

Refinements: Definition and basic facts. We need several definitions that are related to the results of Gopalan, Meka, and Reingold [GMR13]. Their main theorem involves a process of *sparsification* of a depth-2 formula. The sparsification process is iterative: In each iteration, they identify a *quasi-sunflower* in the formula (a notion that was introduced by Rossman [Ros14]), and simplify the quasi-sunflower using one of two operations. The first operation is simply the removal of a clause from the formula; and the second operation is the removal of a set $f_1, ..., f_u$ of $u \geq 2$ clauses, replacing them with a new clause that consists of the set of literals that are shared by all the u clauses (i.e., replacing $f_1, ..., f_u$ with the clause $\bigcap_{j \in [u]} f_j$). The following definition generalizes this sparsification process. [9]

Definition 20 *(refinements of a depth-2 formula). Let $F : \{0,1\}^n \to \{0,1\}$ be a depth-2 formula with at least two clauses. We define the following three syntactic operations on F, which we call* refinement steps.

1. *A* removal step *is simply the removal of a clause from F.*
2. *A* merging step *is the removal of $u \geq 2$ clauses $f_1, ..., f_u$ from F, and the addition of a new clause that consists of the set of literals that appear in all the u clauses (i.e., replacing $f_1, ..., f_u$ with the new clause $\bigcap_{j \in [u]} f_j$). If $\bigcap_{j \in [u]} f_j = \emptyset$, then the new clause computes the constant one function if F is a DNF, and the constant zero function if F is a CNF.*
3. *A* clean-up step *is the removal of one or more clauses that compute the constant zero function from a DNF, or of one or more clauses that compute the constant one function from a CNF.*

We say that a depth-2 formula $F' : \{0,1\}^n \to \{0,1\}$ is a refinement *of another depth-2 formula $F : \{0,1\}^n \to \{0,1\}$ if F' can be obtained from F either by a sequence of removal steps and clean-up steps, or by a sequence of merging steps and clean-up steps.*

We now state some basic facts about refinements, which will be useful for us later on. The following two facts follow from Definition 20:

[9] The reason that we need this generalization is in order to facilitate the proof of Claim 23; this is also the reason that we allow formulas to have redundant clauses that compute constant functions.

Fact 21 *(refinements under negations). Let $F : \{0,1\}^n \to \{0,1\}$ and $F' : \{0,1\}^n \to \{0,1\}$ be depth-2 formulas. Then, F' is a refinement of F if and only if $\neg(F')$ is a refinement of $\neg F$.*

Fact 22 *(sandwiching refinements). Let $F : \{0,1\}^n \to \{0,1\}$ be a DNF. Then, any refinement of F that is obtained by a sequence of removal steps and clean-up steps is lower-sandwiching for F, and any refinement of F that is obtained by a sequence of merging steps clean-up steps is upper-sandwiching for F.*

Loosely speaking, the following claim asserts that if F' is a refinement of F, then for any restriction ρ it holds that $(F')\restriction_\rho$ is a refinement of $F\restriction_\rho$. That is, intuitively, restricting both F and F' by ρ does not affect the fact that the latter formula is a refinement of the former.

Claim 23 *(refinements under restrictions). Let $F : \{0,1\}^n \to \{0,1\}$ be a depth-2 formula of width w and size m, and let $F' : \{0,1\}^n \to \{0,1\}$ be a refinement of F. Then, for any restriction $\rho \in \{0,1,\star\}^n$ it holds that $F\restriction_\rho$ can be computed by a depth-2 formula Φ of width w and size m such that $F'\restriction_\rho$ is a refinement of Φ.*

The proof of Claim 23 relies on an elementary (and tedious) case analysis, so we defer it to Appendix B.

Two theorems from [GMR13]. For $\epsilon > 0$ and two Boolean functions F and F' over a domain $\mathfrak{D}$, we say that F and F' are ϵ-close if $\Pr_{x\in\mathfrak{D}}[F(x) = F'(x)] \geq 1-\epsilon$. We say that F' is an ϵ-refinement of F if F' is both a refinement of F, and ϵ-close to F. Similarly, we say that F' is an ϵ-lower-sandwiching refinement (resp., ϵ-upper-sandwiching refinement) of F if F' is both ϵ-close to F and a lower-sandwiching (resp., upper-sandwiching) refinement of F. Then, the main result of Gopalan, Meka, and Reingold [GMR13] can be stated as follows:

Theorem 24 *([GMR13, Thm 1.2]). Let $F : \{0,1\}^n \to \{0,1\}$ be a depth-2 formula of width w, and let $\beta > 0$. Then, there exist β-lower-sandwiching and β-upper-sandwiching refinements of F, denoted by $F^{\mathtt{low}}$ and $F^{\mathtt{up}}$, respectively, such that the size of $F^{\mathtt{low}}$ and of $F^{\mathtt{up}}$ is at most $m' = 2^{\tilde{O}(w)\cdot\log\log(1/\beta)}$, and their width is at most w.*

We will also need a pseudorandom generator construction from [GMR13]. In fact, we will rely on an assertion from the proof of their generator construction.

Theorem 25 *([GMR13, In the proof of Thm 3.1]). Let $F : \{0,1\}^n \to \{0,1\}$ be a depth-2 formula of width w, and let $\delta_0 > 0$. Then, every δ-almost t-wise independent distribution δ_0-fools F, where $\log(1/\delta) = O(w^2 \cdot \log^2(w) + w \cdot \log(w) \cdot \log(1/\delta_0))$ and $t = O(w^2 \cdot \log(w) + w \cdot \log(1/\delta_0))$.*

5.2.2 Width-dependent derandomizations of the switching lemma

In the proposition statements in this section, the letter n denotes the number of input bits for a formula, the number of clauses (i.e., size) is denoted by m, the width is denoted by w, and $\delta > 0$ is an error parameter (which will typically take the value $\delta = 1/\mathrm{poly}(n)$ in our applications). As a first step, we need to adapt the derandomized switching lemma of Trevisan and Xue [TX13] such that it will depend on the width of the depth-2 formula that we wish to "switch". Then, we will state and prove our new derandomized switching lemma, which is the main technical part in this section.

Proposition 26 *(an adaptation of the derandomized switching lemma of [TX13]). Let $m : \mathbb{N} \to \mathbb{N}$, let $w : \mathbb{N} \to \mathbb{N}$ such that $w(n) \leq O(\log(m(n)))$, and let $\delta : \mathbb{N} \to [0,1)$ such that $\delta(n) \leq$*

$2^{-O(w(n))}$. *Let* $\mathbf{r}$ *be a distribution over* $\{0,1\}^{O(\log(w))\cdot n}$ *that is* δ'*-almost* t'*-wise independent, where* $\log(1/\delta') = O(t') = \tilde{O}(w)\cdot\log(1/\delta)\cdot\log(m) + O(\log(n/\delta))$. *Then, for any depth-2 formula* $F:\{0,1\}^n \to \{0,1\}$ *of width* $w = w(n)$ *and size* $m = m(n)$*, with probability at least* $1-2\delta$ *(where* $\delta = \delta(n)$*) over choice of* $\rho \sim \mathbf{r}$ *it holds that:*

1. *The restricted formula* $F\restriction_\rho$ *can be computed by a decision tree of depth* $D = O(\log(1/\delta))$.
2. *The number of variables that are kept alive by* ρ *is at least* $\Omega(n/w)$.

In particular, a restriction $\rho \sim \mathbf{r}$ *can be sampled using a seed of length* $\tilde{O}(w)\cdot\log(1/\delta)\cdot\log(m) + O(\log(n/\delta))$.

Proof. Loosely speaking, the main lemma of Trevisan and Xue [TX13] reduces the task of finding a restriction that simplifies F to the task of "fooling" a large number of auxiliary CNFs. Going through their proof, we observe is that if F has width w, then each of the auxiliary CNFs also has width (roughly) w; that is, their proof can be adapted to show the following:

Lemma 27 *(a variation on [TX13, Lemma 7]). Let* $F:\{0,1\}^n \to \{0,1\}$ *be a depth-2 formula of size* m *and width* w*. For* $q \in \mathbb{N}$ *and* $p = 2^{-q}$*, let* $\rho \in \{0,1,\star\}^n$ *be a restriction that is chosen according to a distribution over* $\{0,1\}^{(q+1)\cdot n}$ *that* δ_0*-fools all CNFs of width* $w' = w\cdot(q+1)$*. Then, the probability that* $F\restriction_\rho$ *cannot be computed by a decision tree of depth* D *is at most* $2^{D+w+1}\cdot(5pw)^D + \delta_0\cdot 2^{(D+1)\cdot(2\cdot w+\log(m))}$.

The proof of Lemma 27 is a relatively straightforward adaptation of the original proof in [TX13], so we defer it to Appendix B. We will use the lemma with the parameters $p = 1/O(w)$ and $\delta_0 = 2^{-O(D\cdot(w+\log(m)))}$, in order to get the probability of error down to δ. Relying on Theorem 25, the auxiliary CNFs of width w' are δ_0-fooled by $\mathbf{r}$, [10] and therefore with probability $1-\delta$ it holds that $F\restriction_\rho$ can be computed by a decision tree of depth D.

The expected number of variables that the pseudorandom restriction leaves alive is $\Omega(n/w)$ (because the distribution on each block of $O(\log(w))$ bits in $\mathbf{r}$, which corresponds to a variable, is of statistical distance at most δ' from uniform, where $\delta' < 2^{-w}$). Since $\mathbf{r}$ is δ'-almost t'-wise independent, where $\delta' < 1/\mathrm{poly}(n/\delta)$ and $t' > O(\log(w))$, the blocks in $\mathbf{r}$ that correspond to each variable are $\frac{1}{\mathrm{poly}(n/\delta)}$-almost $O(1)$-wise independent. Relying on Fact 13, the probability that $\Omega(n/w)$ variables remain alive is at least $1-\delta$. ∎

We mention that the derandomized switching lemma of Goldreich and Wigderson [GW14, second step of the proof of Lemma 3.3] can also be adapted to depend on the width w of the formula that we want to "switch"; in this case, the required seed length is $\tilde{O}(w)\cdot 2^w\cdot\log(1/\delta)$, where δ is the probability of error (and the target depth of the decision tree is $D = O(\log(1/\delta))$). We provide the details in Appendix B. We now turn to state the new width-dependent derandomization of the switching lemma and prove it:

Proposition 28 *(a new width-dependent derandomization of the switching lemma). Let* $m:\mathbb{N}\to\mathbb{N}$*, let* $w:\mathbb{N}\to\mathbb{N}$ *such that* $w(n) \le O(\log(m(n)))$*, let* $\delta:\mathbb{N}\to[0,1)$*, and let* $\alpha:\mathbb{N}\to[0,1)$*. Let* $\delta' > 0$ *and* $t' \in \mathbb{N}$ *such that* $\log(1/\delta') = O(t') = \tilde{O}(w^2)\cdot\log(1/\delta)\cdot\log\log(m/\alpha\delta) +$

[10] This is because according to Theorem 25, CNFs of width w' are δ_0-fooled by any distribution that is δ''-almost t''-wise independent, where $t'' = O\left((w')^2\cdot\log(w') + w'\cdot\log(1/\delta_0)\right) = \tilde{O}(w)\cdot\log(1/\delta)\cdot\log(m)$ and $\log(1/\delta'') = O\left((w')^2\cdot\log^2(w') + w'\cdot\log(w')\cdot\log(1/\delta_0)\right) = \tilde{O}(w)\cdot\log(1/\delta)\cdot\log(m)$.

