Electronic Colloquium on Computational Complexity, Report No. 121 (2016)

Approximate Degree and the Complexity of Depth Three Circuits

Mark Bun* Justin Thaler†

Abstract

Threshold weight, margin complexity, and Majority-of-Threshold circuit size are basic complexity measures of Boolean functions that arise in learning theory, communication complexity, and circuit complexity. Each of these measures might exhibit a *chasm* at depth three: namely, all polynomial size Boolean circuits of depth two have polynomial complexity under the measure, but there may exist Boolean circuits of depth three that have essentially maximal complexity $\exp(\Theta(n))$. However, existing techniques are far from showing this: for all three measures, the best lower bound for depth three circuits is $\exp(\tilde{\Omega}(n^{2/5}))$. Moreover, current methods appear intrinsically unable to prove lower bounds better than $\exp(\Omega(\sqrt{n}))$ even for depth four circuits, and have yet to prove lower bounds better than $\exp(\tilde{\Omega}(\sqrt{n}))$ for circuits of *any* constant depth.

We take a significant step toward showing that all of these complexity measures indeed exhibit a chasm at depth three. Specifically, for any arbitrarily small constant $\delta > 0$, we exhibit:

- A depth three circuit of polynomial size (in fact, an $O(\log n)$-decision list) of complexity $\exp(\Omega(n^{1/2-\delta}))$ under each of these measures.
- A depth three circuit F of quasi-polynomial size (in fact, an $O(\log^2 n)$-decision list) of complexity $\exp(\Omega(n^{2/3-\delta}))$ under each of these measures. The function F is also computed by a depth four circuit of polynomial size.

Our methods suggest natural candidate functions that may exhibit stronger bounds, of the form $\exp(\tilde{\Omega}(n))$, where the $\tilde{\Omega}$ notation hides factors polylogarithmic in n. The technical core of our results lies in establishing new lower bounds on the uniform approximability of depth three circuits by low-degree polynomials.

*John A. Paulson School of Engineering and Applied Sciences, Harvard University. Supported by an NDSEG Fellowship and NSF grant CNS-1237235. This work was done while the author was visiting Yale University.

†Georgetown University. Parts of this work were performed while the author was at Yahoo Research.

ISSN 1433-8092

1 Introduction

Let $f\colon \{-1,1\}^n \to \{-1,1\}$ be a Boolean function, and let $\mathcal{C}\colon 2^{\{-1,1\}^n} \to \mathbb{N}$ denote a measure of the complexity of f. We say that $\mathcal{C}$ exhibits a *chasm* at depth three if all Boolean circuits[1] of depth two have polynomial complexity under the measure, but there exist circuits of depth three that have essentially maximal complexity $\exp(\Theta(n))$. Examples of measures that may satisfy a chasm at depth three include:

- **Threshold Weight.** A polynomial $p\colon \{-1,1\}^n \to \mathbb{R}$ with integer coefficients is said to sign-represent f if $p(x) \cdot f(x) > 0$ for all $x \in \{-1,1\}^n$. The weight of p, denoted $W(p)$, is the sum of the absolute value of its coefficients. The *threshold weight* of f is the least weight of a sign-representing polynomial for f.

  It is easy to see that all DNF and CNF formulae of size s have threshold weight $O(s)$. The best known upper bound on the threshold weight of depth three circuits is the trivial $2^{O(n)}$ bound. Hence, threshold weight may exhibit a chasm at depth three.

- **Discrepancy and Margin Complexity.** Discrepancy, defined formally in Section 6.2, is a central quantity in communication complexity and circuit complexity.[2] For example, discrepancy is known to characterize the communication complexity class **PP**, and small discrepancy implies large communication complexity in nearly every communication model. The multiplicative inverse of discrepancy is also known to be equivalent to *margin complexity*, a central quantity in learning theory [21].

  All DNF and CNF formulae have at least inverse-polynomial discrepancy. However, the best known lower bound on the discrepancy of depth three circuits is the trivial $2^{-O(n)}$ bound. Hence, margin complexity and (the inverse of) discrepancy may exhibit a chasm at depth three.

- **Majority-of-Threshold Circuit Size**. Since OR and AND can each be computed by a single Majority gate, all DNF and CNF formulae are computed by Majority-of-Threshold (in fact, Majority-of-Majority) circuits of polynomial size. Meanwhile, the best known upper bound on the size of Majority-of-Threshold circuits computing depth three Boolean circuits is the trivial $2^{O(n)}$ bound. Hence, Majority-of-Threshold circuit size may exhibit a chasm at depth three.

We discuss each of these measures, together with applications, in more detail in Section 6.

Unfortunately, we are currently quite far from proving that any of the above complexity measures actually exhibit such a chasm. For each measure, the best known lower bound for depth three circuits is $\exp(\tilde{\Omega}(n^{2/5}))$ [10]. Moreover, as we explain in Section 1.2.4, existing techniques appear intrinsically unable to prove a lower bound better than $\exp(\Omega(n^{1/2}))$ even for circuits of depth four. This barrier stems from the fact that previous work has focused exclusively on analyzing *block-composed* functions. Here, a function $f\colon \{-1,1\}^{N \cdot M} \to \{-1,1\}$ is said to be block-composed if there are two functions $h\colon \{-1,1\}^N \to \{-1,1\}$ and $g\colon \{-1,1\}^M \to \{-1,1\}$ such that $f = h \circ g := h(g,\ldots,g)$. That is, a function is block-composed if it interprets its input as a sequence of N *blocks* $x_1,\ldots,x_N \in \{-1,1\}^M$, applies a Boolean function g independently to each block x_i, and then feeds the N outputs into a different function h.

In this paper, we take a significant step toward showing that all three of these complexity measures indeed exhibit a chasm at depth three. Specifically, for any constant $\delta > 0$, we exhibit:

[1] Throughout this paper, unless otherwise noted, all circuits under consideration are assumed to have polynomial size, and to be over the basis AND, OR and NOT.

[2] Discrepancy is often thought of as a matrix-analytic quantity, rather than as a Boolean function complexity measure. For a function $f\colon \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$, when we refer to the discrepancy of f, we mean the discrepancy of the matrix $[f(x,y)]_{x,y \in \{-1,1\}^n}$.

- A depth three circuit of polynomial size (in fact, an $O(\log n)$-decision list) of complexity $\exp(\Omega(n^{1/2-\delta}))$ under each of these measures.
- A depth three circuit F of quasi-polynomial size (in fact, an $O(\log^2 n)$-decision list) of complexity $\exp(\Omega(n^{2/3-\delta}))$ under each of these measures. Our F is also computed by a depth four circuit of polynomial size.

Our results surpass the aforementioned $\exp(\Omega(n^{1/2}))$ barrier, and improve substantially on the best lower known bounds for circuits of *any* constant depth. Our methods also suggest natural candidate functions that may exhibit stronger bounds, of the form $\exp(\tilde{\Omega}(n))$, where the $\tilde{\Omega}$ notation hides factors polylogarithmic in n (cf. Section 3.3).

Our improvement over prior work stems from the fact that we move beyond block-composed functions. The functions that underly our analysis are rather complicated to define (see Sections 4.2 and 5.2 for details), but here we briefly highlight their novel features. Inspired by prior work of Podolskii [25] (see Section 1.2.2 for further discussion), we define our functions to be "almost" block-composed, but to have mild dependencies between blocks. Just like a block-composed function, each of our functions interprets its input as a sequence of N blocks $x_1, \ldots, x_N \in \{-1,1\}^M$, and applies a Boolean function g to each block, before feeding the N outputs into a different function h. However, for $i \geq 2$, before applying g to x_i we first pass x_i through a "pre-processing function" that *depends on the preceding blocks* $x_1, x_2, \ldots, x_{i-1}$. We ensure that this dependency is simple enough that the final function is computed by a circuit of depth three or four, but complicated enough to break the $\exp(\Omega(n^{1/2}))$ barrier that seems intrinsic to methods focusing exclusively on block-composed functions.

1.1 Our Contributions: Details

The three complexity measures $\mathcal{C}$ described above are intimately related to uniform approximability by low-degree polynomials, as we now explain. Roughly speaking, for each of the three measures, in order to construct a function of complexity at least 2^d under $\mathcal{C}$, it suffices to identify a function $f\colon \{-1,1\}^n \to \{-1,1\}$ such that f cannot be uniformly approximated to error $1-2^{-d}$ by polynomials of degree at most d. One can then apply known transformations [17,18,31] to transform f into a related function $F\colon \{-1,1\}^n \to \{-1,1\}$ such that $\mathcal{C}(F) \geq 2^{\Omega(d)}$. Moreover, these transformations are simple in the following sense: if f is computed by a (polynomial size) depth d circuit with logarithmic bottom fan-in, then so is F. Similarly, if f is computed by a *quasi*-polynomial size depth d circuit with *poly*logarithmic bottom fan-in, then so is F.

Accordingly, the main technical contribution of this paper is to prove a new lower bound on the approximability of suitable constant-depth circuits by low-degree polynomials.

Theorem 1. For any arbitrarily small constant $\delta > 0$ and any arbitrarily large constant $\Gamma > 1$, there is an (explicitly given) function $f\colon \{-1,1\}^n \to \{-1,1\}$ that is computed by Boolean circuit of depth three, with logarithmic bottom fan-in, that satisfies the following property. For any polynomial $p\colon \{-1,1\}^n \to \mathbb{R}$ of total degree at most $n^{1/2-\delta}$, there exists some $x \in \{-1,1\}^n$ such that $|p(x) - f(x)| > 1 - 2^{-n^{\Gamma}}$.

In fact, the function f in Theorem 1 is much simpler than an arbitrary depth three circuit with logarithmic bottom fan-in; it is an $O(\log n)$-decision list of polynomial length. An $O(\log n)$-decision list is a function whose output is determined by a very simple sequential decision process (essentially, a chain of "if-then-else" statements, where each "if" statement is a conjunction on $O(\log n)$ variables – we give a precise definition in Section 2). Decision lists have been studied intensely in learning theory and complexity theory (see, e.g., [6, 8, 13, 16, 17, 27, 28, 36]).

Theorem 2. For any arbitrarily small constant $\delta > 0$ and any arbitrarily large constant $\Gamma > 1$, there is an (explicitly given) function $f\colon \{-1,1\}^n \to \{-1,1\}$ that is computed by a polynomial size Boolean circuit of depth four, with logarithmic bottom fan-in, that satisfies the following property. For any polynomial $p\colon \{-1,1\}^n \to \mathbb{R}$ of total degree at most $n^{2/3-\delta}$, there exists some $x \in \{-1,1\}^n$ such that $|p(x) - f(x)| > 1 - 2^{-n^\Gamma}$. Moreover, f is also computed by a Boolean circuit of depth three and size $n^{O(\log n)}$, with bottom fan-in $O(\log^2 n)$.

Remark 3. In fact, the function f in Theorem 2 is computed by an $O(\log^2 n)$-decision list. Any $O(\log^2 n)$-decision list can be sign-represented by a polynomial of degree $O(\log^2 n)$. Hence, Theorem 2 exhibits a function f that can be sign-represented by polynomials of just polylogarithmic degree, but f requires very large degree (i.e., degree $\Omega(n^{2/3-\delta})$) to uniformly approximate, even if one only wants to approximate it to error *super-exponentially* close to 1 (i.e., to error $1 - 2^{-n^\Gamma}$).

By combining Theorems 1 and 2 with known transformations [18, 31], we obtain depth three circuits F and F' with very large complexity under the three measures described above. In fact, these functions are themselves respectively computed by $O(\log n)$- and $O(\log^2 n)$-decision lists.

Corollary 4. For any constant $\delta > 0$, there is an (explicitly given) function $F\colon \{-1,1\}^n \to \{-1,1\}$ that is computed by a Boolean circuit of depth three and logarithmic bottom fan-in such that:

- The threshold weight of F is $\exp(\Omega(n^{1/2-\delta}))$.
- The discrepancy F is $\exp(-\Omega(n^{1/2-\delta}))$.
- Any Majority-of-Threshold circuit computing F has size $\exp(\Omega(n^{1/2-\delta}))$.

Corollary 5. For any constant $\delta > 0$, there is an (explicitly given) function $F'\colon \{-1,1\}^n \to \{-1,1\}$ that is computed by a Boolean circuit of depth four and logarithmic bottom fan-in, and satisfies:

- The threshold weight of F' is $\exp(\Omega(n^{2/3-\delta}))$.
- The discrepancy F' is $\exp(-\Omega(n^{2/3-\delta}))$.
- Any Majority-of-Threshold circuit computing F' has size $\exp(\Omega(n^{2/3-\delta}))$.

Moreover, F' is also computed by a quasipolynomial size Boolean circuit of depth three and bottom fan-in $O(\log^2 n)$.

Table 1 succinctly compares our results to prior work.

Reference	Threshold Weight Bound	Discrepancy Bound	Majority-of-Threshold Circuit Size Bound	Circuit Depth
[18]	$\exp(\Omega(n^{1/3}))$	N/A	N/A	3
[30]	N/A	$\exp(-\Omega(n^{1/5}))$	$\exp(\Omega(n^{1/5}))$	3
[8,31]	N/A	$\exp(-\Omega(n^{1/3}))$	$\exp(\Omega(n^{1/3}))$	3
[10]	$\exp\left(\Omega(n^{2/5})\right)$	$\exp\left(-\Omega(n^{2/5})\right)$	$\exp\left(\Omega(n^{2/5})\right)$	3
[34]	$\exp\left(\Omega(n^{\frac{k-1}{2k-1}})\right)$	$\exp\left(-\Omega(n^{\frac{k-1}{2k-1}})\right)$	$\exp\left(\Omega(n^{\frac{k-1}{2k-1}})\right)$	$k+1$ (for $k \geq 2$)
[29]	$\exp\left(\Omega(n^{1/2})\right)$	$\exp\left(-\Omega(n^{1/2})\right)$	$\exp\left(\Omega(n^{1/2})\right)$	4
This work	$\exp\left(\Omega(n^{1/2-\delta})\right)$	$\exp\left(-\Omega(n^{1/2-\delta})\right)$	$\exp\left(\Omega(n^{1/2-\delta})\right)$	3
This work	$\exp\left(n^{2/3-o(1)}\right)$	$\exp\left(-n^{2/3-o(1)}\right)$	$\exp\left(n^{2/3-o(1)}\right)$	3 (quasi-polynomial size)
This work	$\exp\left(\Omega(n^{2/3-\delta})\right)$	$\exp\left(-\Omega(n^{2/3-\delta})\right)$	$\exp\left(\Omega(n^{2/3-\delta})\right)$	4

Table 1: Comparison of our new bounds for AC^0 to prior work. The circuit depth column lists the depth of the Boolean circuit used to exhibit the bound, and δ denotes an arbitrarily small positive constant. All Boolean circuits are polynomial size unless otherwise noted.

1.2 Prior Work

In order to discuss prior work, it is helpful to introduce the notions of approximate degree and threshold degree, which both capture the difficulty of point-wise approximation by low-degree polynomials. The ε-approximate degree of a function f, denoted $\widetilde{\deg}_\varepsilon(f)$, is the least degree of a real polynomial that point-wise approximates f to error ε. By convention, $\widetilde{\deg}_{1/3}(f)$ is denoted simply as $\widetilde{\deg}(f)$ and referred to without qualification as the approximate degree of f (the constant $1/3$ is chosen for aesthetic reasons, and could be replaced with any other constant in $(0,1)$ without affecting the theory in any way). The threshold degree of f, denoted $\deg_\pm(f)$, is the least degree of a real polynomial that sign-represents f at all points. When appropriate, we also use subscripts after function symbols to indicate the number of variables over which the function is defined. For example, OR_M denotes the OR function on M inputs.

1.2.1 Early Work on Approximating AC^0 Functions by Low-Degree Polynomials

Minsky and Papert [23] famously proved an $\Omega(n^{1/3})$ lower bound on the threshold degree of the DNF formula $\mathrm{OR}_{n^{1/3}} \circ \mathrm{AND}_{n^{2/3}}$, now known as the Minsky-Papert DNF. Klivans and Servedio [15] proved an essentially matching upper bound of $\tilde{O}(n^{1/3})$ on the threshold degree of *any* polynomial size DNF.

Beigel identified a DNF (in fact, a 1-decision list) known as OMB (short for ODD-MAX-BIT) that has threshold degree 1, but requires large degree to approximate to error bounded away from 1 [6]. OMB will play a central role in this paper, and we define it formally in Section 2. Quantitatively, Beigel showed[3] that for any $d > 0$, there is an $\varepsilon \in 1 - 2^{-\Omega(n/d^2)}$ such that $\widetilde{\deg}_\varepsilon(\mathrm{OMB}_n) \geq d$. For any $\varepsilon > 0$, Klivans and Servedio [16] gave an optimal ε-approximating polynomial for any 1-decision list, showing that Beigel's lower bound is asymptotically tight for all $d > 0$.[4]

1.2.2 Prior Work of Podolskii

Podolskii pioneered a line of work devoted to proving approximate degree lower bounds that hold even when the error parameter ε is allowed to be *super-exponentially* close to 1 [25, 26].[5] In [26], he showed that for any constant $d \geq 2$, there exists a function of threshold degree d that cannot be uniformly approximated to error ε by polynomials of degree at most d, unless $\varepsilon = 1 - n^{-\Omega(n^d)}$. This result is tight, matching an upper bound proved by Burhman et al. [8].

Our construction and analysis are inspired by another related result of Podolskii [25]. For any constant $d > 0$, Podolskii identified a function f of threshold degree d such that, even for $D \gg d$, the following holds: f cannot be uniformly approximated by degree D polynomials to error ε, unless ε is superexponentially close to 1. Quantitatively, he showed that for any constant $d > 0$, there exists a DNF (in fact, a d-decision list) f with threshold degree d, yet for any $D < O(n^{1/5}/\log n)$, there exists an $\varepsilon \in 1 - \exp\left(\Omega((n/D^4)^d)\right)$ for which $\widetilde{\deg}_\varepsilon(f) \geq D$. Unfortunately, Podolskii's construction itself does not yield any new bounds on the complexity measures we are interested in. By introducing new ideas, we are able to prove such improved bounds for depth three circuits.

[3] Beigel describes his result as a lower bound on the *degree-d threshold weight* of OMB_n, which refers to the least weight of a sign-representing polynomial p for f satisfying $\deg(p) \leq d$. However, his argument is easily seen to establish the claimed approximate degree lower bound.

[4] Like Beigel, Klivans and Servedio state their results in terms of degree-d threshold weight. However, their construction is easily seen to imply the claimed upper bound on the approximate degree of OMB_n.

[5] Again, Podolskii describes his work in terms of degree-d threshold weight, but his results hold for approximate degree as well.

1.2.3 Translating Approximate Degree Lower Bounds to Complexity Bounds

Several works have focused on transforming approximate degree and threshold degree lower bounds into bounds on the complexity measures that we focus on in this paper (threshold weight, discrepancy/margin complexity, and Majority-of-Threshold circuit size). Krause and Pudlák [18] showed how to take a function f of threshold degree at least d, and turn f into a related function F of threshold weight[6] at least 2^d. By applying this transformation to the Minsky-Papert DNF, Krause and Pudlák obtained a depth three circuit (with constant bottom fan-in) with threshold weight $\exp(\Omega(n^{1/3}))$.

Subsequent work by Krause [17] showed that for F to have threshold weight $2^{\Omega(d)}$, it is enough for f to satisfy $\widetilde{\deg}_{1-2^{-d}}(f) \geq d$.[7] Krause applied his result to the function $f = \mathrm{OMB}$, to obtain an $\exp(\Omega(n^{1/3}))$ lower bound on the threshold weight of a specific 2-decision list.

