Stronger Connections Between Circuit Analysis and Circuit Lower Bounds, via PCPs of Proximity

Lijie Chen
MIT, Cambridge, MA, USA

R. Ryan Williams
MIT, Cambridge, MA, USA

Abstract

We considerably sharpen the known connections between circuit-analysis algorithms and circuit lower bounds, show intriguing equivalences between the analysis of weak circuits and (apparently difficult) circuits, and provide strong new lower bounds for approximately computing Boolean functions with depth-two neural networks and related models.

- We develop approaches to proving $\mathsf{THR} \circ \mathsf{THR}$ lower bounds (a notorious open problem), by connecting algorithmic analysis of $\mathsf{THR} \circ \mathsf{THR}$ to the *provably weaker* circuit classes $\mathsf{THR} \circ \mathsf{MAJ}$ and $\mathsf{MAJ} \circ \mathsf{MAJ}$, where *exponential* lower bounds have long been known. More precisely, we show equivalences between algorithmic analysis of $\mathsf{THR} \circ \mathsf{THR}$ and these weaker classes. The ε-error CAPP problem asks to approximate the acceptance probability of a given circuit to within additive error ε; it is the "canonical" derandomization problem. We show:
  - There is a non-trivial ($2^n/n^{\omega(1)}$ time) $1/\operatorname{poly}(n)$-error CAPP algorithm for $\operatorname{poly}(n)$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits if and only if there is such an algorithm for $\operatorname{poly}(n)$-size $\mathsf{MAJ} \circ \mathsf{MAJ}$.
  - There is a $\delta > 0$ and a non-trivial SAT (δ-error CAPP) algorithm for $\operatorname{poly}(n)$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits *if and only* if there is such an algorithm for $\operatorname{poly}(n)$-size $\mathsf{THR} \circ \mathsf{MAJ}$.

  Similar results hold for depth-d linear threshold circuits and depth-d MAJORITY circuits.

  These equivalences are proved via new simulations of THR circuits by circuits with MAJ gates.
- We strengthen the connection between non-trivial derandomization (non-trivial CAPP algorithms) for a circuit class $\mathcal{C}$, and circuit lower bounds against $\mathcal{C}$. Previously, [Ben-Sasson and Viola, ICALP 2014] (following [Williams, STOC 2010]) showed that for any polynomial-size class $\mathcal{C}$ closed under projections, non-trivial ($2^n/n^{\omega(1)}$ time) CAPP for $\mathsf{OR}_{\operatorname{poly}(n)} \circ \mathsf{AND}_3 \circ \mathcal{C}$ yields $\mathsf{NEXP} \not\subset \mathcal{C}$. We apply *Probabilistic Checkable Proofs of Proximity* in a new way to show it would suffice to have a non-trivial CAPP algorithm for either $\oplus_2 \circ \mathcal{C}$, $\mathsf{AND}_2 \circ \mathcal{C}$ or $\mathsf{OR}_2 \circ \mathcal{C}$.
- A direct corollary of the first two bullets is that $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$ would follow from either:
  - a non-trivial δ-error CAPP (or SAT) algorithm for $\operatorname{poly}(n)$-size $\mathsf{THR} \circ \mathsf{MAJ}$ circuits, or
  - a non-trivial $1/\operatorname{poly}(n)$-error CAPP algorithm for $\operatorname{poly}(n)$-size $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits.
- Applying the above machinery, we extend lower bounds for depth-two neural networks and related models [R. Williams, CCC 2018] to weak *approximate* computations of Boolean functions. For example, for arbitrarily small $\varepsilon > 0$, we prove there are Boolean functions f computable in nondeterministic $n^{\log n}$ time such that (for infinitely many n) every polynomial-size depth-two neural network N on n inputs (with sign or ReLU activation) must satisfy $\max_{x \in \{0,1\}^n} |N(x) - f(x)| > 1/2 - \varepsilon$. That is, short linear combinations of ReLU gates fail miserably at computing f to within close precision. Similar results are proved for linear combinations of $\mathsf{ACC} \circ \mathsf{THR}$ circuits, and linear combinations of low-degree $\mathbb{F}_p$ polynomials. These results constitute further progress towards $\mathsf{THR} \circ \mathsf{THR}$ lower bounds.

2012 ACM Subject Classification Theory of computation → Circuit complexity

Keywords and phrases PCP of Proximity, Circuit Lower Bounds, Derandomization, Threshold Circuits, ReLU

Digital Object Identifier 10.4230/LIPIcs.CCC.2019.19

Funding Supported by NSF CCF-1741615 (Common Links in Algorithms and Complexity) and a Google Faculty Research Award.

34th Computational Complexity Conference (CCC 2019).
Editor: Amir Shpilka; Article No. 19; pp. 19:1–19:43

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements Part of this work was completed while the authors were visiting the Simons Institute at UC Berkeley, as part of the program on Lower Bounds in Computational Complexity. We thank them for their hospitality and excellent environment. We also thank Josh Alman for helpful last-minute proofreading, and the CCC reviewers for useful comments.

1 Introduction

Recall TC^0 is the class of decision problems that are computable with circuit families of constant depth, composed of MAJORITY and NOT gates. As this class remains the same when "MAJORITY" is replaced by other more expressive functions such as *linear threshold functions* [21, 32], TC^0 naturally captures many mathematical models of neural computing, and contains many natural arithmetic functions (for example, see [9, 38, 26]).

What interesting functions do not have polynomial-size TC^0 circuits? Despite substantial research effort [23, 2, 4, 17, 19, 21, 22, 24, 25, 29, 28, 35, 37, 40, 51, 44, 3, 31] it is consistent with current knowledge that the huge class *nondeterministic exponential time* (NEXP) has polynomial-size $\mathsf{THR} \circ \mathsf{THR}$ circuits, which are depth *two* and can compute arbitrary linear threshold functions at both layers.[1] It seems obvious that "shallow" nets cannot be so powerful, but concrete proofs of their limitations have been elusive.

In 2011, R. Williams [49, 52] proved that NEXP does not have polynomial-size ACC^0 circuits (a presumably weaker circuit class), by showing how circuit lower bounds follow from non-trivial algorithms[2] for problems such as *circuit satisfiability* or *circuit derandomization*. The canonical circuit derandomization problem is CAPP, where the task is to approximate the acceptance probability of a given circuit within an additive constant error (less than $1/3$, say). Along these lines, subsequent works have followed Williams' program [50, 51, 12, 30, 3, 47, 44, 48], and more circuit lower bounds have been proved by either introducing new SAT algorithms, or tightening the algorithms-to-lower-bounds connection. For an example of the latter, Murray and Williams [34] recently showed that nondeterministic *quasi-polynomial time* does not have polynomial-size $\mathsf{ACC}^0 \circ \mathsf{THR}$ circuits using a strengthened connection.

A potential next step in this program would be to prove that NEXP does not have polynomial-size depth-two threshold circuits ($\mathsf{THR} \circ \mathsf{THR}$). Partial progress has already been made: for example, in [44, 3], it is shown that E^{NP} does not have $n^{2-o(1)}$ size $\mathsf{THR} \circ \mathsf{THR}$ circuits. Until now, essentially all lower bounds proven by this program have applied very strong circuit-analysis algorithms, such as circuit satisfiability or #SAT algorithms. It looks difficult to find such strong circuit-analysis algorithms for $\mathsf{THR} \circ \mathsf{THR}$ circuits. Indeed, even for the simpler MAX-k-SAT problem (equivalent to the SAT problem for $\mathsf{MAJ} \circ \mathsf{AND}_k$ circuits), no non-trivial algorithms are known for $k(n) = \omega(\log n)$ (see [8]). For slightly larger circuit classes such as NC^1 (a.k.a. poly(n)-size formulas), it has been conjectured that there are no non-trivial SAT algorithms [1].

An approach based on *derandomizing* a circuit class (finding non-trivial algorithms for CAPP) looks more plausible than one based on SAT solving, because most researchers believe "full" derandomization is possible and that CAPP is in P even for arbitrary circuits. However, there is still a substantial obstacle for proving $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$ via derandomization approaches. While previous works ([30, 13]) have shown that $\mathsf{NEXP} \not\subset \mathcal{C}$ would follow from a non-trivial satisfiability algorithm for $\mathsf{AND}_3 \circ \mathcal{C}$ (i.e., an AND of three $\mathcal{C}$-circuits), the best

[1] See Section 2.1 for formal definitions.

[2] Throughout the paper, we use the term "non-trivial algorithm" to mean that, for every constant $c \geq 1$, the algorithm runs in $2^n/n^{\omega(1)}$ time on circuits of n inputs and n^c gates.

known connection theorem (namely, that of Ben-Sasson and Viola [13]) is that non-trivial derandomization of $\mathsf{OR}_{\mathrm{poly}(n)} \circ \mathsf{AND}_3 \circ \mathcal{C}$ (i.e., a non-trivial CAPP algorithm for a 3-DNF on $\mathrm{poly}(n)$ many $\mathcal{C}$ circuits) implies $\mathsf{NEXP} \not\subset \mathcal{C}$. Finding a tighter correspondence (with no "DNF overheads" in the connection) has been an intriguing open problem.

In this work, applying *Probabilistically Checkable Proofs of Proximity* (PCPPs), we substantially tighten the connection given by Ben-Sasson and Viola, showing that non-trivial derandomization for depth-d TC circuits would directly imply NEXP does not have depth-d TC circuits. That is, in order to show $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$, it suffices to find a non-trivial derandomization of $\mathsf{THR} \circ \mathsf{THR}$. Furthermore, we show that non-trivial derandomization algorithms for $\mathsf{THR} \circ \mathsf{THR}$ is in fact *equivalent* to derandomization for the weaker class $\mathsf{THR} \circ \mathsf{MAJ}$ (tightly) and derandomization for the even weaker class $\mathsf{MAJ} \circ \mathsf{MAJ}$ (with inverse polynomial error). ($\mathsf{THR} \circ \mathsf{MAJ}$ circuits are the special case of $\mathsf{THR} \circ \mathsf{THR}$ circuits where all gates on the bottom layer only compute linear threshold functions with *polynomial* integer weights; $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits have that restriction on both layers. See Section 2.1 for formal definitions.) Therefore, for our desired lower bounds against $\mathsf{THR} \circ \mathsf{THR}$, it suffices to obtain non-trivial CAPP algorithms for $\mathsf{THR} \circ \mathsf{MAJ}$ or $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits, for which exponential-size circuit lower bounds have long been known [19, 23].

As an additional application, we apply our new PCPP approach to strengthen recent depth-two neural network lower bounds of R. Williams [48] for approximate computation of Boolean functions. For example, we show that for every $\varepsilon > 0$ and every (non-uniform) polynomial-size family of depth-two neural nets $\{N_n\}$ with sign or ReLU activation functions, there are Boolean functions f in nondeterministic $n^{O(\log^* n)}$ time such that $\max_{x \in \{0,1\}^n} |N_n(x) - f(x)| > 1/2 - \varepsilon$ for infinitely many n. That is, arbitrary linear combinations of ReLU gates fail miserably at computing f to within any close precision. Versions of the PCP theorem are crucial elements in the proofs of these lower bounds; indeed, our overall argument involves applications of a PCP in two different places. Previously, all concrete circuit lower bounds proved via the algorithmic approach have not required the full power of the PCP theorem [5, 6] for the argument to work.

To formally describe our results, we recall three circuit-analysis problems.

1. **CAPP with error δ:** Given a circuit C on n inputs, estimate the probability that $C(a) = 1$ over uniformly random input $a \in \{0,1\}^n$, to within $\pm\delta$.
2. **SAT:** Given a circuit C, determine if there is an input a such that $C(a) = 1$.
3. **Gap-UNSAT with gap δ:** Given a circuit C, output YES when $C(a) = 0$ for all a, and NO when C has at least $\delta \cdot 2^n$ satisfying assignments. Note that $\mathsf{Gap\text{-}UNSAT}$ with gap δ is easier than the other two problems: either a SAT algorithm or a CAPP algorithm with error δ would immediately imply a $\mathsf{Gap\text{-}UNSAT}$ algorithm with gap δ.

1.1 Equivalence Between Algorithmic Analysis of $\mathsf{THR} \circ \mathsf{THR}$, $\mathsf{THR} \circ \mathsf{MAJ}$, $\mathsf{MAJ} \circ \mathsf{MAJ}$

Our first results give equivalences between algorithmic analysis of $\mathsf{THR} \circ \mathsf{THR}$, $\mathsf{THR} \circ \mathsf{MAJ}$, and $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits. These equivalences are surprising, because the latter two classes are *provably weaker* than $\mathsf{THR} \circ \mathsf{THR}$ [16]. In fact, $2^{\Omega(n)}$-size lower bounds are well-known for them [23, 19].

Poly-Size THR ∘ MAJ and THR ∘ THR are Equivalent for Circuit-Analysis Algorithms

We say an algorithm running on n-input circuits is *non-trivial* if for all c, it runs in $2^n/n^{\omega(1)}$ time for all circuits of size n^c. We first show that, in terms of designing non-trivial SAT or CAPP algorithms, THR ∘ MAJ and THR ∘ THR are *equally hard or easy.*

▶ **Theorem 1.** *The following two statements hold:*

- ***Equivalence of Non-Trivial SAT Algorithms****: There is a non-trivial SAT algorithm for THR ∘ MAJ circuits of* poly(n)*-size* if and only if *there is such an algorithm for* poly(n)*-size THR ∘ THR circuits.*
- ***Equivalence of Non-Trivial CAPP Algorithms With Constant Error****: For any constant $\delta > 0$, If there is a non-trivial CAPP algorithm with error δ for THR ∘ MAJ circuits of* poly(n) *size, then there is a non-trivial CAPP algorithm with error $\delta + 1/n$ for* poly(n)*-size THR ∘ THR circuits.*

Theorem 1 generalizes readily to TC circuits of any constant depth d. Let LT_d be the class of the depth-d circuits consisting entirely of arbitrary linear threshold functions, and let $\widehat{\mathsf{LT}}_d$ be the subclass of LT_d with the restriction that all gates have polynomially-bounded integer weights (see Section 2.1 for formal definitions). E.g., $\widehat{\mathsf{LT}}_2 = \mathsf{MAJ} \circ \mathsf{MAJ}$.

▶ **Corollary 2.** *The following two statements hold for every constant d:*

- ***Equivalence of Non-Trivial SAT Algorithms****: There is a non-trivial SAT algorithm for THR ∘ $\widehat{LT}_{d-1}$ circuits of* poly(n)*-size* if and only if *there is such an algorithm for* poly(n)*-size LT_d circuits.*
- ***Equivalence of Non-Trivial CAPP Algorithms With Constant Error****: For any constant $\varepsilon > 0$, if there is a non-trivial CAPP algorithm with error ε for THR ∘ $\widehat{LT}_{d-1}$ circuits of* poly(n)*-size, then there is a non-trivial CAPP algorithm with error $\varepsilon + 1/n$ for* poly(n)*-size LT_d circuits.*

Weaker Equivalence Between Poly-Size THR ∘ THR and MAJ ∘ MAJ

We also obtain some weaker equivalences between circuit-analysis algorithms for THR ∘ THR and MAJ ∘ MAJ circuits.

▶ **Theorem 3.** *The following two statements hold:*

- ***Equivalence of $2^{(1-\varepsilon)n}$-time SAT Algorithms****: If SAT for* poly(n)*-size MAJ ∘ MAJ circuits is in $2^{(1-\varepsilon)n}$ time for an $\varepsilon > 0$, then SAT for* poly(n)*-size THR ∘ THR circuits is in $2^{(1-\varepsilon')n}$ time for an $\varepsilon' > 0$.*
- ***Equivalence of Non-Trivial CAPP Algorithms with Inverse Polynomial Error****: If there is a non-trivial CAPP algorithm with $1/poly(n)$ error for* poly(n)*-size MAJ ∘ MAJ circuits, then there is a non-trivial CAPP algorithm with $1/$* poly(n) *error for* poly(n)*-size THR ∘ THR circuits.*

Again, a similar result holds for TC circuits of any constant depth d.

▶ **Corollary 4.** *The following two statements hold for any constant d:*

- ***Equivalence of $2^{(1-\varepsilon)n}$-time SAT Algorithms****: If SAT for $\widehat{LT}_d$ circuits of* poly(n)*-size is in $2^{(1-\varepsilon)n}$ time for an $\varepsilon > 0$, then SAT for* poly(n)*-size LT_d circuits is in $2^{(1-\varepsilon')n}$ time for an $\varepsilon' > 0$.*

- ***Equivalence of Non-trivial CAPP Algorithms with inverse polynomial error:*** *If there is a non-trivial CAPP algorithm with $1/poly(n)$ error for $\mathrm{poly}(n)$-size $\widehat{LT}_d$ circuits, then there is a non-trivial CAPP algorithm with $1/\mathrm{poly}(n)$ error for $\mathrm{poly}(n)$-size LT_d circuits.*

1.2 Tighter Connection Between Circuit Lower Bounds and Non-trivial Derandomization

Our next results give a tighter connection between non-trivial circuit-analysis algorithms for $\mathcal{C}$, and circuit lower bounds against $\mathcal{C}$. We say a circuit class $\mathcal{C}$ is *typical* if it is closed under taking negations of the output, and variable projections.[3] We show that, to prove $\mathsf{NEXP} \not\subset \mathcal{C}$, it suffices to obtain non-trivial derandomization of $\mathsf{AND}_3 \circ \mathcal{C}$, $\mathsf{OR}_2 \circ \mathcal{C}$, or $\oplus_2 \circ \mathcal{C}$ (a.k.a. $\mathsf{XOR}_2 \circ \mathcal{C}$).

▶ **Theorem 5** (Lower Bounds From Non-Trivial Gap-UNSAT or CAPP Algorithms). *There is an absolute constant $\delta > 0$, such that for any typical circuit class $\mathcal{C}$, if one of the following holds:*

- *there is a non-trivial Gap-UNSAT algorithm with gap δ for $\mathrm{poly}(n)$-size $AND_3 \circ \mathcal{C}$ circuits, or*
- *there is a non-trivial CAPP algorithm with error δ for $\mathrm{poly}(n)$-size $OR_2 \circ \mathcal{C}$, $\oplus_2 \circ \mathcal{C}$, or $AND_2 \circ \mathcal{C}$ circuits,*

then $NEXP \not\subset \mathcal{C}$. Moreover, in the second bullet, $\mathcal{C}$ does not need to be closed under negation.

Comparison with Ben-Sasson and Viola

Ben-Sasson and Viola [12] showed that non-trivial Gap-UNSAT algorithms for $\mathsf{OR}_{\mathrm{poly}(n)} \circ \mathsf{AND}_3 \circ \mathcal{C}$, or non-trivial satisfiability algorithms for $\mathsf{AND}_3 \circ \mathcal{C}$ would imply $\mathsf{NEXP} \not\subset \mathcal{C}$. Theorem 5 is a strict strengthening of these two connections, as Gap-UNSAT is an easier problem than SAT. In particular, note we avoid the unbounded fan-in OR entirely.

Applying the "easy witness lemma for NP" results of Murray and Williams [34], we can naturally generalize to circuit lower bounds for NP if faster algorithms are used.

▶ **Theorem 6** (NP Lower Bounds From Faster Gap-UNSAT or CAPP Algorithms). *There is an absolute constant $\alpha > 0$, such that for any typical circuit class $\mathcal{C}$, if there is a constant δ such that one of the following holds:*

- *Gap-UNSAT for $2^{\delta n}$-size $AND_3 \circ \mathcal{C}$ circuits with gap α can be solved in $2^{n-\delta n}$ time, or*
- *CAPP for $2^{\delta n}$-size $OR_2 \circ \mathcal{C}$, $\oplus_2 \circ \mathcal{C}$, or $AND_2 \circ \mathcal{C}$ circuits with error α can be solved in $2^{n-\delta n}$ time,*

then for every k there is a function in NP that doesn't have n^k-size $\mathcal{C}$ circuits. Moreover, in the second bullet, $\mathcal{C}$ does not need to be closed under negation.

▶ Remark 7. In Theorems 5 and 6, the desired algorithms can even be non-deterministic, as long as on all computation paths the algorithm either outputs *don't know* or the correct answer, and the correct answer always appears on at least one path.

In terms of techniques, our approach is very different from that of the previous derandomization connection proved by Ben-Sasson and Viola [12]. They constructed a highly efficient PCP for $\mathsf{NTIME}[T(n)]$, where the queries are *projections* of random bits, and the verifier is a 3-CNF. Their results were then obtained by directly plugging this PCP construction into the original argument of [49].

[3] See Section 2.1 for the details.

Our approach is less direct. Our key insight in proving Theorem 5 (and Theorem 6 similarly) is to use PCPs of Proximity to reduce circuit evaluation tasks to derandomization tasks. Using efficient PCPs for $\mathsf{NTIME}[2^n]$ [10], Williams [49] showed $\mathsf{NEXP} \not\subset \mathsf{P/poly}$ follows from non-trivial Gap-UNSAT algorithms for $\mathrm{poly}(n)$-size general circuits. Applying *PCPs of Proximity* to the *Circuit Evaluation Problem*, we design a Gap-UNSAT algorithm for *general* circuits, only assuming that $\mathsf{NEXP} \subset \mathcal{C}$ and a Gap-UNSAT algorithm for $\mathsf{AND}_3 \circ \mathcal{C}$, which results in a contradiction when $\mathcal{C} \subset \mathsf{P/poly}$. Therefore our overall argument applies PCP constructions in two different ways: first on a nondeterministic 2^n-time computation to reduce to a Gap-UNSAT problem, and then on a $\mathrm{poly}(n)$-size circuit evaluation. See Section 1.6 for an overview of the whole argument.

1.3 Potential Approaches to THR ∘ THR Circuit Lower Bounds

As a direct corollary of Theorem 5 and the folklore result that and XOR of two $\mathsf{THR} \circ \mathsf{THR}$ can be written as a polynomially-larger $\mathsf{THR} \circ \mathsf{THR}$,[4] it follows that non-trivial CAPP algorithms for $\mathsf{THR} \circ \mathsf{THR}$ circuits with small constant error would imply $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$. With a little additional work, the same can be shown for non-trivial SAT algorithms for $\mathsf{THR} \circ \mathsf{THR}$ circuits.