$\tilde{O}(w) \cdot \log(m/\alpha\delta) + O(\log(n/\delta))$. *Let* $\mathbf{y}$ *be a distribution over* $\{0,1\}^{O(\log(w))\cdot n}$ *that is* δ'*-almost* t'*-wise independent, and let* $\mathbf{z}$ *be a distribution over* $\{0,1\}^n$ *that is* δ'*-almost* t'*-wise independent. Finally, let* $\rho = \rho_{\mathbf{y},\mathbf{z}}$ *be a restriction that is chosen by using a sample from* $\mathbf{y}$ *to determine which variables are kept alive, and an independent sample from* $\mathbf{z}$ *to determine values for the fixed variables.*

Then, for any depth-2 formula $F : \{0,1\}^n \to \{0,1\}$ *of width* $w = w(n)$ *with* $m = m(n)$ *clauses, with probability at least* $1 - 4\delta$ *(where* $\delta = \delta(n)$*) over choice of* ρ *it holds that:*

1. *There exists a lower-sandwiching refinement* $F^{\mathtt{low}}$ *of* F *such that* $F^{\mathtt{low}}\restriction_\rho$ *and* $F\restriction_\rho$ *are* α*-close (i.e.,* $\Pr_{x\in\mathfrak{C}(\rho)}[F^{\mathtt{low}}(x) = F(x)] \geq 1-\alpha$*) and such that the restricted refinement* $F^{\mathtt{low}}\restriction_\rho$ *can be computed by a decision tree of depth* $D = O(\log(1/\delta))$*.*
2. *There exists an upper-sandwiching refinement* $F^{\mathtt{up}}$ *of* F *such that* $F^{\mathtt{up}}\restriction_\rho$ *and* $F\restriction_\rho$ *are* α*-close and such that* $F^{\mathtt{up}}\restriction_\rho$ *can be computed by a decision tree of depth* $D = O(\log(1/\delta))$*.*
3. *The number of variables that are kept alive by* ρ *is at least* $\Omega(n/w)$*.*

In particular, a restriction ρ *can be sampled using a seed of length* $\tilde{O}(w^2)\cdot\log(1/\delta)\cdot\log\log(m/\alpha\delta) + \tilde{O}(w)\cdot\log(m/\alpha\delta) + O(\log(n/\delta))$.

Note that when $m = \Theta(1/\delta) = \Theta(1/\alpha) = \mathrm{poly}(n)$, the seed length in Proposition 28 is $\tilde{O}(w^2\cdot\log(n))$. As in the overview in Section 2.2, our strategy in the proof of Proposition 28 will be as follows. Let $F^{\mathtt{low}}$ and $F^{\mathtt{up}}$ be the refinements of F from Theorem 24. Using the fact that $F^{\mathtt{low}}$ and $F^{\mathtt{up}}$ are of width w and of size $2^{\tilde{O}(w)\cdot\log\log(m/\alpha\delta)}$, we will rely on Proposition 26 to prove that, with high probability, both $F^{\mathtt{low}}\restriction_\rho$ and $F^{\mathtt{up}}\restriction_\rho$ simplify to depth-D decision trees. The main challenge will be to prove that with high probability it holds that $F^{\mathtt{low}}\restriction_\rho$ (resp., $F^{\mathtt{up}}\restriction_\rho$) and $F\restriction_\rho$ are α-close. The following lemma is the key one needed to establish the latter assertion, and after proving the lemma, we will use it to prove Proposition 28.

Lemma 29. *Let* $m : \mathbb{N} \to \mathbb{N}$*, let* $w : \mathbb{N} \to \mathbb{N}$ *such that* $w(n) \leq O(\log(m(n)))$*, and let* $\delta : \mathbb{N} \to [0,1)$*. Let* $F : \{0,1\}^n \to \{0,1\}$ *be a depth-2 formula of size* $m = m(n)$ *and width* $w = w(n)$*. For* $\alpha > 0$ *and* $\beta \leq \frac{\alpha^6\cdot(\delta/4)^4}{m^4\cdot\log^6(1/\delta)}$*, let* $F' : \{0,1\}^n \to \{0,1\}$ *be a* β*-refinement of* F*.*

Fix $I \subseteq [n]$*, and let* $\mathbf{z}$ *be a distribution over* $\{0,1\}^n$ *that* β*-fools all DNFs of width* w*. Let* $\rho = \rho_{I,\mathbf{z}} \in \{0,1,\star\}^n$ *be the restriction that is obtained by fixing values to the variables indexed by* $[n]\setminus I$ *according to the corresponding bits of* $\mathbf{z}$*. Then, with probability at least* $1-\delta$ *over choice of* $\mathbf{z}$ *it holds that* $F'\restriction_\rho$ *is an* α*-refinement of a depth-2 formula of size* m *and width* w *for* $F\restriction_\rho$*.*

Proof. We will prove the claim assuming that F is a DNF; if F is a CNF, then we can rely on Fact 21 to deduce that the assertion of the lemma holds for F if and only if it holds for the DNF $\neg F$. Also note that by Claim 23, for any $\rho \in \{0,1,\star\}^n$ it holds that $F'\restriction_\rho$ is a refinement of a depth-2 formula of size m and width w for $F\restriction_\rho$. Thus, we only need to prove that with probability at least $1-\delta$ it holds that $F'\restriction_\rho$ is α-close to $F\restriction_\rho$. Recall that $I \subseteq [n]$ is fixed throughout the proof; for brevity of notation, for any $z \in \{0,1\}^n$ denote $\rho_z = \rho_{I,z}$.

In high-level, the proof follows the overview that was presented in Section 2.2, and in particular relies on Lemma 15. We first define a set E of excellent restrictions, which are restrictions ρ such that $F'\restriction_\rho$ is $\sqrt{\beta}$-close to $F\restriction_\rho$, and show that almost all restrictions are excellent. We will then define a set B of bad restrictions, which are restrictions ρ such that $F'\restriction_\rho$ is not α-close to $F\restriction_\rho$. After defining E and B we will define the distribution $\mathbf{T}$ over tests that accepts, with high probability, every restriction in E, and rejects, with high probability, every restriction in B. Then, we will show that the residual tests $T \in \mathbf{T}$ are relatively "simple", in the sense that they can be computed by depth-3 circuits with a

specific structure (i.e., top AND gate and bottom fan-in w). And finally, we will show a hitting-set generator for the set of tests in the support of $\mathbf{T}$ that accept almost all of their input restrictions, and conclude the argument using Lemma 15.

Excellent restrictions and bad restrictions. For any $\rho \in \{0,1,\star\}^n$, let $\mathtt{err}(\rho) = \Pr_{x \in \mathfrak{C}(\rho)}[F'(x) \neq F(x)]$ be the fraction of inputs in $\mathfrak{C}(\rho)$ on which F and F' disagree. Our goal is to show that $\Pr_{z \sim \mathbf{z}}[\mathtt{err}(\rho_z) \leq \alpha] \geq 1 - \delta$. Consider the following two sets:

Definition 29.1. *(excellent and bad restrictions). Let* $E = \{z \in \{0,1\}^n : \mathtt{err}(\rho_z) \leq \sqrt{\beta}\}$ *be the set of* excellent *choices of restrictions, and let* $B = \{z \in \{0,1\}^n : \mathtt{err}(\rho_z) > \alpha\}$ *be the set of* bad *choices of restrictions.*

Since F' is β-close to F, a random restriction $\rho_{I,\mathbf{u}_n}$ is excellent with probability at least $1 - \sqrt{\beta}$. [11] We want to show that a pseudorandom restriction $\rho_{\mathbf{z}} = \rho_{I,\mathbf{z}}$ is not bad, with probability at least $1 - \delta$.

A distribution over simple tests. Let $t = O(\log(1/\delta)/\alpha)$. We now define a distribution $\mathbf{T}$ over tests $\{0,1\}^n \to \{0,1\}$, such that the random variable $\mathbf{T}(z)$ will essentially be the result of the following random test: Given $z \in \{0,1\}^n$, the test uniformly samples t inputs in $\mathfrak{C}(\rho_z)$, and accepts z if and only if F and F' agree on all the t inputs.

For $x \in \{0,1\}^{|I|}$ and $z \in \{0,1\}^n$, denote by $x\restriction_z \in \mathfrak{C}(\rho_z)$ the string that is obtained by fixing the variables indexed by I according to x, and the rest of the variables (i.e., the ones indexed by $[n] \setminus I$) according to the corresponding bits from z. For any $x \in \{0,1\}^{|I|}$, let $T_x : \{0,1\}^n \to \{0,1\}$ be the function such that $T_x(z) = 1$ if and only if $F'(x\restriction_z) = F(x\restriction_z)$. Also, for $\bar{x} = x^{(1)}, ..., x^{(t)} \in \{0,1\}^{t \cdot |I|}$, let $T_{\bar{x}}$ be the function $T_{\bar{x}}(z) = \wedge_{i=1}^{t} T_{x^{(i)}}(z)$. Finally, let $\mathbf{T}$ be the distribution over tests that is obtained by uniformly choosing $\bar{x} \in \{0,1\}^{t \cdot |I|}$ and outputing $T_{\bar{x}}$. Note that $\mathbf{T}(z)$ is indeed the result of uniformly sampling t inputs in $\mathfrak{C}(\rho_z)$, and accepting z if and only if F' and F agree on all the t sampled inputs.

By our choice of the parameter t, and since β is sufficiently small, the distribution $\mathbf{T}$ indeed distinguishes between E and B:

Fact 29.2. *For any* $z \in E$ *it holds that* $\Pr_{T \sim \mathbf{T}}[T(z) = 1] \geq (1 - \sqrt{\beta})^t \geq 1 - t \cdot \sqrt{\beta}$*, and for any* $z \in B$ *it holds that* $\Pr_{T \sim \mathbf{T}}[T(z) = 1] < (1 - \alpha)^t < \delta/3$.

