Depth-d Threshold Circuits vs. Depth-$(d+1)$ AND-OR Trees

Pooya Hatami
Ohio State University
Department of Computer Science and Engineering
Columbus, OH, USA
pooyahat@gmail.com

William M. Hoza
University of California, Berkeley
Simons Institute for the Theory of Computing
Berkeley, CA, USA
williamhoza@berkeley.edu

Avishay Tal
University of California, Berkeley
Department of Electrical Engineering and Computer Sciences
Berkeley, CA, USA
atal@berkeley.edu

Roei Tell
Institute for Advanced Study
School of Mathematics
Princeton, NJ, USA
Rutgers University
DIMACS
Piscataway, NJ, USA
roeitell@gmail.com

ABSTRACT

For any $n \in \mathbb{N}$ and $d = o(\log \log n)$, we prove that there is a Boolean function F on n bits and a value $\gamma = 2^{-\Theta(d)}$ such that F can be computed by a uniform depth-$(d+1)$ AC^0 circuit with $O(n)$ wires, but F cannot be computed by any depth-d TC^0 circuit with $n^{1+\gamma}$ wires. This bound matches the current state-of-the-art lower bounds for computing explicit functions by threshold circuits of depth $d > 2$, which were previously known only for functions outside AC^0 such as the parity function. Furthermore, in our result, the AC^0 circuit computing F is a monotone *read-once formula* (i.e., an AND-OR tree), and the lower bound holds even in the average-case setting with respect to advantage $n^{-\gamma}$.

At a high level, our proof strategy combines two prominent approaches in circuit complexity from the last decade: The celebrated *random projections* method of Håstad, Rossman, Servedio, and Tan (J. ACM 2017), which was previously used to show a tight average-case depth hierarchy for AC^0; and the line of works analyzing the effect of *random restrictions* on threshold circuits. We show that under a modified version of Håstad, Rossman, Servedio, and Tan's projection procedure, any depth-d threshold circuit with $n^{1+\gamma}$ wires simplifies to a near-trivial function, whereas an appropriately parameterized AND-OR tree of depth $d+1$ maintains structure.

CCS CONCEPTS

• **Theory of computation → Circuit complexity**.

KEYWORDS

threshold circuits, AND-OR trees, circuit lower bounds, random projections

STOC '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 978-1-4503-9913-5/23/06.
https://doi.org/10.1145/3564246.3585216

ACM Reference Format:
Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. 2023. Depth-d Threshold Circuits vs. Depth-$(d+1)$ AND-OR Trees. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3564246.3585216

1 INTRODUCTION

The focus of this paper is *linear threshold circuits* (LTF circuits). These are non-uniform circuits in which each gate can compute an arbitrary *linear threshold function* (LTF), of the form

$$\Phi_{w,\theta}(x_1, \ldots, x_n) = 1 \iff \sum_{i \in [n]} w_i \cdot x_i \geq \theta \,,$$

where $w \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$ and the arithmetic is over the reals. We define the *size* of a circuit to be its number of wires.

Proving explicit lower bounds for LTF circuits is one of the most important current challenges in complexity theory. However, despite more than 50 years of research into this circuit class, the best lower bounds known are only for circuits of slightly super-linear size. Specifically, in the 1990s, Impagliazzo, Paturi, and Saks [40] showed that LTF circuits of depth d and size $n^{1+\gamma_d}$ (where $\gamma_d = 2^{-\Theta(d)}$) cannot compute the parity function. This was recently strengthened by Chen, Santhanam, and Srinivasan [26] to an average-case lower bound for circuits of the same size (up to the constant inside the Θ-notation in the expression for γ_d) computing the Generalized Andreev function. The latter work is part of an influential line of works in the last decade, which introduced new ways of analyzing LTFs and LTF circuits (see, e.g., [26, 29, 39, 58, 70]).[1]

Our main result in this paper is a stronger lower bound, where the improvement is that the bound holds for a function that is "even more explicit" than parity (in the sense that it has lower computational complexity). Specifically, we show that LTF circuits fail to compute a simple, read-once, AND-OR tree. In particular, such trees are monotone, read-once AC^0 formulas – arguably one of

[1] Complementing these two lower bounds, Chen and Tell [24] showed that to prove lower bounds against LTF circuits of polynomial size, it suffices to prove lower bounds for LTF circuits of depth d size $n^{1+\delta_d}$, where $\delta_d = 2^{-\Theta(d)}$ differs from γ_d only in the constant hidden inside the Θ-notation.

the weakest complexity classes that has been studied. Moreover, our main result asserts that LTF circuits of depth d and size $n^{1+2^{-\Theta(d)}}$ cannot even compute a read-once AND-OR tree *of depth* $d+1$; that is, the depth difference amounts to a single layer.[2] And lastly, our lower bound also holds on average, rather than just in the worst case.

Theorem 1.1 (LTF circuits cannot compute simple AND-OR trees). *Let $n \in \mathbb{N}$ be sufficiently large, let $d \leq \frac{1}{20} \cdot \log\log(n)$, and let $\gamma_d = 2^{-10 \cdot d}$. Then, there exists an explicit depth-$(d+1)$ read-once AC^0 formula $F = F^{(n)}_{d+1}$ on n input bits such that for every depth-d LTF circuit f with at most $n^{1+\gamma_d}$ wires,*

$$\Pr_{\mathbf{x} \in \{0,1\}^n}[f(\mathbf{x}) = F(\mathbf{x})] \leq \frac{1}{2} + n^{-\gamma_d}.$$

The key contribution underlying Theorem 1.1 is a new and more refined way of analyzing LTF circuits, which paves the way to proving our new lower bound, and which we hope may facilitate further progress in proving lower bounds for LTF circuits. At a high level, our proof strategy combines two main approaches in circuit complexity from the last decade that were separate so far: The celebrated *random projections* method of Håstad, Rossman, Servedio, and Tan [37], which was previously used to show a tight depth hierarchy for AC^0; and the line of works (mentioned above) that introduced new ways of analyzing LTF circuits, and that in particular analyzed the effect of *random restrictions* on LTF circuits.

Combining the two lines of work requires significant technical effort, in order to make them "fit together". From a bird's eye view, our paper analyzes the effect of a very specific (and suitably chosen) random projections procedure on LTF circuits. Replicating the analysis from Håstad, Rossman, Servedio, and Tan's work [37] with a different and careful parameterization, we show that the AND-OR tree F "maintains structure" under this projections procedure; and the crux of our technical contribution is in showing that this projections procedure trivializes every LTF circuit of depth d and size $n^{1+\gamma_d}$, with high probability. We refer the reader to Section 2 for a technical overview of the proof.

1.1 Impossibility of Depth-Reduction Using LTF Gates

Theorem 1.1 can also be viewed in the context of *circuit depth reduction*, which is the task of decreasing the depth of a circuit without significantly increasing its size (and without changing the function that it computes). There are classic, strong *lower bounds* regarding depth reduction of AC^0 circuits. Indeed, for a certain AC^0 circuit of depth $d+1$ and size $O(n)$, Håstad showed that every equivalent depth-d AC^0 circuit has size at least $2^{n^{\Omega(1/d)}}$ [34, Chapter 6], improving earlier work by Sipser [65] and Yao [78]. However, the situation changes if we allow the shallower circuit to use a stronger model. In this case, strong depth reduction *upper bounds* are known for AC^0 circuits and, much more generally, for ACC^0 circuits (i.e., AC^0 circuits augmented with MOD_m gates where m is constant). In particular, building on and improving a long line of work [3–5, 11, 71, 74, 80], Chen and Papakonstantinou showed that for every ACC^0 circuit of depth d and size w, there exists an equivalent $\mathsf{SYM} \circ \mathsf{AND}$ circuit of size $2^{(\log w)^{O(d)}}$ [27]. Depth reduction theorems along these lines have found applications in circuit analysis algorithms [7, 27, 74, 77], circuit lower bounds [21, 23, 25, 27, 51, 73–75, 77], and even graph algorithms [76].

If we insist on the shallower circuit having the specific form $\mathsf{SYM} \circ \mathsf{AND}$, then near-matching lower bounds are known. Indeed, there are constructions of constant-depth polynomial size ACC^0 circuits [55], and even AC^0 circuits [10], for which every equivalent $\mathsf{MAJ} \circ \mathsf{SYM} \circ \mathsf{AND}$ circuit must have super-polynomial size. But what happens if we allow the shallower circuit to use an even stronger model? Observe that a $\mathsf{SYM} \circ \mathsf{AND}$ circuit can be converted into an equivalent depth-three LTF circuit with only a polynomial increase in size, because every symmetric function can be computed by a depth-two polynomial-size LTF circuit. Thus, a special case of Chen and Papakonstantinou's result [27] is that for every constant-depth polynomial-size AC^0 circuit, there is an equivalent depth-three quasipolynomial-size LTF circuit. Indeed, this special case was proven already by Allender in the 1980s [3], who presented the theorem in basically this form. This raises the following question: If we start with an AC^0 circuit – the weakest model that we have discussed – and we wish to convert it to an equivalent LTF circuit – the strongest model that we have discussed – then is depth reduction possible without a significant size blowup?