Sherstov's pattern-matrix method [31] showed how to take a function f satisfying the same condition required by Krause (i.e., $\widetilde{\deg}_{1-2^{-d}}(f) \geq d$), and turn it into a function F with discrepancy $2^{-\Omega(d)}$. By applying this transformation to the Minsky-Papert DNF or to OMB, Sherstov obtained a depth three circuit with discrepancy $\exp(-\Omega(n^{1/3}))$. Buhrman, Vereschagin, and de Wolf independently proved an identical discrepancy bound via very different techniques [8]. These discrepancy bounds also implied corresponding lower bounds on Majority-of-Threshold Circuit Size, through standard transformations [24].

1.2.4 Recent Work on Approximating AC^0 Functions by Low-Degree Polynomials

A handful of recent works have established various forms of "hardness amplification" for approximate degree [9, 10, 20, 29, 29, 32–34]. Roughly speaking, these results show how to take a function g which is hard to approximate by degree d polynomials to error $1/3$, and turn g into a related function f that is hard to approximate by degree d polynomials to error exponentially close to 1. Specifically, in these works, f is obtained from g by block-composing g with another function h.

Let ED_M denote the well-known Element Distinctness function (defined formally in Section 2) and $\overline{\mathrm{ED}}_M$ its negation. ED_M has played a central role in recent works on hardness amplification for approximate degree [10, 29, 34] because it currently exhibits the largest known approximate degree lower bound for any function in AC^0: $\widetilde{\deg}(\mathrm{ED}_M) = \tilde{\Omega}(M^{2/3})$ [1].

Our prior work [10] showed that the function $f = \mathrm{OR}_N \circ \mathrm{ED}_M$ satisfies $\widetilde{\deg}_{\varepsilon}(f) \geq \tilde{\Omega}(M^{2/3})$ for $\varepsilon = 1 - 2^{-N}$, and used this result to obtain a depth three circuit such that $\mathcal{C}(F) = \exp(\tilde{\Omega}(n^{2/5}))$ for the three complexity measures $\mathcal{C}$ that we focus on in this work. Thaler [35] then showed that the function $f = \mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ satisfies an identical lower bound, yielding another depth three circuit F (in fact, an $O(\log n)$-decision list) with $\mathcal{C}(F) = \exp(\tilde{\Omega}(n^{2/5}))$.

Sherstov [34] significantly strengthened the approach of [10] to obtain new *threshold degree* lower bounds for functions in AC^0. Specifically, in [34], for any $k \geq 2$, Sherstov exhibited a read-once formula of depth k (with polynomial bottom fan-in) that has threshold degree $\Omega(n^{\frac{k-1}{2k-1}})$. Applying the transformations of [17, 18, 31] to these circuits increases their depth by 1. In [29], Sherstov exhibited a depth four circuit of logarithmic bottom fan-in and threshold degree $\Omega(n^{1/2})$ – applying the transformation of [17, 18, 31] to this circuit does not increase its depth, yielding a depth four circuit F satisfying $\mathcal{C}(F) = \exp(\Omega(n^{1/2}))$.

The $\exp(\Theta(\sqrt{n}))$ Barrier For Circuits of Depth Four. Recall that for each of the three complexity measures $\mathcal{C}$ in which we are interested, in order to construct a function of complexity at least 2^d under $\mathcal{C}$, it

[6] In fact, Krause and Pudlák showed that F has threshold *length* 2^d, where threshold length is the least number of non-zero Fourier coefficients of any sign-representation for f. The threshold weight of f is always at least as large as its threshold length.

[7] Again, Krause phrased his lower bound in terms of the degree-d threshold weight of OMB, but his result is easily seen to imply the statement here.

suffices to identify a function $f\colon \{-1,1\}^n \to \{-1,1\}$ such that

$$f \text{ cannot be uniformly approximated to error } 1-2^{-d} \text{ by polynomials of degree at most } d. \tag{1}$$

We now argue that for circuits of depth 4, prior techniques cannot accomplish this for $d \gg \sqrt{n}$, *even if they assume the existence of a DNF* $g\colon \{-1,1\}^M \to \{-1,1\}$ *with (one-sided) approximate degree* $\Omega(M)$.[8]

The methods of [10, 35] start with a function $g\colon \{-1,1\}^M \to \{-1,1\}$, and assume nothing about g other than that g has (one-sided) approximate degree at least d. They show how to turn g into a "harder" function $f = h \circ g$ by block-composing g with another function $h \in \{\mathrm{OR}_N, \mathrm{OMB}_N\}$. Quantitatively, the resulting bound is of the form $\widetilde{\deg}_{1-2^{-N}}(f) \geq d$. Clearly, one must set $N \geq d$ to obtain a bound of the form Eq. (1). Hence, even if g has the largest possible (one-sided) approximate degree, $d = M$, the best bound that can be obtained from the methods of [10, 35] is of the form $\widetilde{\deg}_{1-2^{-N}}(f) \geq N$, obtained by setting $M = N$. In this case, f is a function over $n = N^2$ variables, so these methods can only yield complexity bounds of the form $\exp(\Omega(N)) = \exp(\Omega(\sqrt{n}))$.

Both [10, 35] showed that their respective analyses are tight for many functions g. Hence, the $\exp(\sqrt{n})$ barrier is not merely an artifact of the analysis in these works.

Indeed, at least in the case of $h = \mathrm{OR}_N$, the barrier is inherent to any method that attempts to construct an f satisfying Eq. (1) by assuming nothing about a function $g\colon \{-1,1\}^M \to \{-1,1\}$ other than that g has (one-sided) approximate degree at least d, and then block-composing g with h. To see this, first observe that if $M \leq N$, then for *any* function $g\colon \{-1,1\}^M \to \{-1,1\}$, $(1/N)\sum_{i=1}^{N} g(x_i) + (N-1)/N$ is a polynomial of degree at most $M \leq n^{1/2}$ that approximates $f = \mathrm{OR}_N \circ g$ to error $1 - 1/N$.

Even if $M \geq N$, it is often the case that $\mathrm{OR}_N \circ g$ can be approximated to error $1 - 2^{-\tilde{O}(n^{1/2})}$ by a polynomial of degree $O(n^{1/2})$. Indeed, many functions g with large approximate degree (such as ED_M for example) can be approximated to error $1/3N^2$ by a *ratio* $q_1(x)/q_2(x)$ of two polynomials of logarithmic degree and weight quasi-polynomial in M and N. One can use q_1, q_2 to obtain a polynomial approximator p for $f = h \circ g$ such that $\deg(p) = O(N \log(M \cdot N))$, and p uniformly approximates f to error $1 - 2^{-O(N \cdot \mathrm{polylog}(M))}$. We omit the details for brevity, but the construction can be found in [7] (see also [29, Theorem 6.10]).

Sherstov [29,34] introduced sophisticated and demanding methods that can prove stronger lower bounds for constant-depth circuits than [10,35]. However, his methods apply block-composition multiple times, and crucially exploit alternation in the circuits computing the functions being composed; hence, in the context of Eq. (1), his analysis improves over [10,35] only for circuits of greater depth or bottom fan-in than considered in those works. Furthermore, in [29, Section 9.4] Sherstov provides a detailed discussion on barriers facing his methods. In particular, he shows that the $\exp(\Omega(n^{1/2}))$ barrier is inherent to the class of functions he considers in [29], and is not an artifact of the analysis. He does indicate that his methods might be extendable to break the $\exp(\Omega(n^{1/2}))$ barrier by using circuits of depth 5 or greater.

1.3 An Additional Application

Here, we briefly mention one additional application of our results. By combining Theorem 2 with standard machinery, we obtain an improved separation between the analogues of the complexity classes **PP** and $\mathbf{P}^{\mathbf{NP}}$ in communication complexity. Specifically, for a function $F\colon \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$, let $\mathbf{PP}(F)$

[8] One-sided approximate degree is a measure that is intermediate between approximate degree and threshold degree. One-sided approximate degree lower bounds is crucial to the analyses in [10, 29, 34]. However, we will not explicitly utilize one-sided approximate degree in our own results, so we do not formally define it here. The best known one-sided approximate degree lower bound for an AC^0 function is the same as the best known approximate degree lower bound: $\tilde{\Omega}(M^{2/3})$, exhibited by ED_M [10].

and $\mathbf{P}^{\mathbf{NP}}(F)$ respectively denote the least cost of a **PP** and $\mathbf{P}^{\mathbf{NP}}$ communication protocol for F. For any constant $\delta > 0$, we exhibit an F satisfying $\mathbf{PP}(F) = \Omega(n^{2/3-\delta})$ and $\mathbf{P}^{\mathbf{NP}}(F) = O(\log^c n)$ for some constant c, answering a question of Thaler [35]. This improves over prior work that gave an F satisfying $\mathbf{PP}(F) = \tilde{\Omega}(n^{2/5})$ and $\mathbf{P}^{\mathbf{NP}}(F) = O(\log^2 n)$ [35] and earlier work of Buhrman et al. that gave an F satisfying $\mathbf{PP}(F) = \Omega(n^{1/3})$ and $\mathbf{P}^{\mathbf{NP}}(F) = O(\log^2 n)$ [8]. We direct the interested reader to [35] for further details on this application.

2 Preliminaries

Notation. We work with Boolean functions $f : \{-1,1\}^n \to \{-1,1\}$, where -1 corresponds to logical TRUE and $+1$ corresponds to logical FALSE. For a given Boolean function f, the function $\bar{f} := -f$ denotes its negation. The notation $[n]$ refers to the set $\{0,1,\ldots,n\}$. For any $n \in \mathbb{N}$, fix a canonical injection $[n] \to \{-1,1\}^{\lceil \log(n+1) \rceil}$; we refer to the image of any $i \in [n]$ under this injection as the *binary representation* of i. All logarithms in this work are assumed to be taken in base 2.

Decision Lists and OMB**.** A k-decision list D of length L over the Boolean variables $x_1,\ldots,x_n$ is represented by a list of L pairs (C_0,b_0), (C_1,b_1), $\ldots$, (C_{L-1},b_{L-1}) and a bit b_L where each C_i is a conjunction of width at most k, and each b_i is either -1 or 1. Given any $x \in \{-1,1\}^n$, the value of $D(x)$ is b_i if i is the smallest index such that C_i is made true by x; if no C_i is is true then $D(x) = b_L$.

Any k-decision list of length L is computed by a depth three circuit of size $O(L)$ and bottom fan-in $O(k)$. Indeed, letting $S = \{i \leq L : b_i = -1\}$, the circuit is

$$b_L \vee \bigvee_{i \in S} \left(C_i(x) \wedge \bar{C}_1(x) \wedge \cdots \wedge \bar{C}_{i-1}(x) \right).$$

To see that this is indeed a circuit of depth three with bottom fan-in $O(k)$, observe that for any conjunction C_i of width k, $\bar{C}_i$ is computed by a disjunction of width k.

Let $\mathrm{OMB} : \{-1,1\}^N \to \{-1,1\}$ denote a specific 1-decision list known as ODD-MAX-BIT, defined as follows. For $i = 1,2,\ldots,N$, the conjunction $C_i(x) = x_{N-i}$ and $b_i = (-1)^{N-i}$. Finally, define $b_N = 1$. OMB can be equivalently defined in the following manner. On input $x = (x_1,\ldots,x_N)$, let $\beta(x)$ denote the largest index i such that $x_i = -1$, and let $\beta(x) = 0$ if no such index exists. Then

$$\mathrm{OMB}(x_1,\ldots,x_N) = \begin{cases} -1 \text{ if } \beta(x) \text{ is odd} \\ 1 \text{ otherwise.} \end{cases}$$

Beigel [6] showed that OMB has high approximate degree, even when the error parameter is exponentially close to 1. Specifically:

Theorem 6 (Beigel [6])**.** There exists a constant $c > 0$ for which the following holds. Let $p(x)$ be a polynomial of degree at most d such that $|p(x)| \geq 1$ and $p(x) \cdot \mathrm{OMB}_N(x) \geq 0$ for all $x \in \{-1,1\}^N$. Then there exists an $x \in \{-1,1\}^N$ such that $|p(x)| \geq 2^{cN/d^2}$. In particular, $\widetilde{\deg}_\varepsilon(\mathrm{OMB}_N) \geq d$ for some $\varepsilon = 1 - 2^{\Omega(N/d^2)}$.

To prove Theorem 6, Beigel iteratively constructs a sequence of inputs $x^0, x^1, \ldots, x^{cN/d^2}$ for which $|p(x^{t+1})| \geq 2 \cdot |p(x^t)|$. He obtains these inputs by repeatedly applying the following lemma, which we will also make use of directly.

Lemma 7 (Beigel [6])**.** Let $d, N \in \mathbb{N}$ and let $\ell \geq 10d^2$ such that N/ℓ is an integer. Consider the increasing family of sets $S_1 \subset S_2 \subset \cdots \subset S_{N/\ell} \subseteq \{-1,1\}^n$ defined by

$$S_0 = \{1^N\}, S_1 = \{x : x_i = 1 \ \ \forall i > \ell\}, \ldots, S_t = \{x : x_i = 1 \ \forall i > t\ell\}, \ldots, S_{N/\ell} = \{-1,1\}^N.$$

Let $p(x)$ be a polynomial of degree at most d such that $p(x) \cdot \mathrm{OMB}_N(x) > 0$ for all $x \in S_{t+1} \setminus S_t$. Let $z \in S_t$. Then there exists a $z' \in S_{t+1} \setminus S_t$ such that $|p(z')| \geq 2 \cdot |p(z)|$.

ELEMENT DISTINCTNESS. Aaronson and Shi [1], exhibit a function known as $\mathrm{ED}_M : \{-1,1\}^M \to \{-1,1\}$ that is computed by a polynomial size CNF formula of width $O(\log M)$, and satisfies $\widetilde{\deg}_{2/3}(\mathrm{ED}_M) = \Omega((M/\log M)^{2/3})$.[9] This is the best known lower bound (up to logarithmic factors) on the $(1/3)$-approximate degree of *any* function in AC^0.

Specifically, ED_M is defined as follows. Assume for simplicity that $M = m \cdot \log_2 m$ for some $m \in \mathbb{N}$. ED_M takes M bits as input, and interprets its input as m blocks $(x_1, \ldots, x_m)$ with each block consisting of $\log_2 m$ bits. Each block is interpreted as a number in the range $\{1, \ldots, m\}$, and ED_M evaluates to -1 on x if and only if all m numbers are distinct.

Theorem 8 (Aaronson and Shi [1] and Ambanis [2])**.** There exists some constant $c > 0$ such that the following holds. Let $p\colon \{-1,1\}^M \to \{-1,1\}$ be any polynomial of degree at most $c \cdot (M/\log M)^{2/3}$. Then there exists an $x \in \{-1,1\}^M$ such that $|p(x) - \mathrm{ED}_M(x)| > 1/3$. The same statement holds with $\overline{\mathrm{ED}}_M$ in place of ED_M.

3 Intuition and Discussion of Theorems 1 and 2

3.1 Overview of Our Functions

For clarity, we focus in this discussion on the function f that we exhibit in Theorem 2. As mentioned in Section 1, this function is complicated to define. Hence, before formally defining f, we provide here some motivation for our definition.

Recall that [35] showed that $\widetilde{\deg}_\varepsilon(\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M) = \tilde{\Omega}(M^{2/3})$ for $\varepsilon = 1 - 2^{-N}$, Moreover, this lower bound is essentially tight for $\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$: there is in fact a polynomial p of degree $O(\log M)$ that approximates $\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ to error $\varepsilon = 1 - 2^{-O(N \log M)}$.

Our goal is to modify $\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ to obtain an $f : \{-1,1\}^n \to \{-1,1\}$ that is much harder to approximate by low-degree polynomials, while still ensuring that f is computed by an $O(\log^2 n)$ decision list. Specifically, we will require, for some large constant k and small constant $\delta > 0$, $\widetilde{\deg}_\varepsilon(f) = \Omega(M^{2/3-\delta})$ for $\varepsilon = 1 - 2^{-N^k}$.

A natural first attempt to construct such an f is to block-compose $\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ with the parity function on k variables. Specifically, let k be some constant, and consider the following function on $k \cdot N \cdot M$ variables: $\oplus_k \circ \mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$, where $\oplus$ denotes the parity function. However, this function is still too easy to approximate: there is a polynomial of degree $O(k \log M)$ that approximates $\oplus_k \circ \mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ to error $1 - 2^{-O(kN \log M)}$. Indeed, letting p be the polynomial approximation to $\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ described above, the polynomial $q(x_1, \ldots x_k) := 2^{-k} \prod_{i=1}^k p(x_i)$ does the trick.

We instead define f to be "just different enough" from $\oplus_k \circ \mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ to foil this construction of an approximating polynomial. Specifically, our f will first "pre-process" its input $(x_1, \ldots, x_k)$, before feeding

[9] This bound is tight up to a logarithmic factor, as $\widetilde{\deg}_{2/3}(\mathrm{ED}_M) = O(M^{2/3} \log^{1/3}(M))$ [3].

it into $\oplus_k \circ \mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$. The pre-processing step will introduce dependencies between blocks, so that an approximating polynomial for f will be unable to treat them independently in the manner of q.

In more detail, f will interpret its input x as k blocks, $x_1, \ldots, x_k$ (we will refer to $x_1, \ldots, x_k$ as "super-blocks", since each x_i will itself be interpreted as consisting of N blocks, which will themselves each be interpreted as consisting of M "sub-blocks"). For expository purposes, we focus in the remainder of this section on the case $k = 2$, so that there are only two super-blocks x_1, x_2 (The full construction is defined inductively, and described in Section 5.2). Assume for simplicity that $N + 1$ is a power of 2. The two super-blocks will not contain the same number of bits: x_1 will contain $N \cdot M$ bits, while x_2 will contain $N \cdot M \cdot \log(N+1)$ bits. We will ultimately treat x_1 as an input to $\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$; accordingly, let us interpret x_1 as consisting of N blocks, each containing M bits, so that we can write $x_1 = (x_{1,1}, \ldots, x_{1,N}) \in \left(\{-1,1\}^M\right)^N$. Let $\gamma(x_1) \in \{-1,1\}^{\log N}$ be the binary representation of the largest integer j satisfying $\overline{\mathrm{ED}}(x_{1,j}) = -1$, and let $\gamma(x_1) = 0$ if no such j exists. That is, $\gamma(x_1)$ is the index of the "leading TRUE bit" that gets fed into OMB_N when evaluating $(\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M)(x_1)$.

Similarly, we interpret x_2 as consisting of N blocks. However, each block now contains $M \log(N+1)$ bits, and is comprised of M sub-blocks, each consisting of $\log(N+1)$ bits. Let $\mathrm{EQ}_{\gamma(x_1)} \colon \{-1,1\}^{\log(N+1)} \to \{-1,1\}$ denote the function that outputs -1 if and only if its input equals $\gamma(x_1)$. Finally, let $u = (u_1, \ldots u_N) \in (\{-1,1\}^M)^N$ denote the vector obtained by applying $\mathrm{EQ}_{\gamma(x_1)}$ to each sub-block of x_2. That is, u is the vector obtained by first "pre-processing" each sub-block of x_2 with an "equality test" $\mathrm{EQ}_{\gamma(x_1)}$ that is determined by x_1. Finally, we define

$$f = \left((\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M)(x_1)\right) \oplus \left((\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M)(u)\right). \tag{2}$$

Notice that the dependence of this pre-processing function on x_1 is actually quite mild: u only depends on the "leading TRUE bit" that gets fed into OMB_N when evaluating $\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M(x_1)$. This mild dependence is what allows f to be computed by an $O(\log^2 n)$ decision list.