For $\eta = \sqrt{t+1} \cdot \beta^{1/4}$, let $\mathbf{T}'$ be the set of tests $T_{\bar{x}} \in \mathbf{T}$ that accept at least $1 - \eta$ of their inputs (i.e., $\mathbf{T}' = \{T_{\bar{x}} : \Pr_{z \in \{0,1\}^n}[T_{\bar{x}}(z) = 1] \geq 1 - \eta\}$). We will abuse the notations $\mathbf{T}$ and $\mathbf{T}'$, by using them both to denote sets and to denote the uniform distribution over the corresponding set. To see that the set $\mathbf{T}'$ is dense in $\mathbf{T}$, note that

$$\begin{aligned}\mathbb{E}_{T_{\bar{x}} \in \mathbf{T}}\left[\Pr_{z \in \{0,1\}^n}[T_{\bar{x}}(z) = 1]\right] &= \mathbb{E}_{z \in \{0,1\}^n}\left[\Pr_{T_{\bar{x}} \in \mathbf{T}}[T_{\bar{x}}(z) = 1]\right] \\ &\geq \Pr_{z \in \{0,1\}^n}[z \in E] \cdot \min_{z \in E}\left\{\Pr_{T_{\bar{x}} \in \mathbf{T}}[T_{\bar{x}}(z) = 1]\right\},\end{aligned}$$

which is at least $1 - \sqrt{\beta} - t \cdot \sqrt{\beta} = 1 - \eta^2$. Therefore, the probability over $T_{\bar{x}} \in \mathbf{T}$ that $T_{\bar{x}}$ rejects more than η of its input restrictions is at most η.

A hitting-set generator for $\mathbf{T}'$. Towards designing a hitting-set generator with high density for every $T_{\bar{x}} \in \mathbf{T}'$, we first show that each $T_{\bar{x}} \in \mathbf{T}$ can be computed by a depth-3 circuit

[11] Because $\mathbb{E}[\mathtt{err}(\rho_{I,\mathbf{u}_n})] = \Pr_{x \in \{0,1\}^n}[F'(x) \neq F(x)] \leq \beta$, which implies that $\Pr[\mathtt{err}(\rho_{I,\mathbf{u}_n}) > \sqrt{\beta}] < \sqrt{\beta}$.

with a top AND gate and small bottom fan-in. To do so, we first show that for a single $x \in \{0,1\}^{|I|}$ (rather than for $\bar{x} = x^{(1)}, ..., x^{(t)}$) it holds that T_x can be computed by a depth-3 circuit with a top AND gate and small bottom fan-in.

Claim 29.3. *For every fixed $x \in \{0,1\}^{|I|}$, the function $T_x : \{0,1\}^n \to \{0,1\}$ can be computed by a depth-3 circuit with a top AND gate of fan-in at most m such that the bottom fan-in of the circuit is at most w.*

Proof. Denote the number of refinement steps that were applied to F to obtain F' by $k \leq m$. For any $i \in [k]$, let $F^{(i)}$ be the formula in the beginning of the i^{th} refinement step in the transformation of F to F', and let $F^{(k+1)} = F'$. Note that $T_x(z) = 1$ if and only if for every $i \in [k]$ it holds that $F^{(i)}(x|_z) = F^{(i+1)}(x|_z)$ (one direction is immediate, whereas the other direction follows by the monotonicity of the sequence $F^{(1)}(x|_z), ..., F^{(k+1)}(x|_z)$ [12]).

For every $i \in [k]$, let $T_{x,i}$ be the function such that $T_{x,i}(z) = 1$ if and only if $F^{(i)}(x|_z) = F^{(i+1)}(x|_z)$. We will show that each $T_{x,i}$ can be computed by a DNF of width w. This claim suffices to conclude the proof, since it implies that T_x can be computed by a circuit with a top AND gate that is connected to $k \leq m$ DNFs of width w. To prove the claim, fix $i \in [k]$, and let us conduct a case analysis:

- If the i^{th} refinement step was a clean-up step, then $T_{x,i} \equiv 1$.
- If the i^{th} step was a removal step, then let $f^{(i)}$ be the clause that was removed from $F^{(i)}$ in the i^{th} step, and let $F^{(i+1)} = \left(F^{(i)} \setminus f^{(i)}\right)$ be the formula that is obtained by dropping the clause $f^{(i)}$ from $F^{(i)}$. Note that $F^{(i+1)}(x|_z) = F^{(i)}(x|_z)$ if and only if either $f^{(i)}(x|_z) = 0$ or $\left(F^{(i)} \setminus f^{(i)}\right)(x|_z) = 1$. The latter event is a disjunction of at most m events (because $\left(F^{(i)} \setminus f^{(i)}\right)$ is a DNF of size at most $m-1$), each of which depends on the values of at most w bits in $x|_z$. Thus, each of the (at most m) events depends on at most w bits in z, and can therefore be decided by a DNF of width w. It follows that $T_{x,i}$ is the disjunction of width-w DNFs, which is a width-w DNF.
- If the i^{th} refinement step in the transformation of F to F' was a merging step, denote the $u \geq 2$ clauses that were removed from $F^{(i)}$ in the step by $f_1^{(i)}, ..., f_u^{(i)}$, and the new clause that was added in their stead by $h^{(i)}$. Note that $F^{(i+1)}(x|_z) = F^{(i)}(x|_z)$ if and only if either $h^{(i)}(x|_z) = 0$ or $F^{(i)}(x|_z) = 1$. This is a disjunction of at most $m+1$ events, each of which depends on at most w bits in $x|_z$ (and thus on at most w bits in z). Thus, in this case too it holds that $T_{x,i}$ can be computed by a DNF of width w. □

For a fixed $\bar{x} = x^{(1)}, ..., x^{(t)} \in \{0,1\}^{t \cdot |I|}$, we can compute $T_{\bar{x}}$ by taking a conjunction of t circuits for the corresponding T_x's (i.e., $\wedge_{i \in [t]} T_{x^{(i)}}$), which is a depth-3 circuit with bottom fan-in at most w and top fan-in at most $t \cdot m$. We are now ready to prove that $\mathbf{z}$ is a hitting-set generator with density $1 - \delta/3$ for every $T_{\bar{x}} \in \mathbf{T}'$:

Claim 29.4. *For every $T_{\bar{x}} \in \mathbf{T}'$ it holds that $\Pr[T(\mathbf{z}) = 1] \geq 1 - \delta/3$.*

Proof. Fix $T_{\bar{x}} \in \mathbf{T}'$, and recall that by the definition of $\mathbf{T}'$ it holds that $T_{\bar{x}}$ accepts at least $1 - \eta$ of its inputs. Thus, each of the DNFs in the middle layer of the circuit that we

[12] If F' was obtained by merging steps and clean-up steps, then $F^{(1)}(x|_z) \leq ... \leq F^{(k+1)}(x|_z)$, whereas if F' was obtained by removal steps and clean-up steps, then $F^{(1)}(x|_z) \geq ... \geq F^{(k+1)}(x|_z)$.

constructed for $T_{\bar{x}}$ accepts $1-\eta$ of the inputs. It follows that when using the distribution $\mathbf{z}$, which is β-pseudorandom for such DNFs, each of these DNFs accepts with probability at least $1-\eta-\beta$. By a union-bound, it follows that

$$\begin{aligned}\Pr_{z\sim\mathbf{z}}[T_{\bar{x}}(z)=1] &\geq 1-(\eta+\beta)\cdot(t\cdot m)\\ &> 1-(2\cdot t\cdot m)\cdot\eta\\ &= 1-O\left((\log(1/\delta)/\alpha)^{3/2}\cdot m\cdot\beta^{1/4}\right)\,,\end{aligned}$$

which is larger than $1-\delta/3$ by the hypothesis that β is sufficiently small. □

Invoking Lemma 15. We now conclude the argument by invoking Lemma 15. Let E be as in Definition 29.1, and let $G=\{0,1\}^n\setminus B$; recall that for $\epsilon_1=\sqrt{\beta}$ it holds that $\Pr_{z\in\{0,1\}^n}[z\in E]\geq 1-\epsilon_1$. Denoting $\epsilon_2=t\cdot\sqrt{\beta}$ and $\epsilon_3=\delta/3$, according to Fact 29.2, for any $z\in E$ it holds that $\Pr_{T_{\bar{x}}\sim\mathbf{T}}[T_{\bar{x}}(z)=1]\geq 1-\epsilon_2$ and for any $z\notin G$ it holds that $\Pr_{T_{\bar{x}}\sim\mathbf{T}}[T_{\bar{x}}(z)=0]\geq 1-\epsilon_3$.

Finally, for $\epsilon_4=\eta$ it holds that the set $\mathbf{T}'$ is of density at least $1-\epsilon_4$ in $\mathbf{T}$, and for every $T_{\bar{x}}\in\mathbf{T}'$, by Claim 29.4 it holds that $\mathbf{z}$ fools $T_{\bar{x}}$ with error at most $\epsilon_5=\delta/3$ (because $\Pr_{z\in\{0,1\}^n}[T_{\bar{x}}(z)=1]\geq 1-\eta\geq 1-\delta/3$ and $\Pr_{z\sim\mathbf{z}}[T_{\bar{x}}(z)=1]\geq 1-\delta/3$). Relying on Lemma 15, the probability that $\mathbf{z}\notin G$ is at most

$$\sqrt{\beta}+t\cdot\sqrt{\beta}+\delta/3+2\cdot\eta+\delta/3=2\delta/3+\eta^2+2\cdot\eta<\delta\,,$$

where the inequality relied on the fact that β (and hence also η) is sufficiently small. ■

We are now ready to prove Proposition 28.

Proof of Proposition 28. Let $F:\{0,1\}^n\to\{0,1\}$ be a depth-2 formula of width w and size m. Let $F^{\mathtt{low}}:\{0,1\}^n\to\{0,1\}$ and $F^{\mathtt{up}}:\{0,1\}^n\to\{0,1\}$ be the β-lower-sandwiching and the β-upper-sandwiching formulas for F from Theorem 24, respectively, where $\beta=\frac{\alpha^6\cdot(\delta/4)^4}{m^4\cdot\log^6(1/\delta)}$. Note that the width of $F^{\mathtt{low}}$ and of $F^{\mathtt{up}}$ is at most w, and that their size is at most $2^{\tilde{O}(w)\cdot\log\log(m/\alpha\delta)}$.