Theorem 1.1 gives a strong negative answer to the foregoing question, showing that depth-reduction of AC^0 circuits to LTF circuits, even one that attempts to save only a single layer, is impossible without a super-linear increase in size. Thus, although one can achieve a massive depth reduction using LTF circuits with $2^{\mathrm{polylog}(n)}$ wires [3–5, 11, 27, 71, 74, 80], our lower bound asserts that using only $n^{1+c^{-d}}$ wires does not allow for any depth reduction at all. (In fact, as explained below in Section 1.3, we further prove that the complexity of such depth-reduction for our particular AND-OR tree F is *either* precisely super-linear, *or* it is super-polynomial.) Our theorem can thus be interpreted as saying that some functions have an intrinsic "depth complexity" that is robust to changes in the gate set (i.e., from AND/OR gates to LTF gates), at least in the near-linear size regime.

1.2 Hard Functions in Extremely Weak Complexity Classes

Another lens through which to view Theorem 1.1 is via the recent success in proving "super-explicit" lower bounds for circuit classes. Recall that in classical lower bounds (e.g., in [34, 35, 40, 56, 67] and in many other works) the hard function is typically computable in NC^1; the two most well-known examples are the parity function and Andreev's function. A long-standing question, dating back to [34, 66, 78], is whether one can prove lower bounds for functions that are "even more explicit", such as the AND-OR tree in Theorem 1.1.

Our work follows in the footsteps of several influential works in the last decade, which were able to prove lower bounds in which

[2] Needless to say, LTF circuits of depth d and super-linear size (or even linear size) can compute read-once AND-OR trees of depth d or less, and thus to get a lower bound it is necessary for the tree to have at least one additional layer. Our result shows that one additional layer is also sufficient.

the hard function is computable in uniform AC^0.[3] Among these works is the celebrated average-case depth-hierarchy theorem for AC^0 by Håstad, Rossman, Servedio, and Tan [37] mentioned above (which improved several earlier works [34, 53, 66, 72, 78]); results asserting that $\mathsf{AC}^0[\oplus]$ circuits of depth d and size s fail to compute a function computable by uniform AC^0 formulas of depth $d+1$ and linear size,[4] and a function computable by uniform AC^0 formulas of depth d and size $\mathrm{poly}(s)$ [47, 48]; and a recent work by Filmus, Meir, and Tal [30], who showed a function in uniform AC^0 that cannot be computed by De Morgan formulas of sub-cubic size $n^{3-o(1)}$.

The works above "cover" the most widely-studied classes in circuit complexity, the main exception being LTF circuits (i.e., the class TC^0), which are the focus of the current work. We stress that many prior works have shown that AC^0 is hard for various *subclasses* of LTF circuits, which have particular structural restrictions (such as LTF $\circ$ PARITY circuits or MAJ $\circ$ LTF circuits or monotone circuits; see [10, 15–20, 31, 36, 44, 45, 50, 52, 57, 60–64, 79]).[5] However, Theorem 1.1 is the first result showing that AC^0 is hard for LTF circuits of any constant depth (and super-linear size), and without any particular structural restrictions.

As mentioned above, our techniques use Håstad, Rossman, Servedio, and Tan's work on AC^0 [37] as a starting point (and the crux of our technical contribution is in analyzing the effect of a procedure similar to theirs on LTF circuits). However, our techniques are completely different than the techniques used in the works [8, 47, 48, 53] on $\mathsf{AC}^0[\oplus]$ (the latter works build on the line of research on the "coin problem" [1, 2, 12–14, 28, 32, 46–48, 59, 69]), and also completely different than the techniques in [30] on De Morgan formulas.

1.3 Tightness of Our Result

When d is constant, the correlation bound $n^{-\gamma}$ in Theorem 1.1 cannot be significantly improved. The reason is that our hard function is so computationally simple that it can be approximated, to a reasonable extent (i.e., almost matching Theorem 1.1), by shallower circuits:

- Every depth-$(d+1)$ AC^0 circuit with $O(n)$ wires and top fan-in m can be approximated, with success probability $1/2 + \Omega(1/m)$, by a depth-d AC^0 circuit with $O(n)$ wires. (This follows from the discriminator lemma [33].)
- Every monotone function can be approximated, with success probability $1/2 + \Omega((\log n)/n)$, by a constant or a variable. (This follows from the Kahn-Kalai-Linial theorem [43].)
- Every AC^0 circuit (of any constant depth and any polynomial size) can be approximated, with success probability $1/2 + 2^{-\mathrm{polylog}(n)}$, by a depth-1 AC^0 circuit with $\mathrm{polylog}(n)$ wires, i.e., either a conjunction of $\mathrm{polylog}(n)$ literals or else a disjunction of $\mathrm{polylog}(n)$ literals. (This follows from the Linial-Mansour-Nisan theorem [49].)

For completeness, we include proofs of the three preceding bounds in the full version of this paper [38, Appendix A]. It is an interesting problem to close the remaining quantitative gaps between our correlation bound and the three preceding bounds, especially the last one.

In contrast, if we focus on exact (worst-case) simulations, then it is unclear whether the *size bound* $n^{1+\gamma}$ in our theorem is tight. We partially address this question by proving a "hardness magnification" result. Recall that such results assert that specific, seemingly-minor improvements to known lower bounds would imply dramatic, strong lower bounds for powerful models of computation. (An intensive recent interest in such results was sparked by the work of Oliveira and Santhanam [54], who coined the term, following older results such as those by Srinivasan [68] and Allender and Koucký [6].) Regarding TC^0, prior work shows that tiny improvements in the known lower bounds for certain NC^1-complete functions or for MCSP would yield lower bounds for LTF circuits of arbitrarily large polynomial size [6, 22, 24, 39]. In the same spirit, we show that even a very small improvement to the size bound in Theorem 1.1 would imply that our AND-OR tree is hard for LTF circuits of arbitrarily large polynomial size. (Recall that $F^{(n)}_{d+1}$ denotes the depth-$(d+1)$ AND-OR tree from Theorem 1.1.)

Theorem 1.2 (hardness magnification for our construction). *Let $d_0 \in \mathbb{N}$ and $k > 1$ be constants. Suppose that for all sufficiently large n, the function $F^{(n)}_{d_0+1}$ can be computed by depth-d_0 LTF circuits with n^k wires. Then for all sufficiently large n and all $2d_0 \le d \le \frac{1}{20}\log\log n$, the function $F^{(n)}_{d+1}$ can be computed by depth-d LTF circuits with $\tilde{O}(n^{1+k\cdot 10^{-d}})$ wires.*

In particular, suppose that for every sufficiently large constant d and sufficiently large n, the function $F^{(n)}_{d+1}$ cannot be computed by depth-d LTF circuits with $n^{1+2^{-3\cdot d}}$ wires (slightly improving the $n^{1+2^{-10\cdot d}}$ bound from Theorem 1.1). Under that assumption, Theorem 1.2 implies that $F^{(n)}_{d+1}$ cannot even be computed by depth-d LTF circuits with any $\mathrm{poly}(n)$ number of wires. The proof of Theorem 1.2 is simple and relies on the recursive structure of $F^{(n)}_{d+1}$. See the full version of this paper for details [38, Section 9].

The optimal size complexity of depth-d LTF circuits computing our depth-$(d+1)$ AND-OR tree $F^{(n)}_{d+1}$ remains unclear. However, taken together, our results and prior depth-reduction theorems [3–5, 11, 27, 71, 74, 80] essentially narrow down the optimal size complexity to *two* relatively small intervals. Either the optimal size complexity is $n^{1+2^{-\Theta(d)}}$ (for all sufficiently large n and d with $d \le \frac{1}{20}\log\log n$), or else the optimal size complexity is between $n^{\omega(1)}$ and $n^{\mathrm{polylog}(n)}$ (for each constant $d \ge 4$ and infinitely many n).

2 TECHNICAL OVERVIEW

We prove our result using the method of *random projections*, which is a generalization of the traditional method of random restrictions that (to the best of our knowledge) was first used by Impagliazzo and Segerlind [41]. A projection maps each variable either to a constant (0 or 1) or else to another variable. The key feature distinguishing projections from traditional restrictions is that a projection might "merge" multiple variables by mapping them to a single variable, thereby keeping the variables alive but "tying them together."

