Electronic Colloquium on Computational Complexity, Revision 1 of 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 exclusively study *block-composed* functions. Such 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 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.

Our methods go beyond the block-composed functions studied in prior work, and hence may not be subject to the same barriers. In particular, we 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.

*An earlier version of this manuscript claimed to exhibit constant-depth circuits of polynomial size and complexity $\exp(\Omega(n^{2/3-\delta}))$. An anonymous reviewer identified an error in the proof of that claim, and we are retracting it accordingly.

†Princeton University. This work was done while the author was at Harvard University and visiting Yale University, supported by an NDSEG Fellowship and NSF grant CNS-1237235.

‡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 5.2, 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 [22].

  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 5.

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 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 a depth three circuit of polynomial

[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 $[f(x,y)]_{x,y \in \{-1,1\}^n}$.

size (in fact, an $O(\log n)$-decision list) of complexity $\exp(\Omega(n^{1/2-\delta}))$ under each of these measures. Our improvement over prior work stems from the fact that we move beyond block-composed functions, and hence our methods may not be subject to the same barriers. In particular, we 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).

The functions that underly our analysis are rather complicated to define (see Section 4.2 for details), but here we briefly highlight their novel features. Inspired by prior work of Podolskii [26] (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, but complicated enough that the final function has much higher complexity than the block-composed functions considered in prior work.

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 [18,19,32] 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.

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, 14, 17, 18, 28, 29, 39]).

By combining Theorem 1 with known transformations [19, 32], we obtain a depth three circuit F with very large complexity under the three measures described above. In fact, F itself is computed by an $O(\log n)$-decision list.

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

Table 1 succinctly compares our results to prior work.

Reference	Threshold Weight Bound	Discrepancy Bound	Majority-of-Threshold Circuit Size Bound	Circuit Depth
[19]	$\exp(\Omega(n^{1/3}))$	N/A	N/A	3
[31]	N/A	$\exp(-\Omega(n^{1/5}))$	$\exp(\Omega(n^{1/5}))$	3
[8,32]	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
[36]	$\exp\left(\Omega(n^{\frac{k-1}{2k-1}})\right)$	$\exp\left(-\Omega(n^{\frac{k-1}{2k-1}})\right)$	$\exp\left(\Omega(n^{\frac{k-1}{2k-1}})\right)$	$k+1$ (for $k \geq 2$)
[30]	$\exp\left(\Omega(n^{1/2})\right)$	$\exp\left(-\Omega(n^{1/2})\right)$	$\exp\left(\Omega(n^{1/2})\right)$	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

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.

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 pointwise 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 pointwise 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 [24] 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 [16] 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 [17] 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]

[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.

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 [26, 27].[5] In [27], 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 [26]. 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.

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 [19] 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 [18] 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 [32] 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 [25].

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, 21, 30, 30, 33, 35, 36]. 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.

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

[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.

Let ED_M denote the well-known Element Distinctness function and $\overline{\mathrm{ED}}_M$ its negation. ED_M has played a central role in recent works on hardness amplification for approximate degree [10,30,36] 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 [38] 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 [36] significantly strengthened the approach of [10] to obtain new *threshold degree* lower bounds for functions in AC^0. Specifically, in [36], 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 [18, 19, 32] to these circuits increases their depth by 1. In [30], Sherstov exhibited a depth four circuit of logarithmic bottom fan-in and threshold degree $\Omega(n^{1/2})$ – applying the transformation of [18, 19, 32] 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 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, 38] 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, 38] 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, 38] 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

[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, 30, 36]. 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].

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 [30, Theorem 6.10]).

Sherstov [30,36] introduced sophisticated and demanding methods that can prove stronger lower bounds for constant-depth circuits than [10,38]. 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 [10,38] only for circuits of greater depth or bottom fan-in than considered in those works. Furthermore, in [30, 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 [30], 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 Application in Communication Complexity