According to Fact 14, the distribution of strings $\mathbf{r}$ over $\{0,1\}^{O(\log(w))\cdot n}$, which is obtained by combining $\mathbf{y}$ and $\mathbf{z}$ and represents the pseudorandom restriction $\rho=\rho_{\mathbf{y},\mathbf{z}}$, is $(2\cdot\delta')$-almost t'-wise independent. Hence, relying on Proposition 26, with probability at least $1-2\delta$ it holds both that $F^{\mathtt{low}}\restriction_\rho$ and $F^{\mathtt{up}}\restriction_\rho$ can be computed by decision trees of depth D, and that ρ keeps at least $\Omega(n/w)$ variables alive.

According to Theorem 25, all DNFs of width w are β-fooled by the distribution $\mathbf{z}$. [13] Therefore, relying on Lemma 29, for any fixed choice of $y\sim\mathbf{y}$, with probability at least $1-2\delta$ over $z\sim\mathbf{z}$ it holds that both $F^{\mathtt{low}}\restriction_\rho$ and $F^{\mathtt{up}}\restriction_\rho$ are α-close to $F\restriction_\rho$. Thus, the probability over choice of both $\mathbf{y}$ and $\mathbf{z}$ that $F^{\mathtt{low}}\restriction_\rho$ and $F^{\mathtt{up}}\restriction_\rho$ are α-close to $F\restriction_\rho$ is at least $1-2\delta$. ■

[13] Theorem 25 requires that the distribution $\mathbf{z}$ will be δ''-almost t''-wise independent, where $t''=O(w^2\cdot\log(w)+w\cdot\log(1/\beta))=\tilde{O}(w)\cdot\log(m/\alpha\delta)<t'$ and $\log(1/\delta'')=O(w^2\cdot\log^2(w)+w\cdot\log(w)\cdot\log(1/\beta))=\tilde{O}(w)\cdot\log(m/\alpha\delta)<\log(1/\delta')$.

5.2.3 Proofs of Theorems 2 and 3

We are now ready to prove Theorem 3. Recall that Theorem 3 asserts the existence of a hitting-set generator that is parametrized by a parameter $t > 0$.

Theorem 30 *(Theorem 3, restated). Let $d \geq 2$, let $m : \mathbb{N} \to \mathbb{N}$ such that $m(n) \leq \text{poly}(n)$, and let $t : \mathbb{N} \to \mathbb{N}$ such that $c_0 \leq t(n) \leq 2 \cdot \log(m(n))$, where c_0 is a sufficiently large constant. For every $n \in \mathbb{N}$, let $\mathcal{C}_n$ be the class of circuits $C : \{0,1\}^n \to \{0,1\}$ of size $m = m(n)$ and of depth at most d that accept all but at most $B(n)$ of their inputs, where $\log(B(n)) = \Omega\left(n^{1-1/\Omega(t)}/t^{d-2}\right)$ and $t = t(n)$. Then, there exists a hitting-set generator for $\mathcal{C} = \cup_{n\in\mathbb{N}}\mathcal{C}_n$ with seed length $\ell = \ell(n) = \tilde{O}\left(t^2 \cdot \log(n)\right)$.*

Theorem 2 follows as a corollary of Theorem 30, by using the specific parameter value $t = 2 \cdot \log(m)$, in which case $B(n) = 2^{\Omega\left(n/\log^{d-2}(n)\right)}$ and the seed length is $\tilde{O}\left(\log^3(n)\right)$.

Proof. Given input 1^n and a random seed in $\{0,1\}^\ell$, the hitting-set generator works in two steps. In the first step, the generator outputs a restriction $\bar{\rho} \in \{0,1,\star\}^n$ such that for any circuit C over n input bits of depth d and size $m = m(n)$, with high probability it holds that there exists a depth-2 formula C' of size $\text{poly}(n)$ and width t that is both $(1/2)$-close to $C|_{\bar{\rho}}$ and lower-sandwiching for $C|_{\bar{\rho}}$. Moreover, with high probability the restriction $\bar{\rho}$ keeps at least $\log(B(n)) + 2$ variables alive.

Since the subcube $\mathfrak{C}(\bar{\rho})$ contains at least $4 \cdot B(n)$ inputs, the acceptance probability of $C|_{\bar{\rho}}$ is at least $3/4$. Hence, the acceptance probability of C' is at least $1/4$ (because C' is $(1/2)$-close to $C|_{\bar{\rho}}$), and every satisfying input for C' is also satisfying for C (because C' is lower-sandwiching for $C|_{\bar{\rho}}$). Thus, in the second step, we use a pseudorandom generator for depth-2 circuits to "fool" C': The pseudorandom generator outputs a satisfying input for C' in $\mathfrak{C}(\bar{\rho})$ with positive probability, and any such input yields a satisfying input for C.

Parameter settings. Let $\epsilon > 0$ be a sufficiently small constant, and let $\delta = (\epsilon/m)$. Let $D = O(\log(1/\delta)) > 2 \cdot \log(2m/\delta)$, and let $m' = m \cdot 2^D = \text{poly}(n)$. Let $\beta = \left(\frac{\delta}{2dm}\right)^{10^{2d}}$; we will use β as the approximation parameter whenever using Theorem 24. Let $\delta' > 0$ and $t' \in \mathbb{N}$ such that $\log(1/\delta') = O(t') = \tilde{O}\left(t^2 \cdot \log(n)\right)$.

The pseudorandom choice of restrictions. The algorithm that we will describe below constructs a sequence of restrictions. We mention in advance that when describing the algorithm, whenever we will say that we choose a restriction with a parameter $p = 2^{-q}$, the pseudorandom choice of restriction is the following:

- Let $\mathbf{y}$ be a distribution over $\{0,1\}^{\log(1/p)\cdot n}$ that is δ'-almost t'-wise independent.
- Let $\mathbf{z}$ be a distribution over $\{0,1\}^n$ that is δ'-almost t'-wise independent.
- The restriction $\rho = \rho_{\mathbf{y},\mathbf{z}}$ is chosen by sampling $y \sim \mathbf{y}$ in order to determine which variables are kept alive, and independently sampling $z \sim \mathbf{z}$ in order to determine values for the fixed variables.

Note that such a restriction keeps every variable alive with probability approximately p (i.e., with probability $p \pm \delta'$). The above process yields a distribution $\mathbf{r}$ over $\{0,1\}^{(\log(1/p)+1)\cdot n}$, which is obtained by combining $\mathbf{y}$ and $\mathbf{z}$ as detailed in the beginning of Section 5.2.1; according to Fact 14, the distribution $\mathbf{r}$ is $(2 \cdot \delta')$-almost t'-wise independent.

The first step. The generator constructs the restriction $\bar{\rho}$ as the composition of $2d-2$ restrictions $\bar{\rho} = \rho^{(2d-3)} \circ \rho^{(2d-4)} \circ ... \circ \rho^{(1)} \circ \rho^{(0)}$. The initial restriction $\rho^{(0)}$ is chosen with parameter $p = 1/O(1)$, and with probability $1-\epsilon$ it reduces the bottom fan-in of the circuit to $D = O(\log(1/\delta))$. [14] The next $2 \cdot (d-2)$ restrictions are applied in $d-2$ iterations. Loosely speaking, in each iteration, we apply a restriction that reduces the bottom fan-in to t, then define an approximating circuit (by replacing the formulas in the next-to-bottom layer, which have small width at this point, with small lower-sandwiching refinements, using Theorem 24), and finally apply a second restriction in order to "switch" the formulas in the next-to-bottom layer of the approximating circuit, and reduce the depth of the circuit.

Let $C^{(0)} = C\restriction_{\rho^{(0)}}$ be the circuit in the beginning of the first iteration, and note that $C^{(0)}$ is of depth d, size at most $m < m'$, and bottom fan-in at most D. For $i \in [d-2]$, let us describe the i^{th} iteration. Assuming all previous iterations were successful, in the beginning of the i^{th} iteration we start with a circuit $C^{(i-1)}$ of depth at most $d-(i-1)$, bottom fan-in at most D, and with at most $m' = m \cdot 2^D$ gates in its bottom layer. We will produce two restrictions, denoted $\rho^{(2i-1)}$ and $\rho^{(2i)}$, and define a circuit $C^{(i)}$ whose domain is $\mathfrak{C}(\rho^{(2i)} \circ \rho^{(2i-1)} \circ ... \circ \rho^{(0)})$ such that with probability $1 - O(\epsilon)$ it holds that $C^{(i)}$ is of depth at most $d-i$, bottom fan-in D, and the number of gates in its bottom layer is at most m'. (After we finish the description of a single iteration, we will also prove that for any $i \in [d-2]$ it holds that $C^{(i)}\restriction_{\bar{\rho}}$ is close to $C^{(i-1)}\restriction_{\bar{\rho}}$; see Claim 30.2 below.)

The first restriction in iteration i, denoted $\rho^{(2i-1)}$, is chosen with the parameter $p = \left(\epsilon / \left(m \cdot 2^{2D+1}\right)\right)^{1/t} = n^{-1/\Omega(t)}$. We now show that with probability at least $1 - O(\epsilon)$ the bottom fan-in of the circuit $C^{(i-1)}\restriction_{\rho^{(2i-1)}}$ is less than t. To do so, first note the following:

Claim 30.1. *Let S be a fixed set of at most D variables. Then, with probability at least $1 - \epsilon/m'$ it holds that less than t variables in S are kept alive by $\rho^{(2i-1)}$.*

Proof. Recall that the restriction $\rho^{(2i-1)}$ is chosen such that the distribution $\mathbf{y}$ over $\{0,1\}^{\log(1/p) \cdot n}$, which determines which variables will be kept alive, is δ'-almost t'-wise independent. We will only need the fact that the blocks of size $\lceil \log(1/p) \rceil$ in $\mathbf{y}$ are (p^t)-almost t-wise independent; this holds because $t \cdot \lceil \log(1/p) \rceil < O(\log(m/\epsilon)) < t'$, and $\delta' < p^t = 1/\mathrm{poly}(n)$.

For any fixed set of t variables in S, the probability that all variables in the set remain alive after applying a uniformly-chosen restriction with the parameter p is p^t. Since the blocks of size $\lceil \log(1/p) \rceil$ in $\mathbf{y}$ are (p^t)-almost t-wise independent, the probability that $\rho^{(2i-1)}$ keeps all t variables alive is at most $2 \cdot (p^t)$. Thus, the probability that $\rho^{(2i-1)}$ keeps t variables in S alive is at most $\binom{|S|}{t} \cdot 2 \cdot p^t < 2^{D+1} \cdot p^t < \epsilon/m'$. □

Recall that the number of gates in the bottom layer of $C^{(i-1)}$ is at most m', and that each of them is of fan-in at most D. Using Claim 30.1 and a union-bound, with probability at least $1-\epsilon$ it holds that the bottom fan-in of $C^{(i-1)}\restriction_{\rho^{(2i-1)}}$ is less than t.