▶ **Theorem 8.** *There is an absolute constant $\delta > 0$, such that if δ-error CAPP for $\mathrm{poly}(n)$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits can be solved in $2^n/n^{\omega(1)}$ time, then $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$. The same is true with SAT in place of CAPP.*

The above theorem generalizes to TC circuits of any constant depth d (LT_d circuits).

▶ **Theorem 9.** *There is an absolute constant $\delta > 0$, such that for any constant d, if CAPP for $\mathrm{poly}(n)$-size LT_d circuits with error δ can be solved in $2^n/n^{\omega(1)}$ time, then $\mathsf{NEXP} \not\subset \mathsf{LT}_d$. The same is true with SAT in place of CAPP.*

It still appears to be a tough challenge to obtain a non-trivial CAPP algorithm for polynomial-size $\mathsf{THR} \circ \mathsf{THR}$ circuits, as it is usually the case that derandomizations come from circuit lower bounds (and ironically, our goal here is to prove circuit lower bounds for $\mathsf{THR} \circ \mathsf{THR}$!). However, armed with our new equivalence results between algorithmic analysis of $\mathsf{THR} \circ \mathsf{THR}$ circuits and $\mathsf{THR} \circ \mathsf{MAJ}$ or $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits (Theorem 1 and Theorem 3), it suffices for us to obtain non-trivial CAPP algorithms for $\mathsf{THR} \circ \mathsf{MAJ}$ or $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits, for which $2^{\Omega(n)}$ lower bounds are known.

▶ **Corollary 10.** *There is an absolute constant $\delta > 0$, such that if one of the following holds:*
1. *CAPP (or SAT) for $\mathrm{poly}(n)$-size $\mathsf{THR} \circ \mathsf{MAJ}$ circuits with error δ is in $2^n/n^{\omega(1)}$ time, or*
2. *CAPP for $\mathrm{poly}(n)$-size $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits with $1/poly(n)$ error is in $2^n/n^{\omega(1)}$ time,*

then $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$.

Therefore $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$ follows if we can "mine" the known $2^{\Omega(n)}$ lower bounds for $\mathsf{THR} \circ \mathsf{MAJ}$ or $\mathsf{MAJ} \circ \mathsf{MAJ}$, and design non-trivial circuit-analysis algorithms from them!

1.4 Lower Bounds on Representing Boolean Functions Approximately by Linear Combinations of Simple Functions

Finally, we apply our new techniques to strengthen recent depth-two lower bounds of R. Williams [48]. He studied the problem of representing a Boolean function f exactly by a linear combination of simple functions from a class $\mathcal{C}$. Here we introduce an approximate form of such representations.

[4] See e.g. Lemma 50 for a proof.

▶ **Definition 11.** *Let $\mathcal{C}$ be a class of functions from $\{0,1\}^n \to \mathbb{R}$ and $\varepsilon \in [0, 0.5)$. We say $f : \{0,1\}^n \to \{0,1\}$ admits a $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit of sparsity S, if there are S functions $C_1, C_2, \ldots, C_S$ from $\mathcal{C}$, together with S coefficients $\alpha_1, \alpha_2, \ldots, \alpha_S$ in $\mathbb{R}$, such that for all $x \in \{0,1\}^n$,*

$$\left| \sum_{i=1}^{S} \alpha_i \cdot C_i(x) - f(x) \right| \leq \varepsilon.$$

We use $\mathsf{Sum} \circ \mathcal{C}$ to denote the special case of $\varepsilon = 0$, which was the case studied in prior work [48].

When $\mathcal{C}$ is the class of AND gates (or PARITY gates, respectively), we are asking for the *sparsest* ε-approximate polynomial for f, with respect to the standard (or Fourier basis, respectively). This is related to the ε-approximate degree[5] of f, which is already a highly-nontrivial notion; for instance, the approximate degrees of simple natural functions have only recently been determined [15, 14, 42].

In prior work, Williams [48] showed that non-trivial algorithms for the so-called "Sum-Product"[6] of $O(1)$ functions from $\mathcal{C}$ implies sparsity lower bounds against $\mathsf{Sum} \circ \mathcal{C}$, and he obtained sparsity lower bounds against various $\mathsf{Sum} \circ \mathcal{C}$ circuits by designing corresponding Sum-Product algorithms.

Applying our new techniques together with other new ideas, we show that Sum-Product algorithms in fact yield sparsity lower bounds against $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$. That is, we can systematically "lift" the $\mathsf{Sum} \circ \mathcal{C}$ lower bounds in [48] to lower bounds for $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$.

First, we generalize the lower bounds for $\mathsf{Sum} \circ \mathsf{THR}$ in [48] to $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathsf{THR}$. Such circuits are also known in the machine learning literature as *depth-two neural networks with sign activation functions*.

▶ **Theorem 12** (Lower Bound for $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathsf{THR}$)**.** *For all k and constant $\varepsilon < 1/2$, there is a function in NP without $\widetilde{\mathsf{Sum}}_\varepsilon \circ$ THR circuits of n^k sparsity. Furthermore, if $\alpha(n)$ is unbounded such that $n^{\alpha(n)}$ is time-constructible, then $\mathsf{NTIME}[n^{\alpha(n)}] \not\subset \widetilde{\mathsf{Sum}}_\varepsilon \circ$ THR for all constant $\varepsilon < 1/2$.*

A ReLU (rectified linear unit) gate is a function $f : \{0,1\}^t \to \mathbb{R}^+$ such that there is a vector $w \in \mathbb{R}^t$ and scalar $a \in \mathbb{R}$ such that for all x,

$$f(x) = \max\{0, \langle x, w \rangle + a\}.$$

Linear combinations of ReLU gates are also known as *depth-two neural networks with ReLU activation functions*, and they are intensely studied in machine learning.[7]

Next we generalize the lower bounds for $\mathsf{Sum} \circ \mathsf{ReLU}$ in [48] to $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathsf{ReLU}$.

▶ **Theorem 13** (Lower Bound for $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathsf{ReLU}$)**.** *For all k and constant $\varepsilon < 1/2$, there is a function in NP without $\widetilde{\mathsf{Sum}}_\varepsilon \circ$ ReLU circuits of n^k sparsity. Furthermore, if $\alpha(n)$ be unbounded such that $n^{\alpha(n)}$ is time-constructible, then $\mathsf{NTIME}[n^{\alpha(n)}] \not\subset \widetilde{\mathsf{Sum}}_\varepsilon \circ$ ReLU for all constant $\varepsilon < 1/2$.*

We also obtain analogous lower bounds for $\widetilde{\mathsf{Sum}}_\varepsilon \circ (O(1)$-degree $\mathbb{F}_p$-polynomials), strengthening lower bounds for exact linear combinations of $O(1)$-degree polynomials [48].

---

[5] The ε-approximate degree of f is the lowest degree of all polynomial $p : \{0,1\}^n \to \{0,1\}$ such that $\|p - f\|_\infty \leq \varepsilon$. Note that a low degree polynomial is also sparse.

[6] See Section 6 for a formal definition. Intuitively, the "Sum-Product" problem generalizes #SAT.

[7] We refer the readers to [48] and the references therein for more discussion on this topic.

▶ **Theorem 14** (Lower Bound for $\widetilde{\mathsf{Sum}}_\varepsilon \circ (\mathbb{F}_p\text{-polynomials})$). *For all prime p, integers k, d, and constant $\varepsilon < 1/2$, there is a function in NP without $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathsf{MOD}_p \circ \mathsf{AND}_d$ circuits of n^k sparsity. Furthermore, if $\alpha(n)$ be unbounded such that $n^{\alpha(n)}$ is time-constructible, then $\mathsf{NTIME}[n^{\alpha(n)}] \not\subset \widetilde{\mathsf{Sum}}_\varepsilon \circ \mathsf{MOD}_p \circ \mathsf{AND}_d$ for all constant $\varepsilon < 1/2$.*

Finally, using the known #SAT algorithm for $\mathsf{ACC}^0 \circ \mathsf{THR}$ [51], we show a sparsity lower bound for $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathsf{ACC}^0 \circ \mathsf{THR}$ circuits.

▶ **Theorem 15.** *For every $d, m \geq 1$ and $\varepsilon \in [0, 0.5)$, there is a $b \geq 1$ and an $f \in \mathsf{NTIME}[n^{\log^b n}]$ that does not have $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathsf{AC}^0_d[m] \circ \mathsf{THR}$ circuits of n^a size, for every a.*

Therefore, no polynomially-sparse linear combination of $\mathsf{AC}^0_d[m] \circ \mathsf{THR}$ circuits can approximate the value of the hard function in Theorem 15.

This constitutes the strongest known circuit class for which we can presently prove lower bounds for nondeterministic quasi-polynomial time (improving [34]).

1.5 Techniques: Two Structure Lemmas for THR ∘ THR

Two major technical ingredients in our results are structure lemmas for $\mathsf{THR} \circ \mathsf{THR}$, which are of interest in their own right. Informally, our first structure lemma says that every $\mathsf{THR} \circ \mathsf{THR}$ is equivalent to a polynomial-sized OR of Threshold-of-Majority circuits. The second structure lemma says that every $\mathsf{THR} \circ \mathsf{THR}$ circuit is equivalent to a subexponential-sized OR of Majority-of-Majority circuits. For the program of proving $\mathsf{THR} \circ \mathsf{THR}$ lower bounds, this is significant, as exponential-size Majority-of-Majority and Threshold-of-Majority lower bounds are well-known [23, 19].

In the following, DOR refers to a "disjoint" OR gate: an OR gate with the promise that at most one of its inputs is ever true, and $\mathsf{Gap\text{-}OR}_\delta$ refers to a "gapped" OR gate with a error parameter δ: an OR gate with the promise that either all inputs are false or at least a $1 - \delta$ fraction of the inputs are true. We also use Gap-OR to denote $\mathsf{Gap\text{-}OR}_{1/2}$ for simplicity. (See Section 2.1 for formal definitions.)

▶ **Lemma 16** (Structure Lemma I for $\mathsf{THR} \circ \mathsf{THR}$ circuits). *Let n be the number of inputs, let $s = s(n) \geq n$ be a size function, and let $\delta = \delta(n) \in (0, 1)$ be an error function. Every s-size $\mathsf{THR} \circ \mathsf{THR}$ circuit C is equivalent to a $\mathsf{Gap\text{-}OR}_\delta \circ \mathsf{THR} \circ \mathsf{MAJ}$ circuit C' such that:*

- *The top $\mathsf{Gap\text{-}OR}_\delta$ gate of C' has $\mathrm{poly}(s, \delta^{-1})$ fan-in.*
- *Each $\mathsf{THR} \circ \mathsf{MAJ}$ subcircuit of C' has size $\mathrm{poly}(s, \delta^{-1})$.*

The transformation from C to C' can be computed in deterministic $\mathrm{poly}(s, \delta^{-1})$ time.

▶ **Lemma 17** (Structure Lemma II for $\mathsf{THR} \circ \mathsf{THR}$ circuits). *Let n be the number of inputs and let $s = s(n) \leq 2^{o(n)}$ be a size parameter. Let $\varepsilon \in \left(\frac{\log s}{n}, 1\right)$. Every s-size $\mathsf{THR} \circ \mathsf{THR}$ circuit C is equivalent to a $\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{MAJ}$ circuit such that:*

- *The top DOR gate has $2^{O(\varepsilon n)}$ fan-in.*
- *Each sub $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuit has size $s^{O(1/\varepsilon)}$.*

The reduction can be computed in randomized $2^{O(\varepsilon n)} \cdot s^{O(1/\varepsilon)}$ *time.*

Previously, Goldmann-Håstad-Razborov [21] showed that every $\mathsf{THR} \circ \mathsf{THR}$ circuit has an equivalent $\mathsf{MAJ} \circ \mathsf{MAJ} \circ \mathsf{MAJ}$ circuit of polynomially larger size. The top OR gates in our structure lemmas have additional benefits: for instance, an $\mathsf{OR} \circ \mathcal{C}$ circuit is satisfiable, *if and only if* one of its $\mathcal{C}$ subcircuits is satisfiable. Therefore, solving SAT on an $\mathsf{OR} \circ \mathcal{C}$ circuit is easily reduced to solving SAT on $\mathcal{C}$ circuits.

In Appendix C, we discuss more applications of the above two structure lemmas, beyond the algorithmic equivalences for $\mathsf{THR} \circ \mathsf{THR}$, $\mathsf{THR} \circ \mathsf{MAJ}$, and $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits.

1.6 Intuition: Solving Gap-UNSAT with Probabilistic Checkable Proofs of Proximity

Here we provide an overview of the ideas behind our new tightened connection between circuit lower bounds and circuit-analysis algorithms.

Starting Point: Designing Gap-UNSAT Algorithms for General Circuits, Assuming NEXP $\subset \mathcal{C}$

Suppose $\mathcal{C} \subset \mathsf{P}/\mathrm{poly}$. We want to show that a non-trivial Gap-UNSAT algorithm with a constant gap for poly(n)-size $\mathsf{AND}_3 \circ \mathcal{C}$ circuits implies $\mathsf{NEXP} \not\subset \mathcal{C}$. We start with the following connection of R. Williams [49]:

> If Gap-UNSAT with gap $1-1/n^{10}$ for (fan-in 2) circuits with n inputs and poly(n) size is solvable in $O(2^n/n^{\omega(1)})$ nondeterministic time, then NEXP doesn't have poly(n)-size (fan-in 2) circuits.

Our strategy is to assume $\mathsf{NEXP} \subset \mathcal{C}$, and use our non-trivial Gap-UNSAT algorithm for $\mathsf{AND}_3 \circ \mathcal{C}$ to derive a non-trivial Gap-UNSAT algorithm for *general* fan-in-2 circuits. This would imply a contradiction, since by the above connection, it follows that $\mathsf{NEXP} \not\subset \mathsf{P}_{/\mathrm{poly}}$ and therefore $\mathsf{NEXP} \not\subset \mathcal{C}$.

So suppose we are given a poly(n)-size general circuit $C : \{0,1\}^n \to \{0,1\}$ with the promise that either C is unsatisfiable (the YES case) or C has at least $(1 - 1/n^{10}) \cdot 2^n$ satisfying assignments (the NO case), where our goal is to distinguish the two cases in $2^n/n^{\omega(1)}$ non-deterministic time.

To simplify the discussion, we negate the circuit C. Now we are promised C is a tautology, or C has at most $1/n^{10} \cdot 2^n$ satisfying assignments, and we must nondeterministically prove C is a tautology (when that is the case) in $2^n/n^{\omega(1)}$ time.

Review of the Approach in Williams' ACC Lower Bound

It will be useful to review the previous approach ([52]) first, and see where we deviate from it.[8] Let the circuit C be given as above. First, assuming $\mathsf{NEXP} \subset \mathcal{C}$ (which implies Circuit-Eval $\in \mathcal{C}$), there is an equivalent poly(n)-size $\mathcal{C}$ circuit D equivalent to C. Since we are allowed to use *non-deterministic* algorithms, we might try to *guess* a $\mathcal{C}$ circuit D, and *verify* that D is equivalent to C. If this verification can be done in $2^n/n^{\omega(1)}$ time, then we could apply the Gap-UNSAT algorithm for $\mathcal{C}$ to the circuit D, and solve Gap-UNSAT for C. Indeed, this is the original approach of Wiliams [52].

Since the NAND gate ($\mathsf{NAND}(z_1, z_2) := \neg(z_1 \wedge z_2)$) is universal, we may assume C consists of $m = $ poly(n) NAND gates, the first n gates are the inputs (that is, the i-th gate is the input bit x_i for $i \in [n]$), and the m-th gate is the output gate. Let C_i be the subcircuit of C where the i-th gate is the output. Since we are assuming Circuit-Eval $\in \mathcal{C}$, for all C_i there is always an equivalent $\mathcal{C}$-circuit T_i of poly(n) size.

The overall guess-and-verify algorithm works as follows:

- Guess $m - n$ $\mathcal{C}$-circuits $T_{n+1}, T_{n+2}, \ldots, T_m$, such that T_i is intended to be equivalent to C_i. For $i \in [n]$, we set T_i to be a trivial circuit which always outputs the i-th bit of the input.

[8] Our presentation here is slightly different from the original proof.

- For $i \in \{n+1, n+2, \ldots, m\}$, let i_1 and i_2 be the indices of the two gates which are inputs to the i-th gate of C. We want to verify

$$\mathsf{NAND}(T_{i_1}(x), T_{i_2}(x)) = T_i(x) \tag{1}$$

  is true for all $x \in \{0,1\}^n$. This can be reduced to solving SAT for $\mathsf{AND}_3 \circ \mathcal{C}$ circuits.
- If all the above checks pass, then we know T_m is equivalent to C.

The Proof System View

The above approach requires using SAT algorithms to verify (1) is true for all $x \in \{0,1\}^n$, whereas we only want to assume non-trivial $\mathsf{Gap\text{-}UNSAT}$ algorithms (which could be much weaker). Here we present a different perspective on the above approach.

Letting $\pi(x) := (T_{n+1}(x), T_{n+2}(x), \ldots, T_m(x))$, we can view $\pi(x)$ as a certain "locally-checkable proof" for $C(x) = 1$. That is, $C(x) = 1$ if and only if there is a proof $\pi(x) \in \{0,1\}^{m-n}$ such that for the string $z = x \circ \pi(x)$ ($\circ$ means concatenation), we have $\mathsf{NAND}(z_{i_1}, z_{i_2}) = z_i$ for all $i \in \{n+1, n+2, \ldots, m\}$, and $z_m = 1$.

Can we obtain something better from the "locally-checkable" perspective? We may write all the constraints checked in our proof system as a 3-CNF formula φ on $z = x \circ \pi(x)$ of $\ell = O(m) = \mathrm{poly}(n)$ clauses. (Note, this simply mimics the standard reduction from $\mathsf{Circuit\text{-}Eval}$ to 3-SAT.) Suppose the i-th clause is $F_i(z) := \vee_{j=1}^{3}(z_{i_j} \oplus b_{i,j})$.

- As before, we guess $\mathcal{C}$ circuits $T_{n+1}(x), T_{n+2}(x), \ldots, T_m(x)$, but this time with the intention that $T(x) = (T_{n+1}(x), T_{n+2}(x), \ldots, T_m(x))$ is the correct proof for input x.
- When C is a tautology, there is a guess $T(x)$ such that $\mathbb{E}_{x \sim \mathcal{U}_n} \mathbb{E}_{i \in [\ell]}[F_i(x \circ T(x))] = 1$.
- Otherwise, for all guessed $T(x)$, we have $\mathbb{E}_{x \sim \mathcal{U}_n} \mathbb{E}_{i \in [\ell]}[F_i(x \circ T(x))] \leq 1/n^{10} + \frac{\ell-1}{\ell}$, since for at least a $1 - 1/n^{10}$ fraction of inputs, we have $C(x) = 0$, and therefore at most $\ell - 1$ clauses can be satisfied by $x \circ T(x)$.

Note that $F_i(x \circ T(x))$ is an $\mathsf{OR}_3 \circ \mathcal{C}$ circuit. We can try to estimate $\mathbb{E}_{x \sim \mathcal{U}_n}[F_i(x \circ T(x))]$ for each $i \in [\ell]$ to distinguish between the above two cases. Note there is only a $1/\ell = 1/\mathrm{poly}(n)$ gap between the above two cases. Therefore, this argument does show that, if we assume to have non-trivial CAPP algorithms for $\mathsf{OR}_3 \circ \mathcal{C}$ with $1/\mathrm{poly}(n)$ error, the above guess-and-verify approach already suffices to obtain lower bounds against $\mathcal{C}$.

However, in our case, we are only assuming to have a $\mathsf{Gap\text{-}UNSAT}$ algorithm with a *constant* gap. It is not clear how to make further progress with the above idea.

A Better Proof System?

The above idea does not work, essentially because the described "proof system" is a pretty *bad* PCP! Given the pair $(x, T(x))$, if the verifier draws a random $i \in [\ell]$ and checks whether the clause F_i is satisfied, it is only promised to detect an error with probability $\geq 1/\ell$ when $C(x) = 0$ and the proof $T(x)$ is incorrect. In other words, it has a completeness/soundness gap of only $1/\ell = 1/\mathrm{poly}(n)$. A natural response to this observation is to try using a better proof system for proving that $C(x) = 1$; it comes as no surprise that we turn to the PCP Theorem [5, 6].

However, there is a subtle issue. In the above proof system, the verifier does not need to know the input x beforehand, and only needs to query a bit of x when verifying a clause F_i containing that bit. The most important property here is that *the verifier's queries do not depend on the input* x, as otherwise we cannot formulate the condition "the verifier accepts with the random index i and proof $T(x)$ on input x" as a simple function $F_i(x \circ T(x))$ which can be represented by an $\mathsf{OR}_3 \circ \mathcal{C}$ circuit.

Suppose we forced the verifier to access the input x using only $O(1)$ queries, as in the above proof system, but the circuit is computing a highly-sensitive function such as the parity of x. There is no way that a verifier querying x for only $O(1)$ times can correctly infer (with high probability) that the parity of x is odd! This is because if the parity of x is odd, the parity will change if we flip a random bit of x, so it is not possible for a verifier to distinguish between these two cases with constant probability, if the verifier can only query x for $O(1)$ times.

Error Correcting Codes and Probabilistic Checkable Proofs of Proximity

To avoid the above trivial counterexample, our next key idea is to provide the PCP verifier an *error-correcting encoding* of the input. Now we are at the right position to introduce the main technical concept used in this paper: *Probabilistic Checkable Proofs of Proximity* (PCPP) for the Circuit-Eval problem. When properly applied, PCPPs allow us to reduce the error requirement on the CAPP algorithms from *inverse polynomial* to only a *constant*.