[3] Of course, to show that uniform AC^0 is hard for circuits from a certain class C, one needs to give the AC^0 circuit some advantage over C, such as more depth or size (as in Theorem 1.1, in which the AND-OR tree has depth $d+1$).
[4] As explained by Limaye, Sreenivasaiah, Srinivasan, Tripathi, and Venkitesh [47], this follows from works of O'Donnell and Wimmer [53] and Amano [8].
[5] In fact, the works mentioned here managed to prove *super-polynomial* lower bounds for these subclasses.

For this technical overview, let us focus on the problem of proving a *worst-case* separation between depth-d LTF circuits and depth-$(d+1)$ AND-OR trees, and let us focus on the case that d is constant. Such a separation follows from the following theorem.

Theorem 2.1. *Let $d \in \mathbb{N}$ be a constant, let $\gamma = 2^{-10 \cdot d}$, and let $n \in \mathbb{N}$ be sufficiently large. There exists an explicit depth-$(d+1)$ read-once AC^0 formula $F = F^{(n)}_{d+1}$ on n input bits, a random projection $\boldsymbol{\pi}$, and a distribution $\boldsymbol{\sigma}$ such that:*

(1) *(Survival of the* AND-OR *tree) With probability $1 - o(1)$ over $\pi \sim \boldsymbol{\pi}$, the projected function $F\restriction_\pi$ is $o(1)$-unbiased under $\boldsymbol{\sigma}$, i.e.,*
$$\left| \Pr_{\boldsymbol{\sigma}} [F\restriction_\pi(\boldsymbol{\sigma}) = 1] - \frac{1}{2} \right| = o(1).$$
(2) *(Simplification of any* LTF *circuit) For any depth-d* LTF *circuit f on n input bits with at most $n^{1+\gamma}$ wires, with probability $1 - o(1)$ over $\pi \sim \boldsymbol{\pi}$, the projected function $f\restriction_\pi$ is $o(1)$-close to a constant under $\boldsymbol{\sigma}$, i.e., there is some $b \in \{0,1\}$ such that*
$$\Pr_{\boldsymbol{\sigma}} [f\restriction_\pi(\boldsymbol{\sigma}) = b] = 1 - o(1).$$

The distribution $\boldsymbol{\sigma}$ is simply an i.i.d. product distribution (with biased marginals). Furthermore, projecting according to $\boldsymbol{\pi}$ and then assigning values according to $\boldsymbol{\sigma}$ yields the uniform distribution over $\{0,1\}^n$, which is why we actually get an *average-case* separation.

Both our hard function F and our projection $\boldsymbol{\pi}$ are based on the work of Håstad, Rossman, Servedio, and Tan [37]. We do modify the parameters, but still, the fact that the hard function F survives the projection (Item 1 above) follows from a fairly straightforward generalization of their analysis. The main challenge is showing that LTF circuits simplify under the specific random projection procedure $\boldsymbol{\pi}$ (i.e., proving Item 2).

Random Projections and LTF *Circuits.* The last couple of decades have seen the development of what is often referred to as the *structural theory of linear threshold functions*, which can be viewed as a special case of the "structure vs. randomness" paradigm. One of the main applications of this body of knowledge is the analysis of the effect of *random and pseudorandom restrictions* on LTF circuits of depth d and size at most $n^{1+2^{-O(d)}}$ [26, 29, 39, 58, 70].

The main technical contribution of our work is showing that this body of knowledge can be extended so that it works in an inherently different setting, namely the setting of random projections as discussed above. The projections that we analyze are quite different than traditional random restrictions: Not only are variables sometimes merged, but also the assigned values are heavily biased, and the values assigned to them have significant correlations. Indeed, these projections cannot even be considered "pseudorandom," but we nevertheless show that LTF circuits of depth d simplify under these projections. The underlying technical challenges require extending and refining techniques used in previous works in the last decade.

In Section 2.1, we describe the hard function F, the projection $\boldsymbol{\pi}$, and the distribution $\boldsymbol{\sigma}$ in more detail (see the full version of this paper for the full definitions [38, Section 4]), and we briefly explain why F survives the projection (following the analysis in [37]). Then, in Sections 2.2–2.4, we discuss the main part of the proof, which shows that LTF circuits simplify under $\boldsymbol{\pi}$.

2.1 Setup and High-Level Plan

Let us describe our random projection procedure, which is a very careful modification of the one in [37]. One of our contributions is to devise an alternative way of thinking about the projection procedure that hides some complexity, enabling us to analyze its effect on LTF circuits. This abstraction might be useful in other contexts as well. While the new perspective is enough to carry out most of our analysis of LTF circuits, we do still rely on the original perspective for some parts of this analysis. (Readers who are familiar with the work of Håstad *et al.* [37] are encouraged to refer to the full version of this paper, in which we prove the formal connection between the two perspectives [38, Section 6.1].) We now give an overview of the procedure from the new perspective.

The Random Projections. We denote by $\mathbf{b}_{1-\beta}$ a Bernoulli RV that gets value 1 with probability $1-\beta$. The projection procedure works in d iterations. For the first iteration, consider the gates of F just above the inputs, denoted $g_1, \ldots, g_t$, which are all AND gates. We partition the variables into disjoint blocks $B_1, \ldots, B_t$, where B_j consists of the variables that feed into the gate g_j. Then, in each block B_j independently, for suitable parameters $p_1, \beta_1 > 0$:

(1) With probability p_1 the block *survives*, in which case a random subset of its variables of density $\approx \beta_1$ is kept alive,[6] and all the other variables in the block are assigned the value 1 (recall that the g_j is an AND gate).
(2) If the block does not survive, the variables are assigned values from $\mathbf{b}_{1-\beta_1}$ independently, *except* that the probability that all variables are assigned 1 is artificially decreased.

Note that the expected fraction of living variables in such an iteration is approximately $\bar{p}_1 = p_1 \cdot \beta_1$. In the end of the iteration, we merge the living variables in each surviving block; that is, we project these variables to a single new variable. We consider this new variable to be a "level-1 variable" whereas the original input variables are level-0 variables. We denote the end-result of this assignment and merging (i.e., projection) process by $\boldsymbol{\pi}^{(1)}$, and we refer to any projection with the above structure as a *corrupted biased block projection*. (See the full version of this paper [38] for the precise definition.)

After applying $\boldsymbol{\pi}^{(1)}$, we can identify each AND gate g_j either with a constant (in case a variable in B_j was assigned 0, or all variables in B_j were assigned 1), or with the living variable corresponding to the "merged" variables in B_j (in case some variables in B_j were left alive, and all the others were assigned 1). We thus recurse into the next iteration with a circuit of smaller depth on the level-1 variables. However, in our setting (and unlike [37]), subsequent iterations of the projection, denoted $\boldsymbol{\pi}^{(2)}, \ldots, \boldsymbol{\pi}^{(d)}$, will be parameterized by *different values* of $p_2, \ldots, p_d$ and $\beta_2, \ldots, \beta_d$. The projection $\boldsymbol{\pi}^{(i)}$ maps level-$(i-1)$ variables to level-i variables (or constants). The projection $\boldsymbol{\pi}$ referred to in Theorem 2.1 applies $\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(d)}$ successively, thereby mapping level-0 variables to level-d variables or constants.

Loosely speaking, our goal is to prove that each iteration of the procedure above reduces the depth of F by exactly one layer, and simultaneously reduces the depth of any LTF circuit by at least one

[6] Specifically, we include each element with probability β_1 and condition on getting a nonempty set.

layer; thus, after d iterations, F maintains structure whereas any LTF circuit of depth d trivializes. As articulated in Theorem 2.1, our notions of "maintaining structure" and "trivialization" are defined with respect to a suitable distribution $\boldsymbol{\sigma}$. Similarly, at each intermediate stage, we carry out our analysis with respect to a corresponding intermediate distribution $\boldsymbol{\sigma}^{(i)}$ over the level-$(i-1)$ variables. This distribution is also an i.i.d. product distribution with suitably biased marginals (the bias differs from one i to the next).

Specifying the Parameters. Let us now motivate our choice of parameters, informed by prior work on LTF circuits. In [37], the fraction $\bar{p}_i$ of living variables is essentially the same from one iteration to the next: $\bar{p}_1 \approx \bar{p}_2 \approx \cdots \approx \bar{p}_d$. In contrast, previous restriction procedures for LTF circuits apply d rounds of restrictions where the fraction of living variables decreases from one iteration to the next: $\bar{p}_{i+1} = (\bar{p}_i)^C$ for a large constant $C > 1$. We follow the latter line of works, and adapt the AND-OR tree and the random projections above to use corresponding parameters. Accounting for the required changes to maintain the properties above, the resulting AND-OR tree is such that the fan-ins of gates in the tree grow rapidly as we go up the layers.