[14] To see that such a restriction indeed reduces the bottom fan-in, fix a gate in the bottom layer of fan-in more than $2 \cdot \log(2m/\epsilon)$. The probability under a uniformly-chosen restriction with $p = 1/4$ that none of the lexicographically-first $2 \cdot \log(2m/\epsilon)$ variables feeding into the gate is fixed to a satisfying value is $\left(\frac{1+p}{2}\right)^{2 \cdot \log(2m/\epsilon)} < \epsilon/2m$. Since this event depends only on the values that the restriction assigns to $2 \cdot \log(2m/\epsilon)$ variables, and the value for each variable depends on $\log(1/p) = O(1)$ bits, the event depends on at most $O(\log(m/\epsilon))$ bits of the restriction. Thus, the event happens with probability at most ϵ/m when the restriction is chosen from a $1/\mathrm{poly}(m/\epsilon)$-biased set.

Assuming that the bottom fan-in of $C^{(i-1)}\restriction_{\rho^{(2i-1)}}$ is indeed less than t, we now use Theorem 24 to replace each formula F in the next-to-bottom layer of $C^{(i-1)}\restriction_{\rho^{(2i-1)}}$ with a β-lower-sandwiching refinement $F^{\mathtt{low}}$ such that the size of $F^{\mathtt{low}}$ is at most $2^{\tilde{O}(t)\cdot\log\log(1/\beta)}$. Let $\widetilde{C^{(i-1)}\restriction_{\rho^{(2i-1)}}}$ be the circuit that is obtained by replacing all the formulas in the next-to-bottom layer of $C^{(i-1)}\restriction_{\rho^{(2i-1)}}$ in this manner.

The final step in the i^{th} iteration is to apply a restriction $\rho^{(2i)}$ with parameter $p = 1/O(t)$ that is intended to simplify each formula $F^{\mathtt{low}}$ in the next-to-bottom layer of $\widetilde{C^{(i-1)}\restriction_{\rho^{(2i-1)}}}$ to a decision tree of depth at most D. Let $C^{(i)} = \left(\widetilde{C^{(i-1)}\restriction_{\rho^{(2i-1)}}}\right)\restriction_{\rho^{(2i)}}$. Relying on Proposition 26, the restriction $\rho^{(2i)}$ is successful with probability at least $1-O(\epsilon)$, and in this case the circuit $C^{(i)}$ is of depth at most $d-i$, and the bottom layer of $C^{(i)}$ has at most $m' = m\cdot 2^D$ gates, each of fan-in at most D. [15]

We now apply one final restriction $\rho^{(2d-3)}$, with parameter $p = \left(\epsilon/\left(m\cdot 2^{2D+1}\right)\right)^{1/t}$, in order to reduce the bottom fan-in of $C^{(d-2)}$ to t. Using Claim 30.1 and a union-bound, with probability at least $1-O(\epsilon)$ it holds that the width of $C^{(d-2)}\restriction_{\rho^{(2d-3)}}$ is at most t. For convenience, in Table 1 we summarize the restrictions that were applied in the first step.

	Value of p	**Goal of the restriction**
$\rho^{(0)}$	$1/O(1)$	Reduce the bottom fan-in to D
$i = 1,...,d-2$:		
$\rho^{(2i-1)}$	$n^{-1/\Omega(t)}$	Reduce the bottom fan-in to t
$\rho^{(2i)}$	$1/O(t)$	"Switch" the width-t formulas at the next-to-bottom-layer
$\rho^{(2d-3)}$	$n^{-1/\Omega(t)}$	Reduce the bottom fan-in to t

Table 1: Summary of the restrictions that are applied in the first step.

Let $C^{(d-1)} = C^{(d-2)}\restriction_{\rho^{(2d-3)}}$, and recall that $\bar{\rho} = \rho^{(2d-3)}\circ\rho^{(2d-2)}\circ...\circ\rho^{(0)}$. The above shows that if all the iterations are successful, then $C^{(d-1)}$ is a formula of depth 2, size at most m', and width t. Also note that if all the iterations are successful, then $\widetilde{C^{(d-1)}}$ is lower-sandwiching for $C\restriction_{\bar{\rho}}$. This is because for every $i\in[d-2]$ it holds that $\widetilde{C^{(i-1)}\restriction_{\rho^{(2i-1)}}}$ is lower-sandwiching for $C^{(i-1)}\restriction_{\rho^{(2i-1)}}$ (since $\widetilde{C^{(i-1)}\restriction_{\rho^{(2i-1)}}}$ is obtained by replacing every formula F in the next-to-bottom-layer of $C^{(i-1)}\restriction_{\rho^{(2i-1)}}$ with a lower-sandwiching refinement $F^{\mathtt{low}}$), which implies that $C^{(i)}\restriction_{\bar{\rho}} = \left(\widetilde{C^{(i-1)}\restriction_{\rho^{(2i-1)}}}\right)\restriction_{\bar{\rho}}$ is lower-sandwiching for $C^{(i-1)}\restriction_{\bar{\rho}}$.

The main thing that is left to prove in the analysis of the first step is that with probability at least $1-O(\epsilon)$ it holds that $C^{(d-1)}$ is $(1/2)$-close $C\restriction_{\bar{\rho}}$. To do so, we will show that with

[15] Specifically, we rely on Proposition 26 with width parameter t, error parameter δ, size parameter $2^{\tilde{O}(t)\cdot\log\log(1/\beta)}$, and depth bound D for the decision trees. Proposition 26 requires that the distribution $\mathbf{r}$ of restrictions will be δ''-almost t''-wise independent, where $\log(1/\delta'') = O(t'') = \tilde{O}(t^2)\cdot\log(1/\delta)\cdot\log\log(1/\beta) = \tilde{O}(t^2\cdot\log(n))$. The latter holds by our choice of δ' and t'.

probability at least $1 - O(\epsilon)$, for every $i \in [d-2]$ it holds that $C^{(i-1)}\restriction_{\bar{\rho}}$ is $(1/2d)$-close to $C^{(i)}\restriction_{\bar{\rho}}$. Assuming that the latter holds, we can deduce that $C\restriction_{\bar{\rho}} = C^{(0)}\restriction_{\bar{\rho}}$ is $1/2$-close to $C^{(d-1)} = C^{(d-2)}\restriction_{\bar{\rho}}$. Thus, it suffices to prove the following claim:

Claim 30.2. *For any $i \in [d-2]$, with probability at least $1 - O(\epsilon)$ it holds that $C^{(i)}\restriction_{\bar{\rho}}$ is $(1/2d)$-close to $C^{(i-1)}\restriction_{\bar{\rho}}$.*

Proof. Let $i \in [d-2]$, let F be a formula in the next-to-bottom layer of $C^{(i-1)}\restriction_{\rho^{(2i-1)}}$, and let $F^{\mathtt{low}}$ be a β-refinement of F. We will prove that with probability $1 - O(\delta)$ it holds that $F^{\mathtt{low}}\restriction_{\bar{\rho}}$ is $(1/2dm)$-close to $F\restriction_{\bar{\rho}}$. This suffices to prove Claim 30.2, since by a union-bound over m formulas it follows that with probability at least $1 - O(\epsilon)$ it holds that the circuit $\left(\widetilde{C^{(i-1)}\restriction_{\rho^{(2i-1)}}}\right)\restriction_{\bar{\rho}} = C^{(i)}\restriction_{\bar{\rho}}$ is $(1/2d)$-close to $\left(C^{(i-1)}\restriction_{\rho^{(2i-1)}}\right)\restriction_{\bar{\rho}} = C^{(i-1)}\restriction_{\bar{\rho}}$.

For every $j \in \{2i, ..., 2d-3\}$, let $\rho^{(2i,...,j)}$ be the composed restriction $\rho^{(2i,...,j)} = \rho^{(j)} \circ ... \circ \rho^{(2i)}$, and let $\beta_j = (\delta/2dm)^{10^{2d-3-j}}$. We will prove the following statement: For every $j \in \{2i, ..., 2d-3\}$, with probability at least $1 - O(\delta)$ it holds that $F^{\mathtt{low}}\restriction_{\rho^{(2i,...,j)}}$ is a β_j-refinement of a depth-2 formula of size m' and width t for $F\restriction_{\rho^{(2i,...,j)}}$. Invoking this statement with $j = 2d-3$, we can deduce that with probability at least $1 - O(\delta)$ it holds that $F^{\mathtt{low}}\restriction_{\bar{\rho}}$ is β_{2d-3}-close to $F\restriction_{\bar{\rho}}$, where $\beta_{2d-3} < 1/2dm$.

We prove the aforementioned statement by induction on j. For the base case $j = 2i$, we start with a formula F of size m' and width t, and a β-refinement $F^{\mathtt{low}}$ of F, where $\beta < \beta_0 \leq \beta_{j-1}$. Now, $\rho^{(j)}$ is chosen according to a distribution such that for every fixed choice of variables to keep alive (i.e., every fixed $y \sim \mathbf{y}$), the choice of values for the fixed variables (i.e., $z \sim \mathbf{z}$) is δ'-almost t'-wise independent. Relying on Theorem 25 and on our choice of δ' and t', the distribution distribution $\mathbf{z}$ β-fools all DNFs of width w. We can therefore rely on Lemma 29 to deduce that with probability at least $1 - O(\delta)$ it holds that $F^{\mathtt{low}}\restriction_{\rho^{(j)}}$ is a β_j-refinement of $F\restriction_{\rho^{(j)}}$. [16]

The induction step, for $j \geq 2i+1$, is very similar to the base case. By the induction hypothesis, with probability at least $1 - O(\delta)$ it holds that $F^{\mathtt{low}}\restriction_{\rho^{(2i,...,j-1)}}$ is a (β_{j-1})-refinement of a size m' and width w' depth-2 formula for $F\restriction_{\rho^{(2i,...,j-1)}}$. We can then use Theorem 25 and Lemma 29 similarly to the base case. □

To conclude the analysis of the first step, note that with probability at least $1 - O(\epsilon)$ it holds that at least $\log(B(n)) + 2 = \Omega\left(n^{1-1/\Omega(t)}/t^{d-2}\right)$ variables remain alive. To see that this is the case, recall that $\bar{\rho}$ is comprised of one restriction with parameter $p_0 = 1/O(1)$, and $d-1$ restrictions with parameter $p_1 = n^{-1/\Omega(t)}$, and $d-2$ restrictions with parameter $p_2 = 1/O(t)$. Let $\bar{p} = p_0 \cdot p_1^{d-1} \cdot p_2^{d-2} \cdot n$, and note that $\bar{p} = \Omega\left(n^{1-1/\Omega(t)}/t^{d-2}\right)$.