In this type of proof system[9], a circuit E is fixed in advance, the verifier $V(E)$ gets oracle access to the input x of length n and a proof string π, tosses some random coins, and makes at most 3 *non-adaptive* queries. The proof system has constant parameters $\delta > 0$ and $s \in (0, 1)$, and satisfies two important properties:

- (Perfect Completeness.) $E(x) = 1 \Rightarrow$ there is a π such that $\Pr[V(E) \text{ accepts } x \circ \pi] = 1$.
- (Soundness on inputs far from being correct.) If x is δ-far from the set $\{y : E(y) = 1\}$, where δ is the proximity parameter, then for all possible proofs π, $V(E)$ accepts $x \circ \pi$ with probability at most $s < 1$.

To clarify the second point, we are saying that if x has hamming distance more than δn from all y that satisfy E, then $V(E)$ has decent probability of rejection on *any* proof π.

Suppose we use a linear error correcting code with an efficient encoder Enc and decoder Dec, and define the circuit E by $E(y) := C(\mathsf{Dec}(y))$. That is, E treats its input y as an encoding of an input to the circuit C; it first decodes y to a string z, then feeds z to C to get its output.

Let $x \in \{0, 1\}^n$ be an input to C. We instantiate a PCP of proximity proof system with the circuit E and the input $\mathsf{Enc}(x)$. It is not hard to see that when $C(x) = 0$, $\mathsf{Enc}(x)$ is δ_1-far from the accepting inputs for E for a constant δ_1 depending on the error correcting code. We can ensure that $\delta_1 > \delta$.

The Final Reduction

Now, suppose there are ℓ possible outcomes of the random coins, and assume that the proof π is of length ℓ as well. Let $F_i(\mathsf{Enc}(x) \circ T(x))$ be the indicator that given a random outcome $i \in [\ell]$, whether the verifier $V(E)$ accepts the oracle $\mathsf{Enc}(x) \circ T(x)$. By definition, $F_i(\mathsf{Enc}(x) \circ T(x))$ is a function on 3 coordinates of $\mathsf{Enc}(x) \circ T(x)$ (we can assume WLOG that F_i is simply an OR, by using a special PCP of proximity proof system; see Lemma 24). Note that a bit of $\mathsf{Enc}(x)$ is just a parity over a subset of bits in x. For simplicity, let us further assume $\mathcal{C} = \mathsf{THR} \circ \mathsf{THR}$, which can compute parity (note, this assumption can be removed). Then $F_i(\mathsf{Enc}(x) \circ T(x))$ can now be formulated as an $\mathsf{OR}_3 \circ \mathcal{C}$ circuit. Now we proceed similarly as before.

[9] see Definition 20

- We again try to guess $\mathcal{C}$ circuits $T_1(x), T_1(x), \ldots, T_\ell(x)$, but this time with the hope that $T(x) = (T_1(x), T_2(x), \ldots, T_\ell(x))$ is the correct proof for the verifier $V(E)$ given input $\mathsf{Enc}(x)$.
- When C is a tautology, there is a guess $T(x)$ such that $\mathbb{E}_{x\sim\mathcal{U}_n}\mathbb{E}_{i\in[\ell]}[F_i(\mathsf{Enc}(x)\circ T(x))] = 1$.
- Otherwise, for all guesses $T(x)$, $\mathbb{E}_{x\sim\mathcal{U}_n}\mathbb{E}_{i\in[\ell]}[F_i(\mathsf{Enc}(x)\circ T(x))] \leq 1/n^{10} + s$, since for at least a $1 - 1/n^{10}$ fraction of inputs, we have $C(x) = 0$, and therefore at most an s fraction of F_i's can be satisfied by $\mathsf{Enc}(x) \circ T(x)$, because $\mathsf{Enc}(x)$ is δ-far from any accepting input to E.

In this new situation, it now suffices to estimate $\mathbb{E}_{x\sim\mathcal{U}_n}[F_i(x \circ T(x))]$ for each $i \in [\ell]$ within sufficiently small constant error. A careful examination of the above argument shows it suffices to use a non-trivial Gap-UNSAT algorithm for $\mathsf{AND}_3 \circ \mathcal{C}$ circuits with a constant gap (note that the negation of F_i is an $\mathsf{AND}_3 \circ \mathcal{C}$ circuit), because we have perfect completeness in the case where C is a tautology.

Lower Bounds From CAPP Algorithms for $\mathsf{OR}_2 \circ \mathcal{C}$, $\mathsf{AND}_2 \circ \mathcal{C}$, or $\oplus_2 \circ \mathcal{C}$ Circuits

The above shows how to use a non-trivial CAPP algorithm for $\mathsf{OR}_3 \circ \mathcal{C}$; how can we use a non-trivial CAPP algorithm for $\mathsf{OR}_2 \circ \mathcal{C}$, $\mathsf{AND}_2 \circ \mathcal{C}$, or $\oplus_2 \circ \mathcal{C}$? The natural idea is to instead use a 2-query PCPP for Circuit-Eval. Unfortunately, there is no PCPP with only 2 queries with perfect completeness for Circuit-Eval, unless $\mathsf{P} = \mathsf{NP}$.[10] Thus we must use a construction with imperfect completeness. Luckily, there is a 2-query PCPP for Circuit-Eval with a constant soundness/completeness gap (Lemma 25). We use that PCPP in the above argument, together with other ideas, to establish the connection with a non-trivial CAPP algorithm for $\mathsf{OR}_2 \circ \mathcal{C}$, $\mathsf{AND}_2 \circ \mathcal{C}$ or $\oplus_2 \circ \mathcal{C}$ circuits.

1.7 Related Work

For more history on previous works on lower bounds for constant-depth threshold circuits, see the corresponding sections in [51, 31]. We only discuss a few recent results here.

In 2014, Williams [51] showed that NEXP is not contained in $\mathsf{ACC}^0 \circ \mathsf{THR}$, by devising a fast satisfiability algorithm for $\mathsf{ACC}^0 \circ \mathsf{THR}$. The lower bound was recently improved by Murray and Williams [34] to show $\mathsf{NTIME}[n^{\mathrm{polylog}(n)}]$ is not contained in $\mathsf{ACC}^0 \circ \mathsf{THR}$. Tamaki [44], Alman, Chan and Williams [3] proved that E^{NP} does not have $n^{2-o(1)}$ size $\mathsf{THR} \circ \mathsf{THR}$ circuits. Most recently, Williams [48] showed that there are functions in $\mathsf{NTIME}[n^{\log^{\omega(1)}(n)}]$ that can not be represented by a linear combination of polynomially many $\mathsf{ACC}^0 \circ \mathsf{THR}$ circuits.

Tell [46] constructed a *quantified derandomization* algorithm for TC circuits with depth d and $n^{1+\exp(-d)}$ wires, and showed that a modest improvement of his algorithm would imply standard derandomization of TC^0, and consequently $\mathsf{NEXP} \not\subseteq \mathsf{TC}^0$.

Using random restrictions, Kane and Williams [31] proved that any $\mathsf{THR} \circ \mathsf{THR}$ circuits computing Andreev's function requires $\widetilde{\Omega}(n^{1.5})$ gates and $\widetilde{\Omega}(n^{2.5})$ wires. Chattopadhyay and Mande [16] recently showed an exponential size separation between $\mathsf{THR} \circ \mathsf{MAJ}$ and $\mathsf{THR} \circ \mathsf{THR}$, by constructing a function in $\mathsf{THR} \circ \mathsf{THR}$ with exponential *sign-rank*.

[10] A 2-query PCPP for Circuit-Eval with perfect completeness implies a 2-query PCP for NP with perfect completeness [11], which in turn implies $\mathsf{P} = \mathsf{NP}$, as 2-SAT is in P.

1.8 Organization of the Paper

In Section 2 we discuss necessary preliminaries. In Section 3 we prove equivalences between non-trivial circuit analysis tasks of $\mathsf{THR} \circ \mathsf{THR}$ and that of $\mathsf{THR} \circ \mathsf{MAJ}$ or $\mathsf{MAJ} \circ \mathsf{MAJ}$. In Section 4 we establish a tighter connection between non-trivial derandomization and circuit lower bounds. In Section 5, we propose approaches toward proving $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$. In Section 6 we prove lower bounds for various $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuits. In Section 7 we prove two new structure lemmas for $\mathsf{THR} \circ \mathsf{THR}$ circuits.

2 Preliminaries

The *Circuit Evaluation Problem* (Circuit-Eval) is the language of pairs $\{(C, w)\}$ such that when C is a general fan-in-2 circuit and w is a Boolean input, $(C, w) \in$ Circuit-Eval if and only if $C(w) = 1$. For two strings a, b, we use $a \circ b$ to denote their concatenation[11].

2.1 Circuit Classes

Let $\mathcal{C}$ be a circuit class. We use $\mathcal{C}_n^s$ to denote the set of $\mathcal{C}$ of circuits with n inputs and size at most s. Slightly abusing notation, we also use $\mathcal{C}_n^s$ to denote the *Boolean functions* computed by circuits in $\mathcal{C}_n^s$, when convenient.

We say a circuit class $\mathcal{C}$ is *typical*, if given the description of a circuit C from $\mathcal{C}_n^s$, for all indices $1 \leq i, j \leq n$ and $b \in \{0, 1\}$, the following functions are in $\mathcal{C}_n^s$:

$$\neg C,\ C(x_1, \ldots, x_{i-1}, x_j \oplus b, x_{i+1}, \ldots, x_n),\ C(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n).$$

Furthermore, we require that given a description of C, descriptions of all the above circuits can be constructed in $\mathrm{poly}(s)$ time. That is, $\mathcal{C}$ is typical if it is closed under both efficiently-computable *negation* and efficiently-computable *projection*.

Notations for Circuit Classes

As many circuit classes are discussed in this work, we begin with some notation for such classes.

Let $x \in \{0, 1\}^n$. For $w \in \mathbb{R}^n$ and $t \in \mathbb{R}$, we define $\mathsf{THR}_{w,t}(x)$ (the threshold function) to be the indicator function for the condition $w \cdot x \geq t$. Similarly, $\mathsf{ETHR}_{w,t}(x)$ (the exact threshold function) is the indicator function for the condition $w \cdot x = t$. The values in the vector w are called the *weights*, and the real t is called the *threshold* of $\mathsf{THR}_{w,t}$ and $\mathsf{ETHR}_{w,t}$. We say these weights and thresholds are *realizations* of the Boolean functions they define. A fixed Boolean function may have many different realizations. It is known that, without loss of generality, the weights and thresholds are integers of absolute value at most $2^{O(n \log n)}$ [33, 7]. For a threshold or exact threshold function with weight w, we call $L(x) := w \cdot x$ its *associated linear function*.

We use MAJ_n and EMAJ_n to denote the corresponding threshold (exact threshold) functions on n inputs where all weights are 1 and the threshold value is $n/2$. Slightly abusing notation, we also use $\mathsf{THR}, \mathsf{ETHR}, \mathsf{MAJ}, \mathsf{EMAJ}$ to denote the classes of all such functions. We also consider $\oplus_k$ (PARITY), AND_n, and OR_n, with their usual meanings. We use DOR_n to denote the disjoint OR function, that is, an OR function with the promise that at most

[11] Note that we also use $\circ$ for the composition of two circuits, but throughout the paper the meaning of the symbol $\circ$ will always be clear from the context

one input bit is true over all inputs. We use $\mathsf{Gap\text{-}OR}_{n,\delta}$ to denote the gap-OR function on n inputs, that is, an OR function with the promise that either all n inputs are false, or at least a $1-\delta$ fraction of the n inputs are true. (The function may have undefined behavior on other inputs.)

For two classes of functions like THR and MAJ, we use $\mathsf{THR} \circ \mathsf{MAJ}$ to denote the corresponding class of depth-two circuits. Similar notations are used for more than two classes.

We use $\mathsf{AC}^0[m]_d$ to denote depth-d $\mathsf{AC}^0[m]$ circuits (with unbounded fan-in OR, AND, and MOD_m gates). We use LT_d to denote the depth-d THR circuit class, that is, $\mathsf{LT}_d := \underbrace{\mathsf{THR} \circ \ldots \circ \mathsf{THR}}_{d \text{ times}}$. Similarly, we use $\widehat{\mathsf{LT}}_d$ to denote its unweighted version, that is, $\widehat{\mathsf{LT}}_d := \underbrace{\mathsf{MAJ} \circ \ldots \circ \mathsf{MAJ}}_{d \text{ times}}$.

Previous Known Containment Results

We need the following known circuit classes containment results for this paper.

▶ **Proposition 18.** *The following hold:*

1. $\mathsf{THR} \subseteq \mathsf{MAJ} \circ \mathsf{MAJ}$ *[21, 27].*
2. $\mathsf{THR} \subseteq \mathsf{DOR} \circ \mathsf{ETHR}$ *[24]. (also see Appendix B)*
3. $\mathsf{MAJ} \circ \mathsf{THR}$ *and* $\mathsf{MAJ} \circ \mathsf{ETHR}$ *are contained in* $\mathsf{MAJ} \circ \mathsf{MAJ}$ *[21, 24].*
4. $\mathsf{ETHR} \circ \mathsf{ETHR} \subseteq \mathsf{THR} \circ \mathsf{THR}$ *[24].*
5. $\mathsf{AND} \circ \mathsf{ETHR} \subseteq \mathsf{ETHR}$ *[24].*
6. $\mathsf{EMAJ} \subseteq \mathsf{MAJ} \circ \mathsf{AND}_2$ *[24].*
7. $\oplus_k \circ \mathsf{THR} \circ \mathsf{THR} \subseteq \mathsf{THR} \circ \mathsf{THR}$ *for a constant* k*. (see Appendix B)*
8. $\mathsf{THR} \circ \mathsf{EMAJ} \subseteq \mathsf{THR} \circ \mathsf{MAJ}$ *[24].*

Moreover, all the above have corresponding polynomial-time, deterministic constructions.

For the containment $\mathsf{THR} \subseteq \mathsf{DOR} \circ \mathsf{ETHR}$, we present an alternative proof in Appendix B, which is more efficient than the previously known construction of Hansen and Podolskii [24].[12] The last containment is folklore; we present a proof in Appendix B for completeness.

2.2 Approximation Theory

We need the following standard result from approximation theory.

▶ **Lemma 19** ([39] Corollary.1.4.1)**.** *Let* $0 < \varepsilon_1 < \varepsilon_2 < 1/2$ *be two constants, there is an* $O_{\varepsilon_1,\varepsilon_2}(1)$ *degree polynomial* $P : \mathbb{R} \to \mathbb{R}$*, such that:*

- *for all* $z \in [-\varepsilon_2, \varepsilon_2]$*,* $P(z) \in [-\varepsilon_1, \varepsilon_1]$*, and*
- *for all* $z \in [1-\varepsilon_2, 1+\varepsilon_2]$*,* $P(z) \in [1-\varepsilon_1, 1+\varepsilon_1]$*.*

2.3 Probabilistic Checkable Proofs of Proximity

The concept of probabilistically checkable proofs of proximity is crucial for this paper. In the following we introduce its definition and several instantiations useful for this paper.

[12] Hansen and Podolskii [24] proved that a THR gate on n bits with weights of absolute value no greater than W, can be written as a DOR of $O(n^2 \cdot \log W)$ many ETHR gates. In Appendix B we show it can be improved to $O(n \cdot \log W)$ many ETHR gates.

▶ **Definition 20** (Probabilistic Checkable Proofs of Proximity (PCP of proximity, or PCPP)). *For $s, \delta : \mathbb{N} \to [0,1]$ and $r, q : \mathbb{N} \to \mathbb{N}$, a verifier V is a PCP of proximity system for a pair language L with proximity parameter δ, soundness parameter s, number of random bits r and query complexity q if the following holds for all x, y:*

- *If $(x, y) \in L$, there is a proof π such that $V(x)$ accepts oracle $y \circ \pi$ with probability 1.*
- *If y is $\delta(|x|)$-far from $L(x) := \{z : (x, z) \in L\}$, then for all proofs π, $V(x)$ accepts oracle $y \circ \pi$ with probability at most $s(|x|)$.*
- *$V(x)$ tosses $r(|x|)$ random coins, and makes at most $q(|x|)$ non-adaptive queries.*

▶ Remark 21. We can also relax the first condition to be that there is a proof π such that $V(x)$ accepts oracle $y \circ \pi$ with probability at least $c = c(|x|)$, where c is the completeness parameter. In the above definition we assume $c = 1$, i.e., the perfect completeness.

▶ **Lemma 22** (Theorem 3.3 in [11]). *For any constants $0 < \delta, s < 1$, there is a PCP of proximity system for Circuit-Eval with proximity δ, soundness s, number of random bits $r = O(\log n)$ and query complexity $q = O(1)$. Moreover, given the pair $(C, w) \in$ Circuit-Eval, a proof π making $V(C)$ always accepts can be constructed in $\mathrm{poly}(|C| + |w|)$ time.*

▶ Remark 23. The moreover part is not explicitly stated in [11], but it is evident from the constructions.

The exact number of queries used in a PCPP will be significant for us, so we use query-efficient PCPPs. They are already implicit in the literature; for completeness, we provide expositions for them in Appendix A.

▶ **Lemma 24** (3-query PCPP with perfect completeness). *For any constant $\delta > 0$ there is a constant $0 < s < 1$, such that there is a PCP of proximity system for Circuit-Eval with proximity δ, soundness s, random bits $r = O(\log n)$, and query complexity $q = 3$. Moreover, the system satisfies two additional properties:*

(1) *Given the random coins, the verifier simply computes an OR on these 3 queried bits or their negations, and accepts iff the OR is true.*

(2) *Given the pair $(C, w) \in$ Circuit-Eval, we can construct a proof π in $\mathrm{poly}(|C| + |w|)$ time that makes $V(C)$ accept with probability 1.*

▶ **Lemma 25** (2-query PCPP with constant completeness/soundness gap). *For any constant $\delta > 0$ there two constants $0 < s < c < 1$, such that there is a PCP of proximity system for Circuit-Eval with proximity δ, soundness s, completeness c, number of random bits $r = O(\log n)$ and query complexity $q = 2$. Moreover, it satisfies two additional properties:*

(1) *Given the random coins, the verifier computes an OR on the 2 queried bits or their negations, and accepts iff the OR is true.*

(2) *Given the pair $(C, w) \in$ Circuit-Eval, a proof π can be constructed in $\mathrm{poly}(|C| + |w|)$ time that makes $V(C)$ accept with probability at least c.*

2.4 Error Correcting Codes

We also need standard constructions of constant-rate linear error correcting codes.

▶ **Lemma 26** ([43]). *There is a constant $\delta > 0$ such that there is a constant-rate linear error correcting code ECC with minimum relative distance δ, an efficient encoder Enc and an efficient decoder Dec recovering error up to $c_1 \cdot \delta$, where c_1 is a universal constant.*

We use a slight modification of the above construction, which is convenient when we want to guess-and-verify a circuit for the encoder.

▶ **Lemma 27.** *There is a constant $\delta > 0$ such that there is a constant-rate linear error correcting code ECC with minimum relative distance δ, an efficient encoder Enc and an efficient decoder Dec recovering error up to $c_1 \cdot \delta$, where c_1 is a universal constant. Moreover, each bit of the codeword depends on at most $n/2$ bits of the input.*

Proof. Given a message $x \in \{0,1\}^n$, we split it into three parts x_1, x_2, x_3, each of length between $\lfloor n/3 \rfloor$ and $\lceil n/3 \rceil$. Let Enc' and Dec' be the corresponding encoder and decoder of Lemma 26.

We construct our new error correcting code by setting $\mathsf{Enc}(x) := \mathsf{Enc}'(x_1) \circ \mathsf{Enc}'(x_2) \circ \mathsf{Enc}'(x_3)$. Given a codeword y, we split it into three strings y_1, y_2, y_3 of appropriate lengths, and let $\mathsf{Dec}(y) := \mathsf{Dec}'(y_1) \circ \mathsf{Dec}'(y_2) \circ \mathsf{Dec}'(y_3)$. ◀

2.5 Norms and Inequalities for Functions on Boolean Cube

For our lower bounds on approximate sums of functions, we will require a bit of Fourier analysis on Boolean functions. Here we introduce some notations and inequalities for real-valued functions on the Boolean hypercube. (See [36] for an excellent reference on this topic.)

Let $f : \{0,1\}^n \to \mathbb{R}$ be a function and $p \in \mathbb{R}^+$. We define

$$\|f\|_p := \left(\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n} [|f(x)|^p] \right)^{1/p}.$$

We also define the infinity norm in the usual way:

$$\|f\|_\infty = \max_{x \in \{0,1\}^n} |f(x)|.$$

By the standard relations between different L_p-norms, for all $0 < p < q \leq \infty$, we have $\|f\|_p \leq \|f\|_q$.

For two functions $f, g : \{0,1\}^n \to \mathbb{R}$, we define their inner product as

$$\langle f, g \rangle := \mathop{\mathbb{E}}_{x \sim \mathcal{U}_n} [f(x) \cdot g(x)].$$

Note that the Cauchy-Schwarz inequality implies $\langle f, g \rangle \leq \|f\|_2 \cdot \|g\|_2$. We need the following simple lemma for this paper.

▶ **Lemma 28.** *For functions f_1, f_2 and g_1, g_2 from $\{0,1\}^n \to \mathbb{R}$ and positive $\varepsilon, \alpha \in \mathbb{R}$, suppose for all $i \in [2]$ we have:*

- *$\|f_i\|_2 \leq \alpha$ and $\|g_i\|_2 \leq \alpha$,*
- *$\|f_i - g_i\|_2 \leq \varepsilon$.*

Then $\langle f_1, f_2 \rangle - \langle g_1, g_2 \rangle \| \leq 2 \cdot \alpha \cdot \varepsilon$.