It is not hard to see that an equivalent way to write f (that moreover helps reveal its structure as an $O(\log^2 n)$-decision list) is:

$$\begin{aligned} f(x_1, x_2) = \mathrm{OMB}_{N^2+2N}(&\overline{\mathrm{ED}}(u_1), \overline{\mathrm{ED}}(u_2), \ldots, \overline{\mathrm{ED}}(u_N), \\ &\overline{\mathrm{ED}}(x_{1,1}), \overline{\mathrm{ED}}(x_{1,1}) \wedge \overline{\mathrm{ED}}(u_1), \overline{\mathrm{ED}}(x_{1,1}) \wedge \overline{\mathrm{ED}}(u_2), \ldots, \overline{\mathrm{ED}}(x_{1,1}) \wedge \overline{\mathrm{ED}}(u_N), \\ &\vdots \\ &\overline{\mathrm{ED}}(x_{1,N}), \overline{\mathrm{ED}}(x_{1,N}) \wedge \overline{\mathrm{ED}}(u_1), \ldots, \overline{\mathrm{ED}}(x_{1,N}) \wedge \overline{\mathrm{ED}}(u_N)), \end{aligned} \tag{3}$$

where u is defined as above. It turns out that Representation (2) of f is useful for establishing lower bounds on the approximate degree of f, while Representation (3) is more useful for constructing approximating polynomials for f, and gaining intuition about f (cf. Section 5.2.3). In particular, Representation (3) suggests a natural method for approximating f: treat it as a 1-decision list over $N^2 + 2N$ "derived" variables (and then use an optimal method of approximating 1-decision lists, which are well-understood). The proof of our lower bound (Theorem 2) implicitly shows that this simple approach is essentially optimal. The next subsection briefly explains the details of this approximation method.

3.2 A Nearly Matching Upper Bound

We begin by giving the well-known sign-representing polynomial for OMB_N itself. Define $p \colon \{-1,1\}^N \to \mathbb{R}$ via

$$p(x_1, \ldots, x_N) := 1 + \sum_{i=1}^{N} (-2)^i \cdot (1 - x_i)/2.$$

It is easy to see that $\mathrm{OMB}_N(x) \cdot p(x) > 0$ for all $x \in \{-1, 1\}^N$, and in fact $2^{-N-1} \cdot p(x)$ approximates OMB_N to error $\varepsilon = 1 - 2^{-N-1}$.

We now turn to constructing an approximant for the function $\mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$. Our starting point is a polynomial q of degree $O(M^{2/3})$ satisfying the following two properties (cf. [35]).

$$q(x) = 0 \text{ for all } x \in \overline{\mathrm{ED}}_M^{-1}(+1). \tag{4}$$

$$1 \leq q(x) \leq 2 \text{ for all } x \in \overline{\mathrm{ED}}_M^{-1}(-1). \tag{5}$$

Denoting an $(N \cdot M)$-bit input as $(x_1, \ldots, x_N) \in \left(\{-1, 1\}^M\right)^N$, it is easy to check that

$$F(x_1, \ldots, x_N) = \mathrm{sgn}(g(x_1, \ldots, x_N)), \text{ where } g(x_1, \ldots, x_N) = 1 + \sum_{i=1}^{N} (-3)^i \cdot q(x_i).$$

In fact, $3^{-N-1} \cdot g(x)$ approximates F to error $1 - 3^{-N-1}$, and has degree equal to that of q.

Recall (cf. Eq. (3)) that in the case $k = 2$, our function can be written as

$$\mathrm{OMB}_{N^2+2N}(\overline{\mathrm{ED}}(u_1), \overline{\mathrm{ED}}(u_2), \ldots, \overline{\mathrm{ED}}(x_{1,N}) \wedge \overline{\mathrm{ED}}(u_N)).$$

Using techniques similar to the above, one can obtain a degree $\tilde{O}(M^{2/3} \log N)$ polynomial p that approximates this function to error $1 - 2^{-O(N^2)}$. Our lower bound will show this approximation is essentially optimal.

3.3 Prospects for Further Improved Lower Bounds

Recall that the best known lower bound on the approximate degree of any function $f\colon \{-1, 1\}^n \to \{-1, 1\}$ in AC^0 is $\tilde{\Omega}(n^{2/3})$, exhibited by the function ED. A remarkable property of the lower bound in Theorem 2 is that it essentially matches this degree bound, even for approximations that are allowed error *super-exponentially* close to 1 (i.e., error as large as $1 - 2^{-n^\Gamma}$ for any constant $\Gamma > 0$).

Clearly, improving the degree bound in Theorem 1 beyond $\Omega(n^{2/3})$ will require first exhibiting a function in AC^0 whose approximate degree is substantially larger than that of ED. A prime candidate is the k-sum function for $k \geq 3$ (see [3] for a definition of this function; here we merely note that ED is equivalent to k-sum for $k = 2$). In our prior work, we conjectured that for any $k = O(1)$, the approximate degree of the k-sum function is $\tilde{\Omega}(n^{k/(k+1)})$ (an $O(n^{k/(k+1)})$ upper bound was proved in [3]). We further conjecture that for any constants $\delta, \Gamma > 0$, replacing ED with k-sum in the definition the function we constructed to prove Theorem 2 yields a function that cannot be uniformly approximated to error $1 - 2^{-n^\Gamma}$ by any polynomial of degree at most $n^{k/(k+1)-\delta}$. If true, this would imply, for any constant $\delta' > 0$, the existence of a depth four circuit (and a depth three circuit of quasipolynomial size) that has threshold weight, margin complexity, and Majority-of-Threshold circuit size all at least $\exp(\Omega(n^{1-\delta'}))$.

It may also be possible to slightly sharpen our analysis. Specifically, we conjecture that it is possible to improve the parameter c_k appearing in the statement of our Theorem 20 (cf. Section 5.2), from $c_k = \Theta(2^{-k})$

to $c_k = \text{poly}(k)$. If so, then by setting $k = \Theta(\log M)$ and $N = \Theta(1)$ in the statement of Theorem 20, one would obtain a Boolean circuit of constant depth and polynomial size, with complexity $\exp(\tilde{\Omega}(n^{2/3}))$ (this would improve over our stated bound of $\exp(\Omega(n^{2/3-\delta}))$).

Furthermore, this suggests the following candidate AC^0 function that may have essentially maximal complexity $\exp(\tilde{\Omega}(n))$: within the construction of Theorem 20, set $k = \Theta(\log M)$, and $N = \Theta(1)$, and replace ED_M with an AC^0 function conjectured to have approximate degree $\tilde{\Omega}(M)$, such as the SURJECTIVITY function (see [5] for the definition of this function).

4 Proof of Theorem 1

Before stating and proving Theorem 1, we consider an easier statement, the proof of which is much cleaner, while still capturing the main ideas of the general case.

4.1 Simplified Statement and its Proof

The function f that we exhibit in Theorem 1 is defined over k "superblocks", where k is an arbitrarily large constant (see Section 3.1 for motivation for the superblock terminology). Here, we consider the simpler case of exactly $k = 2$ superblocks.

Notation. Recall (cf. Section 2) that for $x = (x_1, \ldots, x_N) \in \{-1,1\}^N$, $\beta(x)$ denotes the largest index $i = 1, \ldots, N$ such that $x_i = -1$ (or $\beta(x) = 0$ if none exists). Assume for simplicity that $N+1$ is a power of two. Given an input $(x, y) \in \{-1,1\}^N \times (\{-1,1\}^{\log(N+1)})^N$, we interpret y as consisting of N blocks $y_1, \ldots, y_N$, each consisting of $\log(N+1)$ bits. Let $\text{EQ}_{\beta(x)} \colon \{-1,1\}^{\log(N+1)} \to \{-1,1\}$ denote the function that outputs -1 if and only if its input equals the binary representation of $\beta(x)$. Finally, let $u = (u_1, \ldots u_N) \in \{-1,1\}^N$ denote the vector obtained by applying $\text{EQ}_{\beta(x)}$ to each block of y. That is, u is the vector obtained by first "pre-processing" each block of y with an "equality test" $\text{EQ}_{\beta(x)}$ that is determined by x.

Function Definition. Define F via:

$$F(x,y) = \text{OMB}_N(x_1, \ldots, x_N) \oplus \text{OMB}_N(u_1(x,y_1), \ldots, u_N(x,y_N)) \tag{6}$$

$$= \text{OMB}_N(x_1, \ldots, x_N) \oplus \text{OMB}_N(\text{EQ}_{\beta(x)}(y_1), \ldots, \text{EQ}_{\beta(x)}(y_N)). \tag{7}$$

Proposition 9. There exists a constant c for which the following holds. Let $d, n \in \mathbb{N}$ where $n = N + N\log(N+1)$. Let p be a polynomial of degree at most d such that $|p(x,y)| \geq 1$ and $p(x,y) \cdot F(x,y) > 0$ for all $(x,y) \in \{-1,1\}^n$. Then there exists an $(x,y) \in \{-1,1\}^n$ such that $|p(x,y)| \geq 2^{(cN/d^2)^2}$.

To ease notation below, we will identify each block $y_i \in \{-1,1\}^{\log(N+1)}$ with the number in $[N]$ for which y_i is the binary representation. That is, while we will write each y_i as though it were a number $0, 1, \ldots, N$, it should always be thought of as the binary string representing that number.

Proof Idea. Let $p(x,y)$ be a polynomial of degree at most d that agrees with F in sign. Building on Beigel's proof of Theorem 6, we iteratively apply Lemma 7 to construct a sequence of inputs to the polynomial p, such that evaluating p on each point yields a value of (at least) twice the magnitude of the previous evaluation. By choosing these inputs carefully (and crucially exploiting the "pre-processing" step in the definition of F_2 that transforms y into the vector $u(x,y)$, before feeding it into OMB_N), we can apply Lemma 7 a total of $(cN/d^2)^2$ times. This is a quadratic improvement over the number of times Beigel is

able to apply Lemma 7 to OMB_N itself. Podolskii [25] used related ideas to obtain a lower bound for a different function, but we are able to avoid the significant quantitative losses that are inherent to his approach.

In more detail, recall that Beigel's lower bound argument (cf. Theorem 6) for OMB_N started with the input $x^0 = 1^N$, and iteratively applied Lemma 7 to obtain inputs $x^1, \ldots, x^{cN/d^2}$ such that $|p(x^t)| \geq 2|p(x^{t-1})|$ for all $t \geq 1$. Roughly speaking, the first input to F that we construct is a point (x^1, y^0) such that $u(x^1, y^0) = 1^N$ and the last $N - 10d^2$ bits of x^1 are all set to 1. Since $u(x^1, y^0)$ is fed into OMB_N in the definition of F, we are able to apply Beigel's argument (Theorem 6) to obtain an input (x^1, y^1) such that $|p(x^1, y^1)| \geq 2^{cn/d^2} \cdot |p(x^1, y^0)|$. We then "use" the second block of $10d^2$ bits of the first superblock to "clean up" u, in the following sense: we find an x^2 whose last $N - 20d^2$ bits are all equal to 1, such that $u(x^2, y^1) = 1^N$ and $|p(x^2, y^1)| \geq |p(x^1, y^1)|$. This enables us to apply Beigel's argument (Theorem 6) a second time, finding an input (x^2, y^2) such that $|p(x^2, y^2)| \geq 2^{cn/d^2} \cdot |p(x^1, y^1)|$. We then use the third $10d^2$ bits of the first superblock to "clean up" u yet again, and repeat. We can continue the argument until we have "used up" all the bits of the first superblock, at which point we have obtained the desired lower bound.

Proof of Proposition 9. Let $\ell = 10d^2$ and consider the increasing family of sets $S_0 \subset S_1 \subset \cdots \subset S_{N/\ell} \subseteq \{-1, 1\}^N$ defined as in Lemma 7. Let p be a polynomial of degree at most d such that $p(x, y) \cdot F(x, y) > 0$ for all $(x, y) \in \{-1, 1\}^n$. We iteratively construct a sequence of inputs $(x^0, y^0), (x^1, y^1), \ldots, (x^{N/\ell}, y^{N/\ell})$ to p such that:

- Each $x^t \in S_t$ and each $y^t \in [t\ell]^N$,
- $|p(x^0, y^0)| \geq 1$, and
- $|p(x^{t+1}, y^{t+1})| \geq 2^{cN/d^2} \cdot |p(x^t, y^t)|$ for each $t = 0, \ldots, N/\ell - 1$.

At the conclusion of this process, we obtain an input $(x^{N/\ell}, y^{N/\ell})$ such that $|p(x^{N/\ell}, y^{N/\ell})| \geq 2^{(cN/d^2)^2}$.

We may take as the first input (x^0, y^0) the point $(1^N, 0^N)$. We construct the remaining inputs (x^t, y^t) iteratively. The following claim formalizes this iterative process.

Claim 10. Let p be a polynomial of degree at most d and suppose $p(x, y) \cdot F(x, y) > 0$ for all $(x, y) \in \{-1, 1\}^n$. Let (x^t, y^t) be an input with $x^t \in S_t$ and $y^t \in [t\ell]^N$. Then there exists an input (x^{t+1}, y^{t+1}) such that $|p(x^{t+1}, y^{t+1})| \geq 2^{cN/d^2} \cdot |p(x^t, y^t)|$, where $x^{t+1} \in S_{t+1}$ and $y^{t+1} \in [(t+1)\ell]^N$.

Proof. We prove the claim in two steps. First, we show that there exists an $x^{t+1} \in S_{t+1}$ for which $|p(x^{t+1}, y^t)| \geq |p(x^t, y^t)|$. Second, we show that there exists a $y^{t+1} \in [(t+1)\ell]^N$ such that $|p(x^{t+1}, y^{t+1})| \geq 2^{cN/d^2} \cdot |p(x^{t+1}, y^t)|$. Putting these steps together yields $|p(x^{t+1}, y^{t+1})| \geq 2^{cN/d^2} \cdot |p(x^t, y^t)|$.

Step 1. We examine the function $F(x, y^t)$ (viewed as a function only of x). By construction, each block $y^t_i \leq t\ell$. Thus, $\mathrm{EQ}_{\beta(x)}(y^t_i) = 1$ for all $x \in S_{t+1} \setminus S_t$, and hence

$$\mathrm{OMB}_N(\mathrm{EQ}_{\beta(x)}(y^t_1), \ldots, \mathrm{EQ}_{\beta(x)}(y^t_N)) = \mathrm{OMB}_N(1^N) = 1$$

for all such inputs. As a result, $F(x, y^t) = \mathrm{OMB}_N(x)$ whenever $x \in S_{t+1} \setminus S_t$.

Now consider the polynomial $q : \{-1, 1\}^N \to \mathbb{R}$ defined by $q(x) = p(x, y^t)$. Then $q(x) \cdot \mathrm{OMB}_N(x) > 0$ for all $x \in S_{t+1} \setminus S_t$. By assumption, $x^t \in S_t$. Thus, by Lemma 7, there exists an $x^{t+1} \in S_{t+1}$ such that $|q(x^{t+1})| \geq 2 \cdot |q(x^t)|$. Unpacking the definition of q, we see that in particular, $|p(x^{t+1}, y^t)| \geq |p(x^t, y^t)|$.

Step 2. We now show that there exists a $y^{t+1} \in [(t+1)\ell]^N$, such that $|p(x^{t+1}, y^{t+1})| \geq 2^{cN/d^2} \cdot |p(x^{t+1}, y^t)|$. For $w \in \{-1, 1\}^N$, define the string y_w by $(y_w)_i = \beta(x^{t+1})$ if $w_i = -1$ and $(y_w)_i = y^t_i$ if $w_i = 1$. Note that since $\beta(x^{t+1}) \leq (t+1)\ell$ and each $y^t_i \in [t\ell]$, we have that $y_w \in [(t+1)\ell]^N$ for every $w \in \{-1, 1\}^N$. Consider the function $E : \{-1, 1\}^N \to \{-1, 1\}$ defined by $E(w) := F(x^{t+1}, y_w)$, and observe that

$$E(w) = \mathrm{OMB}_N(x^{t+1}) \oplus \mathrm{OMB}_N(w).$$

Now consider the polynomial $r(w) := p(x^{t+1}, y_w)$. Observe that y_w is an affine function of w, i.e., we can write

$$r(w) = p\left(x^{t+1}, \left(\frac{1-w_1}{2}\right)\cdot\beta(x^{t+1}) + \left(\frac{1+w_1}{2}\right)\cdot y_1^t, \ldots, \left(\frac{1-w_N}{2}\right)\cdot\beta(x^{t+1}) + \left(\frac{1+w_N}{2}\right)\cdot y_N^t\right).$$

Thus r is an polynomial with $\deg r \leq \deg p \leq d$. Moreover, $r(w)\cdot E(w) > 0$ for all $w \in \{-1,1\}^N$. Since E is either the function OMB_N or its negation, we conclude by Theorem 6 that there exists a w^* such that $|r(w^*)| \geq 2^{cN/d^2}\cdot|r(1^N)|$. Setting $y^{t+1} := y_{w^*}$ thus yields

$$|p(x^{t+1}, y^{t+1})| = |r(w^*)| \geq 2^{cN/d^2}\cdot|r(1^N)| = 2^{cN/d^2}\cdot|p(x^{t+1}, y^t)|,$$

as we wanted to show. □

With Claim 10 established. we conclude the proof of Proposition 9.

□

4.2 Full Proof of Theorem 1

The proof begins by extending the "two-superblock" function F constructed in Section 4.1 (cf. Eq. (6)), to construct a k-superblock function F_k for any constant number of superblocks $k \geq 2$.

4.2.1 Construction of the Function F_k

First, fix a parameter $N \in \mathbb{N}$ and assume for simplicity that $N+1$ is a power of 2. The construction of our function F_k is inductive, and begins with the following sequence of auxiliary functions $u_1, u_2, \ldots, u_k$. For each $i = 1, \ldots, k$, each function

$$u_i : \{-1,1\}^N \times (\{-1,1\}^N \times (\{-1,1\}^{\log(N+1)})^N) \times \cdots \times (\{-1,1\}^N \times (\{-1,1\}^{(i-1)\cdot\log(N+1)})^N) \to \{-1,1\}^N.$$

For $i = 1, \ldots, k$, let $s_i = (s_{i,1}, \ldots, s_{i,N})$ denote an arbitrary input in $\{-1,1\}^N$, and z_i denote an arbitrary input in $(\{-1,1\}^{(i-1)\cdot\log(N+1)})^N$.

The auxiliary functions u_i are defined recursively as follows.

$$u_1(s_1) = s_1 = (s_{1,1}, \ldots, s_{1,N})$$

$$\begin{aligned} u_2(s_1, (s_2, z_2)) &= (s_{2,1} \wedge \mathrm{EQ}_{\beta(u_1)}(z_{2,1}), \ldots, s_{2,N} \wedge \mathrm{EQ}_{\beta(u_1)}(z_{2,N})) \\ &= (s_{2,1} \wedge \mathrm{EQ}_{\beta(s_1)}(z_{2,1}), \ldots, s_{2,N} \wedge \mathrm{EQ}_{\beta(s_1)}(z_{2,N})) \end{aligned}$$

$$\vdots$$

$$u_k(s_1, (s_2, z_2), \ldots, (s_k, z_k)) = (s_{k,1} \wedge \mathrm{EQ}_{\beta(u_1)\circ\cdots\circ\beta(u_{k-1})}(z_{k,1}), \ldots, s_{k,N} \wedge \mathrm{EQ}_{\beta(u_1)\circ\cdots\circ\beta(u_{k-1})}(z_{k,N}))$$

Here, the notation $\circ$ denotes string concatenation. The function u_k should be interpreted as the bitwise AND of $(s_{k,1}, \ldots, s_{k,N})$ with a vector of equality tests between $(z_{k,1}, \ldots, z_{k,N})$ and the complete list of the indices of the "leading TRUE bits" feeding into OMB_N from each of the previous super-blocks $i = 1, \ldots, (k-1)$.