The expected number of living variables under $\bar{\rho}$ is $\Theta(\bar{p})$ (because in each restriction with parameter p, every variable is kept alive with probability $p \pm O(\delta') \in p \pm (p/2)$). Since all the choices of variables to keep alive are according to distributions that are δ'-almost t'-wise independent, we can use Fact 13 to deduce that with probability at least $1 - O(\epsilon)$ it

[16] We invoke Lemma 29 with width parameter t, size bound m', and error parameter δ. We know that $F^{\mathtt{low}}$ is a β_{j-1}-refinement of F, and we want to deduce that with probability at least $1 - O(\delta)$ it holds that $F^{\mathtt{low}}\restriction_{\rho^{(j)}}$ is an α-refinement of $F\restriction_{\rho^{(j)}}$, where $\alpha = \beta^j$. The lemma requires that the distribution $\mathbf{z}$ will (β_{j-1})-fool all DNFs of width t, and that $\beta_{j-1} \leq \frac{\beta_j^6 \cdot (\delta/4)^4}{m^4 \cdot \log^6(1/\delta)}$, both of which indeed hold.

holds that at least $\Omega(\bar{p}) = \Omega\left(n^{1-1/\Omega(t)}/t^{d-2}\right) > \log(B(n)) + 2$ variables remain alive after the first step. (When using Fact 13, we relied on the fact that t is larger than a sufficiently large constant c_0 to deduce that $n^{1-1/\Omega(t)}/t^{d-2} > n^{\Omega(1)}$).

The second step. We now invoke the pseudorandom generator from Theorem 25 for depth-2 circuits of width t, instantiated with error parameter $1/8$, and output the string that the generator outputs, completed to a string of length n according to $\bar{\rho}$. The generator requires a seed of length $O(t^2 \cdot \log^2(t)) = \tilde{O}(t^2)$.

Let us now prove this yields a satisfying input for C, with positive probability. If the first step was successful, then $\bar{\rho}$ kept more than $\log(B(n)) + 2$ live variables, and hence the acceptance probability of $C\restriction_{\bar{\rho}}$ is at least $3/4$. Since $C^{(d-1)}$ is $1/2$-close to $C\restriction_{\bar{\rho}}$, it follows that $\Pr_{x \in \mathfrak{C}(\bar{\rho})}[C^{(d-1)}(x) = 1] \geq 1/4$. Thus, the generator outputs a satisfying input for $C^{(d-1)}$, with positive probability, and this input (when completed to a string of length n according to $\bar{\rho}$) is satisfying for C, because $C^{(d-1)}$ is lower-sandwiching for $C\restriction_{\bar{\rho}}$. ■

6 Constant-depth circuits with parity gates

In this section we prove the claims made in Section 1.3: In Section 6.1 we prove Theorem 5, and in Section 6.2 we prove Theorem 6.

6.1 Proof of Theorem 5

The proof is similar to the proof of Theorem 1, and 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 $C : \{0,1\}^n \to \{0,1\}$ be a CNF that accepts most of its inputs. For $n' = n^{(1/c)+1}$ and $s = O(\log(n))$, let $E : \{0,1\}^{n'} \times \{0,1\}^s \to \{0,1\}^n$ be Trevisan's extractor instantiated for min-entropy $(n')^c = n^{1+\Omega(1)}$ and error parameter $1/4$. We construct a circuit $C' : \{0,1\}^{n'} \to \{0,1\}$ 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, $C'(x) = \vee_{z \in \{0,1\}^s} C(E(x,z))$.

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 $2^{(n')^c}$. 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 $\{E(x,z)\}_{z \in \{0,1\}^s}$.

6.2 Proof of Theorem 6

The current section is organized as follows. In Section 6.2.1 we present two algorithmic tools that will be used in the proof: An adaptation of the approach of Chaudhuri and Radhakrishnan [CR96] to the setting of $\oplus \wedge \oplus$ circuits, 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 6.

We rely on the notion of *affine restrictions*. A restriction of a circuit $C : \{0,1\}^n \to \{0,1\}$ 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 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.

6.2.1 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 31 *(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 32 *(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 $C|_W \not\equiv 0$, 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 $C(y) = 1$. ∎

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

We prove the first item of Theorem 6 by invoking the whitebox algorithm from Proposition 31 with appropriate parameters $d_\wedge, d_\oplus = O(1)$, and then using the generator from Proposition 32.

Proposition 33 *(Theorem 6, 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 31 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 32. ∎

6.2.3 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 6.

Proposition 34 *(Theorem 6, 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 2.3. 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 2.3, the algorithm works in two steps. In the first step, we use Proposition 31 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 2.3); 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 31 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 $\beta = (c/\epsilon) + (1/3) \cdot (1 - c/\epsilon) < \alpha$. [17] Denote the random variable that is the output string of this distribution by $\rho \in [1/p]^{n'}$. For every $\oplus$-gate that has not been restricted by the algorithm from Proposition 31, 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,

[17] 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.

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 12). 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 32, and find a satisfying input for C in W. ∎

6.2.4 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 6. The crucial observation here is that after invoking the algorithm from Proposition 31, 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 35 *(Theorem 6, 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 31 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 $w \in W$. ∎

7 Polynomials that vanish rarely

In the current section we prove Theorem 7 (in Section 7.1) and Theorem 8 (in Section 7.2). Recall that throughout the currnet section we consider a normalized "badness" parameter $b(n) = B(n)/2^n$.

7.1 Proof of Theorem 7

We now prove a more general version of Theorem 7, which depends on additional parameters; after stating this general version, we will spell out the parameter choices that yield Theorem 7. The proof relies on Lemma 16.

Proposition 36 *(Theorem 7, 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 7, 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 36 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 16 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 $g'(x) = \Pi_{i=1}^d \left(1 + \sum_{j \in S_i} (L_i(x) + 1)\right)$. [18]

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

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

[18] 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 $1 - 2^{-d}$.

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 $2/c'$.

Thus, there exists a distribution that is $(1 - 2/c')$-typically in $\mathcal{P}_d^{c'}$ and that rejects every $x \notin C^{-1}(1)$ with probability at least $1 - m \cdot 2^{-d}$. Now, let $\mathbf{w}$ be the output distribution of a hitting-set generator with density $1 - c > (2/c') + m \cdot 2^{-d}$ for $\mathcal{P}_d^{c'}$. Relying on Lemma 16,

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

which concludes the proof. ■

7.2 Proof of Theorem 8

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 37 *(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 37, 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 38 *(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 8. 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 $d' = \mathrm{poly}(d)$ that vanishes very rarely.

Proposition 39 *(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 39 appeared in Section 2.4. 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 $p|_{W_{\vec{u}}} \equiv 0$. We will show that with high probability over choice of $\vec{u}$ it holds that $p|_{W_{\vec{u}}} \not\equiv 0$, 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 $p|_{W_{\vec{u}}} \not\equiv 0$. 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 $p|_{W_{\vec{u}}}(\alpha_1, ..., \alpha_{t-1})$ is of the form

$$\begin{aligned} p|_{W_{\vec{u}}}(\alpha_1, ..., \alpha_{t-1}) &= p\left(u^{(0)} + \alpha_1 \cdot u^{(1)} + ... + \alpha_{t-1} \cdot u^{(t-1)}\right) \\ &= p\left(u_1^{(0)} + \alpha_1 \cdot u_1^{(1)} + ... + \alpha_{t-1} \cdot u_1^{(t-1)}, ..., u_n^{(0)} + \alpha_1 \cdot u_n^{(1)} + ... + \alpha_{t-1} \cdot u_n^{(t-1)}\right) \\ &= \sum_{i_1 + i_2 + ... + i_{t-1} \leq d} c_{i_1, ..., i_{t-1}}(\vec{u}) \cdot \alpha_1^{i_1} \cdot ... \cdot \alpha_{t-1}^{i_{t-1}}, \end{aligned} \tag{7.1}$$

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 $p|_{W_{\vec{u}}}$.

Note that $p|_{W_{\vec{u}}} \equiv 0$ 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 $p|_{W_{\vec{u}}}$ 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 38. 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 39.1. *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. (7.1), 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 12, we deduce that:

Claim 39.2. *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 39.2 amounts to a straightforward calculation, so we defer it to Appendix C. Relying on Claim 39.2 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 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 39 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 40 *(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 $2 \leq c' \leq d$, 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 $2/c'$, 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 39, with the fixed parameter $t = 2$. [19] Let $G = \left\{\vec{u} \in \mathbb{F}^{2 \cdot n} : p\restriction_{W_{\vec{u}}} \not\equiv 0\right\}$ be the set of subspaces on which p is not identically zero. Our goal is to show a distribution $\mathbf{h}$ over polynomials $\mathbb{F}^{2 \cdot n} \to \mathbb{F}$ of degree d that satisfies the following:

[19] 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.

- For every $\vec{u} \notin G$ it holds that $\Pr[\mathbf{h}(\vec{u}) = 0] = 1$.
- The probability that $h \sim \mathbf{h}$ vanishes on more than c'/q of its inputs is at most $2/c'$.

We can then rely on Lemma 16, to deduce that any sufficiently dense hitting-set generator for degree-d polynomials that vanish on at mos c'/q of their inputs also hits G, which allows us to hit p with additional $\lceil \log(q) \rceil$ random bits.

Towards constructing $\mathbf{h}$, recall that for every fixed $\vec{u} \in \mathbb{F}^{2 \cdot n}$, the $d+1$ coefficients of $p\restriction_{W_{\vec{u}}}$ are degree-d polynomials in $\vec{u}$, denoted $c_1(\vec{u}), ..., c_{d+1}(\vec{u})$. The distribution $\mathbf{h}$ is simply 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 $\vec{u} \notin G$ (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 $1 - 1/q$.

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 40.1. *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 40.1 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 40.1 implies that $\Pr_{\vec{u}}[\vec{u} \notin G] \leq 1/q$. Hence, over a random choice of $\vec{\beta}$, the expected fraction of inputs on which $h_{\vec{\beta}}$ vanishes is

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

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 $2/c'$.

