Electronic Colloquium on Computational Complexity, Report No. 39 (2026)

Super-quadratic Lower Bounds for Depth-2 Linear Threshold Circuits

Lijie Chen * Avishay Tal † Yichuan Wang ‡

2026-03-14

Abstract

Proving lower bounds against depth-2 linear threshold circuits (a.k.a. $\mathsf{THR} \circ \mathsf{THR}$) is one of the frontier questions in complexity theory. Despite tremendous effort, our best lower bounds for $\mathsf{THR} \circ \mathsf{THR}$ only hold for *sub-quadratic* number of gates, which was proven a decade ago by Tamaki (ECCC TR16) and Alman, Chan, and Williams (FOCS 2016) for a hard function in E^{NP}.

In this work, we prove that there is a function $f \in \mathsf{E}^{\mathsf{NP}}$ that requires $n^{2.5-\varepsilon}$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits for any $\varepsilon > 0$. We obtain our new results by designing a new $2^{n-n^{\Omega(\varepsilon)}}$-time algorithm for estimating the acceptance probability of an XOR of two $n^{2.5-\varepsilon}$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits, and apply Williams' algorithmic method to obtain the desired lower bound.

*University of California at Berkeley. Email: lijiechen@berkeley.edu.

†University of California at Berkeley. Email: atal@berkeley.edu. Supported by an NSF CAREER Award CCF-2145474.

‡University of California at Berkeley. Email: yichuan-21@berkeley.edu. Supported by an NSF CAREER Award CCF-2145474.

ISSN 1433-8092

Contents

1 Introduction

Proving unconditional circuit lower bounds for explicit functions (with the flagship problem of $\mathsf{NP} \not\subseteq \mathsf{P/poly}$) is one of the central questions in theoretical computer science. Since the 1980s, considerable progress has been made in proving lower bounds against constant-depth circuit classes, such as exponential lower bounds for AC^0 [Ajt83, FSS84, Yao85, Hås89] (constant-depth circuits of unbounded fan-in AND/OR gates) and for $\mathsf{AC}^0[q]$ [Raz87, Smo87] (AC^0 circuits with MOD_q gates) for prime q.

More recently, progress has been made for $\mathsf{AC}^0[m]$ circuits for composite m. Williams [Wil10, Wil11] proved that the class NEXP does not have polynomial-size $\mathsf{AC}^0[m]$ circuits for any composite m. This was later improved by Murray and Williams [MW18] to NQP (non-deterministic quasi-polynomial time) not in $\mathsf{AC}^0[m]$. The natural next step [Aar16] following Williams' approach is to show lower bounds against TC^0, the class of constant-depth circuits of linear threshold gates.[1]

Circuit Lower Bounds against TC^0. Besides being the frontier class for proving circuit lower bounds, TC^0 also captures the class of functions computable by constant-depth neural networks. Given the ubiquitous applications of neural networks, it is important to understand the power and limitations of TC^0.

Unfortunately, our understanding of TC^0 is quite limited even for the simple depth-2 case: a decade ago, Kane and Williams [KW16] proved that there is a function in P that requires $\mathsf{THR}\circ\mathsf{THR}$ circuits with $n^{1.5-\varepsilon}$ size; this was later improved by [Tam16] and [ACW16] to lower bounds against $\mathsf{THR} \circ \mathsf{THR}$ circuits with $n^{2-\varepsilon}$ size, at the cost of having the hard function in E^{NP} instead of P. In particular, it remained an open question whether there exists $f \in \mathsf{E}^{\mathsf{NP}}$ that cannot be computed by $n^{2.001}$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits.

Non-trivial circuit analysis of TC^0. In his seminal work [Wil10, Wil11], Williams not only proved a super-polynomial lower bound against $\mathsf{AC}^0[m]$, but also established a general method for proving circuit lower bounds, termed the *algorithmic method*: a non-trivial circuit-analysis algorithm for a circuit class $\mathcal{C}$ immediately implies a new circuit lower bound for $\mathcal{C}$. Here, the circuit-analysis algorithm can be for either the *satisfiability* (SAT) problem or the *Circuit Acceptance Probability Problem* (CAPP);[2] and by non-trivial we mean the running time should be at most $2^n/n^{\omega(1)}$ (here n is the number of input bits to the circuit C), only slightly better than brute-force search. His lower bound then followed from his new SAT algorithm for $\mathsf{AC}^0[m]$ circuits.