We briefly describe an additional application of our results. By combining Theorem 1 with standard machinery, we obtain an improved separation between the analogues of the complexity classes **PP** and $\mathbf{P}^{\mathbf{NP}}$ in communication and query complexity. Specifically, for a function $F\colon \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$, let $\mathbf{PP}(F)$ and $\mathbf{P}^{\mathbf{NP}}(F)$ respectively denote the least cost of a **PP** and $\mathbf{P}^{\mathbf{NP}}$ communication protocol for F. In both the communication and query complexity settings, for any constant $\delta > 0$, we exhibit an F satisfying $\mathbf{PP}(F) = \Omega(n^{1/2-\delta})$ and $\mathbf{P}^{\mathbf{NP}}(F) = O(\log^2 n)$. This improves over prior work that gave an F satisfying $\mathbf{PP}(F) = \tilde{\Omega}(n^{2/5})$ and $\mathbf{P}^{\mathbf{NP}}(F) = O(\log n)$ [38] and earlier work of Buhrman et al. that gave an F satisfying $\mathbf{PP}(F) = \Omega(n^{1/3})$ and $\mathbf{P}^{\mathbf{NP}}(F) = O(\log n)$ [8]. We direct the interested reader to [38] 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 3 (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 $p(x) \cdot \mathrm{OMB}_N(x) > 0$ for all $x \in \{-1,1\}^N$. Then there exists an $x \in \{-1,1\}^N$ such that $|p(x)| \geq 2^{cN/d^2} \cdot |p(1^N)|$. In particular, $\widetilde{\deg}_\varepsilon(\mathrm{OMB}_N) \geq d$ for some $\varepsilon = 1 - 2^{\Omega(N/d^2)}$.

To prove Theorem 3, 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 4 (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 $|p(z')| \geq 2 \cdot |p(z)|$.

3 Intuition and Discussion of Theorem 1

3.1 Overview of Our Function

As mentioned in Section 1, the function f that we exhibit in Theorem 1 is complicated to define. Hence, before formally defining f, we provide here some motivation for our definition. While the function we describe in this section differs from the f exhibited in Theorem 1, our description here highlights the main ideas underlying the construction of f itself.

Specifically, the function that we describe in this informal overview is a modification of the block-composed function $\mathrm{OMB}_N \circ \mathrm{OR}_M$. This is easily seen to be a sub-function of OMB_{2NM}, and in the formal statement and proof of Theorem 1, our construction relies on the function OMB_{2NM} directly. This is only for the sake of simplicity, and we remark that the proof of Theorem 1 carries over when one uses the function $\mathrm{OMB}_N \circ \mathrm{OR}_M$ instead. We choose to present this overview using the function $\mathrm{OMB}_N \circ \mathrm{OR}_M$ for two reasons. First, reasoning about this function gives the right intuition for both our lower bound and for the approximating polynomial which it matches. Second, as discussed in Section 3.3, replacing the inner function OR_M with other functions (such as $\overline{\mathrm{ED}}_M$) yields natural candidates for further improved lower bounds.

Recall that [38] 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 $\varepsilon = 1 - 2^{-O(N \log M)}$. The methods of [38] also show that

$\widetilde{\deg}_\varepsilon(\mathrm{OMB}_N \circ \mathrm{OR}_M) = \Omega(M^{1/2})$ for $\varepsilon = 1 - 2^{-N}$, and there is a polynomial of degree $O(1)$ that approximates $\mathrm{OMB}_N \circ \mathrm{OR}_M$ to error $\varepsilon = 1 - 2^{-O(N \log M)}$.

Our goal is to modify $\mathrm{OMB}_N \circ \mathrm{OR}_M$ 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 $O(\log n)$ decision list. Specifically, we will require, for some large constant k and small constant $\delta > 0$, $\widetilde{\deg}_\varepsilon(f) = \Omega(M^{1/2-\delta})$ for $\varepsilon = 1 - 2^{-N^k}$.

A natural first attempt to construct such an f is to block-compose $\mathrm{OMB}_N \circ \mathrm{OR}_M$ 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: $\oplus_k \circ \mathrm{OMB}_N \circ \mathrm{OR}_M$, 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 $\oplus_k \circ \mathrm{OMB}_N \circ \mathrm{OR}_M$ to error $1 - 2^{-O(kN \log M)}$. Indeed, letting p be the polynomial approximation to $\mathrm{OMB}_N \circ \mathrm{OR}_M$ 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 $\oplus_k \circ \mathrm{OMB}_N \circ \mathrm{OR}_M$ to foil this construction of an approximating polynomial. Specifically, our f will first "pre-process" its input $(x_1, \ldots, x_k)$, before feeding it into $\oplus_k \circ \mathrm{OMB}_N \circ \mathrm{OR}_M$. 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 4.2). 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 $\mathrm{OMB}_N \circ \mathrm{OR}_M$; 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 $\mathrm{OMB}(x_{1,j}) = -1$, 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 $(\mathrm{OMB}_N \circ \mathrm{OR}_M)(x_1)$.