Now, let $\mathbf{w}$ be the output distribution of a hitting-set generator with density $\mu > 2/c'$ for $\mathcal{P}_{d,c'}$; then, Lemma 16 implies that $\Pr[\mathbf{w} \in G] > \mu - 2/c' > 0$. Finally, similarly to the proof of Proposition 39, after obtaining $\vec{u} \in \mathbb{F}^{2 \cdot n}$ we can use another $\log(q)$ bits to uniformly choose an element in $W_{\vec{u}}$, thus hitting p with positive probability. ■

Let us now formally state Theorem 8, and prove it as a corollary of Propositions 39 and 40.

Theorem 41 *(Theorem 8, 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 $d \in \mathbb{N}$ 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 $2/c'$ for 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 $2/c'$ for polynomials of degree d that vanish on c'/q of their inputs. Relying on Proposition 40, 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 41.1. *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 41.1 appears in Appendix C. It follows from Fact 41.1 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 39, 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 41.1, 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

[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.

[CL16] Kuan Cheng and Xin Li. Randomness extraction in AC0 and with small locality. *Electronic Colloquium on Computational Complexity: ECCC*, 23:18, 2016.

[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.

[GMR13] Parikshit Gopalan, Raghu Meka, and Omer Reingold. Dnf sparsification and a faster deterministic counting algorithm. *Computational Complexity*, 22(2):275–310, 2013.

[Gol17] Oded Goldreich. *Introduction to Property Testing (working draft)*, February 7, 2017. Accessed at http://www.wisdom.weizmann.ac.il/~oded/pt-intro.html, February 14, 2017.

[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. Full version available online at *Electronic Colloquium on Computational Complexity: ECCC*, 20:152 (Rev. 2), 2013.

[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.

[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.

[Ros14] Benjamin Rossman. The monotone complexity of k-clique on random graphs. 43(1):256–279, 2014.

[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 An 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 18.

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 18 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 18, any pseudorandom generator for $\mathcal{H}$ also "fools" p. Let us now formalize and parametrize this argument.

Theorem 42 *($\mathbb{F}_2$-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 $\delta > 0$, any pseudorandom generator with error $\delta/2$ for polynomials of degree $\lceil \log(2c/\delta) \rceil$ is also a pseudorandom generator with error δ for p, where pseudorandom generators for $\mathbb{F}_2$-polynomials are defined in Definition 9.*

Proof. Let $d' = d - \lfloor \log(2c/\delta) \rfloor$, 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 $\Pr[\mathbf{h}(x) = p(x)] > 1 - \delta/2$; 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 $1 - b(n) \cdot (2^{d'} - 1) > 1 - \frac{\delta}{2}$ 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 $\xi(x) = (-1)^x$. Let $\mathbf{w}$ be a distribution that $(\delta/2)$-fools polynomials of degree $\lceil \log(2c/\delta) \rceil$ (which implies that for every such polynomial p' it

holds that $\left|\mathbb{E}[\xi(p'(\mathbf{w}))] - \mathbb{E}[\xi(p'(\mathbf{u}_n))]\right| \leq \delta$). According to Lemma 18, using the parameter values $\delta = \max_{x \in \mathbb{F}_2}\{|\xi(x)|\} = 1$, and $\epsilon_1 = (\delta/2)$, and $\epsilon_2 = 0$, and $\epsilon_3 = \delta$, it holds that $\left|\Pr[p(\mathbf{w}) = 1] - \Pr[p(\mathbf{u}_n) = 1]\right| = \frac{1}{2} \cdot \left|\mathbb{E}[\xi(p(\mathbf{w}))] - \mathbb{E}[\xi(p(\mathbf{u}_n))]\right| \leq \delta$. ■

Appendix B Proofs of claims from Section 5

We prove two claims from Section 5.2.2 (i.e., Lemma 27 and a generalization of the switching lemma of [GW14]) and a technical claim from Section 5.2.1 (i.e., Claim 23). Lemma 27 is an adaptation of the main lemma of Trevisan and Xue [TX13]. Let us now recall the statement of Lemma 27, and prove the lemma.

Lemma 43 *(Lemma 27, restated). Let F be a CNF over n inputs with m clauses, each clause of width at most w. 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 a distribution over $\{0,1\}^{(q+1)\cdot n}$ that δ_0-fools all CNFs of width $w' = w \cdot (q+1)$. Then, the probability that $F\restriction_\rho$ cannot be computed by a decision tree of depth D is at most $2^{D+w+1} \cdot (5pw)^D + \delta_0 \cdot 2^{(D+1)\cdot(2\cdot w + \log(m))}$.*

Proof sketch. We rely on the proof of Lemma 7 in [TX13], and in particular use the same definitions of canonical decision tree, path, and segment. The proof in [TX13] reduces the task of finding a restriction ρ such that $F\restriction_\rho$ can be computed by a shallow decision tree to the task of "fooling" less than $2^{(D+1)\cdot(2w+\log(m))}$ tests: For each path of length $D+1$ (i.e., a sequence of $D+1$ segments), there is a corresponding test $T_P : \{0,1\}^{(q+1)\cdot n} \to \{0,1\}$ that gets as input a restriction $\rho \in \{0,1\}^{(q+1)\cdot n}$, and accepts ρ if and only if the canonical decision tree for $F\restriction_\rho$ contains the path P. Indeed, if all the tests reject ρ, it means that no path of length $D+1$ exists in the canonical decision tree for $F\restriction_\rho$, which implies that the canonical decision tree for $F\restriction_\rho$ is of depth D.

The key claim in the proof is Claim 8, which asserts that for each path P, the test T_P can be computed by a CNF. The goal in [TX13] is to show that the CNF for T_P has few clauses; we focus on showing that the CNF for T_P has small width. To see that this holds, note that T_P is constructed as a conjunction of conditions, where each condition depends only on the assignment that ρ gives to the variables of a single clause of F (either a clause that belongs to a segment in the path, or a clause whose index is between the indices of clauses that belong to segments in the path). Thus, each condition depends only on the assignment that ρ gives to w variables, which means that each condition depends only on $w' = w \cdot (q+1)$ bits of ρ. Hence, each condition can be decided by a CNF of width w', and T_P (which is their conjunction) can also be decided by a CNF of width w'. ■

Let us now formally state the generalization of the switching lemma of Goldreich and Wigderson [GW14] and prove it.

Proposition 44 *(a generalization of the derandomized switching lemma of [GW14]). Let $m : \mathbb{N} \to \mathbb{N}$, let $w : \mathbb{N} \to \mathbb{N}$, and let $\delta : \mathbb{N} \to [0,1)$. Let $\mathbf{z}$ be a distribution over $\{0,1\}^{O(\log(w))\cdot n}$ that is δ'-almost t'-wise independent, where $\log(1/\delta') = O(t') = \tilde{O}(w) \cdot 2^w \cdot \log(1/\delta)$.*

Then, for any depth-2 formula $F : \{0,1\}^n \to \{0,1\}$ of width $w = w(n)$ with $m = m(n)$ clauses, with probability at least $1 - 4\delta$ (where $\delta = \delta(n)$) over choice of $\rho \sim \mathbf{z}$ it holds that the restricted formula $F\restriction_\rho$ can be computed by a decision tree of depth $D = O(\log(1/\delta))$.

Proof. Let $\delta_0 = \delta \cdot 2^{-D} = \text{poly}(\delta)$, and fix a depth-2 formula $F : \{0,1\}^n \to \{0,1\}$; without loss of generality, assume that F is a CNF. [20] Consider a uniformly-chosen restriction ρ that keeps each variable alive with probability $p = 1/O(w)$; Hastad's switching lemma asserts that with probability at least $1 - 2^{-O(D)} \geq 1 - \delta_0$, the *canonical decision tree* of $F\restriction_\rho$ is of depth $D = O(\log(1/\delta))$ (the canonical decision tree is the decision tree that is constructed by the algorithm in Hastad's original proof; for a definition see, e.g., [TX13, Def. 4]).

Given a restriction ρ, we consider the following way to decide whether the canonical decision tree of $F\restriction_\rho$ is of depth D. Associate each string $P \in \{0,1\}^D$ with a potential *positional path* of depth D in the canonical decision tree of F; that is, the string P induces a path from the root to a specific node of depth D in a full binary tree of depth D or more. For each $P \in \{0,1\}^D$, we consider a corresponding test T_P that gets ρ as input, and tests whether or not one of the nodes in the path induced by P along the canonical decision tree of $F\restriction_\rho$ is a leaf node (i.e., whether or not the path ends at depth at most D); if there is indeed a leaf then T_P accepts ρ, and otherwise (i.e., if the path continues to depth $D+1$) then T_P rejects ρ. We will describe T_P in detail in a moment, but for now observe that the canonical decision tree of $F\restriction_\rho$ is of depth D if and only if for each $P \in \{0,1\}^D$ it holds that $T_P(\rho) = 1$.

To describe how each T_P works, fix $P \in \{0,1\}^D$, and let T_P be the following recursive algorithm. The algorithm gets as input a CNF F', a restriction ρ' and a string P' (in the first recursive call $F' = F$, $\rho' = \rho$, and $P' = P$). If the CNF is empty (i.e., has no clauses), then the algorithm accepts; otherwise, the algorithm examines the values that ρ' assigns to the variables in the first clause of F':

- If the first clause is unsatisfied by ρ' (i.e., all variables are fixed to unsatisfying values) then the algorithm accepts and halts.
- If the first clause is satisfied by ρ' (i.e., one or more variables are assigned to satisfying values), then the algorithm simplifies F' by omitting the first clause, and by simplifying the other clauses according to the values that ρ' assigned to the variables in the first clause. Then, the algorithm recurses with with the simplified CNF and with the same restriction ρ' and string P'.
- Otherwise, the first clause is undetermined by ρ'. If the number of living variables in the clause, denoted by k, is greater than the length of P', then the algorithm rejects. [21] If $k \leq |P'|$, let ρ'' be the restriction that fixes the k variables to values according to the k-prefix of P'. The algorithm simplifies F' according to the composition $\rho'' \circ \rho'$, and recurses with the simplified CNF, with the restriction $\rho'' \circ \rho'$, and with the string obtained from P' by omitting its first k bits.

The main point to note in the above description is that in each recursive call, the test T_P needs to read at most w blocks of $\lceil \log(1/p) \rceil = O(\log(w))$ bits in the restriction, corresponding to the (at most w) variables in the clause that it examines. The key observation in [GW14, Lemma 3.3], which we now state in a more general form, is that for each

[20] This is without loss of generality since if F is a DNF, then $F\restriction_\rho$ can be computed by a depth-D decision tree if and only if $(\neg F)\restriction_\rho$ can be computed by such a tree.