Proof. We have

$$\begin{aligned}\|\langle f_1, f_2\rangle - \langle g_1, g_2\rangle\| &\leq \|\langle f_1, f_2\rangle - \langle f_1, g_2\rangle\| + \|\langle f_1, g_2\rangle - \langle g_1, g_2\rangle\| \\ &\leq \|\langle f_1, f_2 - g_2\rangle\| + \|\langle f_1 - g_1, g_2\rangle\| \\ &\leq 2 \cdot \alpha \cdot \varepsilon.\end{aligned}$$

◀

2.6 Connections Between Nondeterministic Gap-UNSAT Algorithms and Circuit Lower Bounds

We also appeal to several known connections between Gap-UNSAT algorithms which improve upon exhaustive search and circuit lower bounds against nondeterministic time classes [49, 30, 41, 13].

▶ **Theorem 29** ([49]). *If Gap-UNSAT with gap $1 - 1/n^{10}$ for (general) circuits with n inputs and $\mathrm{poly}(n)$ size is solvable in $O(2^n/n^{\omega(1)})$ nondeterministic time, then NEXP doesn't have $\mathrm{poly}(n)$-size (general) circuits.*

▶ **Theorem 30** ([34]). *If there is an $\varepsilon > 0$ such that Gap-UNSAT with gap $1 - 1/n^{10}$ for (general) circuits with n inputs and $2^{\varepsilon n}$ size is solvable in $O(2^{n-\varepsilon n})$ nondeterministic time, then for every k there is a function in NP that does not have n^k-size (general) circuits.*

▶ **Theorem 31** (Corollary 12 in Tell [45], following [34]). *If there is a $\delta > 0$ and $c \geq 1$ such that Gap-UNSAT with gap $1 - 1/n^{10}$ for (general) circuits with n variables and m gates is solvable in $O(2^{n(1-\delta)} \cdot m^c)$ nondeterministic time, then for every unbounded $\alpha(n)$ such that $n^{\alpha(n)}$ is time-constructible, there is a function in $\mathsf{NTIME}[n^{\alpha(n)}]$ that is not in P/poly.*

▶ **Theorem 32** ([34]). *If there is an $\varepsilon > 0$ such that Gap-UNSAT with gap $1 - 1/n^{10}$ for (general) circuits with n inputs and 2^{n^ε} size is solvable in $O(2^{n-n^\varepsilon})$ nondeterministic time, then for every k there is a function in $\mathsf{NTIME}[n^{\mathrm{poly}(\log n)}]$ that does not have $n^{\log^k n}$-size (general) circuits.*

3 Equivalence Between Algorithmic Analysis of THR ∘ THR and of THR ∘ MAJ or MAJ ∘ MAJ

In this section, building on our new structure lemmas for THR∘THR circuits. We show several equivalence results between canonical circuit-analysis tasks (SAT or CAPP) of THR ∘ THR circuits and that of THR ∘ MAJ or MAJ ∘ MAJ circuits.

3.1 Poly-Size THR ∘ MAJ and THR ∘ THR are Equivalent for Circuit-Analysis Algorithms

We first show that, in terms of designing non-trivial circuit-analysis algorithms, THR ∘ THR and THR ∘ MAJ circuits are essentially *equivalent*.

▶ **Reminder of Theorem 1.** *The following two statements hold:*

- ***Equivalence of Non-Trivial SAT Algorithms**: There is a non-trivial SAT algorithm for THR ∘ MAJ circuits of $\mathrm{poly}(n)$-size* if and only if *there is such an algorithm for $\mathrm{poly}(n)$-size THR ∘ THR circuits.*
- ***Equivalence of Non-Trivial CAPP Algorithms With Constant Error**: For any constant $\delta > 0$, If there is a non-trivial CAPP algorithm with error δ for THR ∘ MAJ circuits of $\mathrm{poly}(n)$ size, then there is a non-trivial CAPP algorithm with error $\delta + 1/n$ for $\mathrm{poly}(n)$-size THR ∘ THR circuits.*

Proof. We begin with the first equivalence. We only have to show that a $2^n/n^{\omega(1)}$ time SAT algorithm for poly(n)-size THR $\circ$ MAJ circuits implies such an algorithm for THR $\circ$ THR circuits. By Lemma 16, given any THR $\circ$ THR circuit of poly(n) size, in poly(n) time we can construct an equivalent poly(n)-size Gap-OR $\circ$ THR $\circ$ MAJ circuit C. Applying the assumed SAT algorithm for THR $\circ$ MAJ circuits on all THR $\circ$ MAJ subcircuits of C completes the proof of the first equivalence.

For the second equivalence, given any THR $\circ$ THR circuit C of poly(n) size, we construct in poly(n) time a Gap-OR$_{1/n}$ $\circ$ THR $\circ$ MAJ circuit D that is equivalent to C, by Lemma 16. Let $D_1, D_2, \ldots, D_m$ be the THR $\circ$ MAJ subcircuits of C, where $m = \text{poly}(n)$.

By the definition of a Gap-OR$_{1/n}$ gate, for all $x \in \{0,1\}^n$, we have

$$\left| C(x) - \mathop{\mathbb{E}}_{i \in [m]} D_i(x) \right| \leq 1/n.$$

Therefore, to estimate $\mathbb{E}_{x \sim \mathcal{U}_n}[C(x)]$ within error $\delta + 1/n$, it suffices to estimate $\mathbb{E}_{x \sim \mathcal{U}_n}[D_i(x)]$ for each $i \in [m]$ within error δ. Applying the non-trivial CAPP algorithm for THR $\circ$ MAJ circuits from the assumption completes the proof. ◀

With an argument similar to the proof of Theorem 1 and using the fact that MAJ $\circ$ THR $\subseteq$ MAJ $\circ$ MAJ (potentially multiple times), it is not hard to generalize Theorem 1 to hold for TC circuits of any constant depth d.

▶ **Reminder of Corollary 2.** *The following two statements hold for any constant d:*

- ***Equivalence of Non-Trivial SAT Algorithms**: There is a non-trivial SAT algorithm for THR $\circ$ $\widehat{\text{LT}}_{d-1}$ circuits of* poly(n)*-size* if and only if *there is such an algorithm for* poly(n)*-size* LT_d *circuits.*
- ***Equivalence of Non-Trivial CAPP Algorithms With Constant Error**: For any constant $\varepsilon > 0$, if there is a non-trivial CAPP algorithm with error ε for THR $\circ$ $\widehat{\text{LT}}_{d-1}$ circuits of* poly(n)*-size, then there is a non-trivial CAPP algorithm with error $\varepsilon + 1/n$ for* poly(n)*-size* LT_d *circuits.*

3.2 Weaker Equivalence Between Poly-Size THR $\circ$ THR and MAJ $\circ$ MAJ

We also show a weaker equivalence for THR $\circ$ THR and MAJ $\circ$ MAJ circuits.

▶ **Reminder of Theorem 3.** *The following two statements hold:*

- ***Equivalence of $2^{(1-\varepsilon)n}$-time SAT Algorithms**: If SAT for* poly(n)*-size MAJ $\circ$ MAJ circuits is in $2^{(1-\varepsilon)n}$ time for some constant $\varepsilon > 0$, then SAT for* poly(n)*-size THR $\circ$ THR circuits is in $2^{(1-\varepsilon')n}$ time for some $\varepsilon' > 0$.*
- ***Equivalence of Non-Trivial CAPP Algorithms with Inverse Polynomial Error**: If there is a non-trivial CAPP algorithm with $1/poly(n)$ error for* poly(n)*-size MAJ $\circ$ MAJ circuits, then there is a non-trivial CAPP algorithm with* $1/\text{poly}(n)$ *error for* poly(n)*-size THR $\circ$ THR circuits.*

Proof. We begin with the first equivalence.

The first equivalence. Suppose we have a $2^{(1-\varepsilon_1)n}$ time SAT algorithm for poly(n) size MAJ $\circ$ MAJ circuits for a constant $\varepsilon_1 > 0$, and want to design a $2^{n-\Omega(n)}$ time SAT algorithm for poly(n) size THR $\circ$ THR circuits.

Let c be the hidden constant in the big-O of the fan-in of the top DOR gate from Lemma 17. Set $\varepsilon := \varepsilon_1/2c$, and apply Lemma 17 to the given poly(n)-size THR $\circ$ THR

circuit. We obtain an equivalent $\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{MAJ}$ circuit with top fan-in $2^{c\varepsilon n} = 2^{\varepsilon_1/2 \cdot n}$ and $\mathrm{poly}(n)$-size $\mathsf{MAJ} \circ \mathsf{MAJ}$ subcircuits. Then we can apply our SAT algorithm for $\mathrm{poly}(n)$-size $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits to solve the SAT problem for $\mathrm{poly}(n)$ size $\mathsf{THR} \circ \mathsf{THR}$ circuits, which completes the proof of the first equivalence.

The second equivalence. To show the second equivalence, suppose for all constants k', there is a CAPP algorithm for $\mathrm{poly}(n)$-size $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits with error $1/n^{k'}$. We have to design such an algorithm for $\mathrm{poly}(n)$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits.

Given a $\mathsf{THR} \circ \mathsf{THR}$ circuit C of $s = \mathrm{poly}(n) \leq n^{c_1}$ size and a constant k, we want to estimate

$$\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}[C(x)] \tag{2}$$

within error $1/n^k$.

Since $\mathsf{THR} \subseteq \mathsf{DOR} \circ \mathsf{ETHR}$ (item (2) of Proposition 18), we can write C as a DOR of $m = \mathrm{poly}(s) = \mathrm{poly}(n)$ $\mathsf{ETHR} \circ \mathsf{THR}$ subcircuits $C_1, C_2, \ldots, C_m$. By the definition of DOR, we have

$$\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}[C(x)] = \mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}\left[\sum_{i=1}^{m} C_i(x)\right] = \sum_{i=1}^{m} \mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}[C_i(x)].$$

Therefore, in order to estimate (2) within error $1/n^k$, it suffices to estimate

$$\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}[C_i(x)]$$

within error $1/(m \cdot n^k)$ for each $i \in [m]$.

So fix an $i \in [m]$. Let $D = C_i$, and let D's top ETHR gate be G. By construction, G has weights of absolute value at most 2^{n^c}, for a constant c depending on c_1. Define $L : \{0,1\}^n \to \mathbb{Z}$ so that $L(x)$ is the value of the linear function associated with G on input x. That is, $D(x) = 1$ if and only if $L(x) = T$ for the threshold T of G.

Suppose we pick a random prime number p in the interval $[2, M]$, where $M = n^{2c} \cdot (2m \cdot n^k)^2 \leq \mathrm{poly}(n)$. Then for a fixed $x \in \{0,1\}^n$, if $L(x) \neq T$, the probability that $L(x) \equiv T \pmod{p}$ is less than $1/(2m \cdot n^k)$.

Recall that for a prime p and an ETHR gate $G(x) = [\sum_{i=1}^{n} w_i \cdot x_i = T]$, we use G^p to denote its "mod p" version (see Definition 41). Let D^p denote the circuit obtained by replacing the top G gate in D by G^p. For all $x \in \{0,1\}^n$, by the above discussion, we have

$$\left| D(x) - \mathop{\mathbb{E}}_{\text{prime } p \in [2,\, M]}[D^p(x)] \right| \leq 1/(2m \cdot n^k).$$

Therefore, in order to estimate $\mathbb{E}_{x \sim \mathcal{U}_n}[D(x)]$ within error $1/(m \cdot n^k)$, it suffices to estimate

$$\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}[D^p(x)]$$

for all primes $p \leq M$, within error $1/(2m \cdot n^k)$.

By Lemma 42, each D^p can be written as a DOR of $O(n)$ $\mathsf{EMAJ} \circ \mathsf{ETHR}$ circuits of $\mathrm{poly}(n)$ size. Since $\mathsf{EMAJ} \subseteq \mathsf{MAJ} \circ \mathsf{AND}_2$, $\mathsf{AND} \circ \mathsf{ETHR} \subseteq \mathsf{ETHR}$ and $\mathsf{MAJ} \circ \mathsf{ETHR} \subseteq \mathsf{MAJ} \circ \mathsf{MAJ}$ (items (6), (5), and (3) of Proposition 18), D^p can be further written as a DOR of cn $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuits $D_1^p, D_2^p, \ldots, D_{cn}^p$ of $\mathrm{poly}(n)$ size, for a universal constant c.

Therefore, to estimate $\mathbb{E}_{x\sim\mathcal{U}_n}[D^p(x)]$ within error $1/(2m\cdot n^k)$, it suffices to estimate $\mathbb{E}_{x\sim\mathcal{U}_n}[D_i^p(x)]$ within error $1/(2m\cdot n^k\cdot cn)$, for each $i\in[cn]$.
Observe that $2m\cdot n^k\cdot cn\le \mathrm{poly}(n)$, and all D_i^p's are $\mathrm{poly}(n)$-size $\mathsf{MAJ}\circ\mathsf{MAJ}$ circuits. Applying the assumed CAPP algorithm completes the proof of the second equivalence. ◀

Again applying the fact that $\mathsf{MAJ}\circ\mathsf{THR}\subseteq\mathsf{MAJ}\circ\mathsf{MAJ}$, the generalization to TC circuits of any constant depth d is immediate.

▶ **Reminder of Corollary 4.** *The following two statements hold for any constant d:*

- ***Equivalence of $2^{(1-\varepsilon)n}$-time SAT Algorithms**: If SAT for $\widehat{\mathsf{LT}}_d$ circuits of $\mathrm{poly}(n)$-size is in $2^{(1-\varepsilon)n}$ time for a constant $\varepsilon>0$, then SAT for $\mathrm{poly}(n)$-size LT_d circuits is in $2^{(1-\varepsilon')n}$ time for some $\varepsilon'>0$.*
- ***Equivalence of Non-trivial CAPP Algorithms with inverse polynomial error**: If there is a non-trivial CAPP algorithm with $1/poly(n)$ error for $\mathrm{poly}(n)$-size $\widehat{\mathsf{LT}}_d$ circuits, then there is a non-trivial CAPP algorithm with $1/\mathrm{poly}(n)$ error for $\mathrm{poly}(n)$-size LT_d circuits.*

4 Tighter Connection Between Derandomization and Circuit Lower Bounds

In this section we show that $\mathcal{C}$ circuit lower bounds for NEXP or NP follow from better-than-2^n time derandomization of $\mathsf{AND}_3\circ\mathcal{C}$, $\mathsf{OR}_2\circ\mathcal{C}$, $\oplus_2\circ\mathcal{C}$ or $\mathsf{AND}_2\circ\mathcal{C}$ circuits.

▶ **Reminder of Theorem 5.** *There is an absolute constant $\delta>0$, such that for any typical circuit class $\mathcal{C}$, if one of the following holds:*

- *there is a non-trivial Gap-UNSAT algorithm with gap δ for $\mathrm{poly}(n)$-size $\mathsf{AND}_3\circ\mathcal{C}$ circuits, or*
- *there is a non-trivial CAPP algorithm with error δ for $\mathrm{poly}(n)$-size $\mathsf{OR}_2\circ\mathcal{C}$, $\oplus_2\circ\mathcal{C}$, or $\mathsf{AND}_2\circ\mathcal{C}$ circuits,*

then $\mathsf{NEXP}\not\subset\mathcal{C}$. Moreover, in the second bullet, $\mathcal{C}$ does not need to be closed under negation.

Proof. We use $\mathcal{U}_n$ to denote the uniform distribution on $\{0,1\}^n$.

We will show there is an absolute constant $\delta>0$, such that if one of the algorithmic assumptions of the theorem holds and $\mathsf{NEXP}\subset\mathcal{C}$, then $\mathsf{Gap\text{-}UNSAT}$ with gap $1-1/n^{10}$ for $\mathrm{poly}(n)$-size general circuits can be solved in $2^n/n^{\omega(1)}$ non-deterministic time. This proves the theorem, since by Theorem 29, we have $\mathsf{NEXP}\not\subset\mathsf{P}_{/\mathrm{poly}}$, which is a contradiction to $\mathsf{NEXP}\subset\mathcal{C}$.

We are given a $\mathrm{poly}(n)$-size general circuit $C:\{0,1\}^n\to\{0,1\}$ with the promise that either C is unsatisfiable, or C has at least $(1-1/n^{10})\cdot 2^n$ satisfying assignments. Our goal is to distinguish between these two cases in $2^n/n^{\omega(1)}$ non-deterministic time.

Let $\delta_1>0$ be the constant of Lemma 27. We fix a constant-rate linear error correcting code with minimum relative distance δ_1, as guaranteed by Lemma 27. Let $\mathsf{Enc}:\{0,1\}^n\to\{0,1\}^{cn}$ and $\mathsf{Dec}:\{0,1\}^{cn}\to\{0,1\}^n$ be the corresponding encoder and decoder, where $c\ge 1$ is a constant corresponding to the rate of the code. Let $\delta_{\mathsf{Dec}}=c_1\cdot\delta_1$, which is error rate that Dec can recover.

We also need a $\mathcal{C}$ circuit for the parity function on $n/2$ bits for computing Enc (by Lemma 27, the code is linear, and each output bit depends on at most $n/2$ input bits). By the assumption $\mathsf{NEXP}\subset\mathcal{C}$, the parity function must have a $\mathcal{C}$-circuit of $\mathrm{poly}(n)$ size. We can guess a $\mathcal{C}$-circuit $\mathsf{Par}_{n/2}$, and brute-force verify that it is correct in $2^{n/2}\cdot\mathrm{poly}(n)$ time.

Let $D : \{0,1\}^{cn} \to \{0,1\}$ be the circuit defined as $D(y) = \neg C(\mathsf{Dec}(y))$. Since C has $\mathrm{poly}(n)$ size and Dec is efficient, D also has $\mathrm{poly}(n)$ size. Then we can see

$$\Pr_{x\sim\mathcal{U}_n}[C(x)=0] = \Pr_{x\sim\mathcal{U}_n}[D(\mathsf{Enc}(x))=1] = \Pr_{x\sim\mathcal{U}_n}[(D,\mathsf{Enc}(x)) \in \mathsf{Circuit\text{-}Eval}].$$

With non-trivial Gap-UNSAT algorithms for poly-size $\mathsf{AND}_3 \circ \mathcal{C}$ circuits. We first prove the theorem under the first assumption. For that purpose we make use of a PCP of proximity system V for $\mathsf{Circuit\text{-}Eval}$, with $\delta_{\mathsf{PCPP}} < \delta_{\mathsf{Dec}}$, $r = O(\log n)$, $q = 3$ and a constant $s < 1$, whose existence is guaranteed by Lemma 24. We fix the circuit to be D, and write the verifier as $V(D)$.

We can view the verification of $V(D)$ as $m = 2^{r(|D|)} = \mathrm{poly}(n)$ many constraints on the oracle $y \circ \pi$. We can also assume $|\pi| = \ell = \mathrm{poly}(n)$. Suppose there are $F_1, F_2, \ldots, F_m$ constraints on $y \circ \pi$, each constraint is an OR on $q = 3$ variables or their negations.

Then the properties of PCP of proximity system translate to:

- If $y = \mathsf{Enc}(x)$ such that $C(x) = 0$, then $D(y) = 1$ and there is a proof $\pi \in \{0,1\}^\ell$ such that all constraints F_i's are satisfied by $y \circ \pi$.
- If $y = \mathsf{Enc}(x)$ such that $C(x) = 1$, then for all $z \in \{0,1\}^{cn}$ with $\mathsf{dist}(z,y) \le \delta_{\mathsf{Dec}}$, we have $D(z) = C(\mathsf{Dec}(z)) = C(x) = 0$. Therefore, y is δ_{Dec}-far from $\mathsf{Circuit\text{-}Eval}(D) = \{z : z \in \{0,1\}^{cn} \text{ and } (D,z) \in \mathsf{Circuit\text{-}Eval}\}$. Since $\delta_{\mathsf{Dec}} > \delta_{\mathsf{PCPP}}$, we have that for all proofs $\pi \in \{0,1\}^\ell$, at most a s fraction of constraints F_i's are satisfied by $y \circ \pi$.

When C is unsatisfiable, then there is a proof $\pi(x)$ for each $y = \mathsf{Enc}(x)$, such that $V(D)$ accepts $y \circ \pi(x)$ with probability 1. Note that by Lemma 24, such a proof $\pi(x)$ can be computed in polynomial time from y and D, which in particular means that $\pi(x)$ admits a polynomial-size circuit, hence each bit of $\pi(x)$ admits a $n_{\mathsf{proof}} = \mathrm{poly}(n)$ size $\mathcal{C}$ circuit (here we use the assumption that $\mathsf{NEXP} \subset \mathcal{C}$).

Next, we guess a list of n_{proof}-size $\mathcal{C}$ circuits $T_1, T_2, \ldots, T_\ell$ such that

$$T(x) = (T_1(x), T_2(x), \ldots, T_\ell(x))$$

is intended to be the proof $\pi(x)$ for $y = \mathsf{Enc}(x)$. Slightly abusing notation, we also use F_i to denote the function $F_i(x) := F_i(\mathsf{Enc}(x) \circ T(x))$. Since a bit of $\mathsf{Enc}(x)$ is just a parity on at most $n/2$ bits in x, and since $\mathcal{C}$ is typical, each F_i can be written as an $\mathsf{OR}_3 \circ \mathcal{C}$ circuit. We also set $E_i(x) = \neg F_i(x)$, which is an $\mathsf{AND}_3 \circ \mathcal{C}$ circuit.

Therefore, when C is unsatisfiable, by the previous discussion, on some guesses of the T_i's, we have

$$\Pr_{x\sim\mathcal{U}_n}[V(D)^{\mathsf{Enc}(x)\circ T(x)} = 1] = \mathop{\mathbb{E}}_{x\sim\mathcal{U}_n} \mathop{\mathbb{E}}_{i\in[m]}[F_i(x)] = 1.$$

Therefore, for all $i \in [m]$,

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[E_i(x)] = 0.$$

When C has at most $2^n/n^{10}$ unsatisfying assignments, for all possible $T_1, T_2, \ldots, T_\ell$, we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n} \mathop{\mathbb{E}}_{i\in[m]}[F_i(x)] \le 1/n^{10} + s.$$

By an averaging argument, there must be an i such that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[F_i(x)] \le 1/n^{10} + s,$$

or equivalently

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[E_i(x)] \geq 1 - s - 1/n^{10} \geq \frac{1-s}{2}.$$

Next, we set $\delta = \frac{1-s}{2}$. When C is unsatisfiable, all E_i's are unsatisfiable on the correct guesses. When C has at most $1/n^{10} \cdot 2^n$ unsatisfying assignments, then for all guesses, there is at least one i such that E_i has at least $\delta \cdot 2^n$ satisfying assignments. Hence, solving Gap-UNSAT with gap δ for all E_i's suffices to non-deterministically distinguish between the two cases. By the first assumption, that takes $2^n/n^{\omega(1)}$ time, and the theorem follows from Theorem 29.