We are now ready to define F_k. In what follows, $F_1, F_2, \dots F_k$ will denote functions such that

$$F_i : \{-1,1\}^N \times (\{-1,1\}^N \times (\{-1,1\}^{\log(N+1)})^N) \times \cdots \times (\{-1,1\}^N \times (\{-1,1\}^{(i-1)\cdot\log(N+1)})^N) \to \{-1,1\}.$$

The construction is recursive. Define:

$$\begin{aligned} F_1(s_1) &= \mathrm{OMB}_N(u_1) \\ &= \mathrm{OMB}_N(s_{1,1}, \dots, s_{1,N}), \end{aligned}$$

$$\begin{aligned} F_2(s_1, (s_2, z_2)) &= F_1(s_1) \oplus \mathrm{OMB}_N(u_2) \\ &= \mathrm{OMB}_N(s_{1,1}, \dots, s_{1,N}) \oplus \mathrm{OMB}_N(s_{2,1} \wedge \mathrm{EQ}_{\beta(s_1)}(z_{2,1}), \dots, s_{2,N} \wedge \mathrm{EQ}_{\beta(s_1)}(z_{2,N})), \end{aligned}$$

$$\vdots$$

$$\begin{aligned} F_k(s_1, (s_2, z_2), \dots, (s_k, z_k)) &= F_{k-1}(s_1, (s_2, z_2), \dots, (s_{k-1}, z_{k-1})) \oplus \mathrm{OMB}_N(u_k) \\ &= F_{k-1}(s_1, (s_2, z_2), \dots, (s_{k-1}, z_{k-1})) \oplus \mathrm{OMB}_N(\dots, s_{k,i} \wedge \mathrm{EQ}_{\beta(u_1)\circ\cdots\circ\beta(u_{k-1})}(z_{k,i}), \dots) \end{aligned}$$

Remark 11. We clarify that the function F_2 defined in this section differs very slightly from the definition of F given in Section 4.1 (cf. Eq. (6)) in that Eq. (6) did not involve the variables $s_2 \in \{-1,1\}^N$. We omitted the variables s_2 in Eq. (6) for simplicity and clarity, since they are not needed to prove a lower bound on the approximate degree of F_2 itself (cf. Proposition 9). We do, however, need the variables s_k to prove our lower bound for F_k for $k \geq 3$.

4.2.2 Representing F_k as a Decision List

The function F_k is represented by a $O(k^2 \log N)$-decision list

$$(C_0, b_0), (C_1, b_1), \dots, (C_{(N+1)^k-2}, b_{(N+1)^k-2}), b_{(N+1)^k-1},$$

where (C_i, b_i) are as follows:

$$C_0(s;z) = s_{1,N} \wedge (s_{2,N} \wedge \mathrm{EQ}_N(z_{2,N})) \wedge \cdots \wedge (s_{k,N} \wedge \mathrm{EQ}_{\underbrace{N \circ \cdots \circ N}_{k-1 \text{ times}}}(z_{k,N})); \qquad b_0 = (-1)^{k \cdot N}$$

$$C_1(s;z) = s_{1,N} \wedge (s_{2,N} \wedge \mathrm{EQ}_N(z_{2,N})) \wedge \cdots \wedge (s_{k,N-1} \wedge \mathrm{EQ}_{\underbrace{N \circ \cdots \circ N}_{k-1 \text{ times}}}(z_{k,N-1})); \qquad b_1 = (-1)^{k \cdot N - 1}$$

$$\vdots$$

$$C_{N-1}(s;z) = s_{1,N} \wedge (s_{2,N} \wedge \mathrm{EQ}_N(z_{2,N})) \wedge \cdots \wedge (s_{k,1} \wedge \mathrm{EQ}_{\underbrace{N \circ \cdots \circ N}_{k-1 \text{ times}}}(z_{k,1})); \qquad b_{N-1} = (-1)^{(k-1) \cdot N + 1}$$

$$C_N(s;z) = s_{1,N} \wedge (s_{2,N} \wedge \mathrm{EQ}_N(z_{2,N})) \wedge \cdots \wedge (s_{k-1,N} \wedge \mathrm{EQ}_{\underbrace{N \circ \cdots \circ N}_{k-2 \text{ times}}}(z_{k-1,N})); \qquad b_N = (-1)^{(k-1) \cdot N}$$

$$C_{N+1}(s;z) = s_{1,N} \wedge (s_{2,N} \wedge \mathrm{EQ}_N(z_{2,N})) \wedge$$
$$\cdots \wedge (s_{k-1,N-1} \wedge \mathrm{EQ}_{\underbrace{N \circ \cdots \circ N}_{k-2 \text{ times}}}(z_{k-1,N-1})) \wedge (s_{k,N} \wedge \mathrm{EQ}_{\underbrace{N \circ \cdots \circ N}_{k-2 \text{ times}} \circ (N-1)}(z_{k,N})); \; b_{N+1} = (-1)^{k \cdot N - 1}$$

$$\vdots$$

$$C_{(N+1)^k-2}(s;z) = (s_{k,1} \wedge \mathrm{EQ}_{\underbrace{0 \circ \cdots \circ 0}_{k-1 \text{ times}}}(z_{k,1})); \qquad b_{(N+1)^k-2} = -1$$

$$b_{(N+1)^k-1} = 1.$$

Observe that each C_ℓ in the above is indeed a conjunction over $O(k^2 \log N)$ variables (here, we are using the fact that, for any integer $i > 0$ and any fixed string $\tau \in [N]^m$, the function $\mathrm{EQ}_\tau \colon \{-1,1\}^{i \log(N+1)} \to \{-1,1\}$ is a conjunction of width $i \log(N+1)$.)

In general, suppose $\ell = a_{k-1}(N+1)^{k-1} + a_{k-2}(N+1)^{k-2} + \cdots + a_0$ where each $0 \le a_i \le N$. Let $\tilde{a}_i = N - \tilde{a}_i$ for each $i = 0, 1, \ldots, k-1$. then C_ℓ is given by $C_\ell^1 \wedge C_\ell^2 \wedge \cdots \wedge C_\ell^k$ where C_ℓ^i is an empty clause if $\tilde{a}_i = 0$ and otherwise

$$C_\ell^i(s;z) = (s_{i,\tilde{a}_{k-i}} \wedge \mathrm{EQ}_{\tilde{a}_{k-1} \circ \tilde{a}_{k-2} \circ \cdots \circ \tilde{a}_{k-i+1}}(z_{i,\tilde{a}_{k-i}})).$$

The bit $b_\ell = (-1)^{\tilde{a}_{k-1} + \tilde{a}_{k-2} + \cdots + \tilde{a}_0}$.

Since, for any constant $k > 0$, F_k is an $O(\log n)$ decision list of polynomial length, it can be computed by a polynomial size circuit of depth three and logarithmic bottom fan-in.

4.2.3 The Main Proposition

The goal of this section is to prove the following generalization of Proposition 9.

Proposition 12. There exists a universal constant $c > 0$ such that for each $k \in \mathbb{N}$, there exists a $c_k \ge c \cdot 4^{-k^2}$ for which the following holds. Let $d, n \in \mathbb{N}$ where

$$n = N \cdot \sum_{i=1}^{k} (1 + (i-1) \cdot \log(N+1)) = O(k^2 \cdot N \cdot \log N).$$

Let p be a polynomial of degree at most d such that $p(x) \cdot F_k(x) > 0$ for all $x \in \{-1,1\}^n$. Then there exists an $x \in \{-1,1\}^n$ such that $|p(x)| \ge 2^{c_k (N/d^2)^k} \cdot |p(1^n)|$.

Theorem 1 follows easily from Proposition 12.

Proof of Theorem 1, assuming Proposition 12. Let $k = \lceil \Gamma/\delta \rceil$. Observe that F_k is defined on $\{-1,1\}^n$ where $n = O(k^2 N \log N)$. Fix a polynomial p of degree $d = n^{1/2-\delta}$, and suppose that $p(x) \cdot F_k(x) > 0$ for all $x \in \{-1,1\}^n$. By Proposition 12, there exists an $x \in \{-1,1\}^n$ such that

$$|p(x)| \geq 2^{c_k (N/d^2)^k} \cdot |p(1^n)| > 2^{\Omega_k(1) \cdot N^{2 \cdot k \cdot \delta} / \log^{2k} N} \cdot |p(1^n)| > 2^{n^{\Gamma+1}} \cdot |p(1^n)| > 2 \cdot 2^{n^\Gamma} \cdot |p(1^n)|,$$

where the third inequality holds for sufficiently large n. Hence, if $|p(1^n)| > 2^{-n^\Gamma}$, then $|p(x)| > 2$. It follows that p cannot approximate F_k uniformly to within error less than $1 - 2^{-n^\Gamma}$. □

Proof of Proposition 12. The proof is by induction on k. Beginning with $k = 1$, note that the function F_1 is just the OMB_N function. Hence, if p is a polynomial of degree at most d for which $p(x) \cdot F_1(x) > 0$ for all $x \in \{-1,1\}^N$, then by Theorem 6 there exists a universal constant $c > 0$ such that there is an $x \in \{-1,1\}^N$ for which $|p(x)| \geq 2^{cN/d^2} \cdot |p(1^N)|$.

Now assume by way of induction that Proposition 12 holds for F_k, and consider the function F_{k+1}.

Additional Notation. To enable the induction, we need to introduce more detailed notation to represent the inputs to F_{k+1}. Recall that F_{k+1} is defined over a variable set $(s_1, (s_2, z_2), \ldots, (s_{k+1}, z_{k+1}))$ where each $s_i \in \{-1,1\}^N$ and each $z_i \in (\{-1,1\}^{(i-1)\cdot\log(N+1)})^N$. For notational convenience, we make the following relabelings:

$$s_1 \mapsto x$$
$$z_{i,j} \mapsto y_{i,j} \circ w_{i,j} \text{ where } y_{i,j} \in \{-1,1\}^{\log(N+1)} \text{ and } w_{i,j} \in \{-1,1\}^{(i-2)\cdot\log(N+1)}$$

Thus, we can think of F_{k+1} as being defined over variables $(x, (s_2, y_2), (s_3, (y_3, w_3)) \ldots, (s_{k+1}, (y_{k+1}, w_{k+1})))$.

With this notation in mind, we write $F_{k+1}(x; s; y; w)$ as shorthand for

$$F_{k+1}(x, (s_2, y_2), (s_3, (y_3, w_3)), \ldots, (s_{k+1}, (y_{k+1}, w_{k+1}))).$$

Similarly, for a polynomial p, we write $p(x; s; y; w)$ for

$$p(x, (s_2, y_2), (s_3, (y_3, w_3)), \ldots, (s_{k+1}, (y_{k+1}, w_{k+1}))).$$

Here, $x \in \{-1,1\}^N$, while s is shorthand for $s = (s_2, s_3, \ldots, s_{k+1}) \in \left(\{-1,1\}^N\right)^k$, y is shorthand for $(y_2, \ldots, y_{k+1}) \in \left(\left(\{-1,1\}^{\log(N+1)}\right)^N\right)^k$, and w is shorthand for $(w_3, \ldots, w_{k+1})$.

As in the proof of Proposition 9, to ease notation, we will also identify any binary string in $\{-1,1\}^{\log(N+1)}$ with the number in $[N]$ for which the string is the binary representation. That is, while we will write any such binary string as though it were a number $0, 1, \ldots, N$, it should always be thought of as the binary string representing that number.

A Different Expression for F_i. The following claim follows straightforwardly from the definition of F_k (cf. Section 4.2.1).

Claim 13. The function F_{k+1} may be written as

$$F_{k+1}(x, (s_2, y_2), (s_3, (y_3, w_3)), \ldots, (s_{k+1}, (y_{k+1}, w_{k+1}))) =$$
$$\mathrm{OMB}_N(\ldots, x_j, \ldots) \oplus F_k(v_2(x, s_2, y_2), v_3(x, s_3, y_3, w_3), \ldots, v_{k+1}(x, s_{k+1}, y_{k+1}, w_{k+1})),$$

where the functions v_i are defined by:

$$v_2(x, s_2, y_2) = (s_{2,1} \wedge \mathrm{EQ}_{\beta(x)}(y_{2,1}), \ldots, s_{2,N} \wedge \mathrm{EQ}_{\beta(x)}(y_{2,N})),$$

$$v_3(x, s_3, y_3, w_3) = ((\ldots, s_{3,j} \wedge \mathrm{EQ}_{\beta(x)}(y_{3,j}), \ldots), \quad w_3),$$

$$v_i(x, s_i, y_i, w_i) = ((\ldots, s_{i,j} \wedge \mathrm{EQ}_{\beta(x)}(y_{i,j}), \ldots), \quad w_i) \qquad \text{for } i = 3, \ldots, k+1.$$

The Main Argument. Just as in Lemma 7 and Proposition 9, we let $\ell = 10d^2$ and consider the increasing family of sets $S_0 \subset S_1 \subset \cdots \subset S_{N/\ell} \subseteq \{-1,1\}^N$ defined by

$$S_0 = \{1^N\}, S_1 = \{x : x_i = 1 \ \ \forall i > \ell\}, \ldots, S_t = \{x : x_i = 1 \ \forall i > t\ell\}, \ldots, S_{N/\ell} = \{-1,1\}^N.$$

Let p be a polynomial of degree at most d such that $p(x;s;y;w) \cdot F_{k+1}(x;s;y;w) > 0$ for all $(x;s;y;w) \in \{-1,1\}^n$. We iteratively construct a sequence of inputs

$$(x^0;s^0;y^0;w^0), (x^1;s^1;y^1;w^1), \ldots, (x^{N/\ell};s^{N/\ell};y^{N/\ell};w^{N/\ell})$$

to p such that:

- Each $x^t \in S_t$ and each $y^t \in ([t\ell]^N)^k$,
- $(x^0;s^0;y^0;w^0) = (1^N; (1^N)^k; (0^N)^k; (0^N, (0 \circ 0)^N, \ldots, (\underbrace{0 \circ \cdots \circ 0}_{k-1 \text{ times}})^N))$, and
- $|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| \geq 2^{c_k(N/4d^2)^k} \cdot |p(x^t;s^t;y^t;w^t)|$ for each $i = 0, \ldots, N/\ell - 1$.

At the end of this process, we have obtained an input $(x^{N/\ell};s^{N/\ell};y^{N/\ell};w^{N/\ell})$ such that

$$|p(x^{N/\ell};s^{N/\ell};y^{N/\ell};w^{N/\ell})| \geq 2^{c_k \cdot (N/4d^2)^k \cdot (N/\ell)} \geq 2^{c_k \cdot (N/4d^2)^{k+1}},$$

where the last inequality holds for any $k \geq 2$. Letting $c_{k+1} = 4^{-(k+1)} \cdot c_k \geq 4^{-(k+1)} \cdot (4^{-k^2} \cdot c) \geq 4^{-(k+1)^2} \cdot c$, at the conclusion of this process, we obtain an input $(x^{N/\ell};s^{N/\ell};y^{N/\ell};w^{N/\ell})$ such that

$$|p(x^{N/\ell};s^{N/\ell};y^{N/\ell};w^{N/\ell})| \geq 2^{c_{k+1} \cdot (N/d^2)^{k+1}} \cdot |p(x^0;s^0;y^0;w^0)|$$

as desired, completing the induction.

For $t = 1, \ldots, N/\ell$, we construct the inputs $(x^t;s^t;y^t;w^t)$ iteratively. The next claim formalizes this iterative process.

Claim 14. Let p be a polynomial of degree at most d and suppose $p(x;s;y;w) \cdot F_{k+1}(x;s;y;w) > 0$ for all $(x;s;y;w) \in \{-1,1\}^n$. Let $(x^t;s^t;y^t;w^t)$ be an input with $x^t \in S_t$ and $y^t \in ([t\ell]^N)^k$. Then there exists an input $(x^{t+1};s^{t+1};y^{t+1};w^{t+1})$ such that $|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| \geq 2^{c_k(N/4d^2)^k} \cdot |p(x^t;s^t;y^t;w^t)|$, where $x^{t+1} \in S_{t+1}$ and $y^{t+1} \in ([(t+1)\ell]^N)^k$.

Proof. As with Claim 10, we prove this claim in two steps. First, we show that there exists an $x^{t+1} \in \{-1,1\}^N$ supported on S_{t+1} for which $|p(x^{t+1};s^t;y^t;w^t)| \geq |p(x^t;s^t;y^t;w^t)|$. Second, we show that there exists an $s^{t+1} \in (\{-1,1\}^N)^k$, a $y^{t+1} \in ([(t+1)\ell]^N)^k$, and a string w^{t+1} such that $|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| \geq 2^{c_k(N/d^2)^k} \cdot |p(x^{t+1};s^t;y^t;w^t)|$. Putting these steps together yields $|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| \geq 2^{c_k(N/d^2)^k} \cdot |p(x^t;s^t;y^t;w^t)|$.

Step 1. We examine the function $F_{k+1}(x; s^t; y^t; w^t)$, viewed as a function in x. By construction, each block $y^t_{i,j} \leq t\ell$. Thus, for all $x \in S_{t+1} \setminus S_t$, we have $\mathrm{EQ}_{\beta(x)}(y^t_{i,j}) = 1$, and hence $v_2(x, s^t_2, y^t_2) = 1^N$ and $v_i(x, s^t_i, y^t_i, w^t_i) = (1^N, w^t_i)$ for all $i \geq 3$. As a result, whenever $x \in S_{t+1} \setminus S_t$, we have

$$F_{k+1}(x; s^t; y^t; w^t) = \mathrm{OMB}_N(x) \oplus F_k(1^N, (1^N, w^t_3), \ldots, (1^N, w^t_{k+1})),$$

which is either the function $\mathrm{OMB}_N(x)$ or its negation. Without loss of generality, assume $F_{k+1}(x; s^t; y^t; w^t) = \mathrm{OMB}_N(x)$ below.

Now consider the polynomial $q : \{-1, 1\}^N \to \mathbb{R}$ defined by $q(x) = p(x; s^t; y^t; w^t)$. Then $q(x) \cdot \mathrm{OMB}_N(x) > 0$ for all $x \in S_{t+1} \setminus S_t$. By assumption, $x^t \in S_t$. Thus, by Lemma 7, there exists an $x^{t+1} \in S_{t+1}$ such that $|q(x^{t+1})| \geq 2 \cdot |q(x^t)|$. In particular, this means $|p(x^{t+1}; s^t; y^t; w^t)| \geq |p(x^t; s^t; y^t; w^t)|$.