[21] This event means that the path induced by P in the canonical decision tree of $F\restriction_\rho$ is of depth more than $|P| = D$. Recall that by the definition of the canonical decision tree, whenever the algorithm that constructs the canonical decision tree encounters an undetermined clause, it adds the full sub-tree that corresponds to all living variables in the clause to the canonical decision tree.

$P \in \{0,1\}^D$, with high probability it holds that T_P makes at most $D' = O\left(2^w \cdot \log\left(1/\delta_0\right)\right)$ recursive calls; that is, with high probability T_P examines the values that ρ assigns to variables of at most D' clauses. This is the case because for each recursive call, the probability that the clause that is examined is *unsatisfied* is at least 2^{-w}; thus, the probability that after D' recursive calls the algorithm encountered an unsatisfied clause, and thus stopped, is more than $1 - \left(1 - 2^{-w}\right)^{D'} \geq 1 - \delta_0$. It follows that for each $P \in \{0,1\}^D$, with probability at least $1 - 2\delta_0$ over a uniformly-chosen restriction ρ it holds that T_P accepts ρ without making more than D' recursive calls.

Now, consider "truncated" versions of these tests: For each $P \in \{0,1\}^D$, consider a modified version T'_P of T_P that, in addition to the description above, rejects ρ if the depth of the recursion exceeds D'. According to previous paragraph, the test T'_P accepts a uniformly-chosen restriction with probability at least $1 - 2\delta_0$. Since each T'_P reads at most $D'' = O\left(D' \cdot w \cdot \log(w)\right) = \tilde{O}\left(w\right) \cdot \left(2^w \cdot \log(1/\delta)\right)$ bits in the restriction, if instead of the uniform distribution we choose a restriction from the distribution $\mathbf{z}$, which is δ'-almost t'-wise independent, where $\delta' < \left(\delta_0 \cdot 2^{-D''}\right)$ and $t' \geq D''$, then the probability that T'_P will accept is at least $1 - 3\delta_0$. [22] Thus, the probability that all the tests accept (i.e., $\wedge_{P \in \{0,1\}^D} T_P(\rho) = 1$) is at least $1 - 3\delta$. ■

Let us now recall the statement of Claim 23 and prove it.

Claim 45 *(Claim 23, restated). Let $F : \{0,1\}^n \to \{0,1\}$ be a depth-2 formula of width w and size m, and let $F' : \{0,1\}^n \to \{0,1\}$ be a refinement of F. Then, for any restriction $\rho \in \{0,1,\star\}^n$ it holds that $F\restriction_\rho$ can be computed by a depth-2 formula Φ of width w and size m such that $F'\restriction_\rho$ is a refinement of Φ.*

Proof. We prove the claim for the case where F is a DNF; the proof for the case where F is a CNF follows by reduction to the DNF $\neg F$, relying on Fact 21. Let Φ be the DNF for $F\restriction_\rho$ that is obtained by fixing the variables in each clause of F according to ρ, without omitting any clause from the formula (even if a clause becomes a constant function).

When F' was obtained by a sequence of removal steps and clean-up steps, then F' is simply a sub-formula of F. In this case, we can apply the same sequence of removal steps and clean-up steps to Φ, to obtain a corresponding sub-formula of Φ that computes $F'\restriction_\rho$. [23] We thus focus on proving the claim when F' was obtained by a sequence of $k \leq m$ merging steps and clean-up steps.

For every $i \in [k]$, let $F^{(i)}$ be the formula in the beginning of the i^{th} refinement step in the transformation of F to F', and let $F^{(k+1)} = F'$. We will show a sequence of k merging steps and clean-up steps that, when applied to Φ, induce a corresponding sequence of formulas $\Phi = \Phi^{(1)}, ..., \Phi^{(k+1)}$, such that the following holds: For every $i \in [k]$ there exists a bijection between the clauses of $\Phi^{(i)}$ and the clauses of $F^{(i)}\restriction_\rho$ such that every clause φ of the former is mapped to a clause f of the latter such that φ computes the function $f\restriction_\rho$. In

[22] The reason that we use the error parameter $\delta_0 \cdot 2^{-D''}$ instead of the more natural parameter δ_0 is that the tests that we are trying to "fool" are *adaptive*; that is, for each $P \in \{0,1\}^D$, the test T_P does not examine a fixed set of D'' bits in ρ, but rather adaptively chooses which bits to read according to the values of the bits that it read so far. We rely on the fact that any distribution that is $\left(\delta_0 \cdot 2^{-D''}\right)$-almost D''-wise independent also δ_0-fools adaptive tests that only read D'' bits (see, e.g., [Gol17, Exer. 7.4]).

[23] That is, let $F = \vee_{i=1}^m f_i$, and assume that $F' = \vee_{i=k+1}^m f_i$ was obtained from F by removing the clauses $f_1, ..., f_k$. Then it holds that $\Phi = \vee_{i=1}^m (f_i\restriction_\rho)$ and $F'\restriction_\rho = \vee_{i=k+1}^m (f_i\restriction_\rho)$, which implies that we can apply k removal steps to Φ in order to obtain $F'\restriction_\rho$.

particular, this claim implies that for every $i \in [k]$ it holds that $\Phi^{(i)} \equiv F^{(i)}\restriction_\rho$, and therefore $F'\restriction_\rho \equiv \Phi^{(k+1)}$ is a refinement of $\Phi = \Phi^{(1)}$.

The claim is proved by induction on i. The base case $i = 1$ follows immediately from the definition of $\Phi^{(1)} = \Phi$. For the induction step, assume that there is a bijection as above between the clauses of $\Phi^{(i)}$ and the clauses of $F^{(i)}\restriction_\rho$, and let us define the i^{th} refinement step that is applied to $\Phi^{(i)}$. If the i^{th} refinement step of $F^{(i)}$ was a clean-up step, then we can apply an analogous clean-up step to $\Phi^{(i)}$. [24] Otherwise, if the i^{th} refinement step of $F^{(i)}$ was a merging step, let $f_1^{(i)}, ..., f_u^{(i)}$ be the set of clauses that were removed in this step, and let $h^{(i)}$ be the new clause that was added in their stead. For every $j \in [u]$, let $\varphi_j^{(i)}$ be the clause in $\Phi^{(i)}$ that computes $f_j^{(i)}\restriction_\rho$ and exists by the induction hypothesis. We show how apply a single refinement step to $\Phi^{(i)}$ that replaces the clauses $\varphi_1^{(i)}, ..., \varphi_u^{(i)}$ with a new clause $\varphi^{(i)}$ that computes the function $h^{(i)}\restriction_\rho$. This is proved by a case analysis:

1. If $h^{(i)}\restriction_\rho$ is not a constant function, then it follows that $\bigcap_{j\in[u]}(f_j^{(i)}\restriction_\rho) = \bigcap_{j\in[u]} \varphi_j^{(i)} \neq \emptyset$. In this case, we apply a merging step to the clauses $\varphi_1^{(i)}, ..., \varphi_u^{(i)}$ in $\Phi^{(i)}$, and they are replaced with the non-constant clause $\varphi^{(i)} = \bigcap_{j\in[u]} \varphi_j^{(i)} = \bigcap_{j\in[u]}(f_j^{(i)}\restriction_\rho) = h^{(i)}\restriction_\rho$.

2. If $h^{(i)}\restriction_\rho \equiv 0$, then for every $j \in [u]$ it holds that $f_j^{(i)}\restriction_\rho \equiv \varphi_j^{(i)} \equiv 0$. This is the case because $\bigcap_{j\in[u]} f_j^{(i)} \neq \emptyset$ (otherwise $h^{(i)} \equiv 1$ and also $h^{(i)}\restriction_\rho \equiv 1$), whereas $\left(\bigcap_{j\in[u]} f_j^{(i)}\right)\restriction_\rho \equiv 0$, which implies that for every $j \in [u]$ there exists a literal in $f_j^{(i)}$ that is fixed by ρ to an unsatisfying value. Therefore, in this case we can apply a clean-up step to $\Phi^{(i)}$ to remove all but a single constant zero clause among the $f_j^{(i)}$'s.

3. If $h^{(i)}\restriction_\rho \equiv 1$, then it holds that $\bigcap_{j\in[u]} \varphi_j^{(i)} = \emptyset$. To see that this is the case, note that if $\bigcap_{j\in[u]} f_j^{(i)} = \emptyset$ then the latter assertion holds immediately; and otherwise (i.e., $\bigcap_{j\in[u]} f_j^{(i)} \neq \emptyset$), it follows by the assumption that $h^{(i)}\restriction_\rho \equiv 1$ that ρ fixes all the literals that are shared by all the u clauses $f_1^{(i)}, ..., f_u^{(i)}$ to satisfying values, which indeed implies that $\bigcap_{j\in[u]} \varphi_j^{(i)} = \emptyset$. Thus, we can apply a merging step to $\varphi_1^{(i)}, ..., \varphi_u^{(i)}$ to obtain the constant one function. ■

Appendix C Proofs of technical claims from Section 7

In this appendix we prove several technical claims that were made in the proofs of Proposition 39, Proposition 40, and Theorem 41.

Let us first prove a claim that generalizes Claims 39.2 and 40.1, which were made in the proofs of Proposition 39 and Proposition 40, 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:

[24] Specifically, denote by $f_1^{(i)}, ..., f_u^{(i)}$ the constant zero clauses that were removed from $F^{(i)}$ in the i^{th} step. For every $j \in [u]$, let $\varphi_j^{(i)}$ be the clause in $\Phi^{(i)}$ that computes $f_j^{(i)}\restriction_\rho \equiv 0$ and exists by the induction hypothesis. Then, the i^{th} refinement step of $\Phi^{(i)}$ is a clean-up step that removes the constant zero clauses $\varphi_1^{(i)}, ..., \varphi_u^{(i)}$.

Claim 46 *(Claims 39.2 and 40.1, 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 12. 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 12 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 41.1, which was stated in the proof of of Theorem 41:

Fact 47 *(Fact 41.1, restated). 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 $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{n \cdot e}{k}\right)^k$. [25] Similarly, if $(n/t) \leq c' \cdot (n/t + d)$, where $c' = 1/e$, 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. [26]

[25] 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 $\frac{(n/t)+d}{d} \geq t \cdot e$.

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

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 $\log\left(\binom{n+d}{d}\right) \leq H_2(\alpha) \cdot (n + d)$, whereas $\log\left(\binom{n/t+d}{d}\right) \geq (H_2(\alpha') - o(1)) \cdot (n/t + d) = \Omega\left(H_2(\alpha) \cdot (n + d)\right)$. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il