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

$$f = ((\mathrm{OMB}_N \circ \mathrm{OR}_M)(x_1)) \oplus ((\mathrm{OMB}_N \circ \mathrm{OR}_M)(u)) \,. \tag{2}$$

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 $\mathrm{OMB}_N \circ \mathrm{OR}_M(x_1)$. This mild dependence is what allows f to be computed by an $O(\log n)$ decision list.

It is not hard to see that an equivalent way to write f (that moreover helps reveal its structure as an

$O(\log n)$-decision list) is:

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

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. 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 1) 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 $\mathrm{OMB}_N \circ \mathrm{OR}_M$. Our starting point is a polynomial q of degree $O(M^{1/2})$ satisfying the following two properties (cf. [38]).

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

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

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

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

In fact, $3^{-N-1} \cdot g(x)$ approximates $\mathrm{OMB}_N \circ \mathrm{OR}_M$ 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

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

Using techniques similar to the above, one can obtain a degree $\tilde{O}(M^{1/2} \log N)$ 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

Fix a small constant $\delta > 0$ and large constant $\Gamma > 0$. A remarkable feature of the function f exhibited in Theorem 1 is that it simultaneously satisfies the following three properties:

- It has threshold degree $O(\log n)$.
- It has approximate degree $\tilde{\Theta}(n^{1/2})$.
- It has ε-approximate degree $\Omega(n^{1/2-\delta})$ for $\varepsilon = 1 - 2^{n^{\Gamma}}$.

Hence, while f has very low threshold degree, it is essentially as hard to approximate f to error *super-exponentially* close to 1 (i.e., error as large as $1 - 2^{-n^{\Gamma}}$ for any constant $\Gamma > 0$), as it is to approximate to error $1/3$. To the best of our knowledge, ours is the first known function to exhibit these properties.

Clearly, improving the degree bound in Theorem 1 to be polynomially larger than $\Omega(n^{1/2})$ will require considering functions of approximate degree larger than $\Omega(n^{1/2})$. A natural approach is to exhibiting such functions is to simply replace the function OR_M appearing in the construction of Section 3.1 with a function of approximate degree polynomially larger than $\Omega(M^{1/2})$. A prime candidate is the function $\overline{\mathrm{ED}}_M$, which has approximate degree $\tilde{\Omega}(M^{2/3})$ (recall that this is currently the best approximate degree lower bound known for *any* function in AC^0).

Indeed, in a prior version of this paper, we claimed to use precisely this approach to exhibit a polynomial size depth four circuit (and quasipolynomial size depth three circuit) of complexity $\exp(\Omega(n^{2/3-\delta}))$ under each of the measures we consider. Unfortunately, we have retracted this claim because of an error in the proof. Nonetheless, we believe that this is a viable approach to breaking the $\exp(\Theta(n^{1/2}))$ barrier to which methods based on block-composed functions are subject.

It is further natural to conjecture that if OR_M is replaced in the construction with a function of even larger approximate degree then $\overline{\mathrm{ED}}_M$, then the resulting function has yet larger complexity. For example, the SURJECTIVITY function on M bits is computed by a polynomial size Boolean circuit of depth three (see [5] for the definition of this function), and it is reasonable to conjecture that this function has approximate degree $\tilde{\Omega}(M)$ [5,10]. We further conjecture that replacing OR_M with SURJECTIVITY in the construction of Section 3.1 yields a function of complexity $\exp(\Omega(n^{1-\delta}))$ for any $\delta > 0$. Tweaking the parameters in the construction and sharpening the analysis may even yield a lower bound of $\exp(\tilde{\Omega}(n))$.

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 $\mathrm{EQ}_{\beta(x)} \colon \{-1,1\}^{\log(N+1)} \to \{-1,1\}$ 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 $\mathrm{EQ}_{\beta(x)}$ to each block of y. That is, u is the vector obtained by first "pre-processing" each block of y with an "equality test" $\mathrm{EQ}_{\beta(x)}$ that is determined by x.

Function Definition. Define F via:

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

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

Proposition 5. 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 3, we iteratively apply Lemma 4 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 4 a total of $(cN/d^2)^2$ times. This is a quadratic improvement over the number of times Beigel is able to apply Lemma 4 to OMB_N itself. Podolskii [26] 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 3) for OMB_N started with the input $x^0 = 1^N$, and iteratively applied Lemma 4 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 3) 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 3) 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 5. 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 4. 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 6. 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 $y_i^t \leq t\ell$. Thus, $\mathrm{EQ}_{\beta(x)}(y_i^t) = 1$ for all $x \in S_{t+1} \setminus S_t$, and hence

$$\mathrm{OMB}_N(\mathrm{EQ}_{\beta(x)}(y_1^t), \ldots, \mathrm{EQ}_{\beta(x)}(y_N^t)) = \mathrm{OMB}_N(1^N) = 1$$

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 4, 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 $(y_w)_i = y_i^t$ if $w_i = 1$. Note that since $\beta(x^{t+1}) \leq (t+1)\ell$ and each $y_i^t \in [t\ell]$, 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 a 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 3 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 6 established. we conclude the proof of Proposition 5.