Step 2. We now show that there exists an $s^{t+1} \in (\{-1, 1\}^N)^k$, a $y^{t+1} \in ([(t+1)\ell]^N)^k$, and a string w^{t+1} such that $|p(x^{t+1}; s^{t+1}; y^{t+1}; w^{t+1})| \geq 2^{c_k(N/d^2)^k} \cdot |p(x^{t+1}; s^t; y^t; w^t)|$. This is the most complex part of the proof, and is where we invoke the inductive hypothesis (the statement of Proposition 12) using the function F_k. To do so, we introduce a new set of variables $\sigma \in (\{-1, 1\}^N)^k$ and

$$\zeta \in (\{-1, 1\}^{\log(N+1)})^N \times \cdots \times (\{-1, 1\}^{(k-1)\cdot\log(N+1)})^N$$

which should be interpreted as inputs to the function F_k (taking the place of s and z, respectively). We then define a mapping $(\sigma, \zeta) \mapsto (s_\sigma, y_\sigma, w_{\sigma,\zeta})$ taking these inputs to F_k to inputs to F_{k+1} that satisfies the following (informally stated) properties:

- **Property 1.** The mapping is computed by a low-degree polynomial in σ and ζ (in fact, a polynomial of degree 2).
- **Property 2.** The function $F_{k+1}(x^{t+1}; s_\sigma; y_\sigma; w_{\sigma,\zeta})$ simply computes either $F_k(\sigma; \zeta)$ or its negation.
- **Property 3.** When the pair $(\sigma; \zeta)$ is the "starting input" $(x^0; s^0; y^0; w^0)$ to F_k, then the resulting image $(s_\sigma, y_\sigma, w_{\sigma,\zeta}) = (s^t, y^t, w^t)$.

Properties 1 and 2 taken together show that the projected polynomial $r(\sigma; \zeta) := p(x^{t+1}; s_\sigma; y_\sigma; w_{\sigma,\zeta})$ satisfies the hypotheses of Proposition 12 with respect to the function $F_k(\sigma; \zeta)$, and moreover $\deg(r) \leq 2 \cdot \deg(p) \leq 2d$. Thus, by the inductive hypothesis, there is some pair $(\sigma^*; \zeta^*)$ such that $|r(\sigma^*; \zeta^*)| \geq 2^{c_k(N/(2d)^2)^k} \cdot |r(1^{n_k})|$, where the latter quantity is $2^{c_k(N/(2d)^2)^k} \cdot |p(x^{t+1}; s^t; y^t; w^t)|$ by Property 3 above.

Now we carry out the full details of Step 2. Define the mapping $(\sigma, \zeta) \mapsto (s_\sigma, y_\sigma, w_{\sigma,\zeta})$ as follows. For strings

$$\sigma = (\sigma_2, \sigma_3, \ldots, \sigma_{k+1}) \in (\{-1, 1\}^N)^k, \text{and}$$
$$\zeta = (\zeta_3, \zeta_4, \ldots, \zeta_{k+1}) \in \left(\{-1, 1\}^{\log(N+1)}\right)^N \times \left(\{-1, 1\}^{2\cdot\log(N+1)}\right)^N \times \cdots \times \left(\{-1, 1\}^{(k-1)\cdot\log(N+1)}\right)^N$$

define the strings s_σ, y_σ and $w_{\sigma,\zeta}$ by

- For each $i = 2, \ldots, k+1$,

$$(s_\sigma)_{i,j} = \begin{cases} -1 & \text{if } \sigma_{i,j} = -1, \\ s^t_{i,j} & \text{if } \sigma_{i,j} = 1, \end{cases}$$

- For each $i = 2, \dots, k+1$,

$$(y_\sigma)_{i,j} = \begin{cases} \beta(x^{t+1}) & \text{if } \sigma_{i,j} = -1, \\ y^t_{i,j} & \text{if } \sigma_{i,j} = 1, \end{cases}$$

- For each $i = 3, \dots, k+1$,

$$(w_{\sigma,\zeta})_{i,j} = \begin{cases} \zeta_{i,j} & \text{if } \sigma_{i,j} = -1, \\ w^t_{i,j} & \text{if } \sigma_{i,j} = 1. \end{cases}$$

Observe that this parametrization has the property that if $\sigma = (1^N)^k$, then

$$(x^{t+1}; s_\sigma; y_\sigma; w_{\sigma,\zeta}) = (x^{t+1}; s^t; y^t; w^t).$$

That is, Property 3 above holds under this definition of s_σ, y_σ and $w_{\sigma,\zeta}$.

We now need to show that $F_{k+1}(x^{t+1}; s_\sigma; y_\sigma; w_{\sigma,\zeta})$ indeed collapses to $F_k(\sigma; \zeta)$ (i.e., that Property 2 above holds). We will do this by applying the decomposition of Claim 13. Note that since $\beta(x^{t+1}) \leq (t+1)\ell$ and each $y^t_{i,j} \in [t\ell]$, we have that $y_\sigma \in ([(t+1)\ell]^N)^k$ for every $\sigma \in (\{-1,1\}^N)^k$. We can thus calculate

$$v_2(x^{t+1}, (s_\sigma)_2, (y_\sigma)_2) = (\dots, (s_\sigma)_{2,j} \wedge \mathrm{EQ}_{\beta(x^{t+1})}((y_\sigma)_{2,j}), \dots) = (\dots, \sigma_{2,j}, \dots),$$

where the final equality exploits the fact that $y^t_{i,j} \leq t\ell$ for all i, j, and $\beta(x^{t+1}) > t\ell$. Moreover, for $i = 3, \dots, k+1$,

$$\begin{aligned} v_i(x^{t+1}, (s_\sigma)_i, (y_\sigma)_i, (w_{\sigma,\zeta})_i) &= ((\dots, (s_\sigma)_{i,j} \wedge \mathrm{EQ}_{\beta(x^{t+1})}((y_\sigma)_{i,j}), \dots), (\dots, (w_{\sigma,\zeta})_{i,j}, \dots)) \\ &= ((\dots, \sigma_{i,j}, \dots), (\dots, (w_{\sigma,\zeta})_{i,j}, \dots)). \end{aligned}$$

Consider the function $E(\sigma; \zeta) := F_{k+1}(x^{t+1}; s_\sigma; y_\sigma; w_{\sigma,\zeta})$. By the calculations above,

$$\begin{aligned} E(\sigma; \zeta) &= \mathrm{OMB}_N(x^{t+1}) \oplus F_k(v_2(x^{t+1}, (s_\sigma)_2, (y_\sigma)_2), \dots, v_{k+1}(x^{t+1}, (s_\sigma)_{k+1}, (y_\sigma)_{k+1}, (w_{\sigma,\zeta})_{k+1})) \\ &= \mathrm{OMB}_N(x^{t+1}) \oplus F_k(\sigma_2, (\sigma_3, (w_{\sigma,\zeta})_3), \dots, (\sigma_{k+1}, (w_{\sigma,\zeta})_{k+1})) \\ &= \mathrm{OMB}_N(x^{t+1}) \oplus F_k(\sigma_2, (\sigma_3, \zeta_3), \dots, (\sigma_{k+1}, \zeta_{k+1})), \end{aligned}$$

where the last equality follows because, for any string τ, we have

$$\begin{aligned} \sigma_{i,j} \wedge \mathrm{EQ}_\tau((w_{\sigma,\zeta})_{i,j}) &\iff \sigma_{i,j} \wedge \mathrm{EQ}_\tau((w_{\sigma,\zeta})_{i,j}) \wedge ((w_{\sigma,\zeta})_{i,j} = \zeta_{i,j}) \\ &\iff \sigma_{i,j} \wedge \mathrm{EQ}_\tau(\zeta_{i,j}). \end{aligned}$$

Now consider the polynomial $r(\sigma; \zeta) := p(x^{t+1}; s_\sigma; y_\sigma; w_{\sigma,\zeta})$. Since the variables s_σ, y_σ, and $w_{\sigma,\zeta}$ can be written as linear or quadratic functions of σ and ζ, the polynomial r satisfies $\deg r \leq 2 \deg p \leq 2d$. Moreover, $r(\sigma; \zeta) \cdot E(\sigma; \zeta) > 0$ for all $(\sigma; \zeta)$. Since E is either the function F_k or its negation, the inductive hypothesis (the statement of Proposition 12) allows us to conclude that there exists a $(\sigma^*; \zeta^*)$ such that $|r(\sigma^*; \zeta^*)| \geq 2^{c_k(N/4d^2)^k} \cdot |r(1^{n_k})|$, where n_k is the number of Boolean variables on which F_k is defined. Setting $s^{t+1} := s_{\sigma*}$, $y^{t+1} := y_{\sigma*}$, and $w^{t+1} := w_{\sigma^*,\zeta^*}$ thus yields

$$|p(x^{t+1}; s^{t+1}; y^{t+1}; w^{t+1})| = |r(\sigma^*; \zeta^*)| \geq 2^{c_k(N/4d^2)^k} \cdot |r(1^{n_k})| = 2^{c_k(N/4d^2)^k} \cdot |p(x^{t+1}; s^t; y^t; w^t)|,$$

as we wanted to show. This completes the proof of Claim 14. □

With Claim 14 completed, we conclude the proof of Proposition 12.

□

5 Proof of Theorem 2

5.1 ELEMENT DISTINCTNESS Preliminaries

Let $\overline{\mathrm{ED}}_M : \{-1,1\}^M \to \{-1,1\}$ denote the negation of the function ED defined in Section 2. As in Section 2, assume for simplicity that $M = m \log m$ for some m which is a power of 2. Define an equivalence relation $\sim$ on $\{-1,1\}^M$ as follows. Let $\mathcal{S}_m$ denote the symmetric group on $\{1,\dots,m\}$. Given an $x = (x_1,\dots,x_m) \in \{-1,1\}^M$ where each block $x_i \in \{-1,1\}^{\log m}$, and an $x' \in \{-1,1\}^M$, we say $x \sim x'$ if there exists a permutation $\pi \in \mathcal{S}_m$ such that $x_i = x'_{\pi(i)}$ for every $i = 1,\dots,m$. This equivalence relation is defined so that $\overline{\mathrm{ED}}_M(x) = \overline{\mathrm{ED}}_M(x')$ whenever $x \sim x'$. Moreover, there is a unique equivalence class $T \subseteq \{-1,1\}^M$ such that $\overline{\mathrm{ED}}_M(\eta) = 1$ for any $\eta \in T$. For the remainder of the paper, we let $\eta \in T$ denote a fixed, representative element of this equivalence class. We also say that a function $q\colon \{-1,1\}^M \to \{-1,1\}$ is *symmetric* with respect to $\sim$ if $q(x) = q(x')$ whenever $x \sim x'$.

Let $G = \mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ denote the block composition of OMB_N with $\overline{\mathrm{ED}}_M$, which is a function on $N \cdot M$ variables. The next theorem shows that composed function G is much harder to approximate by low degree polynomials than OMB itself. Namely, $\widetilde{\deg}_\varepsilon(G) \geq \tilde{\Omega}(M^{2/3})$ for $\varepsilon = 1 - 2^{-\Omega(N)}$.

Theorem 15 (Thaler [35])**.** There exists a universal constant $c > 0$ such that the following holds. Let $p : (\{-1,1\}^M)^N \to \mathbb{R}$ be a polynomial of degree $d = c \cdot (M/\log M)^{2/3}$. Suppose that $|p(x)| \geq 1$ and $p(x) \cdot G(x) > 0$ for all $x \in (\{-1,1\}^M)^N$. Then there exists an $x \in (\{-1,1\}^M)^N$ such that $|p(x)| \geq 2^{N/2}$.

Theorem 15 will be used as a building block in our proof of Theorem 2. Theorem 15 is a special case of a more general result proved by Thaler in [35]. The proof in [35] is *dual* (in the sense of linear programming duality) – it constructs a "witness" for the bound in Theorem 15 by exhibiting a solution to the dual of a linear program capturing the approximate degree of G. Our proofs in this paper are *primal*, in the sense that we reason about approximating polynomials directly. For completeness and expository purposes, we reprove Theorem 15 below using a primal argument. To do so, we first introduce the following notion of symmetry for polynomials with respect to the structure of G.

Definition 16. A function $\psi : (\{-1,1\}^M)^N \to \mathbb{R}$ is *intra-block symmetric (with respect to $\sim$)* if for every $i = 1,\dots,N$ and every pair $x_i, x'_i \in \{-1,1\}^\ell$ with $x_i \sim x'_i$,

$$\psi(x_1,\dots,x'_i,\dots,x_N) = \psi(x_1,\dots,x_i,\dots,x_N).$$

Lemma 17. Let $p : (\{-1,1\}^M)^N \to \mathbb{R}$ be a polynomial. Then the polynomial

$$\tilde{p}(x_1,\dots,x_N) := \mathop{\mathbb{E}}_{x'_1 \sim x_1,\dots,x'_N \sim x_N} [p(x'_1,\dots,x'_N)]$$

is intra-block symmetric and satisfies $\deg \tilde{p} \leq \deg p$.

Proof. By linearity, it suffices to prove the lemma for factored polynomials of the form

$$p(x) = p_1(x_1)p_2(x_2)\dots p_N(x_N).$$

Then

$$\tilde{p}(x_1,\dots,x_N) = \prod_{i=1}^N \mathop{\mathbb{E}}_{x'_i \sim x_i} p_i(x_i).$$

Write $x_i = (x_{i,1}, \dots x_{i,m})$ where each $x_{i,j} \in \{-1,1\}^{\log m}$. Then

$$\mathop{\mathbb{E}}_{x_i' \sim x_i} p_i(x_i) = \mathop{\mathbb{E}}_{\pi \in \mathcal{S}_m} p_i(x_{i,\pi(1)}, \dots x_{i,\pi(m)})$$

which is a polynomial of degree at most $\deg p_i$. Therefore, $\deg \tilde{p} \leq \deg p$. □

Clearly, if Theorem 15 holds for $\tilde{p}$, then it holds for p as well. Hence, Lemma 17 allows us to assume without loss of generality that a sign-representing polynomial p for G is intra-block symmetric (if not, then we apply the argument to $\tilde{p}$ rather than to p itself). Theorem 15 therefore follows from the proposition below.

Proposition 18. There exists a universal constant $c > 0$ such that the following holds. Let $p : (\{-1,1\}^M)^N \to \mathbb{R}$ be a polynomial of degree at most $d = c \cdot (M/\log M)^{2/3}$, and suppose p is intra-block symmetric. Suppose that $p(x) \cdot G(x) > 0$ for all $x \in (\{-1,1\}^M)^N$. Then there exists an $x \in (\{-1,1\}^M)^N$ such that $|p(x)| \geq 2^{N/2} \cdot |p(\eta^N)|$.

As in Beigel's original proof for the ODD-MAX-BIT function itself, the proof of Proposition 18 is via induction using the following lemma.

Lemma 19. Consider the increasing family of sets $S_0 \subset S_1 \subset \cdots \subset S_{N/2} \subseteq (\{-1,1\}^M)^N$ defined by

$$\begin{aligned}
&S_0 = \{\eta^N\},\\
&S_1 = \{x : x_3 = x_4 = \cdots = x_N = \eta\},\\
&\vdots\\
&S_i = \{x : x_{2i+1} = x_{2i+2} = \cdots = x_N = \eta\},\\
&\vdots\\
&S_{N/2} = (\{-1,1\}^M)^N.
\end{aligned}$$

There is some constant $c > 0$ such that the following holds. Let $p : (\{-1,1\}^M)^N \to \mathbb{R}$ be a polynomial of degree at most $d = c \cdot (M/\log M)^{2/3}$ that is intra-block symmetric, and suppose that $p(x) \cdot G(x) > 0$ for all $x \in S_{i+1} \setminus S_i$. Let $z \in S_i$. Then there exists a $z' \in S_{i+1} \setminus S_i$ such that $|p(z')| \geq 2 \cdot |p(z)|$.

Proof. Fix any polynomial $p : (\{-1,1\}^M)^N \to \mathbb{R}$, and assume that $p(z) = -T$, where $T > 0$ (the case where $T < 0$ is similar). Suppose that $|p(x)| \leq 2T$ for every $x \in S_{i+1} \setminus S_i$. Define the polynomial $q : \{-1,1\}^M \to \mathbb{R}$ as follows. Given a string $y \in \{-1,1\}^M$, define a string x_y by

- $(x_y)_j = z_j$ for all blocks with $j \leq 2i$,
- $(x_y)_{2i} = \eta$,
- $(x_y)_{2i+1} = y$,
- $(x_y)_j = \eta$ for all $j \geq 2i+2$.

Now define the polynomial $q : \{-1,1\}^M \to \mathbb{R}$ by $q(y) := p(x_y)$. By construction, $\deg q \leq \deg p \leq d$, and q is symmetric with respect to $\sim$. Moreover, $q(\eta) = p(z) = -T$ while $q(y) \in [0, 2T]$ for every $y \in \overline{\mathrm{ED}}^{-1}(-1)$. The polynomial $q' := -\frac{1}{T} \cdot q$ satisfies $q'(x) = 1$ for all $x \in \overline{\mathrm{ED}}_M^{-1}(+1)$, and $q'(x) \in [-2, 0]$ for every $x \in \overline{\mathrm{ED}}_M^{-1}(-1)$. Observe that $q'' := \frac{2q'-1}{5}$ satisfies $|q''(x) - \overline{\mathrm{ED}}_M| \leq 4/5$ for all $x \in \{-1,1\}^M$.

Using standard error reduction techniques (see for example [11, Claim 4.3]), q'' can be transformed into a polynomial of degree $O(\deg(q'')) = O(\deg(q)) = O(\deg(p))$ that uniformly approximates $\overline{\mathrm{ED}}_M$ to error at most $1/3$. Hence, the approximate degree lower bound for ED_M (cf. Theorem 8) implies that $\deg(p) \geq c \cdot (M/\log M)^{2/3}$ for some constant $c > 0$. □

5.2 The Main Argument

Fix parameters $M, N \in \mathbb{N}$, and a constant $k > 0$. We construct a family of functions $H_1, H_2, \ldots, H_k$, each of which is computed by a polynomial size Boolean circuit of depth 4, with logarithmic bottom fan-in, as well as by a quasipolynomial size Boolean circuit of depth 3, with polylogarithmic bottom fan-in. Theorem 2 follows easily from the following claim, which is the main result of this section.

Theorem 20. There exists a universal constant $c > 0$ such that the following holds. For each $k \in \mathbb{N}$, let $c_k = 2^{-k} \cdot c$. Let $n \in \mathbb{N}$ where

$$n = N \cdot M \cdot \sum_{i=1}^{k} (1 + (i-1) \cdot \log(N+1)) = O(k^2 \cdot N \cdot M \cdot \log N).$$

Let p be a polynomial of degree at most $d = c_k \cdot (M/\log M)^{2/3}$ such that $p(x) \cdot H_k(x) > 0$ for all $x \in \{-1,1\}^n$ and $|p(x)| \geq 1$ for all $x \in \{-1,1\}^n$. Then there exists an $x \in \{-1,1\}^n$ such that $|p(x)| \geq 2^{(N/2)^k}$.

Proof of Theorem 2, assuming Theorem 20. Let $N = 2M^\delta$ and Let $k = \lceil 2 \cdot (1+\delta)\Gamma/\delta \rceil$ in the statement of Theorem 20, and observe that under this setting, H_k is defined over the domain $\{-1,1\}^n$ for $n = O(k^2 M^{1+\delta} \log M)$. Fix a polynomial p of degree at most $d = c_k \cdot (M/\log M)^{2/3}$.

Let $v = \min_{x \in \{-1,1\}^n} |p(x)|$, and suppose that $p(x) \cdot H_k(x) > 0$ for all $x \in \{-1,1\}^n$. By Theorem 20, there exists an $x \in \{-1,1\}^n$ such that

$$|p(x)| \geq 2^{(N/2)^k} \cdot v > 2^{M^{\delta \cdot k}} \cdot v \geq 2^{M^{2\Gamma(1+\delta)}} \cdot v \geq 2 \cdot 2^{n^\Gamma} \cdot v,$$

where the last inequality holds for sufficiently large M. Hence, if $v > 2^{-n^\Gamma}$, then $|p(x)| > 2$. It follows that p cannot approximate H_k uniformly to within error less than $1 - 2^{-n^\Gamma}$. To complete the proof, we observe that for sufficiently large M, it holds that $d \geq n^{2/3-\delta}$. □

Remark 21. Rather than setting $N = 2M^\delta$ as in the proof above, if we set N to be slightly subpolynomial in M, and k to be a slowly growing function in M such that $N^k \approx n^\Gamma$, then we can obtain a function H_k that is still computed by a depth three Boolean circuit of quasipolynomial size, and satisfies Theorem 2 with $n^{2/3-\delta}$ replaced by $n^{2/3-o(1)}$. For example, to obtain such a function it suffices to set $N = M^{1/\log\log M}$ and $k = \Gamma \cdot \log\log M$. We omit further details for brevity.