Being more specific, recall that we are assuming that the depth $d+1$ is constant for simplicity. For a parameter M (where $M \approx n^{\epsilon}$ for a small constant $\epsilon > 0$), we define a sequence of parameters $M_1 = M$ and $M_{i+1} = M_i^{100}$ for $i = 1, \ldots, d-1$. We choose the fan-ins in the AND-OR tree such that under a uniform random input, for each gate g at distance $i \le d$ from the inputs, the subformula rooted at g has acceptance probability roughly $1/M_i$ if i is even or $1 - 1/M_i$ if i is odd, and overall, the AND-OR tree has acceptance probability roughly $1/2$. In more detail, we set f_i, p_i, β_i, and $\boldsymbol{\pi}^{(i)}$ as described in Figure 1. For the precise values, see the full version of this paper [38, Sections 4 and 6.1]. For intuition, we remark that our AND-OR tree corresponds to a d-fold composition that alternates between the well-known *tribes function* (a read-once DNF) and its Boolean dual (a read-once CNF), with widths approximately $\log(M_1), \ldots, \log(M_d)$. The tribes function and its dual are approximately balanced, so the composition is also approximately balanced. The top fan-ins of these CNFs and DNFs are approximately $M_1 \cdot \ln(2), \ldots, M_d \cdot \ln(2)$, and hence after merging adjacent layers when possible, we get a depth-$(d+1)$ tree with the fan-ins as described above.

Observe that for $i \ge 2$, the values that $\boldsymbol{\pi}^{(i)}$ assigns to fixed variables are highly biased, alternately toward 1 or toward 0.

As mentioned above, the proof that F survives the projection procedure simply generalizes the analysis in Håstad, Rossman, Servedio, and Tan's work [37] to our different parameter setting. Intuitively, the "blockwise" correlations that are present in the projections $\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(d)}$ (e.g., recall that when $\boldsymbol{\pi}^{(1)}$ keeps a variable alive, it assigns 1 to all of the non-surviving variables in that block) are tailored to the AND-OR-tree and designed to keep it alive. See the full version of this paper for details [38, Section 5]. The innovative part in our argument is showing that LTF circuits simplify under the projections.

LTF *Circuits Simplify under Projections: The High-Level Plan.* Our argument has a high-level structure similar to the ones in prior work [26, 39, 70], but instead of arguing about the effects of traditional random restrictions, we now argue about the effects of each random projection $\boldsymbol{\pi}^{(i)}$ above (for any fixed $i \in [d]$). For simplicity, we assume from now on that i is odd, in which case the assigned values of $\boldsymbol{\pi}^{(i)}$ are biased toward 1 and the corresponding gate in F is an AND gate.

Similarly to the analysis in [39] (also implicit in the work of Chen, Santhanam, and Srinivasan [26]), we will work with *hybrid computational models*. That is, on the way to proving that any LTF circuit f becomes (close to) a constant, we argue that after intermediate projections $\boldsymbol{\pi}^{(i)}$ the circuit can be computed by a computational model that combines decision trees and LTF circuits; specifically, the tree queries variables to reach a leaf, and the leaf is labeled by an LTF circuit that is then applied to the input.[7]

Our proof has three main steps:

(1) As a first step, we prove that applying $\boldsymbol{\pi}^{(i)}$ to any LTF (i.e., any single gate in f) makes the LTF extremely close to a constant, with somewhat high probability. This probability is high, but not enough to allow a union bound on all gates. We will elaborate in Section 2.2.
(2) Our second step is to argue that applying $\boldsymbol{\pi}^{(i)}$ to any LTF circuit of depth $d+1-i$ with $n^{1+2^{-O(d)}}$ wires simplifies the circuit, with somewhat high probability, to be very close to a decision tree with LTF circuits of depth $d-i$ at its leaves, where the depth of the tree is significantly smaller than the number of living variables under $\boldsymbol{\pi}^{(i)}$. We will elaborate in Section 2.3.
(3) The final step is to show that applying $\boldsymbol{\pi}^{(i)}$ to a decision tree with LTF circuits of depth $d+1-i$ at its leaves simplifies the tree, with high probability, such that it is close to a shallower decision tree in which the leaves are labeled by depth $d-i$ LTF circuits. We will elaborate in Section 2.4.

We stress that in all the statements above, the notion of "approximating a function" (i.e., when saying that a restricted function is close to a simpler function) refers to the distribution $\boldsymbol{\sigma}^{(i+1)}$, rather than to the uniform distribution. Having proved the three steps above, the proof will analyze the applications of $\boldsymbol{\pi}^{(i)}$ for $i = 1, \ldots, d$, arguing at each iteration i that the circuit simplifies with respect to the "next" distribution $\boldsymbol{\sigma}^{(i+1)}$. In the last step the circuit will be a decision tree that queries only a sub-constant fraction of its variables. Such a tree cannot approximate the AND (or OR) function $F\restriction_{\boldsymbol{\pi}}$ under $\boldsymbol{\sigma}^{(d+1)}$. Indeed, with high probability over $\boldsymbol{\pi}$, the tree is close to a constant under $\boldsymbol{\sigma}^{(d+1)}$, whereas $F\restriction_{\boldsymbol{\pi}}$ is nearly balanced (because $\boldsymbol{\sigma}^{(d+1)}$ is biased).

2.2 Random Projections Simplify Any Single LTF

Chen, Santhanam, and Srinivasan showed that a random restriction that keeps a p-fraction of the variables alive simplifies any single LTF to be $\exp(-p^{-\Omega(1)})$-close to a constant, with probability at least $1 - p^{\Omega(1)}$ [26]. A motivating observation for our analysis is that a *biased restriction*, which keeps a p-fraction of variables alive and fixes the other variables independently by $\mathbf{b}_{1-\beta}$, *simplifies any unweighted threshold function* to be $\exp(-p^{-\Omega(1)})$-close to a constant with respect to any product distribution, with probability

[7] This hybrid model is simpler than the one considered in [39], since the latter also allowed queries to LTF gates at internal nodes of the tree.

$$
\begin{aligned}
f_i &= \text{fan-ins at distance } i \text{ from inputs} && \approx \begin{cases} \log(M_1) & i = 1 \\ M_{i-1} \cdot \ln(M_i) & 2 \le i \le d \\ M_d \cdot \ln(2) & i = d+1 \end{cases} \\
p_i &= \text{probability that a block survives under } \boldsymbol{\pi}^{(i)} && \approx 1/\sqrt{M_i} \\
\beta_i &= \text{bias parameter of } \boldsymbol{\pi}^{(i)} \text{ (biased toward 1 if } i \text{ is odd, 0 if } i \text{ is even)} && = 1/\sqrt{M_{i-1}} \\
\boldsymbol{\sigma}^{(i)} &= \text{distribution over level-}(i-1) \text{ variables} && = \begin{cases} \mathbf{b}_{1-\beta_i} \text{ to each variable (i.i.d.)} & i \text{ odd} \\ \mathbf{b}_{\beta_i} \text{ to each variable (i.i.d.)} & i \text{ even.} \end{cases}
\end{aligned}
$$

Figure 1: The main parameters of our construction and analysis.

$1 - (p/\beta)^{\Omega(1)}$.[8] In other words, for any bias $\beta > 0$ of assignments to the fixed variables, the probability that an unweighted LTF fails to simplify is $(p/\beta)^{\Omega(1)}$, compared to the $p^{\Omega(1)}$ bound that Chen, Santhanam, and Srinivasan showed in their setting [26].

We extend this statement to hold for an *arbitrary* (weighted) LTF rather than just the unweighted LTFs; to hold when the values assigned to fixed variables are correlated (i.e., within each block, if the block survives then all fixed variables are simultaneously assigned the value 1, and otherwise the probability of the all-ones string is artificially decreased); and to hold also when considering a merging of the variables after applying the restriction. In this more challenging setting, we show a slightly worse error bound of $(p/\beta^2)^{\Omega(1)}$ compared to $(p/\beta)^{\Omega(1)}$:

Theorem 2.2 (LTFs simplify under corrupted biased block projections; informal). *Let Φ be an* LTF *on n variables, let $\boldsymbol{\pi}$ be a corrupted biased block projection with parameters p and β, and let $\boldsymbol{\sigma}$ be a product distribution (possibly with biased marginals). Assume that each block B_j satisfies $\epsilon/n \le (1-\beta)^{|B_j|} \le p$. Then the probability that $\Phi\restriction_{\boldsymbol{\pi}}$ is not ϵ-close to a constant under $\boldsymbol{\sigma}$ is*

$$O\left(\left(\frac{p}{\beta^2}\right)^{1/3} \cdot \log\left(\frac{n}{\epsilon}\right)\right).$$

Before explaining the ideas in the proof, let us comment on the subtlety of the parameters obtained in Theorem 2.2. First, in Theorem 2.2, we assume both upper and lower bounds on the quantity $(1-\beta)^{|B_j|}$. That is, we assume that the block size $|B_j|$ is neither too big nor too small. Fortunately, this "Goldilocks" condition is indeed satisfied by our projections $\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(d)}$ with high probability.[9] (It is also satisfied by Håstad, Rossman, Servedio, and Tan's projections [37].) Thus, Theorem 2.2 applies to $\boldsymbol{\pi}^{(i)}$ with failure probability $O((p_i/\beta_i^2)^{1/3} \cdot \log(n/\epsilon))$.