□

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, $F_1, F_2, \ldots F_k$ 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:

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

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

$$\vdots$$

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

Remark 7. 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 5). 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

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

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 5.

Proposition 8. There exists a universal constant $c > 0$ such that for each $k \in \mathbb{N}$, there exists a $c_k \geq c \cdot 4^{-k^2}$ 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 $|p(x)| \geq 2^{c_k (N/d^2)^k} \cdot |p(1^n)|$.

Theorem 1 follows easily from Proposition 8.

Proof of Theorem 1, assuming Proposition 8. 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 8, 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 8. 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 3 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 8 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 5, 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 9. 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:

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

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

The Main Argument. Just as in Lemma 4 and Proposition 5, 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 10. 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 6, 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 4, 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 8) 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 8 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 $i = 2, \dots, k+1$,

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

- 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 9. Note that since $\beta(x^{t+1}) \le (t+1)\ell$ 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 $y^t_{i,j} \le t\ell$ 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 $\deg r \le 2 \deg p \le 2d$.

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 8) 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 10. □

With Claim 10 completed, we conclude the proof of Proposition 8.

□

5 Applications

5.1 Threshold Weight of Depth Three Circuits

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 [24] and Winnow [23] 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 11. 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, $|p(x)| \geq 1$ on $\{-1,1\}^n$. Moreover, $|p(x)| \leq w$ by the weight bound, so the polynomial $\frac{1}{w} \cdot p(x)$ satisfies $|\frac{1}{w} \cdot p(x) - f(x)| \leq 1 - \frac{1}{w}$ 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.

Corollary 12. For any arbitrarily small constant $\delta > 0$ and any arbitrarily large constant $\Gamma > 1$, there exists a depth three Boolean circuit $f\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that $W(f_1, n^{1/2-\delta}) > 2^{n^\Gamma}$.

Moreover, a result of Krause [18] allows us to translate each of these lower bounds into a *degree independent* threshold weight lower bound for a related function.

Lemma 13 ([18], 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

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

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.

Corollary 14. For any arbitrarily small constant $\delta > 0$, there exists a depth three Boolean circuit $F\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that $W(F) > \exp(\Omega(n^{1/2-\delta}))$.

While the weight bounds of Corollaries 12 and 14 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 [20] 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.

5.2 Discrepancy of Depth Three Circuits

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.

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 [32] 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 15 (cf. [32], 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 12. By the same argument as in Section 5.1, the pattern matrix method does not increase the depth of the circuits computing these functions. We thus obtain a new discrepancy upper bound for circuits of depth three:

Corollary 16. For any arbitrarily small constant $\delta > 0$, there exists a depth three Boolean circuit $F\colon \{-1,1\}^n \to \{-1,1\}$ (with logarithmic bottom fan-in) such that $\mathrm{disc}(F) < \exp(-\Omega(n^{1/2-\delta}))$.

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 [13, 15, 25, 31]. Indeed, the exponential Majority-of-Threshold circuit size lower bounds of [8, 10, 30–32, 36] for AC^0 are all proved using discrepancy. Our discrepancy upper bound of Corollary 16 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 $\exp\left(\Omega(n^{1/2-\delta})\right)$.

Corollary 17. For any arbitrarily small constant $\delta > 0$, there exists 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 $\exp(\Omega(n^{1/2-\delta}))$.

Combining Corollaries 14, 16, and 17 yields Corollary 2 from the introduction.

Acknowledgements. We are indebted to the anonymous reviewer from STOC 2017 who identified an error in an earlier version of this manuscript.

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] Arkadev Chattopadhyay and Nikhil Mande. Small error versus unbounded error protocols in the NOF model. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:95, 2016.

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

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

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

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

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

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

[22] 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.

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

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

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

[26] 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.

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

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

[29] 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.

[30] 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/`.

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

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

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

[34] Alexander A. Sherstov. Communication lower bounds using directional derivatives. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 921–930. ACM, 2013.

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

[36] 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.

[37] Alexander A. Sherstov. On multiparty communication with large versus unbounded error. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:138, 2016.

[38] 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.

[39] 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