With non-trivial CAPP algorithms for $\mathrm{poly}(n)$-size $\mathsf{AND}_2 \circ \mathcal{C}$, $\oplus_2 \circ \mathcal{C}$ or $\mathsf{OR}_2 \circ \mathcal{C}$ circuits. The theorem under the second assumption can be proved similarly if we use the 2-query PCP of proximity system for Circuit-Eval instead, which is given by Lemma 25. The proof here is similar in parts to the one we just described for $\mathsf{AND}_3 \circ \mathcal{C}$; for completeness we will give the proof in full.

Now we make use of a PCP of proximity system V for Circuit-Eval, with $\delta_{\mathsf{PCPP}} < \delta_{\mathsf{Dec}}$, $r = O(\log n)$, $q = 2$ and constants $0 < s < c < 1$, whose existence is guaranteed by Lemma 25. We again fix the circuit to be D, and write the verifier as $V(D)$.

Similarly, we can view the verification of $V(D)$ as $m = 2^{r(|D|)} \leq \mathrm{poly}(n)$ many constraints on the oracle $y \circ \pi$. We can also assume $|\pi| = \ell \leq \mathrm{poly}(n)$. Suppose there are $F_1, F_2, \ldots, F_m$ constraints on $y \circ \pi$, where each constraint is a function on $q = 2$ coordinates of $y \circ \pi$.

Then the properties of PCP of proximity system translate to:

- If $y = \mathsf{Enc}(x)$ such that $C(x) = 0$, then $D(y) = 1$ and there is a proof $\pi \in \{0,1\}^\ell$ such that at least a c-fraction of F_i's are satisfied by $y \circ \pi$.
- If $y = \mathsf{Enc}(x)$ such that $C(x) = 1$, then for all $z \in \{0,1\}^{cn}$ with $\mathsf{dist}(z, y) \leq \delta_{\mathsf{Dec}}$, we have $D(z) = C(\mathsf{Dec}(z)) = C(x) = 0$. Therefore, y is δ_{Dec}-far from Circuit-Eval(D). Since $\delta_{\mathsf{Dec}} > \delta_{\mathsf{PCPP}}$, we have that for all proofs $\pi \in \{0,1\}^\ell$, at most an s-fraction of F_i's are satisfied by $y \circ \pi$.

If C is unsatisfiable, then there is a proof $\pi(x)$ for each $y = \mathsf{Enc}(x)$ that makes $V(D)$ accept $y \circ \pi(x)$ with probability at least c. By Lemma 24, such a proof $\pi(x)$ can be computed in polynomial time from y and D, which in particular means that $\pi(x)$ has a polynomial-size circuit. Therefore each output bit of $\pi(x)$ has an $n_{\mathsf{proof}} = \mathrm{poly}(n)$ size $\mathcal{C}$-circuit, from the assumption that $\mathsf{NEXP} \subset \mathcal{C}$.

The next step is to guess a list of n_{proof}-size $\mathcal{C}$ circuits $T_1, T_2, \ldots, T_\ell$ such that $T(x) = (T_1(x), T_2(x), \ldots, T_\ell(x))$ is supposed to the proof $\pi(x)$ given input $y = \mathsf{Enc}(x)$. Slightly abusing notation, F_i is also used to denote the function $F_i(x) := F_i(\mathsf{Enc}(x) \circ T(x))$.

When C is unsatisfiable, by the previous discussion, there is a guess of T_i's such that

$$\Pr_{x\sim\mathcal{U}_n}[V(D)^{\mathsf{Enc}(x)\circ T(x)} = 1] = \mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}\mathop{\mathbb{E}}_{i\in[m]}[F_i(x)] \geq c.$$

When C has at most $2^n/n^{10}$ unsatisfying assignments, then for all possible $T_1, T_2, \ldots, T_\ell$, we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}\mathop{\mathbb{E}}_{i\in[m]}[F_i(x)] \leq 1/n^{10} + s.$$

Now set $\delta_1 := \frac{c-s}{2}$. In order for us to non-deterministically distinguish between the above two cases, it suffices to estimate

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[F_i(x)]$$

to within δ_1, for each $i \in [m]$.

Since each output bit of $\mathsf{Enc}(x)$ is simply a parity on at most $n/2$ bits of x, each F_i can be written as a function $F_i(x) = P(C_1(x), C_2(x))$, where C_1, C_2 are two $\mathcal{C}$ circuits, and P is a function from $\{0,1\}^2 \to \{0,1\}$. (Recall that in this case, we do not require $\mathcal{C}$ to be closed under negation.)

Now we write P as a polynomial:

$$P(z_1, z_2) = \sum_{S \subseteq [2]} \alpha_S \cdot \prod_{i \in S} z_i = \sum_{S \subseteq [2]} \alpha_S \cdot \bigwedge_{i \in S} z_i,$$

where each coefficient $\alpha_S \in [-4, 4]$. Given two $\mathcal{C}$ circuits C_1, C_2, to estimate

$$\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}[P(C_1(x), C_2(x))] = \sum_{S \subseteq [2]} \alpha_S \cdot \mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}\left[\bigwedge_{i \in S} C_i(x)\right]$$

within error δ_1, it suffices to estimate each

$$\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}\left[\bigwedge_{i \in S} C_i(x)\right]$$

within error $\delta = \delta_1/16$. Finally, we can apply our assumed non-trivial CAPP algorithm for $\mathrm{poly}(n)$-size $\mathsf{AND}_2 \circ \mathcal{C}$ circuits to non-deterministically distinguish the two cases, and the theorem follows from Theorem 29.

When we only have non-trivial CAPP algorithms for $\oplus_2 \circ \mathcal{C}$ or $\mathsf{OR}_2 \circ \mathcal{C}$ circuits, we can simply write P in the basis of OR functions or $\oplus$ functions instead. That is, we can write

$$P(z_1, z_2) = \sum_{S \subseteq [2]} \alpha'_S \cdot \bigoplus_{i \in S} z_i,$$

or

$$P(z_1, z_2) = \sum_{S \subseteq [2]} \alpha''_S \cdot \bigvee_{i \in S} z_i.$$

The rest of the argument is the same as the case of $\mathsf{AND}_2 \circ \mathcal{C}$ circuits. ◀

Using Theorem 30, the following theorem can be proved with the same argument as of Theorem 5.

▶ **Reminder of Theorem 6.** *There is an absolute constant $\alpha > 0$, such that for any typical circuit class $\mathcal{C}$, if there is a constant δ such that one of the following holds:*

- *Gap-UNSAT for $2^{\delta n}$-size $\mathsf{AND}_3 \circ \mathcal{C}$ circuits with gap α can be solved in $2^{n-\delta n}$ time, or*
- *CAPP for $2^{\delta n}$-size $\mathsf{OR}_2 \circ \mathcal{C}$, $\oplus_2 \circ \mathcal{C}$, or $\mathsf{AND}_2 \circ \mathcal{C}$ circuits with error α can be solved in $2^{n-\delta n}$ time,*

then for every k there is a function in NP that doesn't have n^k-size $\mathcal{C}$ circuits. Moreover, in the second bullet, $\mathcal{C}$ does not need to be closed under negation.

5 Approaches For THR ∘ THR Circuit Lower Bounds

In this section we propose approaches for proving $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$. We will see that surprisingly weak algorithms suffice for proving this lower bound.

Applying Theorem 5 and the fact that $\oplus_2 \circ \mathsf{THR} \circ \mathsf{THR} \subseteq \mathsf{THR} \circ \mathsf{THR}$, we first show that $\mathsf{NEXP} \not\subset \mathsf{THR} \circ \mathsf{THR}$ would follow from a non-trivial CAPP algorithm for $\mathrm{poly}(n)$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits.

▶ **Reminder of Theorem 8.** *There is an absolute constant $\delta > 0$, such that if δ-error CAPP for $\mathrm{poly}(n)$-size THR∘THR circuits can be solved in $2^n/n^{\omega(1)}$ time, then NEXP $\not\subset$ THR∘THR. The same is true with SAT in place of CAPP.*

Proof. The theorem for CAPP follows directly from the fact that $\oplus_2 \circ \mathsf{THR} \circ \mathsf{THR} \subseteq \mathsf{THR} \circ \mathsf{THR}$ (item (7) of Proposition 18) and Theorem 5.

Suppose SAT for $\mathrm{poly}(n)$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits can be solved in $2^n/n^{\omega(1)}$ time. By Theorem 5, it suffices to give a $2^n/n^{\omega(1)}$ time algorithm for solving SAT for $\mathsf{AND}_3 \circ \mathsf{THR} \circ \mathsf{THR}$ circuits of $\mathrm{poly}(n)$ size (note that Gap-UNSAT is easier than SAT).

Given such an $\mathsf{AND}_3 \circ \mathsf{THR} \circ \mathsf{THR}$ circuit C, we first use the fact that $\mathsf{THR} \subseteq \mathsf{DOR} \circ \mathsf{ETHR}$ (item (2) of Proposition 18) to transform it into a $\mathrm{poly}(n)$ size $\mathsf{AND}_3 \circ \mathsf{DOR} \circ \mathsf{ETHR} \circ \mathsf{ETHR}$ circuit C'.

Treating the DOR as an addition gate (that has at most one true input), and the AND_3 as a multiplication, we can apply distributivity to the circuit. Together with the fact that $\mathsf{AND} \circ \mathsf{ETHR} \subseteq \mathsf{ETHR}$ (item (5) of Proposition 18), C' is then equivalent to a $\mathsf{DOR} \circ \mathsf{ETHR} \circ \mathsf{ETHR}$ circuit C'' of $\mathrm{poly}(n)$ size.

Finally, observe that solving SAT for C'' can be reduced to solving SAT for its $\mathrm{poly}(n)$ $\mathsf{ETHR} \circ \mathsf{ETHR}$ subcircuits, and note that $\mathsf{ETHR} \circ \mathsf{ETHR}$ can be converted efficiently into $\mathsf{THR} \circ \mathsf{THR}$ (item (4) of Proposition 18). Therefore, applying the $2^n/n^{\omega(1)}$ time SAT algorithm for $\mathrm{poly}(n)$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits from the assumption completes the proof. ◀

In fact, similar results apply to TC circuits of any constant depth d (i.e., LT_d circuits). The following theorem can be proved in exactly the same way.

▶ **Reminder of Theorem 9.** *There is an absolute constant $\delta > 0$, such that for any constant d, if CAPP for $\mathrm{poly}(n)$-size LT_d circuits with error δ can be solved in $2^n/n^{\omega(1)}$ time, then NEXP $\not\subset \mathsf{LT}_d$. The same is true with SAT in place of CAPP.*

Now, combing Theorem 8 and our equivalence theorems (Theorem 1 and Theorem 3), the following corollary follows immediately.

▶ **Reminder of Corollary 10.** *There is an absolute constant $\delta > 0$, such that if one of the following holds:*

1. *CAPP (or SAT) for $\mathrm{poly}(n)$-size THR∘MAJ circuits with error δ can be solved in $2^n/n^{\omega(1)}$ time, or*
2. *CAPP for $\mathrm{poly}(n)$-size MAJ∘MAJ circuits with $1/poly(n)$ error can be solved in $2^n/n^{\omega(1)}$ time.*

Then NEXP $\not\subset$ THR∘THR.

6 Lower Bounds for $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ Circuits

We now present our lower bounds for various $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuits. In the following we slightly abuse notation, by also using $\mathcal{C}$ to denote a class of functions from $\{0,1\}^n \to \mathbb{R}$. Note that Boolean circuit classes are special cases of real-valued function classes. We also assume $\mathcal{C}$ contains the constant functions $\mathbf{0}$ and $\mathbf{1}$ for simplicity.

We first define the Sum-Product problem over functions from $\mathcal{C}$.

Sum-Product over $\mathcal{C}$

Given k functions $f_1, \ldots, f_k : \{0,1\}^n \to \mathbb{R}$ from $\mathcal{C}$, compute

$$\sum_{x \in \{0,1\}^n} \prod_{i=1}^{k} f_i(x).$$

6.1 The Main Challenge: Guessed $\mathsf{Sum}_\varepsilon \circ \mathcal{C}$ Circuits Could be Invalid

The main idea is to follow the proof of Theorem 5. Suppose we are given $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuits C_1 and C_2 computing two Boolean functions f_1 and f_2, respectively. One can see that their product $C(x) := C_1(x) \cdot C_2(x)$ (which is a real function on $\{0,1\}^n$), is an $(3 \cdot \varepsilon)$-approximation to $f_1 \wedge f_2$, for small enough $\varepsilon > 0$.

Therefore, if we simply computed

$$\sum_{x \in \{0,1\}^n} C_1(x) \cdot C_2(x), \tag{3}$$

we would have estimated $\Pr_{x \in \{0,1\}^n}[(f_1(x) \wedge f_2(x)) = 1]$ within 3ε. Note that if C_1 (C_2) is a linear sum of m_1 (m_2) $\mathcal{C}$-circuits, then the above quantity can be reduced to $m_1 \cdot m_2$ instances of the sum-product problem over $\mathcal{C}$.

However, the above reasoning does not complete the proof. The problem is that in the proof of Theorem 5, one actually has to *guess* the $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuits which are supposed to compute the PCPP proofs. It could well be the case that our guessed representations are *not valid at all*. That is, it could be that for some $x \in \{0,1\}^n$, $\sum_{i=1}^S \alpha_i \cdot C_i(x)$ is much larger than 1, or much less than 0. If C_1 and C_2 are not valid $\mathsf{Sum}_\varepsilon \circ \mathcal{C}$ circuits to begin with, then the quantity (3) would not be useful.

6.2 Testing Whether a Linear Representation is Close to Boolean

This issue also occurs in Williams' lower bounds on $\mathsf{Sum} \circ \mathcal{C}$ circuits [48]. There, the problem is solved by using a clever algorithm to verify whether a given $\mathsf{Sum} \circ \mathcal{C}$ circuit is valid. In particular, the test of whether a $\mathsf{Sum} \circ \mathcal{C}$ outputs 0 or 1 on every Boolean input is effectively reduced to a small number of Sum-Product calls. But this argument crucially uses the fact that the $\mathsf{Sum} \circ \mathcal{C}$ must output one of two discrete values on every Boolean input. It appears to be much harder to verify that a given $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit is valid.

We will later show that it suffices to test whether a given $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit is close to a Boolean function *with respect to ℓ_2 distance*, in which case we know how to get an algorithm.

It will be convenient to introduce some notation. Let $d_{\mathsf{bin}}(z) = \min_{b \in \{0,1\}} |z - b|$. Intuitively, $d_{\mathsf{bin}}(z)$ measures how close z is to a bit-value. For a function $f : \{0,1\}^n \to \mathbb{R}$, define its closest binary function bin_f as follows: for all $x \in \{0,1\}$, if $f(x) \geq 1/2$, $\mathsf{bin}_f(x) := 1$, otherwise $\mathsf{bin}_f(x) := 0$. By definition, for any $p > 0$ we have

$$\|f - \mathsf{bin}_f\|_p = \left(\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n} [|d_{\mathsf{bin}}(f(x))|^p] \right)^{1/p},$$

and

$$\|f - \mathsf{bin}_f\|_\infty = \max_{x \in \{0,1\}^n} |d_{\mathsf{bin}}(f(x))|.$$

Let $f = \sum_{i=1}^S \alpha_i \cdot C_i$ be a linear combination of functions from $\mathcal{C}$; we wish to verify that f is a $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit for some Boolean function. With respect to the above definitions, f is a valid $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit for some Boolean function if and only if $\|f - \mathsf{bin}_f\|_\infty \leq \varepsilon$.

The following algorithm shows that, given an algorithm for evaluating the Sum-Product of 4 functions from $\mathcal{C}$, the algorithm can be used to distinguish between the case that $\|f - \mathsf{bin}_f\|_\infty$ is small and the case that $\|f - \mathsf{bin}\|_2$ is large.

▶ **Lemma 33.** *For $S \in \mathbb{N}$, suppose we are given S reals $\{\alpha_i\}_{i\in[S]}$, S functions from $\mathcal{C}$ $\{C_i\}_{i\in[S]}$, and parameter $\varepsilon < 0.01$. Suppose Sum-Product of 4 functions on n bits from $\mathcal{C}$ can be solved in $T(n)$ time. Let $f = \sum_{i=1}^{S} \alpha_i \cdot C_i$.*

There is an algorithm A such that:

- *If $\|f - \mathsf{bin}_f\|_\infty \le \varepsilon$, then A always accepts. (That is, if $\sum_{i=1}^{S} \alpha_i \cdot C_i$ is a valid $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit for* some *Boolean function, then A always accepts.)*
- *If $\|f - \mathsf{bin}_f\|_2 \ge 3 \cdot \varepsilon$, then A always rejects.*
- *Otherwise, A can output anything.*
- *A runs in $T(n) \cdot (S+1)^4 + 2^{o(n)}$ time.*

Proof. We define a polynomial of degree 4,

$$P(z) := z^2 \cdot (1-z)^2,$$

to approximate $d_{\mathsf{bin}}(z)$. Simple calculations confirm the following facts about $P(z)$:

- $P(z) \le d_{\mathsf{bin}}(z)^2 \cdot (1 + d_{\mathsf{bin}}(z))^2$, and
- $P(z) \ge d_{\mathsf{bin}}(z)^2 \cdot 2^{-2}$.

When $d_{\mathsf{bin}}(z) \le \varepsilon$, we have $P(z) \le \varepsilon^2 \cdot (1+\varepsilon)^2$. This means that if $\|f - \mathsf{bin}_f\|_\infty \le \varepsilon$, then we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[P(f(x))] \le \varepsilon^2 \cdot (1+\varepsilon)^2 \le \varepsilon^2 \cdot (1+0.01)^2.$$

On the other hand, if $\|f - \mathsf{bin}_f\|_2 \ge 3 \cdot \varepsilon$, then by definition we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}\left[d_{\mathsf{bin}}(f(x))^2\right] \ge (3\cdot\varepsilon)^2,$$

therefore

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[P(f(x))] \ge (3/2)^2 \cdot \varepsilon^2.$$

Therefore, it suffices to compute

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[P(f(x))] \tag{4}$$

to distinguish between these two cases.

Expanding out $P(f(x)) = P(\sum_{i=1}^{S} \alpha_i \cdot C_i)$, it can be written as a $\mathbb{R}$-sum of at most $(S+1)^4$ products of 4 functions from $\mathcal{C}$. By rearranging the order of summation (summing all $(S+1)^4$ terms first), we see that (4) can be evaluated by making at most $(S+1)^4$ calls to the assumed Sum-Product algorithm. Assuming that algorithm runs in $T(n)$ time, the sum (4) can be evaluated in time $T(n) \cdot (S+1)^4 + 2^{o(n)}$. ◀

6.3 Meta-Theorem for $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ Lower Bounds

Now we are ready to prove the following meta theorem for lower bounds on $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuits.

▶ **Theorem 34.** *Suppose every $C \in \mathcal{C}$ has a $\mathrm{poly}(n)$-bit representation, where each C can be evaluated in $\mathrm{poly}(n)$ time. Assume there is a $\delta > 0$ such that for all constant integers $k > 0$, there is a $\mathrm{poly}(n) \cdot 2^{n-\delta n}$-time algorithm for computing the Sum-Product of k functions on n bits from $\mathcal{C}$. Then:*

- *For every k and constant $\varepsilon < 1/2$, there is a function in NP without $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuits of n^k sparsity.*
- *For every unbounded function $\alpha(n)$ such that $n^{\alpha(n)}$ is time-constructible, NTIME$[n^{\alpha(n)}]$ doesn't have $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuits of polynomial sparsity for all constant $\varepsilon < 1/2$.*

The most important component in the proof of the above meta-theorem is an argument that we can solve Gap-UNSAT faster, given a non-trivial algorithm for evaluating Sum-Product of functions from $\mathcal{C}$ and the assumption that Circuit-Eval has a small $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit. Formally, we have the following lemma, whose proof follows similar reasoning as the proof of Theorem 5, while taking care of the issue that a guessed $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit may not be a valid one with the algorithm from Lemma 33.

▶ **Lemma 35.** *There is an absolute constant $\varepsilon > 0$ such that if:*

- *there is a $\delta > 0$ such that for all integers $k \leq 4$, there is a $\mathrm{poly}(n) \cdot 2^{n-\delta n}$-time algorithm for computing the Sum-Product of k functions on n bits from $\mathcal{C}$, and*
- *Circuit-Eval has a $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit of sparsity n^k for some $k > 0$,*

then there is a non-deterministic $2^{n-\delta n} \cdot \mathrm{poly}(n, s)$ time algorithm for Gap-UNSAT with gap $1 - 1/n^{10}$, on general (fan-in 2) circuits with n inputs and s gates.

Proof. Suppose we are given an s-size general circuit $C : \{0,1\}^n \to \{0,1\}$ with the promise that either C is unsatisfiable or C has at least $(1 - 1/n^{10}) \cdot 2^n$ satisfying assignments. We want to distinguish between these two cases in $2^{n-\delta n} \cdot \mathrm{poly}(n, s)$ non-deterministic time.