Secondly, the projections in Håstad, Rossman, Servedio, and Tan's original work [37] satisfy $p \approx \beta$. For such parameters, the bound of Theorem 2.2 would be useless. However, our modified projections have parameters p and β that vary from one iteration to the next, and crucially, the bias parameter β in each iteration is approximately equal to the block survival probability p in the *previous* iteration, i.e., $\beta_i \approx p_{i-1}$. A key property of our projections is that they are "increasingly aggressive" in the sense that $p_i \ll p_{i-1}$ (specifically $p_i \approx p_{i-1}^{100}$). Therefore, the bound of Theorem 2.2 is indeed small when we apply it to our projections.

To prove Theorem 2.2, we generalize the "structure vs. randomness" approach that Chen, Santhanam, and Srinivasan used to show that LTFs simplify under traditional random restrictions [26]. Loosely speaking, their proof first analyzes "regular" LTFs, i.e., LTFs in which the weights are reasonably well-spread (this is the "random" case). Under the assumption of regularity, they argue that the weighted sum of assigned values is *anti-concentrated*, and thus unlikely to land in the small interval that would cause the restricted LTF to be somewhat balanced. To complement this analysis, they analyze LTFs that have a small number of "heavy" variables (this is the "structure" case). If the number of heavy variables is sufficiently small, then it is possible to fix them and reduce to the regular case, and otherwise they show anti-concentration among these "heavy" variables alone.

To make this approach work in our setting, the main challenge is establishing anti-concentration in the regular case.[10] Recall that in the projection $\boldsymbol{\pi}^{(i)}$, after the set of living variables has been fixed, non-surviving variables in surviving blocks are always assigned the value 1; this deterministic assignment does not contribute any anti-concentration at all. In non-surviving blocks, the assignment is random, but the assigned values are not independent, because the probability of the all-ones assignment is artificially decreased. The effect of this "corruption" within a single block is limited (because the all-ones assignment would be rare even without corruption). However, there are many blocks, and the *overall* effect is statistically significant; nevertheless, our goal is to show anti-concentration despite these corruptions.

To prove anti-concentration of the weighted sum of the assigned values, we first observe that with high probability, the variables

[8] To see this, let Φ be an n-bit unweighted LTF, and let $1/n \le p \le 1/2$. Consider a random restriction $\boldsymbol{\rho}$ that keeps a random subset S of $p \cdot n$ variables alive and fixes the variables in $[n] \setminus S$ independently according to $\mathbf{b}_{1-\beta}$. Hoeffding's inequality implies that (with respect to any fixed product distribution) the function is ϵ-close to a constant only if the sum of values assigned to variables in $[n] \setminus S$ falls in an interval of length $O(\sqrt{\log(1/\epsilon) \cdot p \cdot n})$. By the Berry-Esseen theorem, the probability of this event is at most $O(\sqrt{\log(1/\epsilon) \cdot (p/\beta)})$. See the full version of this paper for further details [38].

[9] Actually, in the rare event that a block is an unacceptable size, our projection assigns values to all variables in that block from an i.i.d. product distribution independently of the other blocks, and we show that this does not affect the rest of the analysis. See the full version of this paper for details [38].

[10] By comparison, our analysis of non-regular LTFs is a relatively straightforward adaptation of techniques from prior work. Note that both the regular and the non-regular cases contribute to the final error bound in Theorem 2.2, and as discussed, that error bound forces us to use a careful choice of parameters in our construction.

in non-surviving blocks have a constant fraction of the total ℓ_2-weight. (By "ℓ_2-weight," we mean the sum of the squares of the weights.) We may therefore focus on such variables (and ignore the non-surviving variables in surviving blocks). To handle the corruptions in non-surviving blocks, we show that the weighted sum of assigned values to fixed variables in non-surviving blocks can be represented as the weighted sum of truly independent Bernoulli variables, plus an error term. The sum of independent variables is anti-concentrated by the Berry-Esseen theorem; see the full version of this paper for details [38, Lemma 6.9]. To handle the error term, we bound its variance. (This is where we use the assumption that the blocks are not too small, in which case the all-ones assignment would be rare even without corruption. See the full version of this paper [38, Lemma 6.7] for details.) We thereby show that the error term is *concentrated*, and therefore it does little harm to the anti-concentration property of the sum of independent variables.

The anti-concentration established by the preceding arguments must be compared to the ℓ_2-weight of the living variables. Here we face another potential pitfall: When variables are merged, their weights effectively add, which typically increases the ℓ_2-weight of the living variables (making the LTF more balanced). This potential pitfall is the reason that we assume that the blocks are not too big. The assumption in Theorem 2.2 implies that with high probability, the number of variables that are merged in each block is small – only $O(\log(n/\epsilon))$ – and therefore the detrimental effect of mergings is limited.

2.3 Random Projections Simplify LTF Circuits (If We Allow Some Queries)

The next step is to argue that for every LTF circuit f of slightly super-linear size, with high probability, $\boldsymbol{\pi}^{(i)}$ "simplifies" the entire bottom layer of f. Ideally, we are hoping that the gates in the bottom layer become close to constants. We cannot simply apply a union bound to claim that they are all close to constants simultaneously, because the failure probability in Theorem 2.2 might be too large. Instead, following prior work, we argue that *after querying a sub-linear number of the remaining variables*, each gate in the bottom layer is either close to a constant (over the distribution $\boldsymbol{\sigma}^{(i+1)}$) or has fan-in one. Thus, the projected circuit $f\restriction_{\boldsymbol{\pi}^{(i)}}$ can be approximated (over $\boldsymbol{\sigma}^{(i+1)}$) by a decision tree whose leaves are labeled by shallower LTF circuits.

Given appropriate techniques from prior work [9, 26, 39, 42, 70], this is the easiest part of our argument, because those techniques do not depend on the assignments to fixed variables, but rather only on concentration properties of the number of living variables inside certain sets. We include a brief explanation of the argument here for completeness; see the full version of this paper for details [38, Section 6.3].

We partition the gates in the bottom layer of f into "heavy gates" and "light gates" based on their fan-in. Most light gates have only one (or zero) living variable feeding into them after the projection, so they can be replaced with a wire (or eliminated), and we query the variables feeding into the remaining light gates (there are few such variables, because these gates are light). Most heavy gates become close to a constant by Theorem 2.2, and we query all the variables feeding into the remaining heavy gates. The total number of such queries is bounded because the total number of wires in the circuit is bounded. (The latter argument is carried out by a standard bucketing technique, looking at all gates with fan-in roughly 2^i for each i and using the simple observation that there can be at most $w/2^i$ such gates for each i.)

2.4 Random Projections Simplify Decision Trees with LTF Circuits at Their Leaves

The previous step yielded a decision tree T with LTF circuits at its leaves. The last key piece in our proof is arguing that each such decision tree simplifies, under $\boldsymbol{\pi}^{(i)}$, to a shallower decision tree with shallower LTF circuit at its leaves. (Indeed, we need the tree depth to decrease by a factor of $\approx \bar{p}_i$, and we need the circuits to decrease by one layer.)

First, we show that the tree depth indeed shrinks, with high probability, by a factor of $\approx \bar{p}_i$. This turns out to be not as straightforward as it might seem, due to correlations and mergings in $\boldsymbol{\pi}^{(i)}$; see the full version of this paper for details [38, Lemma 7.1]. Nonetheless, the more interesting part of the argument is arguing that we can use shallower LTF circuits at the leaves. The natural strategy to try and prove this is to claim that for each leaf, the corresponding circuit simplifies under $\boldsymbol{\pi}^{(i)}$ with high probability, and thus the fraction of "bad" leaves is small and we can replace those by constants, obtaining a tree with similar functionality.

The problem with this approach is the correlations between variables in the same block under the projection $\boldsymbol{\pi}^{(i)}$. At each *fixed* leaf, simplification occurs with high probability, but we must analyze the *random* leaf reached when we apply $\boldsymbol{\pi}^{(i)}$ to T and then plug in an input sampled from $\boldsymbol{\sigma}^{(i+1)}$. In particular, the leaf is determined in part by $\boldsymbol{\pi}^{(i)}$, and the event of reaching a particular leaf can be *correlated* with the event that simplification fails at that leaf. It is therefore not clear how to show that simplification occurs with high probability at the random leaf that we reach.[11]

Dealing with this issue is the most subtle part of our argument, and it involves a two-step approach.