Following Williams' framework, many subsequent papers obtained non-trivial circuit-analysis algorithms for interesting subclasses of TC^0 circuits [IPS13, Wil14, Tam16, ACW16, CSS16, SSTT15, SSTT16, Tel18, Wil18, CW19, CR20, CLW20, KL18, HHTT21, BKK+22]. However, despite much effort in designing non-trivial algorithms for TC^0 circuits, prior to this work, we still do not have any non-trivial circuit-analysis algorithms for *super-quadratic-size* $\mathsf{THR} \circ \mathsf{THR}$ circuits (say, size $n^{2.001}$) for either SAT or CAPP. In particular, the aforementioned lower bounds against $n^{2-\varepsilon}$-size $\mathsf{THR} \circ \mathsf{THR}$ [Tam16, ACW16] were proven by designing corresponding circuit-analysis algorithms for *sub-quadratic-size* $\mathsf{THR} \circ \mathsf{THR}$ circuits.

[1] A function $f\colon \{0,1\}^n \to \{0,1\}$ is a *linear threshold function* (LTF) if there are weights $w_1, \ldots, w_n, t \in \mathbb{R}$ such that $f(x) = 1$ if and only if $\sum_{i=1}^n w_i \cdot x_i \geq t$.

[2] Given an input circuit $C\colon \{0,1\}^n$: the SAT problem asks whether there is a satisfying input $x \in \{0,1\}^n$ that makes $C(x) = 1$; the CAPP problem asks to estimate $\Pr_{x\sim\{0,1\}^n}[C(x) = 1]$ within an additive error ε.

1.1 Our Results

In this work, we prove new $\mathsf{THR} \circ \mathsf{THR}$ lower bounds after a nearly decade-long hiatus in progress, by designing new non-trivial circuit-analysis algorithms for $\mathsf{THR} \circ \mathsf{THR}$ circuits.

1.1.1 New circuit lower bounds against $\mathsf{THR} \circ \mathsf{THR}$

Our main result is the following:

Theorem 1.1. *For every constant $\varepsilon \in (0,1)$, there is a function $f \in \mathsf{E}^{\mathsf{NP}}$ that cannot be computed by $n^{2.5-\varepsilon}$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits. The same holds for $\mathsf{SYM} \circ \mathsf{THR}$.*

Comparison with [Tam16] and [ACW16]. Our new lower bound is a direct improvement over [Tam16]'s lower bound.[3] Compared with the lower bound from [ACW16], we note that their lower bound also holds for $\mathsf{AC}^0 \circ \mathsf{THR} \circ \mathsf{THR}$ with sub-exponential-size AC^0 circuits at the top and $n^{2-\varepsilon}$ many THR circuits at the bottom layer, with no restriction on the number of THR gates in the second layer.

1.1.2 New circuit-analysis algorithms for $\mathsf{THR} \circ \mathsf{THR}$

Following [Wil10, Wil11], Chen and Williams [CW19] and Bathie and Williams [BW24] showed that to prove lower bounds against n^{α}-size $\mathsf{THR} \circ \mathsf{THR}$ circuits, it suffices to give a non-trivial algorithm for estimating the acceptance probability of an XOR of two n^{α}-size $\mathsf{THR} \circ \mathsf{THR}$ circuits.

Our main technical result is exactly that: a new CAPP algorithm for $n^{2.5-\varepsilon}$-size $\mathsf{THR} \circ \mathsf{THR}$ circuits that runs in non-trivial time.

Theorem 1.2. *For every constant $\varepsilon \in (0,1)$, there is a deterministic algorithm for solving CAPP for $\oplus_2 \circ \mathsf{THR}_{O(n^{2.5-\varepsilon})} \circ \mathsf{THR}$ with error $o(1)$ that runs in $O\left(2^{n-n^{\Omega(\varepsilon)}}\right)$ time.*

We note that both [Tam16] and [ACW16], indeed obtained non-trivial #SAT algorithms for the corresponding circuit classes, for which one can exactly count the number of accepting inputs to a given circuit C. This is stronger than both CAPP and SAT. We only obtain the weaker CAPP algorithm, which is sufficient nonetheless for the algorithmic method.