Section Roadmap. The remainder of this section is devoted to defining the functions H_k and proving Theorem 20. In Section 5.2.1, we review necessary notation. In Section 5.2.2, we exhibit the main ideas underlying the construction of H_k and the proof of Theorem 20 by considering a simplified case. Specifically, we fix $k = 2$, and establish that the bound of Theorem 20 holds for a slightly simplified version of H_2. In Section 5.2.3, we define H_k for any $k \geq 1$ and explain why H_k be efficiently computed by decision lists and Boolean circuits. Finally, in Section 5.2.4, we prove Theorem 20.

5.2.1 Notation

Recall that $G = \mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$ denotes the block composition of OMB_N with $\overline{\mathrm{ED}}_M$. G is a function on $N \cdot M$ variables. For two binary strings $a, b \in \{-1,1\}^N$, $a \oplus b$ denotes the entrywise XOR of a and b. Recall (cf. Section 3.1) that for any binary vector $a \in \left(\{-1,1\}^M\right)^N$, we interpret a as consisting of N blocks $a_1, \dots, a_N$, each in $\{-1,1\}^M$, and we let $\gamma(a) \in \{-1,1\}^{\log(N+1)}$ denote the binary representation of the largest integer j satisfying $\overline{\mathrm{ED}}(a_j) = -1$.

5.2.2 Proof of Theorem 20 When $k = 2$

Consider the following function $H : \left(\{-1,1\}^M\right)^N \times \left(\left(\{-1,1\}^{\log(N+1)}\right)^M\right)^N$, which should be viewed as a slightly simplified version of the function H_2 defined later (cf. Section 5.2.3).

$$H(x,y) = \mathrm{OMB}_N(\overline{\mathrm{ED}}_M(x_1), \dots, \overline{\mathrm{ED}}_M(x_N)) \oplus \mathrm{OMB}_N(\dots, \overline{\mathrm{ED}}_M(\dots, \eta_j \oplus \mathrm{EQ}_{\gamma(x)}(y_{i,j}), \dots), \dots).$$

One can think of $H(x,y)$ as computing $G(x_1, \dots, x_N) \oplus G(u_2(x,y))$, where $u_2(x,y)$ "pre-processes" $y = (y_{1,1}, \dots, y_{N,M})$ by first testing each $y_{i,j}$ for equality with $\gamma(x)$, and then XOR-ing the result with η.

Our goal is to prove the following special case of Theorem 20.

Proposition 22. Let $n = M \cdot N + M \cdot N \cdot \log(N+1)$ for some $M, N \in \mathbb{N}$. There exists a universal constant $c > 0$ such that the following holds. Let

$$p \colon (\{-1,1\}^M)^N \times \left(\left(\{-1,1\}^{\log(N+1)}\right)^M\right)^N \to \mathbb{R}$$

be a polynomial of degree at most $d = c \cdot (M/\log M)^{2/3}$ such that $|p(x,y)| \geq 1$ and $p(x,y) \cdot H(x,y) > 0$ for all (x,y) in its domain. Then there exists an (x,y) in its domain such that $|p(x,y)| \geq 2^{(N/2)^2}$.

Proof. As with Lemma 19 we consider the increasing family of sets $S_0 \subset S_1 \subset \cdots \subset S_{N/2} \subseteq (\{-1,1\}^M)^N$ defined by

$$\begin{aligned}
&S_0 = \{\eta^N\},\\
&S_1 = \{x : x_3 = x_4 = \cdots = x_N = \eta\},\\
&\vdots\\
&S_t = \{x : x_{2t+1} = x_{2t+2} = \cdots = x_N = \eta\},\\
&\vdots\\
&S_{N/2} = (\{-1,1\}^M)^N.
\end{aligned}$$

Assuming Intra-Block Symmetry. Let p be a polynomial of degree at most d such that $|p(x,y)| \geq 1$ and $p(x,y) \cdot H(x,y) > 0$ for all $(x,y) \in \{-1,1\}^n$. First, we use the following simple lemma to show that it is without loss of generality to assume a symmetric structure on p. The proof is similar to Lemma 17 and is omitted for brevity.

Lemma 23. Let $m \in \mathbb{N}$, and $M = m \log m$. Let $p : (\{-1,1\}^M)^N \times \left(\left(\{-1,1\}^{\log(N+1)}\right)^M\right)^N \to \mathbb{R}$ be a polynomial. For $x = (x_1, \dots, x_N) \in (\{-1,1\}^M)^N$, let us write each $x_i = (x_{i,1}, \dots, x_{i,m}) \in \{-1,1\}^M$,

where each $x_{i,j} \in \{-1,1\}^{\log m}$. For permutation $\pi \in \mathcal{S}_m$, let $\pi(x_i) = (x_{i,\pi(1)}, \ldots, x_{i,\pi(m)})$. Similarly, for $y = (y_1, \ldots, y_N) \in \left(\left(\{-1,1\}^{\log(N+1)}\right)^M\right)^N$, let us write each

$$y_i = (y_{i,1}, \ldots, y_{i,m}) \in \left(\left(\{-1,1\}^{\log(N+1)}\right)^{\log m}\right)^m,$$

where each $y_{i,j} \in (\{-1,1\}^{\log(N+1)})^{\log m}$. For $\pi \in \mathcal{S}_m$, let

$$\pi(y_i) = (y_{i,\pi(1)}, \ldots, y_{i,\pi(m)}).$$

Let

$$\tilde{p}(x,y) = \mathop{\mathbb{E}}_{\pi_1,\ldots,\pi_{2N} \in \mathcal{S}_m}[p(\pi_1(x_1), \ldots, \pi_N(x_N), \pi_{N+1}(y_1), \ldots, \pi_{2N}(y_N))].$$

Then $\deg(\tilde{p}) \leq \deg(p)$.

If p is a polynomial of degree at most d such that $|p(x,y)| \geq 1$ and $p(x,y) \cdot H(x,y) > 0$ for all $(x,y) \in \{-1,1\}^n$, then the transformation of the above lemma results in a polynomial $\tilde{p}$ with the same properties. Moreover, this polynomial is $\tilde{p}$ is invariant with respect to permutations of the form described in Lemma 23. Formally, $\tilde{p}$ satisfies the following property: for any two inputs $(x,y), (x',y')$ to $\tilde{p}$, if there exist permutations $\pi_1, \ldots, \pi_{2N}$ such that $(x_1, \ldots, x_N, y_1, \ldots, y_N) = (\pi_1(x'_1), \ldots, \pi_N(x'_N), \pi_{N+1}(y'_1), \ldots, \pi_{2N}(y'_N))$, then $\tilde{p}(x,y) = \tilde{p}(x',y')$. We call any polynomial defined on $(\{-1,1\}^M)^N \times \left(\left(\{-1,1\}^{\log(N+1)}\right)^M\right)^N$ that satisfies this invariance property *intra-block symmetric*. This in in analogy with the notion of intra-block symmetry that we defined for polynomials on $(\{-1,1\}^M)^N$ (cf. Definition 16).

If $\tilde{p}$ satisfies Proposition 22, then so does p. Hence, we assume for the remainder of the proof that p itself is intra-block symmetric; if not, then we apply the argument below to $\tilde{p}$ rather than to p.

The Proof of Proposition 22 for Intra-Block Symmetric Polynomials. As in the proofs of Propositions 9 and 12, to ease notation, we will identify any binary string in $\{-1,1\}^{\log(N+1)}$ with the number in $[N]$ for which the string is the binary representation. That is, while we will write any such binary string as though it were a number $0, 1, \ldots, N$, it should always be thought of as the binary string representing that number.

Let c be the universal constant appearing in the statement of Proposition 18. Given an intra-block symmetric polynomial p of degree at most $c \cdot (M/\log M)^{2/3}$ we iteratively construct a sequence of inputs $(x^0, y^0), (x^1, y^1), \ldots, (x^{N/2}, y^{N/2})$ to p such that:

- Each $x^t \in S_t$ and each $y^t \in ([2t]^M)^N$,
- $|p(x^0, y^0)| \geq 1$, and
- $|p(x^{t+1}, y^{t+1})| \geq 2^{N/2} \cdot |p(x^t, y^t)|$ for each $t = 0, \ldots, (N/2) - 1$.

At the conclusion of this process, we obtain an input $(x^{N/2}, y^{N/2})$ such that $|p(x^{N/2}, y^{N/2})| \geq 2^{(N/2)^2}$.

We may take as the first input (x^0, y^0) the point $(\eta^N, (0^M)^N)$, and we construct the remaining inputs iteratively. The following claim formalizes this iterative process.

Claim 24. Let c be the universal constant appearing in the statement of Proposition 18. Let p be an intra-block symmetric polynomial of degree at most $d = c \cdot (M/\log M)^{2/3}$ and suppose $p(x,y) \cdot H(x,y) > 0$ for all $(x,y) \in \{-1,1\}^n$. Let (x^t, y^t) be an input with $x^t \in S_t$ and $y^t \in ([2t]^M)^N$. Then there exists an input (x^{t+1}, y^{t+1}) such that $|p(x^{t+1}, y^{t+1})| \geq 2^{N/2} \cdot |p(x^t, y^t)|$, where $x^{t+1} \in S_{t+1}$ and $y^{t+1} \in ([2(t+1)]^M)^N$.

Proof. We prove the claim in two steps. First, we show that there exists an $x^{t+1} \in S_{t+1}$ for which $|p(x^{t+1}, y^t)| \geq |p(x^t, y^t)|$. Second, we show that there exists a $y^{t+1} \in ([2(t+1)]^M)^N$ such that $|p(x^{t+1}, y^{t+1})| \geq 2^{N/2} \cdot |p(x^{t+1}, y^t)|$. Putting these steps together yields $|p(x^{t+1}, y^{t+1})| \geq 2^{N/2} \cdot |p(x^t, y^t)|$.

Step 1. To complete the first step, we examine the function $H(x, y^t)$, viewed as a function of x. By construction, each block $y_i^t \in [2t]^M$. Thus, $\mathrm{EQ}_{\gamma(x)}(y_{i,j}^t) = 1$ for all $x \in S_{t+1} \setminus S_t$, and hence

$$\begin{aligned} \forall i = 1, \ldots, N \quad & \overline{\mathrm{ED}}_M(\ldots, \eta_j \oplus \mathrm{EQ}_{\gamma(x)}(y_{i,j}^t), \ldots) = \overline{\mathrm{ED}}_M(\eta) = 1 \\ & \implies \mathrm{OMB}_N(\ldots, \overline{\mathrm{ED}}_M(\ldots, \eta_j \oplus \mathrm{EQ}_{\gamma(x)}(y_{i,j}^t), \ldots), \ldots) = 1 \end{aligned}$$

for all such inputs. As a result, $H(x, y^t) = G(x)$ whenever $x \in S_{t+1} \setminus S_t$.

Now consider the polynomial $q : (\{-1,1\}^M)^N \to \mathbb{R}$ defined by $q(x) = p(x, y^t)$. By construction, $\deg q \leq \deg p \leq d$. Moreover, since p is intra-block symmetric, so is q as per Definition 16. In addition, $q(x) \cdot G(x) > 0$ for all $x \in S_{t+1} \setminus S_t$. By assumption, $x^t \in S_t$. Thus, by Lemma 19, there exists an $x^{t+1} \in S_{t+1}$ such that $|q(x^{t+1})| \geq 2 \cdot |q(x^t)|$. By the definition of q, we see that in particular, $|p(x^{t+1}, y^t)| \geq |p(x^t, y^t)|$.

Step 2. We now show that there exists a $y^{t+1} \in ([2(t+1)]^M)^N$, such that $|p(x^{t+1}, y^{t+1})| \geq 2^{N/2} \cdot |p(x^{t+1}, y^t)|$. For $w \in (\{-1,1\}^M)^N$, define the string y_w by $(y_w)_{i,j} = \gamma(x^{t+1})$ if $w_{i,j} = -\eta_j$ and $(y_w)_{i,j} = y_{i,j}^t$ if $w_{i,j} = \eta_j$. Note that since $\gamma(x^{t+1}) \leq 2(t+1)$ and each $y_i^t \in [2t]^M$, we have that $y_w \in ([2(t+1)]^M)^N$ for every $w \in (\{-1,1\}^M)^N$. Consider the function $E(w) := H(x^{t+1}, y_w)$, and observe that

$$E(w) = \mathrm{OMB}_N(\ldots, \overline{\mathrm{ED}}_M(x_i^{t+1}), \ldots) \oplus \mathrm{OMB}_N(\ldots, \overline{\mathrm{ED}}_M(w_i), \ldots).$$

Now consider the polynomial $r : (\{-1,1\}^M)^N \to \mathbb{R}$ given by $r(w) := p(x^{t+1}, y_w)$. Since we can write

$$r(w) = p\left(x^{t+1}, \ldots, \left(\frac{1 - \eta_j \cdot w_{i,j}}{2}\right) \cdot \gamma(x^{t+1}) + \left(\frac{1 + \eta_j \cdot w_{i,j}}{2}\right) \cdot y_{i,j}^t \cdots\right),$$

r satisfies $\deg r \leq \deg p \leq d$. Moreover, since p is intra-block symmetric, so is r. Finally, observe there $r(w) \cdot E(w) > 0$ for all $w \in (\{-1,1\}^M)^N$. Since E is either the function G or its negation, we conclude by Proposition 18 that there exists a w^* such that $|r(w^*)| \geq 2^{N/2} \cdot |r(\eta^N)|$. Setting $y^{t+1} := y_{w^*}$ thus yields

$$|p(x^{t+1}, y^{t+1})| = |r(w^*)| \geq 2^{N/2} \cdot |r(\eta^N)| = 2^{N/2} \cdot |p(x^{t+1}, y^t)|,$$

as we wanted to show (here, the final equality holds because p is intra-block symmetric). This concludes the proof of Claim 24. □

With Claim 24 established, we conclude the proof of Proposition 22.

□

5.2.3 Construction of the Function H_k

Definition of Auxiliary Functions. The construction of the functions $H_1, H_2, \ldots, H_k$ begins with the following sequence of auxiliary "pre-processing" functions $u_1, u_2, \ldots, u_k$. Each function u_i maps

$$(\{-1,1\}^M)^N \times \left(\left(\{-1,1\}^M\right)^N \times \left(\left(\{-1,1\}^{\log(N+1)}\right)^M\right)^N\right) \times \ldots$$
$$\cdots \times \left(\left(\{-1,1\}^M\right)^N \times \left(\left(\{-1,1\}^{(i-1)\cdot\log(N+1)}\right)^M\right)^N\right)$$

to

$$(\{-1,1\}^M)^N.$$

For $i = 1, \ldots, k$, let $s_i = (s_{i,1}, \ldots, s_{i,N})$ denote an arbitrary input in $(\{-1,1\}^M)^N$, and z_i denote an arbitrary input in $\left(\left(\{-1,1\}^{(i-1)\cdot\log(N+1)}\right)^M\right)^N$. The auxiliary functions u_i are defined recursively by:

$$u_1(s_1) = (\eta \oplus s_{1,1}, \ldots, \eta \oplus s_{1,N})$$

$$u_2(s_1,(s_2,z_2)) = (\eta \oplus (\ldots, s_{2,1,\ell} \wedge \mathrm{EQ}_{\gamma(u_1)}(z_{2,1,\ell}), \ldots), \ldots, \eta \oplus (\ldots, s_{2,N,\ell} \wedge \mathrm{EQ}_{\gamma(u_1)}(z_{2,N,\ell}), \ldots))$$

$$\vdots$$

$$u_k(s_1,(s_2,z_2),\ldots,(s_k,z_k)) = (\ldots, \eta \oplus (\ldots, s_{k,j,\ell} \wedge \mathrm{EQ}_{\gamma(u_1)\circ\cdots\circ\gamma(u_{k-1})}(z_{k,j,\ell}), \ldots), \ldots)$$

Definition of the H_i's. We now recursively define H_i. Each function H_i is defined on the same domain as u_i. Define:

$$\begin{aligned} H_1(s_1) &= G(u_1) \\ &= \mathrm{OMB}_N(\overline{\mathrm{ED}}_M(\eta \oplus s_{1,1}), \ldots, \overline{\mathrm{ED}}_M(\eta \oplus s_{1,N})) \end{aligned}$$

$$\begin{aligned} &H_2(s_1,(s_2,z_2)) = H_1(s_1) \oplus G(u_2) \\ &= \mathrm{OMB}_N(\ldots, \overline{\mathrm{ED}}_M(\eta \oplus s_{1,j}), \ldots) \oplus \mathrm{OMB}_N(\ldots, \overline{\mathrm{ED}}_M(\ldots, \eta_j \oplus (s_{2,j,\ell} \wedge \mathrm{EQ}_{\gamma(u_1)}(z_{2,j,\ell})), \ldots), \ldots) \end{aligned}$$

$$\vdots$$

$$H_k(s_1,(s_2,z_2),\ldots,(s_k,z_k)) = H_{k-1}(s_1,(s_2,z_2),\ldots,(s_{k-1},z_{k-1})) \oplus G(u_k)$$

Representing H_k as an $O(\log^2 n)$ Decision List and As Boolean Circuits. The function H_k is represented by both a depth-four circuit of polynomial size and bottom fan-in $O(k^2 \log N)$, and by a $O(k^2 \cdot \log M \cdot \log N)$-decision list of quasi-polynomial length $2^{O(k^2 \cdot \log M \cdot \log N)}$ (the latter is itself computed by a Boolean circuit of quasi-polynomial size and polylogarithmic bottom fan-in). To see why this is true, it is instructive to first examine the function

$$\begin{aligned} \tilde{H}_k(u_1,u_2,\ldots,u_k) &= G(u_1) \oplus G(u_2) \oplus \cdots \oplus G(u_k) \\ &= \mathrm{OMB}_N(\ldots, \overline{\mathrm{ED}}_M(u_{1,j}), \ldots) \oplus \cdots \oplus \mathrm{OMB}_N(\ldots, \overline{\mathrm{ED}}_M(u_{k,j}), \ldots). \end{aligned}$$

We will first argue that this function $\tilde{H}_k$ is an $O(k \log M)$-decision list of polynomial length in the variables u_i.