Preprocessing the Tree. As a first step, we "preprocess" the tree T, transforming it into a new tree $\tilde{T}$. The new tree $\tilde{T}$ simulates T and in fact refines T in the following way: if T ever queries too many variables in a block, or if T ever queries a variable in some block and observes a 0 (a somewhat unlikely event as bits are biased towards 1), then $\tilde{T}$ queries all variables in that block. It turns out that these modifications are not too costly, in the sense that after applying the projection $\boldsymbol{\pi}^{(i)}$, the two projected trees $T\restriction_{\boldsymbol{\pi}^{(i)}}$ and $\tilde{T}\restriction_{\boldsymbol{\pi}^{(i)}}$ have similar query complexities. Briefly, this holds for the following two reasons:

- The event of querying too many variables in a single block can only happen so many times given T's depth bound, and most such blocks don't survive the projection, so these events only cause $\tilde{T}\restriction_{\boldsymbol{\pi}^{(i)}}$ to perform a few additional queries compared to $T\restriction_{\boldsymbol{\pi}^{(i)}}$.

[11] In previous work [39] a similar challenge occurred, since the path to each leaf contained LTFs. However, the challenge there was significantly easier, since the number of LTFs on a path was small and thus it was possible to easily upper bound their effect on the resulting distribution.

- If a variable x_j in block B is observed to be 0, we have two cases. If the block B is non-surviving under $\boldsymbol{\pi}^{(i)}$, then $\tilde{T}\restriction_{\boldsymbol{\pi}^{(i)}}$ does not need to query any variable in B, because they are all assigned. On the other hand, if the block B is surviving, then the individual variable $x_j \in B$ must survive in $\boldsymbol{\pi}^{(i)}$, because non-surviving variables in surviving blocks are assigned the value 1. Therefore, in this case, both $T\restriction_{\boldsymbol{\pi}^{(i)}}$ and $\tilde{T}\restriction_{\boldsymbol{\pi}^{(i)}}$ query the single "merged" variable corresponding to the entire block B. Thus, in either case, observing a 0 ultimately does not cause $\tilde{T}\restriction_{\boldsymbol{\pi}^{(i)}}$ to perform any additional queries compared to $T\restriction_{\boldsymbol{\pi}^{(i)}}$.

Conditional Analysis of Corrupted Biased Block Projections. For the second step, consider the process of applying $\boldsymbol{\pi}^{(i)}$ to $\tilde{T}$ and then plugging in an input sampled from $\boldsymbol{\sigma}^{(i+1)}$. We analyze the joint distribution of $\boldsymbol{\pi}^{(i)}$ and $\boldsymbol{\sigma}^{(i+1)}$ *conditioned* on the event of reaching a leaf ℓ. Because of the preprocessing step, we can make a "win-win" argument: for each block, either (a) the tree queries every single variable in the block, or (b) the tree only makes a few queries to the block and observes 1 each time. In case (a), we can assume without loss of generality that the circuit C_ℓ labeling the leaf ℓ ignores all variables in that block, hence we can ignore the block. In case (b), the constraints on the queries help us to bound the extent to which conditioning distorts the distributions of $\boldsymbol{\pi}^{(i)}$ and $\boldsymbol{\sigma}^{(i+1)}$.

For example, we show that the event we are conditioning on in case (b) is a high-probability event regardless of whether the block survives, and hence the conditioning has little effect on the block's survival probability. By analyzing our projection distribution in more detail, we show that instead of applying $\boldsymbol{\pi}^{(i)}$ to the circuit C_ℓ, plugging in an input sampled from $\boldsymbol{\sigma}^{(i+1)}$, and conditioning on the event of reaching ℓ, we can equivalently imagine applying *another corrupted biased block projection* $\tilde{\boldsymbol{\pi}}$, plugging in an input sampled from *another product distribution* $\tilde{\boldsymbol{\sigma}}$, and *not conditioning on anything*. The parameters of $\tilde{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\sigma}}$ are slightly different than the parameters of $\boldsymbol{\pi}^{(i)}$ and $\boldsymbol{\sigma}^{(i+1)}$, but our analysis of a single circuit is sufficiently robust against these small distortions to conclude that T simplifies with high probability.

2.5 Putting It All Together

To summarize our discussion so far, we show that when we apply the random projection $\boldsymbol{\pi}^{(i)}$ to a decision tree $\mathbf{T}_{i-1}$ with LTF circuits at its leaves, we get another decision tree $\mathbf{T}_i$ with LTF circuits at its leaves that is "simpler" in the sense that the circuit-depth decreases by 1. The tree $\mathbf{T}_i$ agrees with the projected function $\mathbf{T}_{i-1}\restriction_{\boldsymbol{\pi}^{(i)}}$ with high probability under the product distribution $\boldsymbol{\sigma}^{(i+1)}$. To finish the proof, we need to apply some type of triangle inequality. For example, we know that $\mathbf{T}_1 \approx \mathbf{T}_0\restriction_{\boldsymbol{\pi}^{(1)}}$ and $\mathbf{T}_2 \approx \mathbf{T}_1\restriction_{\boldsymbol{\pi}^{(2)}}$; we want to conclude that $\mathbf{T}_2 \approx \mathbf{T}_0\restriction_{\boldsymbol{\pi}^{(2)}\circ\boldsymbol{\pi}^{(1)}}$.

We are indeed able to show that $\mathbf{T}_d \approx \mathbf{T}_0\restriction_{\boldsymbol{\pi}^{(d)}\circ\cdots\circ\boldsymbol{\pi}^{(1)}}$ by relying upon a crucial feature of the projections $\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(d)}$ and the product distributions $\boldsymbol{\sigma}^{(1)}, \ldots, \boldsymbol{\sigma}^{(d+1)}$. These projections and product distributions are *compatible* with each other, in the sense that applying $\boldsymbol{\pi}^{(i)}$ and then assigning values sampled from $\boldsymbol{\sigma}^{(i+1)}$ yields exactly the distribution $\boldsymbol{\sigma}^{(i)}$ [38, Lemma 5.2].

The same feature (the "completion property") is also crucial in the work of Håstad, Rossman, Servedio, and Tan [37]. However, the completion property plays a different role in their work than it does in ours. Their work is focused on *average-case* lower bounds; they rely on the fact that applying $\boldsymbol{\pi}^{(d)} \circ \cdots \circ \boldsymbol{\pi}^{(1)}$ and then assigning values sampled from $\boldsymbol{\sigma}^{(d+1)}$ yields the uniform distribution over inputs. The completion property is likewise an essential ingredient of our average-case separation, but the distinction is that in our setting, the completion property would still be crucial even if we were merely aiming for a *worst-case* separation. After all, the simplification we achieve at intermediate stages of our argument is itself only approximate, forcing us to use techniques designed for average-case separations.

The completion property holds trivially in the traditional setting of truly random restrictions, because the values assigned by the restriction are themselves independent and uniform. In both our work and the work of Håstad *et al.* [37], there are correlations between the values assigned to different variables, which are essential for ensuring that the AND-OR tree F survives. In both works, the *purpose of merging variables* (i.e., the purpose of using projections rather than restrictions) is to achieve the completion property despite these correlations.

ACKNOWLEDGMENTS

P.H. is supported by NSF grant CCF-1947546. Part of this work was done while W.M.H. was visiting the Simons Institute for the Theory of Computing, and part of this work was done while W.M.H. was a graduate student at the University of Texas at Austin, supported by the NSF GRFP under Grant DGE-1610403 and by a Harrington Fellowship from the University of Texas at Austin. A.T. is supported by NSF CAREER award CCF-2145474 and by a Sloan Fellowship from the Sloan Foundation. R.T. is partially supported by the National Science Foundation under grant number CCF-1445755 and under grant number CCF-1900460. Part of this work was conducted while R.T. was at DIMACS, and part of this work was conducted while R.T. was a fellow at the Simons Institute for the Theory of Computing.