Let δ_1 be the constant from Lemma 27. Fix a constant-rate linear error correcting code with minimum relative distance δ_1, as guaranteed by Lemma 27, letting $\mathsf{Enc} : \{0,1\}^n \to \{0,1\}^{cn}$ and $\mathsf{Dec} : \{0,1\}^{cn} \to \{0,1\}^n$ be the corresponding encoder and decoder, for a constant c corresponding to the rate of the code. Let $\delta_{\mathsf{Dec}} = c_1 \cdot \delta_1$, which is error rate that Dec can recover.

We also need a $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit for the parity function on $n/2$ bits for computing Enc (by Lemma 27, the code is linear and each output bit depends on at most $n/2$ input bits). Applying the second assumption of the theorem, and the fact that parity reduces easily to Circuit-Eval (parity has linear-size circuits), there is a $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit of sparsity $n_{\mathsf{parity}} = n^{O(k)}$ for the parity function on $n/2$ inputs. We can guess such a $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit $\mathsf{Par}_{n/2}$ of n_{parity} size, and verify it is correct in $2^{n/2} \cdot \mathrm{poly}(n_{\mathsf{parity}}) = 2^{n/2} \cdot \mathrm{poly}(n)$ time, as in the proof of Theorem 5.

Let $D : \{0,1\}^{cn} \to \{0,1\}$ be the circuit defined as $D(y) = \neg C(\mathsf{Dec}(y))$. Since C is of s size and Dec is efficient, D is of size $n_D = \mathrm{poly}(n, s)$.

Then we observe that

$$\Pr_{x \sim \mathcal{U}_n}[C(x) = 0] = \Pr_{x \sim \mathcal{U}_n}[D(\mathsf{Enc}(x)) = 1] = \Pr_{x \sim \mathcal{U}_n}[(D, \mathsf{Enc}(x)) \in \mathsf{Circuit\text{-}Eval}].$$

Now we make use of a PCP of proximity system V for Circuit-Eval, with parameters $\delta_{\mathsf{PCPP}} < \delta_{\mathsf{Dec}}$, $r = O(\log n)$, $q = 2$, and constants $0 < s < c < 1$, with existence guaranteed by Lemma 25. We fix the circuit to be D, and write the verifier as $V(D)$.

We can view the verification of $V(D)$ as $m = 2^{r(|D|)} = \mathrm{poly}(n_D) \leq \mathrm{poly}(n, s)$ constraints on the oracle $y \circ \pi$. We can also assume $|\pi| = \ell \leq \mathrm{poly}(n, s)$. Suppose there are m constraints $F_1, F_2, \ldots, F_m$ on $y \circ \pi$, where each constraint is a function on two coordinates of $y \circ \pi$. Then the properties of the PCP of proximity yield the following consequences:

- If $y = \mathsf{Enc}(x)$ such that $C(x) = 0$, then $D(y) = 1$ and there is a proof $\pi \in \{0,1\}^\ell$ such that such that at least a c-fraction of the F_i's are satisfied by $y \circ \pi$.

- If $y = \mathsf{Enc}(x)$ such that $C(x) = 1$, then for all $z \in \{0,1\}^{cn}$ with $\mathsf{dist}(z, y) \leq \delta_{\mathsf{Dec}}$, we have $D(z) = C(\mathsf{Dec}(z)) = C(x) = 0$. Therefore, y is δ_{Dec}-far from $\mathsf{Circuit\text{-}Eval}(D)$. Since $\delta_{\mathsf{Dec}} > \delta_{\mathsf{PCPP}}$, we have that for all proofs $\pi \in \{0,1\}^{\ell}$, at most an s-fraction of the F_i's are satisfied by $y \circ \pi$.

When C is unsatisfiable, it means there is a proof $\pi(x)$ for each $y = \mathsf{Enc}(x)$, so that $V(D)$ accepts $y \circ \pi(x)$ with probability at least c. Note that by Lemma 25, such a proof $\pi(x)$ can be computed in polynomial time from y and D, which in particular means that $\pi(x)$ admits a $\mathrm{poly}(n, n_D) = \mathrm{poly}(n, s)$-size circuit. By our second assumption, each bit of $\pi(x)$ therefore has a $\widetilde{\mathsf{Sum}}_{\varepsilon} \circ \mathcal{C}$ circuit of sparsity $n_{\mathsf{proof}} = \mathrm{poly}(n, s)^{O(k)} \leq \mathrm{poly}(n, s)$.

Now, we guess a list of (presumably) $\widetilde{\mathsf{Sum}}_{\varepsilon} \circ \mathcal{C}$ circuits $T_1, T_2, \ldots, T_{\ell}$ each of sparsity n_{proof}, and denote $H_i = \mathsf{bin}_{T_i}$. We want $H(x) = (H_1(x), H_2(x), \ldots, H_{\ell}(x))$ to be the proof $\pi(x)$ for the input $y = \mathsf{Enc}(x)$. Let each $T_i = \sum_{j=1}^{n_{\mathsf{proof}}} \alpha_{i,j} \cdot E_{i,j}$, where each $\alpha_{i,j} \in \mathbb{R}$ and $E_{i,j} \in \mathcal{C}$. Slightly abusing notation, we also use F_i to denote the function $F_i(x) := F_i(\mathsf{Enc}(x) \circ H(x))$.

First, for all i, we apply the test of Lemma 33 on T_i with parameter ε. We reject immediately if some test rejects. Note that by Lemma 33 and our first assumption, all the tests take $2^{n-\delta n} \cdot \mathrm{poly}(n, s)$ time in total.

If all guesses are valid $\widetilde{\mathsf{Sum}}_{\varepsilon} \circ \mathcal{C}$ circuits, (that is, $\|H_i - T_i\|_{\infty} \leq \varepsilon$ for all i), then all the tests are passed by Lemma 33. Furthermore if all tests passed, we know that $\|H_i - T_i\|_2 \leq 3 \cdot \varepsilon$ for all $i \in [\ell]$ by Lemma 33.

Therefore, when C is unsatisfiable, by the previous discussion, there is some guess of T_i's such that

$$\Pr_{x \sim \mathcal{U}_n}[V(D)^{\mathsf{Enc}(x) \circ H(x)} = 1] = \Pr_{x \sim \mathcal{U}_n} \Pr_{i \in [m]}[F_i(x) = 1] \geq c.$$

When C has at most $2^n/n^{10}$ unsatisfying assignments, then for all possible $T_1, T_2, \ldots, T_{\ell}$, we have

$$\Pr_{x \sim \mathcal{U}_n} \Pr_{i \in [m]}[F_i(x)] \leq 1/n^{10} + s.$$

Note that F_i is a function on two coordinates of $\mathsf{Enc}(x) \circ H(x)$. In particular, since each bit of $\mathsf{Enc}(x)$ is a just a parity of some inputs in x (it is a linear code), we only need to estimate the following quantity for a function $P : \{0,1\}^2 \to \{0,1\}$:

$$\mathop{\mathbb{E}}_{x \sim \mathcal{U}_n}[P(L_1(x), L_2(x))], \tag{5}$$

where for each function $L_i(x)$, we have an approximate linear representation $T_i = \sum_{j=1}^{n_{\mathsf{final}}} \alpha_{i,j} \cdot E_{i,j}$, such that $\|T_i - L_i\|_2 \leq 3 \cdot \varepsilon$, where each $E_{i,j} \in \mathcal{C}$ and $n_{\mathsf{final}} = \max(n_{\mathsf{proof}}, n_{\mathsf{parity}}) \leq \mathrm{poly}(n, s)$.[13] (Note that when $L_i(x)$ is a bit in the error correcting code, we can simply use the guessed circuit $\mathsf{Par}_{n/2}$.)

Let $\varepsilon_2 = \frac{c-s}{2}$. In order to non-deterministically distinguish between the above two cases, we only have to estimate (5) within error ε_2.

We can write $P : \{0,1\}^2 \to \{0,1\}$ as a multi-linear polynomial, with

$$P(z) := \sum_{S \subseteq [2]} \alpha_S \cdot \prod_{i \in S} z_i,$$

[13] Here we don't need the fact that F_i is an OR on variables or their negations.

where each $\alpha_S \in [-2^2, 2^2]$. Therefore, to estimate (5) within ε_2, we only need to estimate

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}\left[\prod_{i\in S} L_i(x)\right]$$

within error $\varepsilon_2/16$, for each $|S| \geq 1$. (when $S = \emptyset$, it is 1 by definition.)

Now, instead of the above, we compute

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}\left[\prod_{i\in S} T_i(x)\right]. \tag{6}$$

When $|S| = 1$ and in particular $S = \{i\}$, we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[L_i(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{U}_n}[T_i(x)] \leq \|L_i - T_i\|_1 \leq \|L_i - T_i\|_2 \leq 3\cdot\varepsilon.$$

When $|S| = 2$, we want to bound

$$|\langle L_1, L_2\rangle - \langle T_1, T_2\rangle|.$$

Since L_i is Boolean, we have $\|L_i\|_2 = 1$, and therefore $\|T_i\|_2 \leq 1 + 3\cdot\varepsilon$ by the triangle inequality. By Lemma 28, we have

$$|\langle L_1, L_2\rangle - \langle T_1, T_2\rangle| \leq (1+3\varepsilon)\cdot 2\cdot 3\varepsilon.$$

Now we set ε such that $(1+3\varepsilon)\cdot 2\cdot 3\varepsilon = \varepsilon_2/16$.

Finally, for each $S \subseteq [2]$, computing (6) can be reduced to $n_{\mathsf{final}}^{|S|} \leq \mathrm{poly}(n,s)$ evaluations of Sum-Products of $|S| \leq 2$ functions on n bits from $\mathcal{C}$. By assumption, thse evaluations can be computed in $2^{n-\delta n}\cdot\mathrm{poly}(n,s)$ time, which completes the proof. ◀

Now we are ready to prove Theorem 34.

Proof of Theorem 34. For the first consequence, assume every function in NP has a $\widetilde{\mathsf{Sum}}_\varepsilon$ circuit of n^k sparsity, for some fixed $k > 0$ and $0 < \varepsilon < 0.5$. Let ε_1 be the absolute constant specified in Lemma 35. By Lemma 19, there is a polynomial P of degree $d = O(1)$, such that for all $b \in \{0,1\}$, if $|z - b| \leq \varepsilon$ then $|P(z) - b| \leq \varepsilon_1$.

Let $L : \{0,1\}^\star \to \{0,1\}$ be any function in NP and $\sum_{i=1}^{n^k} \alpha_i \cdot C_i$ be the $\widetilde{\mathsf{Sum}}_\varepsilon \circ \mathcal{C}$ circuit for $L_n : \{0,1\}^n \to \{0,1\}$, where each $C_i \in \mathcal{C}$.

Consider the function

$$P\left(\sum_{i=1}^{n^k} \alpha_i \cdot C_i\right). \tag{7}$$

By the definition of P, for all $x \in \{0,1\}^n$, we have

$$\left|P\left(\sum_{i=1}^{n^k} \alpha_i \cdot C_i(x)\right) - L_n(x)\right| \leq \varepsilon_1.$$

Expanding the expression of (7) into a sum of products, we obtain a $\widetilde{\mathsf{Sum}}_{\varepsilon_1} \circ \mathcal{C}^{\otimes d}$ circuit for L_n of sparsity $n^{k'}$, where $\mathcal{C}^{\otimes d}$ consists of all possible products of d functions from $\mathcal{C}$, and $k' \leq O(d\cdot k) \leq O(k)$. Observe that the Sum-Product problem for at most 4 functions from

$\mathcal{C}^{\otimes d}$ is simply the Sum-Product problem for at most $4 \cdot d$ functions from $\mathcal{C}$, which admits a $\text{poly}(n) \cdot 2^{n-\delta n}$ time algorithm by assumption.

Since Circuit-Eval $\in$ NP, both conditions of Lemma 35 are now satisfied for $\mathcal{C}^{\otimes d}$. Therefore Gap-UNSAT with gap $1 - 1/n^{10}$ for s-gate n-input circuits of fan-in 2 has a $2^{n-\delta n} \cdot \text{poly}(n, s)$ time non-deterministic algorithm. It follows from Lemma 30 that, for every k, there is a function in NP which does not have general circuits of n^k size and fan-in 2. This is a contradiction, since every $\widetilde{\text{Sum}}_\varepsilon \circ \mathcal{C}$ circuit of n^k sparsity can be simulated by an $n^{\alpha \cdot k}$-size general fan-in-2 circuit, for a universal constant $\alpha \geq 1$.

The second consequence follows the same way, by applying Theorem 31 instead. ◀

6.4 Lower Bounds for $\widetilde{\text{Sum}}_\varepsilon \circ$ THR/ReLU/$\mathbb{F}_p$-polynomials

In order to apply Theorem 34, we need the following algorithms from [48], for computing the Sum-Product of $O(1)$ functions from the function classes we care about.

▶ **Lemma 36** (Theorem 4.1 in [48]). *The Sum-Product of k THR functions on n variables (with weight in $[-n^n, n^n]$) can be computed in $2^{n/2} \cdot n^{O(k)}$ time.*

▶ **Lemma 37** (Theorem 5.1 in [48]). *The Sum-Product of k ReLU functions on n variables (with weight in $[-W, W]$) can be computed in $2^{n/2} \cdot n^{O(k)} \cdot \text{poly}(k, n, \log W)$ time.*

▶ **Lemma 38** (Theorem 6.1 in [48]). *The Sum-Product of k degree-d polynomials $p_1, \ldots, p_k \in \mathbb{F}_p[x_1, \ldots, x_n]$ can be computed in $p^{2k} \cdot (1.9^n + 2^{n-n/(6dp)}) \cdot \text{poly}(n)$ time.*

Applying Theorem 34 with the above algorithms for computing the Sum-Product for functions from THR, ReLU and $O(1)$-degree $\mathbb{F}_p$-polynomials, Theorem 12, Theorem 13 and Theorem 14 follow immediately.

6.5 Lower Bounds for $\widetilde{\text{Sum}}_\varepsilon \circ \text{ACC}^0 \circ$ THR

Theorem 15 follows via a similar argument as Theorem 34, and the known #SAT algorithms for $\text{ACC}^0 \circ$ THR [52]. Formally, we prove

▶ **Reminder of Theorem 15.** *For every $d, m \geq 1$ and $\varepsilon \in [0, 0.5)$, there is a $b \geq 1$ and an $f \in \text{NTIME}[n^{\log^b n}]$ that does not have $\widetilde{\text{Sum}}_\varepsilon \circ \text{AC}^0_d[m] \circ \text{THR}$ circuits of n^a size, for every a.*

Using the argument of Lemma 35, we can show:

▶ **Lemma 39.** *There is an absolute constant $\varepsilon > 0$ such that if the following two conditions hold:*

- *there is a $\delta > 0$ such that for all integer $k \leq 4$, there is a $\text{poly}(n) \cdot 2^{n-n^\delta}$-time algorithm for computing the Sum-Product of k functions on n bits from $\mathcal{C}$, and*
- *Circuit-Eval has a $\widetilde{\text{Sum}}_\varepsilon \circ \mathcal{C}$ of sparsity n^k for some $k > 0$.*

Then there is non-deterministic $2^{n-n^\delta} \cdot \text{poly}(n, s)$ time algorithm for Gap-UNSAT with gap $1 - 1/n^{10}$ and a general fan-in-2 circuit with n input and s gates.

Theorem 15 then follows from exactly the same arguments as that of Theorem 34, combining the following two facts:

1. For every depth d and integer $m \geq 2$, there is an $\varepsilon > 0$ such that the Sum-Product of $O(1)$ $\text{AC}^0_d[m] \circ$ THR circuits of 2^{n^ε} size can be computed in 2^{n-n^ε} time. This simply applies the algorithm for counting satisfying assignments of $\text{AC}^0_d[m] \circ$ THR circuits ([51]).

2. If for some $\alpha > 0$ there is a nondeterministic $2^{n-n^{\alpha}}$-time Gap-UNSAT algorithm with gap $1 - 1/n^{10}$ for $2^{n^{\alpha}}$-size circuits, then for every $a \geq 1$, there is a $b \geq 1$ such that $\mathsf{NTIME}[n^{\log^b n}]$ does not have $n^{\log^a n}$-size circuits (this is a theorem of Murray and Williams [34]).

6.6 A Note on the Coefficients in the $\widetilde{\mathsf{Sum}}_{\varepsilon} \circ \mathcal{C}$ Circuits

In our proof, we have to *guess* a $\widetilde{\mathsf{Sum}}_{\varepsilon} \circ \mathcal{C}$ circuit, so it is crucial that all $\widetilde{\mathsf{Sum}}_{\varepsilon} \circ \mathcal{C}$ circuits we consider have "reasonable" coefficients, with in $\mathrm{poly}(n)$-bit complexity. When all functions in $\mathcal{C}$ are Boolean-valued, the following proposition provides this guarantee.

▶ **Proposition 40.** *Let $\varepsilon \in [0, 0.5)$ be a constant of bit complexity b.*[14] *Let $\mathcal{C}$ be a class of functions with co-domain $\{0,1\}$, and let C be a $\mathsf{Sum}_{\varepsilon} \circ \mathcal{C}$ circuit of sparsity s for a Boolean function $f : \{0,1\}^n \to \{0,1\}$. There is an equivalent $\widetilde{\mathsf{Sum}}_{\varepsilon} \circ \mathcal{C}$ circuit C' such that every weight in the linear combination of $\mathcal{C}$ has the form j/k, where both j and k are integers in $[-s^{\mathrm{poly}(s,b)}, s^{\mathrm{poly}(s,b)}]$.*

Proof. Let C be a linear combination of s functions from $\mathcal{C}$. We may assume without loss of generality that these s functions are linearly independent. The problem of finding coefficients for these s functions to ε-approximate a given boolean function f is equivalent to finding a solution to a certain linear programming instance $\|Ax - b\|_{\infty} \leq \varepsilon$ in s unknowns over the rationals, where $b \in \{0,1\}^{2^n}$ represents the truth-table of the function f and $A \in \{0,1\}^{2^n \times s}$.

Standard results from the theory of linear programming show that, if the instance is feasible, then there is a valid solution corresponding to the unique solution of a linear system where some of the inequalities are tight (that is $(Ax - b)_i = \varepsilon$ or $(Ax - b)_i = -\varepsilon$). Then proposition then follows from Cramer's rule. ◀

The case for $\widetilde{\mathsf{Sum}}_{\varepsilon} \circ \mathsf{ReLU}$ circuits is more involved. Luckily, Maass [32] showed that the weights for such a circuit of sparsity s needs only $\mathrm{poly}(s, n)$ bits of precision.

7 Structure Lemmas for THR ∘ THR Circuits

In this section we present our structure lemmas for $\mathsf{THR} \circ \mathsf{THR}$ circuits. We first need a simple construction, which will be used in both proofs.

▶ **Definition 41** (Mod p Exact Threshold Gate)**.** *Let G be an ETHR gate with n inputs, p be a prime and G^p be the "mod p" version of G. That is, let L and T be the corresponding linear function and threshold of G, $G^p(x) := [L(x) \equiv T \pmod p]$.*

▶ **Lemma 42.** *Let G be an ETHR gate with n inputs and p be a prime. Then G^p can be written as a DOR ∘ ETHR circuit such that*

- *The top DOR gate has $O(n)$ fan-in.*
- *All ETHR gates have positive weights and thresholds smaller than $O(np)$.*[15]

Proof. Let $w_1, w_2, \ldots, w_n$ and T be the corresponding weights and threshold of G. Reduce each weight w_i in G to $w_i \bmod p$ (the corresponding integer between 0 and $p-1$). This yields another circuit with associate top linear function $L'(x)$, whose value is always at most np. Setting $t = T \bmod p$, $L(x) \equiv T \pmod p$ is equivalent to $L'(x) = t + k \cdot p$ for

[14] That is, we assume ε can be specified as the ratio of two b-bit integers.
[15] Therefore, when $p \leq \mathrm{poly}(n)$, the ETHR gate can be seen as an EMAJ gate.

some $k \in \{0, 1, 2, \ldots, n\}$. Therefore, by taking an OR over all possible k on the condition $L'(x) = t + k \cdot p$, it is a disjoint OR, and we obtain the equivalent DOR $\circ$ ETHR circuit. ◀

7.1 Proof of Structure Lemma I

We begin with the proof of Structure Lemma I for THR $\circ$ THR circuits (restated below).

▶ **Reminder of Lemma 16.** *Let n be number of inputs, $s = s(n) \geq n$ be a size parameter and $\delta = \delta(n)$ be the error parameter. Every s-size THR $\circ$ THR circuit C is equivalent to a Gap-OR$_\delta$ $\circ$ THR $\circ$ MAJ circuit such that:*

- *The top Gap-OR$_\delta$ gate has* $\text{poly}(s, \delta^{-1})$ *fan-in.*
- *Each sub THR $\circ$ MAJ circuit has size* $\text{poly}(s, \delta^{-1})$.

Moreover, the reduction can be computed in deterministic $\text{poly}(s, \delta^{-1})$ *time.*

Proof. Let C' be the given THR $\circ$ THR circuit. By negating some of its input gates (THR is closed under negation), we may assume all weights in the top THR gate of C' are ≤ 0. Since every THR can be converted into a DOR $\circ$ ETHR (item (2) of Proposition 18), C' can be transformed into an equivalent THR $\circ$ ETHR circuit C of size $t = \text{poly}(s)$.

Let $G_1, G_2, \ldots, G_t$, $w_1, w_2, \ldots, w_t$ be the ETHR gates on the bottom layer and their corresponding weights in the top gate of C. By assumption, we also have $w_i \leq 0$ for all i. Let T be the threshold of the top gate. For all inputs x of n bits, we have

$$C(x) = \left[\sum_{i=1}^{t} w_i \cdot G_i(x) \geq T \right].$$

By construction, we may assume that the weights in G_i are bounded by 2^{n^c} for a constant c. Suppose we fix an input x, and let p be a random prime from 2 to $n^{2c} \cdot t^2 \cdot \delta^{-1} = \text{poly}(s, \delta^{-1})$. With probability at least $1 - \delta/t$, we have $G_i^p(x) = G_i(x)$. Let C^p be the circuit obtained by replacing all G_i's in C by corresponding G_i^p's.