Before writing $\tilde{H}_k$ as a decision list, we first write it as a "generalized" decision list $(f_0, b_0), (f_1, b_1), \ldots$. Here, a generalized decision list is simply a decision list where the decision rules may be made by arbitrary

functions instead of conjunctions. For $\tilde{H}_k$, it suffices to take

$$\begin{aligned}
f_0(u) &= \overline{\mathrm{ED}}_M(u_{1,N}) \wedge \overline{\mathrm{ED}}_M(u_{2,N}) \wedge \cdots \wedge \overline{\mathrm{ED}}_M(u_{k,N}); && b_0 = (-1)^{k\cdot N}\\
f_1(u) &= \overline{\mathrm{ED}}_M(u_{1,N}) \wedge \overline{\mathrm{ED}}_M(u_{2,N}) \wedge \cdots \wedge \overline{\mathrm{ED}}_M(u_{k,N-1}); && b_1 = (-1)^{k\cdot N-1}\\
&\vdots\\
f_{N-1}(u) &= \overline{\mathrm{ED}}_M(u_{1,N}) \wedge \overline{\mathrm{ED}}_M(u_{2,N}) \wedge \cdots \wedge \overline{\mathrm{ED}}_M(u_{k,1}); && b_{N-1} = (-1)^{(k-1)\cdot N+1}\\
f_N(u) &= \overline{\mathrm{ED}}_M(u_{1,N}) \wedge \overline{\mathrm{ED}}_M(u_{2,N}) \wedge \cdots \wedge \overline{\mathrm{ED}}_M(u_{k-1,N}); && b_N = (-1)^{(k-1)\cdot N}\\
f_{N+1}(u) &= \overline{\mathrm{ED}}_M(u_{1,N}) \wedge \overline{\mathrm{ED}}_M(u_{2,N}) \wedge \cdots \wedge \overline{\mathrm{ED}}_M(u_{k-1,N-1}) \wedge \overline{\mathrm{ED}}_M(u_{k,N}); && b_{N+1} = (-1)^{k\cdot N-1}\\
&\vdots\\
f_{(N+1)^k-2}(u) &= \overline{\mathrm{ED}}_M(u_{k,1}); && b_{(N+1)^k-2} = 1\\
& && b_{(N+1)^k-1} = 1.
\end{aligned}$$

Intuitively, this generalized decision list corresponds to walking through the tuples of indices $(N, N, \ldots, N), (N, N, \ldots, N-1), \cdots \in [N]^k$ in decreasing lexicographic order. The function f corresponding to a tuple $(a_1, a_2, \ldots, a_k)$ is given by $f^1(u_1) \wedge f^2(u_2) \wedge \cdots \wedge f^k(u_k)$, where $f^i(u_i)$ is an empty clause if $a_i = 0$, and $f^i(u_i) = \overline{\mathrm{ED}}_M(u_{i,a_i})$ otherwise.

Now we argue that the generalized decision list $(f_0, b_0), (f_1, b_1), \ldots$ can be compiled into a true $O(k \log n)$-decision list with length $n^{O(k)}$. Our starting point is a natural DNF representation for the $\overline{\mathrm{ED}}_M$ function with size m^3 and with $2 \log m$:

$$\overline{\mathrm{ED}}_M(x_1, \ldots, x_m) = \bigvee_{r=1}^{m} \bigvee_{i \neq j} (x_i = r) \wedge (x_j = r).$$

Each decision rule f is a conjunction of at most k of these $\overline{\mathrm{ED}}_M$ functions. Thus, each f is itself computed by a DNF of size m^{3k} and width $2k \log m$. Write each function $f = C^1 \vee C^2 \vee \cdots \vee C^{m^{3k}}$, where each C^i is a conjunction of width $2k \log m$. Then we may replace each rule (f, b) by the sequence of conjunctive rules

$$(C^1, b), (C^2, b), \ldots, (C^{m^{3k}}, b).$$

This yields a true $O(k \log M)$-decision list for $\tilde{H}_k$ with length at most $m^{3k} \cdot (N+1)^k = n^{O(k)}$.

What changes when we consider the function H_k as a function of variables $(s; z)$ instead of as a function of the derived variables $u_1, \ldots, u_k$? At first glance, it may not seem possible to write H_k as a w-decision list in $(s; z)$ for $w = \mathrm{polylog}(n)$, because for all i, the derived variable u_i depends on $\gamma(u_j)$ for all $i' < i$. A crucial observation is that because of the order in which we are evaluating the decision rules, we may equivalently evaluate each function $\overline{\mathrm{ED}}_M(u_{i,j})$ *as if of the values* $\gamma(u_{i'})$ *for* $i' < i$ *are fixed to constants*. (Namely, in a rule corresponding to a tuple $(a_1, a_2, \ldots, a_k)$, we may consider each $\gamma(u_{i'})$ as if it were fixed to $a_{i'}$). When the preceding values of $\gamma(u_1), \ldots, \gamma(u_{i-1})$ are fixed, each variable $u_{i,j,\ell}$ is simply computed by a conjunction of width $k \log(N+1) + 1$ over $(s; z)$.

Thus, each decision rule f, as a function of variables $(s; z)$, may be computed by a depth three circuit, with a top OR gate of fan-in m^{3k}, a middle level of AND gates with fan-in $2k \log m$, and a bottom level of OR and AND gates with fan-in $k \log(N+1) + 1$. Using the decomposition above implies that H_k is a polynomial-length generalized decision list where each decision rule is a CNF of width $O(k \log N)$.

This immediately shows that H_k is computed by a polynomial size circuit of depth four and bottom fan-in $O(k \log N)$.

We now see why H_k is also computed by a decision list of quasi-polynomial length. For each decision rule f, the subfunction of f computed by each AND gate in the middle level depends on at most $O(k^2 \cdot \log m \cdot \log N)$ variables. Thus, we can replace each of these subfunctions by a DNF of size $2^{O(k^2 \cdot \log m \cdot \log N)}$ and width $O(k^2 \cdot \log m \cdot \log N)$. Merging the top two levels of OR gates shows that f is itself computed by a DNF of size $2^{O(k^2 \cdot \log m \cdot \log N)}$ and width $O(k^2 \cdot \log m \cdot \log N)$ over the variables $(s; z)$.

Each function f can be then be decomposed as above to yield a true $O(k^2 \cdot \log M \cdot \log N)$-decision list for H_k with length $2^{O(k^2 \cdot \log M \cdot \log N)}$.

5.2.4 Proof of Theorem 20, General k

While the proof requires additional cumbersome notation, all of the main ideas were already contained in the proofs of Theorem 1 and Proposition 22.

Assuming Intra-Block Symmetry. Let p be a polynomial of degree at most d satisfying the hypothesis of Theorem 20. As in the proofs of Theorem 15 and Proposition 22, it is without loss of generality to assume that p satisfies the natural generalization of intra-block symmetry to inputs to H_k for $k \geq 2$. More specifically, recall that H_k is defined on inputs that can be expressed as $(s_1, (s_2, z_2), \ldots, (s_k, z_k))$, where each $s_i \in \left(\{-1,1\}^M\right)^N$ and $z_i \in \left(\left(\{-1,1\}^{(i-1)\cdot\log(N+1)}\right)^M\right)^N$, where $M = m \log m$. We think of each s_i and z_i as having a two-level hierarchical structure. At the first level, write $s_i = (s_{i,1}, \ldots, s_{i,N})$, where each $s_{i,j} \in \{-1,1\}^M$, and write $z_i = (z_{i,1}, \ldots, z_{i,N})$, where each $z_{i,j} \in \left(\{-1,1\}^{(i-1)\cdot\log(N+1)}\right)^M$. At the second level, write each $s_{i,j} = (s_{i,j,1}, \ldots, s_{i,j,m})$, where each $s_{i,j,\ell} \in \{-1,1\}^{\log m}$, and similarly write each $z_{i,j} = (z_{i,j,1}, \ldots, z_{i,j,m})$, where each $z_{i,j,\ell} \in \{-1,1\}^{(i-1)\cdot\log(N+1)\cdot\log m}$.

For any permutation $\pi \in \mathcal{S}_m$, let

$$\pi(s_{i,j}) = (s_{i,j,\pi(1)}, \ldots, s_{i,j,\pi(m)}),$$

and similarly let

$$\pi(z_{i,j}) = (z_{i,j,\pi(1)}, \ldots, z_{i,j,\pi(m)}).$$

For any two inputs $(s_1, (s_2, z_2), \ldots, (s_k, z_k))$ and $(s'_1, (s'_2, z'_2), \ldots, (s'_k, z'_k))$ to H_k, write

$$(s_1, (s_2, z_2), \ldots, (s_k, z_k)) \sim (s'_1, (s'_2, z'_2), \ldots, (s'_k, z'_k))$$

if there exist permutations $\pi_{i,j} : 1 \leq i \leq N, 1 \leq j \leq m$ such that $s_{i,j} = \pi_{i,j}(s'_{i,j})$ and $z_{i,j} = \pi_{i,j}(z'_{i,j})$ for all i, j. By a simple averaging argument analogous to Lemmas 17 and 23, we may assume that p is invariant under the relation $\sim$, i.e., that if $(s_1, (s_2, z_2), \ldots, (s_k, z_k)) \sim (s'_1, (s'_2, z'_2), \ldots, (s'_k, z'_k))$, then $p(s_1, (s_2, z_2), \ldots, (s_k, z_k)) = p(s'_1, (s'_2, z'_2), \ldots, (s'_k, z'_k))$. We refer to any such polynomial as intra-block symmetric.

Indeed, if p is does not satisfy intra-block symmetry, then it is possible, by averaging, to define a polynomial $\tilde{p}$ that does (this is analogous to Lemmas 17 and 23)). Moreover $\deg(\tilde{p}) \leq \deg(p)$, the polynomial $\tilde{p}$ satisfies the hypotheses of Theorem 20 if p does, and p satisfies the conclusion of Theorem 20 if $\tilde{p}$ does. Hence, if p is not intra-block symmetric, we apply the proof below to $\tilde{p}$ instead to conclude that $\tilde{p}$ satisfies the conclusion of Theorem 20, which implies that p does as well.

By the above discussion, Theorem 20 is an immediate consequence of the following proposition that is tailored to intra-block symmetric polynomials.

Proposition 25. There exists a universal constant $c > 0$ such that the following holds. For each $k \in \mathbb{N}$, let $c_k = 2^{-k} \cdot c$. Let p be an intra-block symmetric polynomial of degree at most $d = c_k \cdot (M/\log M)^{2/3}$ such that $p(x) \cdot H_k(x) > 0$ for all $x \in \{-1,1\}^n$. Then there exists an $x \in \{-1,1\}^n$ such that $|p(x)| \geq 2^{(N/2)^k} \cdot |p(1^n)|$.

Proof. The proof is by induction on k. Beginning with $k = 1$, note that the function H_1 is the function $G = \mathrm{OMB}_N \circ \overline{\mathrm{ED}}_M$, with each input pre-processed by an XOR with an appropriate bit of η. Hence, letting c be the universal constant appearing in the statements of Proposition 18 and Lemma 19, Proposition 18 implies that if p is a polynomial of degree $d \leq c \cdot (M/\log M)^{2/3}$ for which $p(x) \cdot H_1(x) > 0$ for all $x \in (\{-1,1\}^M)^N$, then there exists an $x \in (\{-1,1\}^M)^N$ for which $|p(x)| \geq 2^{N/2} \cdot |p((1^M)^N)|$.

Now assume the inductive hypothesis for H_k, and consider the function H_{k+1}.

Additional Notation. To enable the induction, we need to introduce more detailed notation to represent the inputs to H_{k+1}. Recall that H_{k+1} is defined over a variable set $(s_1, (s_2, z_2), \ldots, (s_{k+1}, z_{k+1}))$ where each $s_i \in (\{-1,1\}^M)^N$ and each $z_i \in \left(\left(\{-1,1\}^{\log(N+1)}\right)^{(i-1)\cdot M}\right)^N$. For convenience, we make the following relabelings:

$$s_{1,j} \oplus \eta \mapsto x_j \in \{-1,1\}^M$$
$$z_{i,j} \mapsto y_{i,j} \circ w_{i,j} \text{ where } y_{i,j} \in \left(\{-1,1\}^{\log(N+1)}\right)^M \text{ and } w_{i,j} \in \left(\{-1,1\}^{\log(N+1)}\right)^{(i-2)\cdot M}$$

Thus, we can think of H_{k+1} as being defined over variables $(x, (s_2, y_2), (s_3, (y_3, w_3)) \ldots, (s_{k+1}, (y_{k+1}, w_{k+1})))$ The following claim can be established straightforwardly from the recursive definition of H_{k+1}.

Claim 26. The function H_{k+1} may be written as

$$H_{k+1}(x, (s_2, y_2), (s_3, (y_3, w_3)), \ldots, (s_{k+1}, (y_{k+1}, w_{k+1}))) =$$
$$G(\ldots, \eta \oplus x_i, \ldots) \oplus H_k(v_2(x, s_2, y_2), v_3(x, s_3, y_3, w_3), \ldots, v_{k+1}(x, s_{k+1}, y_{k+1}, w_{k+1})),$$

where the functions v_i are defined by:

$$v_2(x, s_2, y_2) = (s_{2,j,\ell} \wedge \mathrm{EQ}_{\gamma(x)}(y_{2,j,\ell}))_{1 \leq j \leq N, 1 \leq \ell \leq M},$$

$$v_i(x, s_i, y_i, w_i) = ((s_{i,j,\ell} \wedge \mathrm{EQ}_{\gamma(x)}(y_{i,j,\ell}))_{1 \leq j \leq N, 1 \leq \ell \leq M}, \quad w_i) \qquad \text{for } i = 3, \ldots, k+1,$$

With this decomposition in mind, we use the notation $H_{k+1}(x; s; y; w)$ as shorthand for $H_{k+1}(x, (s_2, y_2), (s_3, (y_3, w_3)), \ldots, (s_{k+1}, (y_{k+1}, w_{k+1})))$. Here, $x \in (\{-1,1\}^M)^N$, while s is shorthand for

$$s = (s_2, s_3, \ldots, s_{k+1}) \in \left(\left(\{-1,1\}^M\right)^N\right)^k,$$

y is shorthand for

$$(y_2, \ldots, y_{k+1}) \in \left(\left(\left(\{-1,1\}^{\log(N+1)}\right)^M\right)^N\right)^k,$$

and w is shorthand for $(w_3, \ldots, w_{k+1})$. Similarly, for a polynomial p, we write $p(x; s; y; w)$ for $p(x, (s_2, y_2), (s_3, (y_3, w_3)), \ldots, (s_{k+1}, (y_{k+1}, w_{k+1})))$.

Consider the increasing family of sets $S_0 \subset S_1 \subset \cdots \subset S_{N/2} \subseteq (\{-1,1\}^M)^N$ as in the proof of Proposition 22. That is,

$$\begin{aligned}
&S_0 = \{\eta^N\},\\
&S_1 = \{x : x_3 = x_4 = \cdots = x_N = \eta\},\\
&\vdots\\
&S_t = \{x : x_{2t+1} = x_{2t+2} = \cdots = x_N = \eta\},\\
&\vdots\\
&S_{N/2} = (\{-1,1\}^M)^N.
\end{aligned}$$

Let p be a polynomial of degree at most d such that $p(x;s;y;w) \cdot H_{k+1}(x;s;y;w) > 0$ for all $(x;s;y;w) \in \{-1,1\}^n$.

As in the proof of Propositions 9, 12, and 22, to ease notation, we will identify any binary string in $\{-1,1\}^{\log(N+1)}$ with the number in $[N]$ for which the string is the binary representation. That is, while we will write any such binary string as though it were a number $0, 1, \ldots, N$, it should always be thought of as the binary string representing that number.

The Core Argument for Proposition 25. We iteratively construct a sequence of inputs

$$(x^0;s^0;y^0;w^0), (x^1;s^1;y^1;w^1), \ldots, (x^{N/2};s^{N/2};y^{N/2};w^{N/2})$$

to p such that:

- Each $x^t \in S_t$ and each $y^t \in (([2t]^M)^N)^k$,
- $(x^0;s^0;y^0;w^0) = (\eta^N; ((1^M)^N)^k; ((0^M)^N)^k; ((0^M)^N, ((0 \circ 0)^M)^N, \ldots, ((\underbrace{0 \circ \cdots \circ 0}_{k-1 \text{ times}})^M)^N))$, and
- $|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| \geq 2^{(N/2)^k} \cdot |p(x^t;s^t;y^t;w^t)|$ for each $i = 0, \ldots, N/2-1$.

At the conclusion of this process, we obtain an input $(x^{N/2};s^{N/2};y^{N/2};z^{N/2})$ such that

$$|p(x^{N/2};s^{N/2};y^{N/2};z^{N/2})| \geq 2^{(N/2)^k} \cdot |p(x^0;s^0;y^0;z^0)|.$$

The following claim formalizes this process.

Claim 27. Let p be an intra-block symmetric polynomial defined on the same domain as H_{k+1}. Let $c_k = 2^{-(k+1)} \cdot c$, where c is the universal constant appearing in the statement of Lemma 19. Suppose the degree of p is at most $d \leq c_{k+1}(M/\log M)^{2/3}$ and suppose $p(x;s;y;w) \cdot H_{k+1}(x;s;y;w) > 0$ for all $(x;s;y;w) \in \{-1,1\}^n$. Let $(x^t;s^t;y^t;w^t)$ be an input with $x^t \in S_t$ and $y^t \in (([2t]^M)^N)^k$. If $t+1 \leq N/2$, then there exists an input $(x^{t+1};s^{t+1};y^{t+1};w^{t+1})$ such that $|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| \geq 2^{(N/2)^k} \cdot |p(x^t;s^t;y^t;w^t)|$, where $x^{t+1} \in S_{t+1}$ and $y^{t+1} \in (([2(t+1)]^M)^N)^k$.

Proof. As usual, we prove this claim in two steps. First, we show that there exists an $x^{t+1} \in (\{-1,1\}^M)^N$ supported on S_{t+1} for which $|p(x^{t+1};s^t;y^t;w^t)| \geq |p(x^t;s^t;y^t;w^t)|$. Second, we show that there exists an $s^{t+1} \in ((\{-1,1\}^M)^N)^k$, a $y^{t+1} \in (([2(t+1)]^M)^N)^k$, and a string w^{t+1} such that $|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| \geq 2^{(N/2)^k} \cdot |p(x^{t+1};s^t;y^t;w^t)|$. Putting these steps together yields $|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| \geq 2^{(N/2)^k} \cdot |p(x^t;s^t;y^t;w^t)|$.

Step 1. To complete the first step, we examine the function $H_{k+1}(x; s^t; y^t; w^t)$, viewed as a function in x. By construction, each block $y^t_{i,j,\ell} \le 2t$. Thus, for all $x \in S_{t+1} \setminus S_t$, we have $\mathrm{EQ}_{\gamma(x)}(y^t_{i,j,\ell}) = 1$, and hence $v_2(x, s^t_2, y^t_2) = (1^M)^N$ and $v_i(x, s^t_i, y^t_i, w^t_i) = ((1^M)^N, w^t_i)$ for all $i \ge 3$. As a result, whenever $x \in S_{t+1} \setminus S_t$, we have

$$H_{k+1}(x; s^t; y^t; w^t) = G(x) \oplus H_k(1^N, (1^N, w^t_3), \ldots, (1^N, w^t_{k+1})),$$

which is either the function $G(x)$ or its negation. Without loss of generality, assume $H_{k+1}(x; s^t; y^t; w^t) = G(x)$ below.

Now consider the polynomial $q : (\{-1,1\}^M)^N \to \mathbb{R}$ defined by $q(x) = p(x; s^t; y^t; w^t)$. Then $q(x) \cdot G(x) > 0$ for all $x \in S_{t+1} \setminus S_t$. Moreover, $\deg(q) \le \deg(p)$, and since p is intra-block symmetric, so is q. By assumption, $x^t \in S_t$. Thus, by Lemma 19, there exists an $x^{t+1} \in S_{t+1}$ such that $|q(x^{t+1})| \ge 2 \cdot |q(x^t)|$. In particular, this means $|p(x^{t+1}; s^t; y^t; w^t)| \ge |p(x^t; s^t; y^t; w^t)|$.