REFERENCES

[1] Scott Aaronson. 2010. BQP and the polynomial hierarchy. In *Proc. 42nd Annual ACM Symposium on Theory of Computing (STOC)*. ACM, New York, 141–150. https://doi.org/10.1145/1806689.1806711

[2] Rohit Agrawal. 2020. Coin theorems and the Fourier expansion. *Chicago Journal of Theoretical Computer Science* (2020), Art. 4, 15. https://doi.org/10.4086/cjtcs.2020.004

[3] Eric Allender. 1989. A note on the power of threshold circuits. In *Proc. 30th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 580–584. https://doi.org/10.1109/SFCS.1989.63538

[4] Eric Allender and Vivek Gore. 1994. A Uniform Circuit Lower Bound for the Permanent. *SIAM J. Comput.* 23, 5 (1994), 1026–1049. https://doi.org/10.1137/S0097539792233907

[5] Eric Allender and Ulrich Hertrampf. 1994. Depth reduction for circuits of unbounded fan-in. *Inform. and Comput.* 112, 2 (1994), 217–238. https://doi.org/10.1006/inco.1994.1057

[6] Eric Allender and Michal Koucký. 2010. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM* 57, 3 (2010), 14, 36. https://doi.org/10.1145/1706591.1706594

[7] Josh Alman, Timothy M. Chan, and Ryan Williams. 2016. Polynomial representations of threshold functions and algorithmic applications. In *Proc. 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 467–476. https://doi.org/10.1109/FOCS.2016.57

[8] Kazuyuki Amano. 2009. Bounds on the size of small depth circuits for approximating majority. In *Proc. 36th International Colloquium on Automata, Languages*

and Programming (ICALP). 59–70. https://doi.org/10.1007/978-3-642-02927-1_7

[9] Swapnam Bajpai, Vaibhav Krishan, Deepanshu Kush, Nutan Limaye, and Srikanth Srinivasan. 2022. A #SAT algorithm for small constant-depth circuits with PTF gates. *Algorithmica* 84, 4 (2022), 1132–1162. https://doi.org/10.1007/s00453-021-00915-7

[10] Paul Beame and Trinh Huynh. 2012. Multiparty Communication Complexity and Threshold Circuit Size of AC^0. *SIAM Journal of Computing* 41, 3 (2012), 484–518. https://doi.org/10.1137/100792779

[11] Richard Beigel and Jun Tarui. 1994. On ACC. *Computational Complexity* 4, 4 (1994), 350–366. https://doi.org/10.1007/BF01263423

[12] Mark Braverman, Sumegha Garg, and David P. Woodruff. 2020. The coin problem with applications to data streams. In *Proc. 61st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Soc., Los Alamitos, CA, 318–329. https://doi.org/10.1109/FOCS46700.2020.00038

[13] Mark Braverman, Sumegha Garg, and Or Zamir. 2022. Tight space complexity of the coin problem. In *Proc. 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Soc., Los Alamitos, CA, 1068–1079. https://doi.org/10.1109/FOCS52979.2021.00106

[14] Joshua Brody and Elad Verbin. 2010. The coin problem, and pseudorandomness for branching programs. In *Proc. 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 30–39. https://doi.org/10.1109/FOCS.2010.10

[15] Jehoshua Bruck and Roman Smolensky. 1992. Polynomial threshold functions, AC^0 functions, and spectral norms. *SIAM J. Comput.* 21, 1 (1992), 33–42. https://doi.org/10.1137/0221003

[16] Harry Buhrman, Nikolay Vereshchagin, and Ronald de Wolf. 2007. On Computation and Communication with Small Bias. In *Proc. 22nd Annual IEEE Conference on Computational Complexity (CCC)*. 24–32. https://doi.org/10.1109/CCC.2007.18

[17] Mark Bun and Justin Thaler. 2015. Hardness amplification and the approximate degree of constant-depth circuits. In *Proc. 42nd International Colloquium on Automata, Languages and Programming (ICALP)*. Lecture Notes in Comput. Sci., Vol. 9134. Springer, Heidelberg, 268–280. https://doi.org/10.1007/978-3-662-47672-7_22

[18] Mark Bun and Justin Thaler. 2016. Improved bounds on the sign-rank of AC^0. In *Proc. 43rd International Colloquium on Automata, Languages and Programming (ICALP)*. LIPIcs. Leibniz Int. Proc. Inform., Vol. 55. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, Art. No. 37, 14. https://doi.org/10.4230/LIPIcs.ICALP.2016.37

[19] Mark Bun and Justin Thaler. 2021. The large-error approximate degree of AC^0. *Theory of Computing* 17 (2021), Paper No. 7, 46. https://doi.org/10.4086/toc.2021.v017a007

[20] Arkadev Chattopadhyay. 2007. Discrepancy and the Power of Bottom Fan-in in Depth-three Circuits. In *Proc. 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 449–458. https://doi.org/10.1109/FOCS.2007.30

[21] Lijie Chen. 2019. Non-deterministic Quasi-Polynomial Time is Average-case Hard for ACC Circuits. In *Proc. 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. https://doi.org/10.1109/FOCS.2019.00079

[22] Lijie Chen, Ce Jin, and Richard Ryan Williams. 2020. Sharp threshold results for computational complexity. In *Proc. 52nd Annual ACM Symposium on Theory of Computing (STOC)*. 1335–1348. https://doi.org/10.1145/3357713.3384283

[23] Lijie Chen and Hanlin Ren. 2022. Strong Average-Case Circuit Lower Bounds from Nontrivial Derandomization. *SIAM J. Comput.* 51, 3 (2022), STOC20–115–STOC20–173. https://doi.org/10.1137/20M1364886

[24] Lijie Chen and Roei Tell. 2019. Bootstrapping results for threshold circuits "just beyond" known lower bounds. In *Proc. 51st Annual ACM Symposium on Theory of Computing (STOC)*. 34–41. https://doi.org/10.1145/3313276.3316333

[25] Ruiwen Chen, Igor C. Oliveira, and Rahul Santhanam. 2018. An average-case lower bound against ACC^0. In *LATIN 2018: Theoretical informatics*. Lecture Notes in Comput. Sci., Vol. 10807. Springer, Cham, 317–330. https://doi.org/10.1007/978-3-319-77404-6_24

[26] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. 2018. Average-case lower bounds and satisfiability algorithms for small threshold circuits. *Theory of Computing* 14 (2018), Paper No. 9, 55. https://doi.org/10.4086/toc.2018.v014a009

[27] Shiteng Chen and Periklis A. Papakonstantinou. 2019. Depth reduction for composites. *SIAM J. Comput.* 48, 2 (2019), 668–686. https://doi.org/10.1137/17M1129672

[28] Gil Cohen, Anat Ganor, and Ran Raz. 2014. Two sides of the coin problem. In *Proc. 18th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM) (LIPIcs. Leibniz Int. Proc. Inform., Vol. 28)*. 618–629. https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2014.618

[29] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco A. Servedio, and Emanuele Viola. 2010. Bounded independence fools halfspaces. *SIAM Journal of Computing* 39, 8 (2010), 3441–3462. https://doi.org/10.1137/100783030

[30] Yuval Filmus, Or Meir, and Avishay Tal. 2021. Shrinkage under random projections, and cubic formula lower bounds for AC0. In *Proc. 12th Conference on Innovations in Theoretical Computer Science (ITCS)*, Vol. 185. Art. No. 89, 7. https://doi.org/10.4230/LIPIcs.ITCS.2021.89

[31] Jürgen Forster, Matthias Krause, Satyanarayana V. Lokam, Rustam Mubarakzjanov, Niels Schmitt, and Hans Ulrich Simon. 2001. Relations Between Communication Complexity, Linear Arrangements, and Computational Complexity. In *Proc. 21st Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*. 171–182. https://doi.org/10.1007/3-540-45294-X_15

[32] Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. 2019. $AC^0[p]$ lower bounds against MCSP via the coin problem. In *Proc. 46th International Colloquium on Automata, Languages and Programming (ICALP) (LIPIcs. Leibniz Int. Proc. Inform., Vol. 132)*. Art. No. 66, 15. https://doi.org/10.4230/LIPIcs.ICALP.2019.66

[33] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. 1993. Threshold Circuits of Bounded Depth. *Journal of Computer and System Sciences* 46, 2 (1993), 129–154. https://doi.org/10.1016/0022-0000(93)90001-D

[34] Johan Håstad. 1987. *Computational Limitations for Small-Depth Circuits*. MIT Press.

[35] Johan Håstad. 1998. The shrinkage exponent of De Morgan formulas is 2. *SIAM J. Comput.* 27, 1 (1998), 48–64. https://doi.org/10.1137/S0097539794261556

[36] Johan Håstad and Mikael Goldmann. 1991. On the power of small-depth threshold circuits. *Computational Complexity* 1, 2 (1991), 113–129. https://doi.org/10.1007/BF01272517

[37] Johan Håstad, Benjamin Rossman, Rocco A Servedio, and Li-Yang Tan. 2017. An average-case depth hierarchy theorem for boolean circuits. *Journal of the ACM (JACM)* 64, 5 (2017), 1–27. https://doi.org/10.1145/3095799

[38] Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. 2022. Depth-d Threshold Circuits vs. Depth-$(d+1)$ AND-OR Trees. *Electronic Colloquium on Computational Complexity: ECCC* 87 (2022), 71. https://eccc.weizmann.ac.il/report/2022/087/

[39] Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. 2022. Fooling constant-depth threshold circuits. In *Proc. 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 104–115. https://doi.org/10.1109/FOCS52979.2021.00019

[40] Russell Impagliazzo, Ramamohan Paturi, and Michael E. Saks. 1997. Size-depth tradeoffs for threshold circuits. *SIAM Journal of Computing* 26, 3 (1997), 693–707. https://doi.org/10.1137/S0097539792282965

[41] Russell Impagliazzo and Nathan Segerlind. 2001. Counting axioms do not polynomially simulate counting gates. In *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. https://doi.org/10.1109/SFCS.2001.959894