When $C(x) = 1$, it follows from a union bound that $C^p(x) = C(x) = 1$ with probability at least $1 - \delta$. When $C(x) = 0$, note that for all primes p, we have $G_i^p(x) \geq G_i(x)$ for all i, therefore we must have $\sum_i^s w_i \cdot G_i^p(x) \leq \sum_i^s w_i \cdot G_i(x) < T$ (all w_i's are ≤ 0) and $C^p(x) = 0$.

Therefore C is equivalent to a Gap-OR$_\delta$ over *all* C^p's, for every prime p (recall their total number is $\text{poly}(s, \delta^{-1})$). By Lemma 42, each C^p can be expressed as a $\text{poly}(s, \delta^{-1})$-size THR $\circ$ EMAJ circuit. Converting each THR $\circ$ EMAJ into a THR $\circ$ MAJ (item (8) of Proposition 18) completes the proof. ◀

7.2 Proof of Structure Lemma II

Now we turn to proving Lemma 17. The proof has two steps, provided by Lemma 43 and Lemma 45.

▶ **Lemma 43** (Weight Reduction at the Top THR gate)**.** *Every size-s THR$_d$ $\circ$ $\mathcal{C}$ circuit (having a top THR gate of fan-in d) is equivalent to a DOR $\circ$ ETHR $\circ$ $\mathcal{C}$ circuit such that:*

- *The top DOR gate has* $\text{poly}(d)$ *fan-in.*
- *Each ETHR gate has fan-in d, with positive weights and threshold value, all of which are less than* $\text{poly}(d) \cdot 2^n$.
- *The $\mathcal{C}$-part is unchanged.*

The same statement also holds for ETHR$_d$ $\circ$ $\mathcal{C}$ circuits. Moreover, the reductions can be computed in randomized $\text{poly}(s)$ *time.*

Proof. We only consider the $\mathsf{THR}_d \circ \mathcal{C}$ case (the $\mathsf{ETHR}_d \circ \mathcal{C}$ case is even easier).

Let C be the given circuit. First, by the fact that $\mathsf{THR} \subseteq \mathsf{DOR} \circ \mathsf{ETHR}$ (item (2) of Proposition 18), C can be transformed to an equivalent $\mathsf{DOR} \circ \mathsf{ETHR} \circ \mathcal{C}$ circuit C'.

Let G be a ETHR gate in C'; note that G has fan-in d. Let D be the subcircuit with top gate G. By construction, G has weights of absolute value at most $M_{\mathsf{old}} = 2^{\mathrm{poly}(d)}$.

Next, we define $L : \{0,1\}^n \to \mathbb{Z}$ such that $L(x)$ is the value of the linear function associated with the gate G when the input is x. That is $D(x) = 1$ if and only if $L(x) = T$ for the threshold T of G.

Pick a random prime number m in the interval $[2, M_{\mathsf{new}}]$, where $M_{\mathsf{new}} = d^c \cdot 2^n$ and c is a sufficiently large constant. For a fixed $x \in \{0,1\}^n$, if $L(x) \neq T$, the probability that $L(x) \equiv T \pmod m$ is smaller than

$$\frac{\log(M_{\mathsf{old}})}{M_{\mathsf{new}}/\ln(M_{\mathsf{new}})} = \frac{\mathrm{poly}(d)}{\Theta(2^n \cdot d^c/(n + c\log d))} \leq d^{-c/2}/2^n,$$

for a sufficiently large c. Applying the union bound over all inputs x, with probability at least $1 - d^{-c/2}$, we have $L(x) \equiv T \pmod m$ if and only if $L(x) = T$ for all $x \in \{0,1\}^n$.

Finally, applying Lemma 42 with prime m, we can replace G with an equivalent $\mathsf{DOR} \circ \mathsf{ETHR}$ subcircuit, whose ETHR gates have positive weights and thresholds smaller than $\mathrm{poly}(d) \cdot 2^n$.

Union-bounding over all ETHR gates, and choosing c to be a large enough constant, this completes the randomized reduction. ◀

▶ Remark 44. One can observe that the above reduction indeed only introduces *one-sided error*. That is, even if it chooses some "bad" primes, the resulting circuit D satisfies the property that $D(x) = 1$ whenever $C(x) = 1$.

▶ **Lemma 45** (Decomposition of the top ETHR gate). *Given an $\mathsf{ETHR}_d \circ \mathcal{C}$ circuit C (a circuit with a top ETHR gate of fan-in d) of size s and a real $\varepsilon \in \left(\frac{\log d}{n}, 1\right)$, suppose the top ETHR gate in C has positive weights and threshold smaller than 2^{2n}. C is equivalent to a $\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{AND}_2 \circ \mathcal{C}$ circuit such that:*

- *The top DOR gate has $2^{O(\varepsilon n)}$ fan-in.*
- *Each MAJ gate has fan-in $d^{O(1/\varepsilon)}$.*
- *The $\mathcal{C}$ part is unchanged.*

Moreover, the reduction can be computed in deterministic

$$2^{O(\varepsilon n)} \cdot d^{O(1/\varepsilon)} + \mathrm{poly}(s)$$

time.

Proof. Let G_{top} be the top ETHR in C, and let $G_1, G_2, \ldots, G_d$ be its input gates. Let w_i's and T be the weights and the threshold of G_{top} and $L(x)$ be the associated linear function. We have for all $x \in \{0,1\}^n$ that

$$L(x) = \sum_{i=1}^{d} w_i \cdot G_i(x).$$

Observe that the binary representations of w_i's and T are of length at most $\log(2^{2n}) \leq 2n$. Break each of their binary representations into $D = \left\lceil \frac{\varepsilon \cdot n}{\log d} \right\rceil$ blocks, where each block has $B \leq 2/\varepsilon \cdot \log d$ bits. Let $w_{i,j}, T_j \in [2^B - 1]$ be the values of w_i's and T's j-th block, respectively (where blocks are numbered from the least significant bit to the most significant bit).

Consider adding the $w_i \cdot G_i(x)$'s in base 2^B, keeping track of all $D-1$ carries on each position, except for the highest one. Let $c = (c_1, c_2, \ldots, c_{D-1}) \in \{0, 1, \ldots, d-1\}^{D-1}$ be such a carry sequence. Observe that $\sum_{i=1}^{d} w_i \cdot G_i(x) = T$ with carry sequence c if and only if for all $j \in [D]$:

$$\sum_{i=1}^{d} w_{i,j} \cdot G_i(x) + c_{j-1} = T_j + 2^B \cdot c_j,$$

where we set C_D and C_0 to be 0 for notational convenience. That is, after we fix the carries c_j's for all j, the sums $\sum_{i=1}^{d} w_{i,j} \cdot G_i(x)$ are also forced to be $T_j^c = T_j + 2^B \cdot c_j - c_{j-1}$. Therefore, consider the sum

$$\sum_{j=1}^{\varepsilon \cdot n} \left(\sum_{i=1}^{d} w_{i,j} \cdot G_i(x) - T_j^c \right)^2.$$

Checking whether this sum is at most 0 can be formulated as a $\mathrm{poly}(d) \cdot 2^{O(B)} = d^{O(1/\varepsilon)}$ size $\mathsf{MAJ} \circ \mathsf{AND}_2$ subcircuit, with input gates $G_1, G_2, \ldots, G_d$.

Each of these addition checks corresponds to one carry sequence. By enumerating all possible d^{D-1} carry sequences, the above transforms G_{top} into a $\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{AND}_2$ subcircuit with input gates $G_1, G_2, \ldots, G_d$, having top fan-in:

$$d^{D-1} = d^{O(\varepsilon \cdot n / \log d)} = 2^{O(\varepsilon \cdot n)},$$

which completes the proof. ◀

Finally, the Structure Lemma II for $\mathsf{THR} \circ \mathsf{THR}$ circuits follows from applying Lemma 43 and Lemma 45 in the appropriate way.

▶ **Reminder of Lemma 17.** *Let n be the number of inputs and let $s = s(n) \leq 2^{o(n)}$ be a size parameter. Let $\varepsilon \in \left(\frac{\log s}{n}, 1\right)$. Every s-size $\mathsf{THR} \circ \mathsf{THR}$ circuit C is equivalent to a $\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{MAJ}$ circuit such that:*

- *The top DOR gate has $2^{O(\varepsilon n)}$ fan-in.*
- *Each sub $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuit has size $s^{O(1/\varepsilon)}$.*

The reduction can be computed in randomized $2^{O(\varepsilon n)} \cdot s^{O(1/\varepsilon)}$ *time.*

Proof. First, since $\mathsf{THR} \subseteq \mathsf{DOR} \circ \mathsf{ETHR}$ (item (2) of Proposition 18), C is equivalent to a $\mathrm{poly}(s)$-size $\mathsf{THR} \circ \mathsf{ETHR}$ circuit C_1. Moreover, we can convert C into C_1 in polynomial time.

Second, we apply Lemma 43 to transform C_1 into a $\mathsf{DOR}_{\mathrm{poly}(s)} \circ \mathsf{ETHR} \circ \mathsf{ETHR}$ circuit C_2, such that all middle-layer ETHR gates have positive weights and thresholds smaller than $\mathrm{poly}(s) \cdot 2^n < 2^{2n}$.

Third, we apply Lemma 45 to C_2, which changes all middle-layer ETHR gates of C_2 into $\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{AND}_2$ subcircuits, with top gate fan-in $2^{O(\varepsilon \cdot n)}$. This yields a $\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{AND} \circ \mathsf{ETHR}$ circuit. Converting the remaining $\mathsf{AND} \circ \mathsf{ETHR}$ subcircuits into ETHR's (item (5) of Proposition 18), we obtain a $\mathsf{DOR}_{2^{O(\varepsilon \cdot n)}} \circ \mathsf{MAJ} \circ \mathsf{ETHR}$ circuit where all $\mathsf{MAJ} \circ \mathsf{ETHR}$ subcircuits have size at most $s^{O(1/\varepsilon)}$.

Finally, converting each $\mathsf{MAJ} \circ \mathsf{ETHR}$ into a $\mathsf{MAJ} \circ \mathsf{MAJ}$ (item (3) of Proposition 18) completes the reduction. The running time bound follows from plugging in the time bounds of Lemma 43 and Lemma 45. ◀

Setting the parameter ε carefully in Lemma 17, we have the following corollary.

▶ **Corollary 46.** *Let n be the number of inputs and let $s = s(n) \leq 2^{o(n)}$ be a size parameter. Let $\varepsilon \in \left(\frac{\log s}{n}, 1\right)$. Every s-size $\mathsf{THR} \circ \mathsf{THR}$ circuit C is equivalent to a $\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{MAJ}$ circuit C' such that:*

- *The top DOR gate of C' has $s^{O(1/\varepsilon)}$ fan-in.*
- *Every sub $\mathsf{MAJ} \circ \mathsf{MAJ}$ circuit of C' has size $2^{O(\varepsilon \cdot n)}$.*

The reduction can be computed in randomized $2^{O(\varepsilon n)} \cdot s^{O(1/\varepsilon)}$ time.

References

1. Amir Abboud and Karl Bringmann. Tighter Connections Between Formula-SAT and Shaving Logs. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 8:1–8:18, 2018. doi:10.4230/LIPIcs.ICALP.2018.8.
2. Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3):14:1–14:36, 2010. doi:10.1145/1706591.1706594.
3. Josh Alman, Timothy M. Chan, and R. Ryan Williams. Polynomial Representations of Threshold Functions and Algorithmic Applications. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 467–476, 2016. doi:10.1109/FOCS.2016.57.
4. Kazuyuki Amano and Akira Maruoka. On the Complexity of Depth-2 Circuits with Threshold Gates. In *Mathematical Foundations of Computer Science 2005, 30th International Symposium, MFCS 2005, Gdansk, Poland, August 29 - September 2, 2005, Proceedings*, pages 107–118, 2005. doi:10.1007/11549345_11.
5. Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof Verification and the Hardness of Approximation Problems. *J. ACM*, 45(3):501–555, 1998. doi:10.1145/278298.278306.
6. Sanjeev Arora and Shmuel Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.
7. László Babai, Kristoffer Arnsfelt Hansen, Vladimir V Podolskii, and Xiaoming Sun. Weights of exact threshold functions. In *International Symposium on Mathematical Foundations of Computer Science*, pages 66–77. Springer, 2010.
8. Swapnam Bajpai, Vaibhav Krishan, Deepanshu Kush, Nutan Limaye, and Srikanth Srinivasan. A #SAT Algorithm for Small Constant-Depth Circuits with PTF Gates. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 8:1–8:20, 2019. doi:10.4230/LIPIcs.ITCS.2019.8.
9. Paul Beame, Stephen A. Cook, and H. James Hoover. Log Depth Circuits for Division and Related Problems. *SIAM J. Comput.*, 15(4):994–1003, 1986. doi:10.1137/0215070.
10. Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Short PCPs Verifiable in Polylogarithmic Time. In *20th Annual IEEE Conference on Computational Complexity (CCC 2005), 11-15 June 2005, San Jose, CA, USA*, pages 120–134, 2005. doi:10.1109/CCC.2005.27.
11. Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of Proximity, Shorter PCPs, and Applications to Coding. *SIAM J. Comput.*, 36(4):889–974, 2006. doi:10.1137/S0097539705446810.
12. Eli Ben-Sasson and Emanuele Viola. Short PCPs with Projection Queries. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 163–173, 2014. doi:10.1007/978-3-662-43948-7_14.
13. Eli Ben-Sasson and Emanuele Viola. Short PCPs with projection queries. In *International Colloquium on Automata, Languages, and Programming*, pages 163–173. Springer, 2014.

14 Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: tight quantum query bounds via dual polynomials. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 297–310, 2018. doi:10.1145/3188745.3188784.

15 Mark Bun and Justin Thaler. A Nearly Optimal Lower Bound on the Approximate Degree of AC^0. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 1–12, 2017. doi:10.1109/FOCS.2017.10.

16 Arkadev Chattopadhyay and Nikhil S. Mande. A Short List of Equalities Induces Large Sign Rank. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 47–58, 2018. doi:10.1109/FOCS.2018.00014.

17 Ruiwen Chen and Rahul Santhanam. Improved Algorithms for Sparse MAX-SAT and MAX-k-CSP. In *Theory and Applications of Satisfiability Testing - SAT 2015 - 18th International Conference, Austin, TX, USA, September 24-27, 2015, Proceedings*, pages 33–45, 2015. doi:10.1007/978-3-319-24318-4_4.

18 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.

19 Jürgen Forster, Matthias Krause, Satyanarayana V. Lokam, Rustam Mubarakzjanov, Niels Schmitt, and Hans Ulrich Simon. Relations Between Communication Complexity, Linear Arrangements, and Computational Complexity. In *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science, 21st Conference, Bangalore, India, December 13-15, 2001, Proceedings*, pages 171–182, 2001. doi:10.1007/3-540-45294-X_15.

20 M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some Simplified NP-Complete Graph Problems. *Theor. Comput. Sci.*, 1(3):237–267, 1976. doi:10.1016/0304-3975(76)90059-1.

21 Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. Majority Gates VS. General Weighted Threshold Gates. *Computational Complexity*, 2:277–300, 1992. doi:10.1007/BF01200426.

22 Hans Dietmar Groeger and György Turán. A linear lower bound for the size of threshold circuits. *Bulletin-European Association For Theoretical Computer Science*, 50:220–220, 1993.

23 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. doi:10.1016/0022-0000(93)90001-D.

24 Kristoffer Arnsfelt Hansen and Vladimir V. Podolskii. Exact Threshold Circuits. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, June 9-12, 2010*, pages 270–279, 2010. doi:10.1109/CCC.2010.33.

25 Kristoffer Arnsfelt Hansen and Vladimir V. Podolskii. Polynomial threshold functions and Boolean threshold circuits. *Inf. Comput.*, 240:56–73, 2015. doi:10.1016/j.ic.2014.09.008.

26 William Hesse, Eric Allender, and David A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *J. Comput. Syst. Sci.*, 65(4):695–716, 2002. doi:10.1016/S0022-0000(02)00025-9.

27 Thomas Hofmeister. A Note on the Simulation of Exponential Threshold Weights. In *Computing and Combinatorics, Second Annual International Conference, COCOON '96, Hong Kong, June 17-19, 1996, Proceedings*, pages 136–141, 1996. doi:10.1007/3-540-61332-3_146.

28 Russell Impagliazzo, Ramamohan Paturi, and Michael E. Saks. Size-Depth Tradeoffs for Threshold Circuits. *SIAM J. Comput.*, 26(3):693–707, 1997. doi:10.1137/S0097539792282965.

29 Russell Impagliazzo, Ramamohan Paturi, and Stefan Schneider. A Satisfiability Algorithm for Sparse Depth Two Threshold Circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 479–488, 2013. doi:10.1109/FOCS.2013.58.

30 Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local Reductions. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 749–760, 2015. doi:10.1007/978-3-662-47672-7_61.

31 Daniel M. Kane and Ryan Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 633–643, 2016. doi:10.1145/2897518.2897636.

32 Wolfgang Maass. Bounds for the Computational Power and Learning Complexity of Analog Neural Nets. *SIAM J. Comput.*, 26(3):708–732, 1997. doi:10.1137/S0097539793256041.

33 Saburo Muroga, Iwao Toda, and Satoru Takasu. Theory of majority decision elements. *Journal of the Franklin Institute*, 271(5):376–418, 1961.

34 Cody Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 890–901, 2018. doi:10.1145/3188745.3188910.

35 Noam Nisan. The communication complexity of threshold gates. *Combinatorics, Paul Erdos is Eighty*, 1:301–315, 1993.

36 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. URL: http://www.cambridge.org/de/academic/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g/analysis-boolean-functions.

37 Ramamohan Paturi and Michael E. Saks. Approximating Threshold Circuits by Rational Functions. *Inf. Comput.*, 112(2):257–272, 1994. doi:10.1006/inco.1994.1059.

38 John H. Reif and Stephen R. Tate. On Threshold Circuits and Polynomial Computation. *SIAM J. Comput.*, 21(5):896–908, 1992. doi:10.1137/0221053.

39 Theodore J Rivlin. *An introduction to the approximation of functions*. Courier Corporation, 2003.

40 Vwani P. Roychowdhury, Alon Orlitsky, and Kai-Yeung Siu. Lower bounds on threshold and related circuits via communication complexity. *IEEE Trans. Information Theory*, 40(2):467–474, 1994. doi:10.1109/18.312169.

41 Rahul Santhanam and Ryan Williams. On Medium-Uniformity and Circuit Lower Bounds. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 15–23, 2013. doi:10.1109/CCC.2013.40.

42 Alexander A. Sherstov. Algorithmic polynomials. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 311–324, 2018. doi:10.1145/3188745.3188958.

43 Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Information Theory*, 42(6):1723–1731, 1996. doi:10.1109/18.556668.

44 Suguru Tamaki. A Satisfiability Algorithm for Depth Two Circuits with a Sub-Quadratic Number of Symmetric and Threshold Gates. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:100, 2016. URL: http://eccc.hpi-web.de/report/2016/100.

45 Roei Tell. Proving that prBPP=prP is as hard as "almost" proving that P $\neq$ NP. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:3, 2018. URL: https://eccc.weizmann.ac.il/report/2018/003.

46 Roei Tell. Quantified derandomization of linear threshold circuits. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 855–865, 2018. doi:10.1145/3188745.3188822.

47 R. Ryan Williams. Natural Proofs versus Derandomization. *SIAM J. Comput.*, 45(2):497–529, 2016. doi:10.1137/130938219.

48 Richard Ryan Williams. Limits on Representing Boolean Functions by Linear Combinations of Simple Functions: Thresholds, ReLUs, and Low-Degree Polynomials. In *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, pages 6:1–6:24, 2018. doi:10.4230/LIPIcs.CCC.2018.6.

49 Ryan Williams. Improving Exhaustive Search Implies Superpolynomial Lower Bounds. *SIAM J. Comput.*, 42(3):1218–1244, 2013. doi:10.1137/10080703X.

50 Ryan Williams. Towards NEXP versus BPP? In *Computer Science - Theory and Applications - 8th International Computer Science Symposium in Russia, CSR 2013, Ekaterinburg, Russia, June 25-29, 2013. Proceedings*, pages 174–182, 2013. doi:10.1007/978-3-642-38536-0_15.

51 Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 194–202, 2014. doi:10.1145/2591796.2591858.

52 Ryan Williams. Nonuniform ACC circuit lower bounds. *Journal of the ACM (JACM)*, 61(1):2, 2014.

A Constructions of Super Query-Efficient PCP of Proximity Systems

In this appendix we present proofs for Lemma 24 and Lemma 25 for completeness. We remark that we did not make any effort to optimize the soundness/completeness constants s and c in our construction; any universal constant suffices in our applications.[16]

We start with Lemma 24 (restated below).

▶ **Reminder of Lemma 24** (3-query PCPP with perfect completeness)**.** *For any constant $\delta > 0$ there is a constant $0 < s < 1$, such that there is a PCP of proximity system for Circuit-Eval with proximity δ, soundness s, random bits $r = O(\log n)$, and query complexity $q = 3$. Moreover, the system satisfies two additional properties:*

(1) *Given the random coins, the verifier simply computes an OR on these 3 queried bits or their negations, and accepts if the OR is true.*

(2) *Given the pair $(C, w) \in$ Circuit-Eval, we can construct a proof π in $\mathrm{poly}(|C| + |w|)$ time that makes $V(C)$ accept with probability 1.*

Proof. By Lemma 22, there is a PCP of proximity system V for Circuit-Eval with proximity δ, soundness $s = 1/2$, number of random bits $r = O(\log n)$ and query complexity $q = O_\delta(1)$.

Let the circuit be C. Suppose we are given random bits $R \in \{0,1\}^r$, so that $V(C)$ queries positions $k_1 = k_1(C, R), k_2 = k_2(C, R), \ldots, k_q = k_q(C, R)$ of the oracle $z = w \circ \pi$, computes a predicate $P = P(C, R)$ on these bits, and outputs $P(z_{k_1}, z_{k_2}, \ldots, z_{k_q})$.