Step 2. We now move on to complete the second step, i.e., to show that there exists an $s^{t+1} \in (((\{-1,1\}^M)^N)^k$, a $y^{t+1} \in (([2(t+1)]^M)^N)^k$, and a string w^{t+1} such that $|p(x^{t+1}; s^{t+1}; y^{t+1}; w^{t+1})| \ge 2^{(N/2)^k} \cdot |p(x^{t+1}; s^t; y^t; w^t)|$. For strings

$$\sigma = (\sigma_2, \sigma_3, \ldots, \sigma_{k+1}) \in (((\{-1,1\}^M)^N)^k, \text{ and}$$
$$\zeta = (\zeta_3, \zeta_4, \ldots, \zeta_{k+1}) \in \left(\left(\{-1,1\}^{\log(N+1)}\right)^M\right)^N \times \left(\left(\{-1,1\}^{\log(N+1)}\right)^{2\cdot M}\right)^N \times \ldots \times \left(\left(\{-1,1\}^{\log(N+1)}\right)^{(k-1)\cdot M}\right)^N$$

define the strings s_σ, y_σ and $w_{\sigma,\zeta}$ by

- For each $i = 2, \ldots, k+1, \quad 1 \le j \le N, 1 \le \ell \le M$:

$$(s_\sigma)_{i,j,\ell} = \begin{cases} -1 & \text{if } \sigma_{i,j,\ell} = -1, \\ s^t_{i,j,\ell} & \text{if } \sigma_{i,j,\ell} = 1, \end{cases}$$

- For each $i = 2, \ldots, k+1, \quad 1 \le j \le N, 1 \le \ell \le M$:

$$(y_\sigma)_{i,j,\ell} = \begin{cases} \gamma(x^{t+1}) & \text{if } \sigma_{i,j,\ell} = -1, \\ y^t_{i,j,\ell} & \text{if } \sigma_{i,j,\ell} = 1, \end{cases}$$

- For each $i = 3, \ldots, k+1, \quad 1 \le j \le N, 1 \le \ell \le (i-2) \cdot M$:

$$(w_{\sigma,\zeta})_{i,j,\ell} = \begin{cases} \zeta_{i,j,\ell} & \text{if } \sigma_{i,j,\ell} = -1, \\ w^t_{i,j,\ell} & \text{if } \sigma_{i,j,\ell} = 1. \end{cases}$$

Observe that this parametrization has the property that if $\sigma = ((1^M)^N)^k$, then

$$(x^{t+1}; s_\sigma, y_\sigma; w_{\sigma,\zeta}) = (x^{t+1}; s^t; y^t; w^t).$$

Note that since $\gamma(x^{t+1}) \le 2(t+1)$ and each $y^t_{i,j,\ell} \in [2t]$, we have that $y_\sigma \in (([2(t+1)]^M)^N)^k$ for every $\sigma \in (((\{-1,1\}^M)^N)^k$. We can thus calculate

$$v_2(x^{t+1}, (s_\sigma)_2, (y_\sigma)_2) = ((s_\sigma)_{2,j,\ell} \wedge \mathrm{EQ}_{\gamma(x^{t+1})}((y_\sigma)_{2,j,\ell}))_{1\le j\le N, 1\le \ell\le M} = (\sigma_{2,j,\ell})_{1\le j\le N, 1\le \ell\le M},$$

$$v_i(x^{t+1},(s_\sigma)_i,(y_\sigma)_i,(w_{\sigma,\zeta})_i) = (\quad ((s_\sigma)_{i,j} \wedge \mathrm{EQ}_{\gamma(x^{t+1})}((y_\sigma)_{i,j}))_{1\le j\le N,1\le \ell\le M},\quad (w_{\sigma,\zeta})_i \quad)$$
$$= (\quad (\sigma_{i,j})_{1\le j\le N,1\le \ell\le M},\quad (w_{\sigma,\zeta})_i \quad).$$

Consider the function $E(\sigma;\zeta) := H_{k+1}(x^{t+1};s_\sigma;y_\sigma;w_{\sigma,\zeta})$. By the calculations above,

$$\begin{aligned}E(\sigma;\zeta) &= G(x^{t+1}) \oplus H_k(v_2(x^{t+1},(s_\sigma)_2,(y_\sigma)_2),\ldots,v_{k+1}(x^{t+1},(s_\sigma)_{k+1},(y_\sigma)_{k+1},(w_{\sigma,\zeta})_{k+1}))\\ &= G(x^{t+1}) \oplus H_k(\sigma_2,(\sigma_3,(w_{\sigma,\zeta})_3),\ldots,(\sigma_{k+1},(w_{\sigma,\zeta})_{k+1}))\\ &= G(x^{t+1}) \oplus H_k(\sigma_2,(\sigma_3,\zeta_3),\ldots,(\sigma_{k+1},\zeta_{k+1})).\end{aligned}$$

Now consider the polynomial $r(\sigma;\zeta) := p(x^{t+1};s_\sigma;y_\sigma;w_{\sigma,\zeta})$. Since the variables s_σ, y_σ, and $w_{\sigma,\zeta}$ can be written as linear or quadratic functions of σ and ζ, the polynomial r satisfies $\deg r \le 2\deg p \le c_k \cdot (M/\log M)^{2/3}$. Moreover, $r(\sigma;\zeta)\cdot E(\sigma;\zeta) > 0$ for all (σ,ζ), and r is intra-block symmetric. Since E is either the function H_k or its negation, the inductive hypothesis for Proposition 25 (i.e, the statement of Proposition 25 for k instead of $k+1$) allows us to conclude that there exists a (σ^*,ζ^*) such that $|r(\sigma^*;\zeta^*)| \ge 2^{(N/2)^k}\cdot|r(1^{n_k})|$, where n_k is the number of Boolean variables on which H_k is defined. Setting $s^{t+1} := s_{\sigma*}$, $y^{t+1} := y_{\sigma*}$ and $w^{t+1} := w_{\sigma^*,\zeta^*}$ thus yields

$$|p(x^{t+1};s^{t+1};y^{t+1};w^{t+1})| = |r(\sigma^*;\zeta^*)| \ge 2^{(N/2)^k}\cdot|r(1^{n_k})| = 2^{(N/2)^k}\cdot|p(x^{t+1};s^t;y^t;w^t)|,$$

as we wanted to show. □

With Claim 27 established, the inductive proof of Proposition 25 is complete. □

6 Applications

6.1 Threshold Weight of AC^0

A *polynomial threshold function* (PTF) for a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ is a polynomial $p : \{-1,1\}^n \to \mathbb{R}$ with *integer coefficients* that agrees in sign with f on all Boolean inputs. The *weight* of an n-variate polynomial p is the sum of the absolute values of its coefficients. The *degree-d threshold weight* of a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, denoted $W(f,d)$, is defined to be the least weight of a degree-d PTF for f. We let $W(f)$ denote the quantity $W(f,n)$, i.e., the least weight of any threshold function for f regardless of its degree. Threshold weight upper bounds underly some of the most powerful techniques in computational learning theory based on the classic Perceptron [23] and Winnow [22] algorithms (see [10, Section 8.3] for a discussion). Thus, our threshold weight lower bounds impose limitations on how efficiently such algorithms can learn depth three circuits.

Degree-d threshold weight is closely related to ε-approximate degree when ε is very close to 1:

Lemma 28. Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function, and let $w > 0$. If $\widetilde{\deg}_{1-\frac{1}{w}}(f) > d$, then $W(f,d) > w$.

Proof. We prove the contrapositive, i.e., that any PTF p for f having weight w and degree d can be transformed into a uniform approximation to f with error $1-\frac{1}{w}$. Let p be such a PTF. Since p has integer coefficients and is nonzero on Boolean inputs, $|p(x)| \ge 1$ on $\{-1,1\}^n$. Moreover, $|p(x)| \le w$ by the weight bound, so the polynomial $\frac{1}{w}\cdot p(x)$ satisfies $|\frac{1}{w}\cdot p(x) - f(x)| \le 1-\frac{1}{w}$ for every $x \in \{-1,1\}^n$. □

Thus, our main results yield new lower bounds on the degree-d threshold weight of circuits of depth three and four.

Corollary 29. For any arbitrarily small constant $\delta > 0$ and any arbitrarily large constant $\Gamma > 1$, there exist:

1. A depth three Boolean circuit $f\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that $W(f_1, n^{1/2-\delta}) > 2^{n^\Gamma}$.
2. A depth four Boolean circuit $f;\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that $W(f, n^{2/3-\delta}) > 2^{n^\Gamma}$. The function f' is also computed by a depth three circuit of quasi-polynomial size and bottom fan-in $O(\log^2 n)$.

Moreover, a result of Krause [17] allows us to translate each of these lower bounds into a *degree independent* threshold weight lower bounds for a related function.

Lemma 30 ([17], Lemma 3.4)**.** Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function, and define $F : \{-1,1\}^{3n} \to \{-1,1\}$ by

$$F(x_1,\dots,x_n,y_1,\dots,y_n,z_1,\dots,z_n) := f(\dots,(\bar{z}_i \wedge x_i) \vee (z_i \wedge y_i),\dots).$$

Then $W(F) \geq W(f,d)$ for all d for which $2^d \geq W(f,d)$.

When this transformation is applied to a function f computed by a Boolean circuit of depth d with logarithmic bottom fan-in, the resulting function F is also computed by a depth d circuit with logarithmic bottom fan-in. To see this, note that if g is any function that depends on $O(\log n)$ variables, then $G(x,y,z) := g((\bar{z} \wedge x) \vee (z \wedge y))$ also depends on $O(\log n)$ variables. Hence, G is computed by either a DNF or CNF of size $\mathrm{poly}(n)$ and bottom fan-in $O(\log n)$. So while F is naturally computed by a circuit of depth $d+2$, the bottom three levels of gates can be replaced by such DNF or CNF formulae so as to merge a layer of gates and obtain a depth d circuit with logarithmic bottom fan-in.

The same argument shows that if f is computed by a quasi-polynomial size circuit of depth d with polylogarithmic bottom fan-in, then the resulting function F is also computed by a depth d circuit with quasi-polynomial size and polylogarithmic bottom fan-in.

Corollary 31. For any arbitrarily small constant $\delta > 0$, there exist:

1. A depth three Boolean circuit $F\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that $W(F) > \exp(\Omega(n^{1/2-\delta}))$.
2. A depth four Boolean circuit $F'\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that $W(F') > \exp(\Omega(n^{2/3-\delta}))$. The function F' is also computed by a depth three circuit of quasi-polynomial size and bottom fan-in $O(\log^2 n)$.

While the weight bounds of Corollaries 29 and 31 are stated for polynomial threshold functions over $\{-1,1\}^n$ (i.e., for polynomials that are integer linear combinations of parities), a now standard transformation [19] shows that the same threshold weight lower bound also holds for polynomials over $\{0,1\}^n$ (i.e., for integer linear combinations of conjunctions) up to polynomial factors.

6.2 Discrepancy of $\mathbf{AC}^0$

Discrepancy is a central quantity in communication complexity and circuit complexity. For instance, an upper bound on the discrepancy of a Boolean function $f : X \times Y \to \{-1,1\}$ yields lower bounds for computing f in essentially every model of communication complexity. In particular, the discrepancy of

f essentially characterizes it's small-bias communication complexity in the **PP** model of Babai et al. [4]. Theorem 1 yields a new exponentially small upper bound on the discrepancy of a depth three circuit, while Theorem 2 yields a new upper bound for any function in AC^0.

For a Boolean function $f : X \times Y \to \{-1, 1\}$, let $M^{(f)}$ be its communication matrix $M^{(f)} = [f(x, y)]_{x \in X, y \in Y}$. A combinatorial rectangle of $X \times Y$ is a set of the form $A \times B$ with $A \subseteq X$ and $B \subseteq Y$. For a distribution μ over $X \times Y$, the discrepancy of f with respect to μ is defined to be the maximum over all rectangles R of the *bias* of f on R. That is:

$$\mathrm{disc}_\mu(f) = \max_R \left| \sum_{(x,y) \in R} \mu(x, y) f(x, y) \right|.$$

The discrepancy of f, denoted $\mathrm{disc}(f)$, is defined to be $\min_\mu \mathrm{disc}_\mu(f)$.

Sherstov's pattern matrix method [31] shows how to generically transform an AC^0 function with high threshold degree or high threshold weight into another AC^0 function with low discrepancy.

Theorem 32 (cf. [31], adapted from Corollary 1.2 and Theorem 7.3)**.** Let $f : \{-1, 1\}^n \to \{-1, 1\}$ be given, and define the communication problem $F : \{-1, 1\}^{4n} \times \{-1, 1\}^{4n} \to \{-1, 1\}$ by

$$F(x, y) = f(\ldots, \vee_{j=1}^4 (x_{i,j} \wedge y_{i,j}), \ldots).$$

Then for every integer $d \geq 0$, we have

$$\mathrm{disc}(F)^2 \leq \max \left\{ \frac{2n}{W(f, d-1)}, 2^{-d} \right\}.$$

Recall that $W(f, d-1)$ is the least weight of any degree $d-1$ PTF for f. We apply the pattern matrix method to the functions f, f' of Corollary 29. By the same argument as in Section 6.1, the pattern matrix method does not increase the depth of the circuits computing these functions. We thus obtain new discrepancy upper bounds for circuits of depth three and four:

Corollary 33. For any arbitrarily small constant $\delta > 0$, there exist:

1. A depth three Boolean circuit $F\colon \{-1, 1\}^n \to \{-1, 1\}$ (with logarithmic bottom fan-in) such that $\mathrm{disc}(F) < \exp(-\Omega(n^{1/2-\delta}))$.
2. A depth four Boolean circuit $F'\colon \{-1, 1\}^n \to \{-1, 1\}$ (with logarithmic bottom fan-in) such that $\mathrm{disc}(F') < \exp(-\Omega(n^{2/3-\delta}))$. The function F' is also computed by a depth three Boolean circuit of quasi-polynomial size and bottom fan-in $O(\log^2 n)$.

Application to Circuit Complexity. It is well-known that a discrepancy upper bound for a function F yields a lower bound on the size of Majority-of-Threshold circuits computing F [12, 14, 24, 30]. Indeed, the exponential Majority-of-Threshold circuit size lower bounds of [8,10,29–31,34] for AC^0 are all proved using discrepancy. Our discrepancy upper bound of Corollary 33 sharpens these previous lower bounds by yielding a depth three Boolean circuit F of polynomial size such that any Majority-of-Threshold circuit computing F requires size $\exp\left(\Omega(n^{1/2-\delta})\right)$, and a depth four Boolean circuit requiring size $\exp\left(\Omega(n^{2/3-\delta})\right)$, for any $\delta > 0$.

Corollary 34. For any arbitrarily small constant $\delta > 0$, there exist:

1. A depth three Boolean circuit $F\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that any Majority-of-Threshold circuit computing F has size at least $\exp(\Omega(n^{1/2-\delta}))$.

2. A depth four Boolean circuit $F'\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that any Majority-of-Threshold circuit computing F' has size at least $\exp(\Omega(n^{2/3-\delta}))$. The function F' is also computed by a depth three Boolean circuit of quasi-polynomial size and bottom fan-in $O(\log^2 n)$.

Combining Corollaries 31, 33, and 34 yields Corollaries 4 and 5 from the introduction.

References

[1] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004.

[2] Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1(1):37–46, 2005.

[3] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM J. Comput.*, 37(1):210–239, 2007.

[4] László Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory (preliminary version). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 337–347. IEEE Computer Society, 1986.

[5] Paul Beame and Widad Machmouchi. The quantum query complexity of ac^0. *Quantum Information & Computation*, 12(7-8):670–676, 2012.

[6] Richard Beigel. Perceptrons, PP, and the Polynomial Hierarchy. *Computational Complexity*, 4:339–349, 1994.

[7] Richard Beigel, Nick Reingold, and Daniel A. Spielman. PP is closed under intersection. *J. Comput. Syst. Sci.*, 50(2):191–202, 1995.

[8] Harry Buhrman, Nikolai K. Vereshchagin, and Ronald de Wolf. On computation and communication with small bias. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 24–32. IEEE Computer Society, 2007.

[9] Mark Bun and Justin Thaler. Dual lower bounds for approximate degree and markov-bernstein inequalities. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 303–314. Springer, 2013.

[10] Mark Bun and Justin Thaler. Hardness amplification and the approximate degree of constant-depth circuits. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 268–280. Springer, 2015. Full version available at `http://eccc.hpi-web.de/report/2013/151`.

[11] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco A. Servedio, and Emanuele Viola. Bounded independence fools halfspaces. *SIAM J. Comput.*, 39(8):3441–3462, 2010.

[12] Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. Majority gates VS. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992.

[13] Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:49, 2015.

[14] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993.

[15] Adam R. Klivans and Rocco A. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. *J. Comput. Syst. Sci.*, 68(2):303–318, 2004.

[16] Adam R. Klivans and Rocco A. Servedio. Toward attribute efficient learning of decision lists and parities. *Journal of Machine Learning Research*, 7:587–602, 2006.

[17] Matthias Krause. On the computational power of boolean decision lists. *Computational Complexity*, 14(4):362–375, 2006.

[18] Matthias Krause and Pavel Pudlák. On the computational power of depth-2 circuits with threshold and modulo gates. *Theor. Comput. Sci.*, 174(1-2):137–156, 1997.

[19] Matthias Krause and Pavel Pudlák. Computing boolean functions by polynomials and threshold circuits. *Computational Complexity*, 7(4):346–370, 1998.

[20] Troy Lee. A note on the sign degree of formulas. *CoRR*, abs/0909.4607, 2009.

[21] Nati Linial and Adi Shraibman. Learning complexity vs. communication complexity. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, 23-26 June 2008, College Park, Maryland, USA*, pages 53–63. IEEE Computer Society, 2008.

[22] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1987.

[23] Marvin Minsky and Seymour Papert. *Perceptrons - an introduction to computational geometry*. MIT Press, 1969.

[24] Noam Nisan. The communication complexity of threshold gates. In *Combinatorics, Paul Erdos is Eighty*, pages 301–315, 1994.

[25] Vladimir V. Podolskii. A uniform lower bound on weights of perceptrons. In Edward A. Hirsch, Alexander A. Razborov, Alexei L. Semenov, and Anatol Slissenko, editors, *Computer Science - Theory and Applications, Third International Computer Science Symposium in Russia, CSR 2008, Moscow, Russia, June 7-12, 2008, Proceedings*, volume 5010 of *Lecture Notes in Computer Science*, pages 261–272. Springer, 2008.

[26] Vladimir Vladimirovich Podolskii. Perceptrons of large weight. *Problems of Information Transmission*, 45(1):46–53, 2009.

[27] Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

[28] Rocco A. Servedio, Li-Yang Tan, and Justin Thaler. Attribute-efficient learning and weight-degree tradeoffs for polynomial threshold functions. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *COLT*, volume 23 of *JMLR Proceedings*, pages 14.1–14.19. JMLR.org, 2012.

[29] A. A. Sherstov. The power of asymmetry in constant-depth circuits. In *FOCS*, 2015. Full version available at `http://eccc.hpi-web.de/report/2015/147/`.

[30] Alexander A. Sherstov. Separating AC^0 from depth-2 majority circuits. *SIAM J. Comput.*, 38(6):2113–2129, 2009.

[31] Alexander A. Sherstov. The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011.

[32] Alexander A. Sherstov. Approximating the and-or tree. *Theory of Computing*, 9(20):653–663, 2013.

[33] Alexander A. Sherstov. The intersection of two halfspaces has high threshold degree. *SIAM J. Comput.*, 42(6):2329–2374, 2013.

[34] Alexander A. Sherstov. Breaking the Minsky-Papert barrier for constant-depth circuits. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 223–232. ACM, 2014.

[35] Justin Thaler. Lower bounds for the approximate degree of block-composed functions. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:150, 2014. To appear in *ICALP*, 2016.

[36] Leslie G. Valiant. A theory of the learnable. In Richard A. DeMillo, editor, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 436–445. ACM, 1984.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de