[42] Valentine Kabanets and Zhenjian Lu. 2018. Satisfiability and derandomization for small polynomial threshold circuits. In *Proc. 22nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*. Art. No. 46, 19. https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2018.46

[43] J. Kahn, G. Kalai, and N. Linial. 1988. The influence of variables on Boolean functions. In *Proc. 29th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 68–80. https://doi.org/10.1109/SFCS.1988.21923

[44] Matthias Krause and Pavel Pudlák. 1997. On the computational power of depth-2 circuits with threshold and modulo gates. *Theoretical Computer Science* 174, 1-2 (1997), 137–156. https://doi.org/10.1016/S0304-3975(96)00019-9

[45] Matthias Krause and Pavel Pudlák. 1998. Computing Boolean functions by polynomials and threshold circuits. *Computational Complexity* 7, 4 (1998), 346–370. https://doi.org/10.1007/s000370050015

[46] Chin Ho Lee and Emanuele Viola. 2018. The coin problem for product tests. *ACM Transactions on Computation Theory* 10, 3 (2018), Art. 14, 10. https://doi.org/10.1145/3201787

[47] Nutan Limaye, Karteek Sreenivasaiah, Srikanth Srinivasan, Utkarsh Tripathi, and S. Venkitesh. 2021. A fixed-depth size-hierarchy theorem for $AC^0[\oplus]$ via the coin problem. *SIAM Journal of Computing* 50, 4 (2021), 1461–1499. https://doi.org/10.1137/19M1276467

[48] Nutan Limaye, Srikanth Srinivasan, and Utkarsh Tripathi. 2019. More on $AC^0[\oplus]$ and Variants of the Majority Function. In *Proc. 39th Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*. 22:1–22:14. https://doi.org/10.4230/LIPIcs.FSTTCS.2019.22

[49] Nathan Linial, Yishay Mansour, and Noam Nisan. 1993. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM* 40, 3 (1993), 607–620. https://doi.org/10.1145/174130.174138

[50] Marvin Minsky and Seymour Papert. 1969. *Perceptrons: an Introduction to Computational Geometry*. MIT Press.

[51] Cody D. Murray and R. Ryan Williams. 2020. Circuit lower bounds for nondeterministic quasi-polytime from a new easy witness lemma. *SIAM J. Comput.* 49, 5 (2020), STOC18–300–STOC18–322. https://doi.org/10.1137/18M1195887

[52] Ryan O'Donnell and Rocco A. Servedio. 2010. New degree bounds for polynomial threshold functions. *Combinatorica* 30, 3 (2010), 327–358. https://doi.org/10.1007/s00493-010-2173-3

[53] Ryan O'Donnell and Karl Wimmer. 2007. Approximation by DNF: examples and counterexamples. In *Proc. 34th International Colloquium on Automata, Languages and Programming (ICALP)*. Lecture Notes in Comput. Sci., Vol. 4596. Springer, Berlin, 195–206. https://doi.org/10.1007/978-3-540-73420-8_19

[54] Igor C. Oliveira and Rahul Santhanam. 2018. Hardness magnification for natural problems. In *Proc. 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Soc., Los Alamitos, CA, 65–76. https://doi.org/10.1109/FOCS.2018.00016

[55] Alexander Razborov and Avi Wigderson. 1993. $n^{\Omega(\log n)}$ lower bounds on the size of depth-3 threshold circuits with AND gates at the bottom. *Inform. Process. Lett.* 45, 6 (1993), 303–307. https://doi.org/10.1016/0020-0190(93)90041-7

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

[57] Alexander A. Razborov and Alexander A. Sherstov. 2010. The Sign-Rank of AC^0. *SIAM Journal of Computing* 39, 5 (2010), 1833–1855. https://doi.org/10.1137/080744037

[58] Rocco A. Servedio. 2007. Every linear threshold function has a low-weight approximator. *Computational Complexity* 16, 2 (2007), 180–209. https://doi.org/10.1007/s00037-007-0228-7

[59] Ronen Shaltiel and Emanuele Viola. 2010. Hardness amplification proofs require majority. *SIAM J. Comput.* 39, 7 (2010), 3122–3154. https://doi.org/10.1137/080735096

[60] Alexander A. Sherstov. 2009. Separating AC^0 from Depth-2 Majority Circuits. *SIAM Journal of Computing* 38, 6 (2009), 2113–2129. https://doi.org/10.1137/08071421X

[61] Alexander A. Sherstov. 2011. The pattern matrix method. *SIAM Journal of Computing* 40, 6 (2011), 1969–2000. https://doi.org/10.1137/080733644

[62] Alexander A. Sherstov. 2018. Breaking the Minsky-Papert barrier for constant-depth circuits. *SIAM Journal of Computing* 47, 5 (2018), 1809–1857. https://doi.org/10.1137/15M1015704

[63] Alexander A. Sherstov. 2018. The power of asymmetry in constant-depth circuits. *SIAM Journal of Computing* 47, 6 (2018), 2362–2434. https://doi.org/10.1137/16M1064477

[64] Alexander A. Sherstov and Pei Wu. 2021. Near-Optimal Lower Bounds on the Threshold Degree and Sign-Rank of AC^0. *SIAM Journal of Computing* online ahead of print (2021), STOC19–1–STOC19–86. https://doi.org/10.1137/20M1312459

[65] Michael Sipser. 1983. Borel Sets and Circuit Complexity. In *Proc. 15th Annual ACM Symposium on Theory of Computing (STOC)*. 61–69. https://doi.org/10.1145/800061.808733

[66] Michael Sipser. 1983. A complexity theoretic approach to randomness. In *Proc. 15th Annual ACM Symposium on Theory of Computing (STOC)*. 330–335. https://doi.org/10.1145/800061.808762

[67] Roman Smolensky. 1987. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. 19th Annual ACM Symposium on Theory of Computing (STOC)*. 77–82. https://doi.org/10.1145/28395.28404

[68] Aravind Srinivasan. 2003. On the approximability of clique and related maximization problems. *Journal of Computer and System Sciences* 67, 3 (2003), 633–651. https://doi.org/10.1016/S0022-0000(03)00110-7

[69] John Steinberger. 2013. The distinguishability of product distributions by read-once branching programs. In *Proc. 28th Annual IEEE Conference on Computational Complexity (CCC)*. 248–254. https://doi.org/10.1109/CCC.2013.33

[70] Roei Tell. 2018. Quantified Derandomization of Linear Threshold Circuits. In *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*. 855–865. https://doi.org/10.1145/3188745.3188822

[71] Seinosuke Toda. 1991. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.* 20, 5 (1991), 865–877. https://doi.org/10.1137/0220053

[72] Emanuele Viola. 2014. Randomness Buys Depth for Approximate Counting. *Computational Complexity* 23, 3 (2014), 479–508. https://doi.org/10.1007/s00037-013-0076-6

[73] Nikhil Vyas and R. Ryan Williams. 2020. Lower bounds against sparse symmetric functions of ACC circuits: expanding the reach of #SAT algorithms. In *Proc. 37th Symposium on Theoretical Aspects of Computer Science (STACS) (LIPIcs. Leibniz Int. Proc. Inform., Vol. 154)*. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, Art. No. 59, 17. https://doi.org/10.4230/LIPIcs.STACS.2020.59

[74] Ryan Williams. 2014. Nonuniform ACC circuit lower bounds. *J. ACM* 61, 1 (2014), Art. 2, 32. https://doi.org/10.1145/2559903

[75] R. Ryan Williams. 2016. Natural proofs versus derandomization. *SIAM J. Comput.* 45, 2 (2016), 497–529. https://doi.org/10.1137/130938219

[76] R. Ryan Williams. 2018. Faster All-Pairs Shortest Paths via Circuit Complexity. *SIAM J. Comput.* 47, 5 (2018), 1965–1985. https://doi.org/10.1137/15M1024524

[77] R. Ryan Williams. 2018. New algorithms and lower bounds for circuits with linear threshold gates. *Theory of Computing* 14 (2018), Paper No. 17, 25. https://doi.org/10.4086/toc.2018.v014a017

[78] Andrew C-C. Yao. 1985. Separating the Polynomial-time Hierarchy by Oracles. In *Proc. 26th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 1–10. https://doi.org/10.1109/SFCS.1985.49

[79] Andrew Chi-Chih Yao. 1989. Circuits and Local Computation. In *Proc. 21st Annual ACM Symposium on Theory of Computing (STOC)*. 186–196. https://doi.org/10.1145/73007.73025

[80] Andrew Chi-Chih Yao. 1990. On ACC and threshold circuits. In *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Comput. Soc. Press, Los Alamitos, CA, 619–627. https://doi.org/10.1109/FSCS.1990.89583

Received 2022-11-07; accepted 2023-02-06