We construct a new PCP of proximity system V' as follows. Fix the circuit C. and let $P_R = P(C, R)$. Slightly abusing notation, we let x_j denote the bit z_{k_j}. P_R can be computed by a circuit D_R of $O_q(1)$ size; therefore P_R can computed in size S for a universal constant S only depending on q. We can construct a group of auxiliary variables $\{y_{R,\ell}\}_{\ell \in [S]}$, and a group of constraints $\{F_{w,\ell}\}_{\ell \in [S]}$, where each constraint is an OR of three bits (or their negations) from the x_j's and $y_{R,\ell}$'s, such that $P_w(x_1, x_2, \ldots, x_q) = 1$ if and only if there exists an assignment to the $y_{R,\ell}$'s such that all constraints $F_{R,\ell}$'s are satisfied.

The verifier $V'(C)$ treats its oracle as three parts. The first two parts are w and π (where π is supposed to be a proof for $V(C)$), while the third part π_y is supposed to contain assignments to all $y_{R,\ell}$'s for all $R \in \{0,1\}^r$ and $\ell \in [S]$. $V'(C)$ first tosses r random coins to get a random string $R \in \{0,1\}^r$, then tosses $\log(S)$ more coins to pick a random integer $\ell \in [S]$. Then $V'(C)$ queries the 3 bits appearing in the constraint $F_{R,\ell}$, and accepts if and only if the constraint is satisfied by those 3 bits. We denote its proof to be $\pi' = (\pi, \pi_y)$.

We claim that $V'(C)$ is a correct PCP of Proximity system. If $(C, w) \in$ Circuit-Eval, let π be a proof such that $V(C)$ accepts $w \circ \pi$ with probability 1. Then by our construction of $V'(C)$, there is a π_y such that $V'(C)$ accepts $w \circ (\pi \circ \pi_y)$ with probability 1.

[16] Here we are actually composing the PCPP from [11] with some trivial PCPP constructions for constant-size functions. There are much better constructions, see e.g. [18].

Otherwise, suppose w is δ-far from the set $\{z : C(z) = 1\}$. Then for any proof π, $V(C)$ accepts $(w \circ \pi)$ with probability at most $1/2$. This means for all additional proofs π_y, at least a $1/2$-fraction of $R \in \{0,1\}^r$ are such that at least one constraint from $\{F_{R,\ell}\}_{\ell \in [S]}$ is not satisfied by $w \circ (\pi \circ \pi_y)$. Therefore, $V'(C)$ rejects with probability at least $1/2 \cdot 1/S = \Omega_q(1) = \Omega_\delta(1)$, which completes the proof. ◀

In order to get a 2-query PCP of proximity system from the above, we use the following classical gadget by Garey, Johnson, and Stockmeyer [20], originally used to prove the NP-hardness of MAX-2-SAT.

▶ **Lemma 47.** *Let X_1, X_2, X_3 and Y be 4 Boolean variables. Consider the following 10 constraints:*

$$X_1, X_2, X_3, \neg X_1 \vee \neg X_2, \neg X_2 \vee \neg X_3, \neg X_3 \vee \neg X_1,$$
$$Y, X_1 \vee \neg Y, X_2 \vee \neg Y, X_3 \vee \neg Y.$$

If $X_1 \vee X_2 \vee X_3$, then there exists an assignment to Y such that 7 of the above constraints are satisfied. Otherwise, all assignments to Y satisfy at most 6 of the above constraints.

▶ **Reminder of Lemma 25** (2-query PCPP with constant completeness/soundness gap)**.** *For any constant $\delta > 0$ there two constants $0 < s < c < 1$, such that there is a PCP of proximity system for Circuit-Eval with proximity δ, soundness s, completeness c, number of random bits $r = O(\log n)$ and query complexity $q = 2$. Moreover, the system satisfies two additional properties:*

(1) *Given the random coins, the verifier computes an OR on the 2 queried bits or their negations, and accepts iff the OR is true.*

(2) *Given the pair $(C, w) \in$ Circuit-Eval, a proof π can be constructed in $\mathrm{poly}(|C| + |w|)$ time that makes $V(C)$ accept with probability at least c.*

Proof. By Lemma 24, there is a PCP of proximity system V for Circuit-Eval with proximity δ, soundness $s = s(\delta) < 1$, number of random bits $r = O(\log n)$ and query complexity $q = 3$. The verifier computes an OR on these 3 queried bits or their negations, and accepts if it is true.

Let the circuit be C. We begin as in the previous proof. Suppose we have randomness $R \in \{0,1\}^r$, and $V(C)$ queries positions $k_1 = k_1(C,R), k_2 = k_2(C,R), k_3 = k_3(C,R)$ of the oracle $z = w \circ \pi$, computes a predicate $P = P(C,R)$ on these bits, then outputs $P(z_{k_1}, z_{k_2}, z_{k_3})$. Slightly abusing notation, we use x_j to denote the bit z_{k_j}. By Lemma 24, we can assume

$$P(x_1, x_2, x_3) = \vee_{j \in [3]}(x_j \oplus b_j),$$

where $b_j = b_j(C,R)$ is whether it negates the bit x_j.

Our new PCP of proximity system V' works as follows. Fix the circuit C and let $P_R = P(C,R)$. By Lemma 47, we can construct an auxiliary variable y_R and a group of constraints $\{F_{R,\ell}\}_{\ell \in [10]}$, each is an OR of 2 bits (or their negations) from x_j's and y_R[17] such that if $P_w(x_1, x_2, x_3) = 1$ then there is an assignment to the y_R such that 7 constraints from $\{F_{R,\ell}\}_{\ell \in [10]}$ are satisfied; otherwise, for all assignments to y_R, at most 6 constraints from $\{F_{R,\ell}\}_{\ell \in [10]}$ are satisfied.

[17] In Lemma 47, constraint X_i can be written as $X_i \vee X_i$, which is an OR of 2 bits.

As in the 3-query PCPP, $V'(C)$ treats its oracle as three parts: the first two are w and π (π is intended to be a proof in $V(C)$), and the third part π_y is intended to contain assignments to all y_R's, for all $R \in \{0,1\}^r$. Our $V'(C)$ first tosses r random coins to get $R \in \{0,1\}^r$, then tosses $O(1)$ more coins to pick a random integer $\ell \in [10]$. Then it simply queries the 2 bits appearing in the constraint $F_{R,\ell}$, and accepts iff that constraint is satisfied. We denote its proof to be $\pi' = (\pi, \pi_y)$.

Let us argue $V'(C)$ satisfies our requirement. If $(C, w) \in$ Circuit-Eval, let π be a proof such that $V(C)$ accepts $w \circ \pi$ with probability 1. Then by our construction of $V'(C)$, there is a π_y such that $V'(C)$ accepts $w \circ (\pi \circ \pi_y)$ with probability at least $7/10$.

Now suppose w is δ-far from the set $\{z : C(z) = 1\}$. Then for all proofs π, $V(C)$ accepts $(w \circ \pi)$ with probability at most s. This means that for any additional proof π_y, there is at most an s-fraction of $R \in \{0,1\}^r$ such that 7 constraints from $\{F_{R,\ell}\}_{\ell \in [S]}$ are satisfied by $w \circ (\pi \circ \pi_y)$; for the remaining R's, at most 6 constraints from $\{F_{R,\ell}\}_{\ell \in [S]}$ are satisfied. Therefore, $V'(C)$ accepts with probability at most $s \cdot 7/10 + (1-s) \cdot 6/10 < 7/10$, which completes the proof. ◀

B Proofs for THR ⊆ DOR ∘ ETHR and $\oplus_k$ ∘ THR ∘ THR ⊆ THR ∘ THR

Here we present an alternative proof that THR ⊆ DOR ∘ ETHR, which has a better weight dependence than prior work [24] and is arguably simpler. We first give a construction for the special case when all the weights and the threshold value are non-negative. Then we show that the general case can be easily reduced to this case.

▶ **Lemma 48.** *Let G be a THR gate on n bits defined as $G(x) := [\sum_{i=1}^n w_i \cdot x_i > T]$, such that all w_i's and T are integers in $[0, 2^L - 1]$ for some $L \in \mathbb{N}$. Then G can be written as a DOR of $O(n \cdot L)$ many ETHR gates, each with weights and threshold from $[0, 2^{L+1} - 1]$.*

Proof. For each weight $w_i \in [0, 2^L - 1]$, write it in its binary representation

$$w_{i,L}, w_{i,L-2}, \ldots, w_{i,1} \in \{0,1\}^L,$$

such that

$$w_i = \sum_{j=1}^{L} 2^{j-1} \cdot w_{i,j}.$$

In this way, we can view w as a Boolean matrix from $\{0,1\}^{n \times L}$. For each position $(a,b) \in [n] \times [L]$, we build a partial matrix $w^{(a,b)}$ as follows: for $(i,j) \in [n] \times [L]$,

$$w^{(a,b)}_{i,j} = \begin{cases} w_{i,j} & (j > b) \text{ or } ((j = b) \text{ and } (i \geq a)) \\ 0 & \text{otherwise} \end{cases}$$

That is, $w^{(a,b)}$ is the sub-matrix of w, consisting of entries which are either to the right of (a,b), or directly above (a,b). (We number the rows of the matrix from bottom to top, and the columns of the matrix from left to right.)

Given $x \in \{0,1\}^n$, we define

$$w^{(a,b)} \cdot x := \sum_{i=1}^{n} \left(\sum_{j=1}^{L} 2^{j-1} \cdot w^{(a,b)}_{i,j} \right) \cdot x_i.$$

That is, we treat each row of $w^{(a,b)}$ as the L-bit binary representation of the corresponding weight on x_i. By definition, we have $w^{(1,1)} \cdot x = w \cdot x = \sum_{i=1}^{n} w_i \cdot x_i$.

Now, fix $x \in \{0,1\}^n$, and consider the sequence S, defined as:

$$w^{(n,L)} \cdot x, w^{(n-1,L)} \cdot x, \ldots, w^{(1,L)} \cdot x, w^{(n,L-1)} \cdot x, \ldots, w^{(1,L-1)} \cdot x, \ldots, w^{(n,1)} \cdot x, \ldots, w^{(1,1)} \cdot x.$$

By definition of $w^{(a,b)}$, we are including 1-entries of the matrix w one-by-one, hence the sequence S is non-decreasing (and begins with 0).

Suppose $w \cdot x = w^{(1,1)} \cdot x > T$. Then there must be a *unique* position $(a,b) \in [n] \times [L]$ such that $w^{(a,b)} \cdot x$ is the first value in the sequence S which is greater than T. For each $(a,b) \in [n] \times [L]$, we will use an ETHR gate $E^{(a,b)}$ to specify the condition that $w^{(a,b)} \cdot x$ is the first value greater than T from S. Then when $G(x)$ is true, exactly one of the $E^{(a,b)}(x)$'s is true, and when $G(x)$ is false, all of the $E^{(a,b)}(x)$'s are false.

To see that an ETHR gate suffices, we observe that $w^{(a,b)} \cdot x$ is the first value greater than T from the sequence S, if and only if the following conditions hold:

1. $w^{(a,b)} \cdot x > T$ *(it is greater than T)*,
2. $(w_{a,b} = 1) \wedge (x_a = 1)$ *(it is bigger than the previous value)*, and
3. $w^{(a,b)} \cdot x - 2^{b-1} \leq T$ *(the previous value is no greater than T)*.

We crucially observe that $w^{(a,b)} \cdot x$ is a multiple of 2^{b-1}. In the matrix $w^{(a,b)}$, we only include nonzero $w_{i,j}$'s where $j \geq b$. Thus in $w^{(a,b)} \cdot x$, every 1 in x is getting multiplied by a power of two which is at least 2^{b-1}.

By division, $T = 2^{b-1} \cdot T_b + T_r$, for some $0 \leq T_r < 2^{b-1}$ and $T_b > 0$. Then $w^{(a,b)} \cdot x > T$ if and only if $w^{(a,b)} \cdot x \geq 2^{b-1} \cdot (T_b + 1)$. Furthermore, $w^{(a,b)} \cdot x - 2^{b-1} \leq T$ if and only if $w^{(a,b)} \cdot x \leq 2^{b-1} \cdot (T_b + 1)$. Therefore, the above conditions are equivalent to

1. $(w_{a,b} = 1) \wedge (x_a = 1)$, and
2. $w^{(a,b)} \cdot x = 2^{b-1} \cdot (T_b + 1)$.

Now all these conditions are linear equations, so we can define an ETHR function $E^{(a,b)}$ that checks all of them. In particular, set $E^{(a,b)}$ to be the constant function $\mathbf{0}$ if $w_{a,b} = 0$; otherwise set

$$E^{(a,b)}(x) := \left[(2 \cdot w^{(a,b)} \cdot x) + x_a = 2^b \cdot (T_b + 1) + 1 \right].$$

This completes the proof. ◀

Now we reduce the general case to the non-negative weights and thresholds case, and complete the reduction from THR to $\mathsf{DOR} \circ \mathsf{ETHR}$.

▶ **Lemma 49.** *Let G be a THR gate on n bits, $G(x) := [\sum_{i=1}^{n} w_i \cdot x_i > T]$, such that all w_i's and T are integers from $[-W, W]$ for some $W \in \mathbb{N}$. Then G can be written as a DOR of $O(n \cdot \log W)$ many ETHR gates, each with weights and threshold from $[-\Theta(W), \Theta(W)]$.*

Proof. We start by defining n new variables $z_1, z_2, \ldots, z_n \in \{0,1\}^n$. Set $z_i := x_i$ if $w_i \geq 0$, and $z_i := 1 - x_i$ otherwise. Letting $S = \{i : w_i \geq 0\}$, we have

$$\begin{aligned} \sum_{i=1}^{n} w_i \cdot x_i &= \sum_{i \in S} w_i \cdot z_i + \sum_{i \notin S} w_i \cdot (1 - z_i) \\ &= \sum_{i \in S} w_i \cdot z_i + \sum_{i \notin S} -w_i \cdot z_i + \sum_{i \notin S} w_i. \end{aligned}$$

Let $\widehat{w}_i = |w_i|$, and $\widehat{T} = T - \sum_{i \notin S} w_i$. Observe that

$$\left[\sum_{i=1}^{n} w_i \cdot x_i > T \right] \Leftrightarrow \left[\sum_{i=1}^{n} \widehat{w}_i \cdot z_i > \widehat{T} \right].$$

If $\widehat{T}$ is negative, then $G(x) = 1$ on all Boolean inputs x (since all $\widehat{w}_i$'s are non-negative, and the z_i are Boolean) and we are done. Otherwise, we can apply Lemma 48 with $\widehat{G}(z) := \left[\sum_{i=1}^{n} \widehat{w}_i \cdot z_i > \widehat{T}\right]$. Substituting each z_i by $1 - x_i$, we obtain the desired DOR decomposition for $G(x)$. ◀

▶ **Lemma 50.** *Let k be a constant, a $\oplus_k \circ$ THR $\circ$ THR circuit of $s = s(n)$ size on n bits is equivalent to a THR $\circ$ THR circuit of $s^{O(k)}$ size. Moreover, the corresponding THR $\circ$ THR circuit can be constructed deterministically in $s^{O(k)}$ time.*

Proof. First, by Lemma 49, the given $\oplus_k \circ \mathsf{THR} \circ \mathsf{THR}$ circuit can be transformed into a $\oplus_k \circ \mathsf{THR} \circ \mathsf{ETHR}$ circuit C of $t = \mathrm{poly}(s)$ size.

Let $C_1, C_2, \ldots, C_k$ be the $\mathsf{THR} \circ \mathsf{ETHR}$ subcircuits of C. For each C_i, let $E_{i,1}, \ldots, E_{i,t}$ be its ETHR gates, $w_{i,1}, \ldots, w_{i,t}$ be the corresponding weights in the output threshold function, and T_i be the threshold value of the output threshold function. We have

$$C_i(x) := \left[\sum_{j=1}^{t} w_{i,j} \cdot E_{i,j}(x) > T_i\right].$$

By slightly perturbing the $w_{i,j}$'s and T_i, we can ensure that $\sum_{j=1}^{t} w_{i,j} \cdot E_{i,j}(x) - T_i$ is never equal to 0, over all $x \in \{0,1\}^n$. Next, we define

$$F(x) = \left[\prod_{i=1}^{k} \left(T_i - \sum_{j=1}^{t} w_{i,j} \cdot E_{i,j}(x)\right) < 0\right]. \tag{8}$$

Noting that $C_i(x) = 1$ if and only if $T_i - \sum_{j=1}^{t} w_{i,j} \cdot E_{i,j}(x) < 0$, we observe that $F(x) = 1$ when an odd number of $C_i(x)$'s are 1, and $F(x) = 0$ otherwise. Therefore, F computes the same function as the original circuit C.

Finally, expanding the product of k sums in (8) into a sum of $s^{O(k)}$ products, and recalling that $\mathsf{AND} \circ \mathsf{ETHR} \subseteq \mathsf{ETHR}$ (Proposition 18), F can be written as a $\mathsf{THR} \circ \mathsf{ETHR}$ circuit. Converting this back to a $\mathsf{THR} \circ \mathsf{THR}$ circuit (Proposition 18), the proof is complete. ◀

C More Applications of Structure Lemmas for THR ∘ THR Circuits

Here we discuss more applications of Lemma 16 and Lemma 17.

C.1 Generalization to Threshold Circuits of Constant Depth

Lemma 17 generalizes readily to threshold circuits of any constant depth. In the following LT_d denotes threshold circuits of depth-d, while $\widehat{\mathsf{LT}}_d$ denotes depth-d majority circuits (see Section 2.1 for formal definitions).

▶ **Corollary 51.** *Let n be number of inputs and $s = s(n)$ be a size parameter. Let $\varepsilon \in \left(\frac{\log s}{n}, 1\right)$ and d be a constant. For $s = 2^{o(n)}$, every s-size LT_d circuit is equivalent to a DOR $\circ$ $\widehat{\mathsf{LT}}_d$ circuit such that:*

- *The top DOR gate has $2^{O(\varepsilon \cdot n)}$ fan-in.*
- *Each sub $\widehat{\mathsf{LT}}_d$ circuit has size $O\left(s^{O(1/\varepsilon)}\right)$.*

Proof. We apply Lemma 17 to the top 2 layers, and then apply item (5) of Proposition 18 recursively to obtain an equivalent $\mathsf{DOR} \circ \widehat{\mathsf{LT}}_d$ circuit. ◀

C.2 A Structure Lemma for Polynomial Threshold Functions

Our ideas can also be used to derive a structure lemma for polynomial threshold functions of degree k, i.e., $\mathsf{THR} \circ \mathsf{AND}_k$ circuits:

▶ **Corollary 52.** *Let n be number of inputs and $s = s(n)$ be a size parameter. Let $\varepsilon \in \left(\frac{\log s}{n}, 1\right)$ and k be a constant. Assuming $s = 2^{o(n)}$, an s-size $\mathit{THR} \circ \mathit{AND}_k$ circuit is equivalent to a $\mathit{DOR} \circ \mathit{MAJ} \circ \mathit{AND}_{2k}$ circuit such that:*

- *The top DOR gate has $2^{O(\varepsilon \cdot n)}$ fan-in.*
- *Each sub $\mathit{MAJ} \circ \mathit{AND}_{2k}$ circuit has size $O\left(s^{O(1/\varepsilon)}\right)$.*

The above still holds if we replaced both AND_k and AND_{2k} by unbounded fan-in AND gates.

Proof. We simply apply Lemma 43 and Lemma 45, and merge each $\mathsf{AND}_2 \circ \mathsf{AND}_k$ subcircuits into a single AND_{2k} gate. ◀

That is, every polynomial threshold function of degree k with arbitrary weights can be simulated by a *subexponential-size* disjoint OR of polynomial threshold functions of degree $2k$ with small weights.

The following corollary follows from that the SAT problem for $\mathsf{THR} \circ \mathsf{AND}_k$ circuits is equivalent to the *weighted* MAX-k-SAT problem (given a CNF formula φ with weights on each clause, find an assignment satisfying clauses of maximum total weight), and that SAT for $\mathsf{MAJ} \circ \mathsf{AND}_{2k}$ is equivalent to the (unweighted) MAX-$2k$-SAT problem.

▶ **Corollary 53.** *For any integer k, if there is a $2^{(1-\Omega(1))n}$ time algorithm for polynomial size unweighted MAX-$2k$-SAT, then so does polynomial size weighted MAX-k-SAT.*[18]

To prove the above corollary, we need the following folklore lemma, which helps us to transform between $\mathsf{MAJ} \circ \mathsf{AND}$ circuits and $\mathsf{MAJ} \circ \mathsf{OR}$ circuits.

▶ **Lemma 54.** *Let $x = x_1, x_2, \ldots, x_k$ be the inputs, there are k OR functions $O_1, O_2, \ldots, O_k$ on the inputs (or their negations) such that:*

$$\mathit{AND}(x) = \left(\sum_{i=1}^{k} O_i(x)\right) - (k-1).$$

Proof. We define

$$O_i(x) := \left(\bigvee_{j=1}^{i-1} \neg x_j\right) \vee x_i.$$

That is, $O_i(x) = 0$ if and only if the first $i-1$ bits are 1, and the i-th bit is 0. Now, note that if $\mathsf{AND}(x) = 1$, then all bits are 1, which means all $O_i(x)$'s are 1. When $\mathsf{AND}(x) = 0$, let i be the index of the first 0-bit, it is easy to see that $O_i(x) = 0$ and all other $O_j(x)$'s are 1, and hence $\sum_{i=1}^{k} O_i(x) = k - 1$. ◀

Proof of Corollary 53. We can use Lemma 54 to transform the bottom AND gates to OR gates for $\mathsf{THR} \circ \mathsf{AND}$ and $\mathsf{MAJ} \circ \mathsf{AND}$ circuits. From there, the proof is the same as of Theorem 3. ◀

[18] We assume the weights are at most $2^{\mathrm{poly}(n)}$ for making the input polynomial size.