2 Technical Overview

Below we will give a high-level overview of the techniques used to prove the main result. First, we quickly review algorithms from prior work and explain why they cannot extend to super-quadratic-size $\mathsf{SYM} \circ \mathsf{THR}$ circuits or $\mathsf{THR} \circ \mathsf{THR}$ circuits.

We will then explain our new algorithm for $\oplus_2 \circ \mathsf{SYM} \circ \mathsf{THR}$ circuits, which essentially covers most of the new ideas that are needed for our new $\mathsf{THR} \circ \mathsf{THR}$ algorithms. Finally, we briefly explain how to generalize that further to $\mathsf{THR} \circ \mathsf{THR}$ circuits.

[3] We remark that [Tam16]'s lower bound also holds for $n^{2-\varepsilon}$-size $\mathsf{SYM} \circ \mathsf{SYM}$ circuits, while our lower bounds do not extend to such a circuit class.

2.1 Review of Prior Works and Potential Improvements

We will first focus on the case of $\mathsf{SYM} \circ \mathsf{THR}$ circuits, since its proof is simpler. It would be instructive to first review the approach in [ACW16, Tam16] that gives non-trivial algorithms for sub-quadratic-size $\mathsf{SYM} \circ \mathsf{THR}$ circuits. Since we are only aiming for a CAPP algorithm, while prior work gave a $\#\mathsf{SAT}$ algorithm, we provide a simplified sketch of their approach sufficient for CAPP.

The algorithm from [ACW16, Tam16]. Given an $\mathsf{SYM} \circ \mathsf{THR}$ circuit C of size $m = n^{2-\varepsilon}$. Let the top SYM gate be $S\colon \{0,1\}^m \to \{0,1\}$ and the list of bottom THR gates be $T = \{T_1, \ldots, T_m\}$. We also write $T(x) = (T_1(x), \ldots, T_m(x))$ for any $x \in \{0,1\}^n$, so we have that $C(x) = S(T(x))$ for all $x \in \{0,1\}^n$.

The key technical ingredient in [ACW16, Tam16] is that an m-size SYM gate can be approximated by an $\widetilde{O}(\sqrt{m})$-degree probabilistic polynomial $\{P_r\}_{r \in \{0,1\}^{\mathsf{polylog}(m)}}$ over $\mathbb{F}_2$. Here r denotes the randomness used to sample the $\mathbb{F}_2$-polynomial and one only needs $\mathsf{polylog}(m)$ many random bits for that.

So we have that for any $u \in \{0,1\}^m$,

$$\Pr_{r \sim \{0,1\}^{\mathsf{polylog}(m)}}[P_r(u) = S(u)] \geq 1 - 1/n^{\omega(1)}.$$

This also implies that for any $x \in \{0,1\}^n$,

$$\Pr_{r \sim \{0,1\}^{\mathsf{polylog}(m)}}[P_r(T(x)) = S(T(x))] \geq 1 - 1/n^{\omega(1)}.$$

As $\deg(P_r) = \widetilde{O}(\sqrt{m}) \leq n^{1-\varepsilon/3}$ (for n large enough), to solve the CAPP problem for C, it suffices to solve the CAPP problem for $2^{\mathsf{polylog}(m)}$ many $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/3}] \circ \mathsf{THR}$ circuits[4] and take the average.

Circuit-analysis algorithm for $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/3}] \circ \mathsf{THR}$ circuits. There is a standard algorithm for solving even the $\#\mathsf{SAT}$ of $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/3}] \circ \mathsf{THR}$ circuits in non-trivial time, this idea dates back to [Wil11] and is also used in [ACW16].

Take a $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/3}] \circ \mathsf{THR}$ circuit C as input and set $\delta = \varepsilon/10$. The idea is to partition the n-bit input x as (y, z), where $|y| = n^\delta$ and $|z| = n - n^\delta$, and reduce the $\#\mathsf{SAT}$ problem for C to the batch-evaluation problem for another $\mathsf{POLY}_{\mathbb{Z}}[n^{1-\varepsilon/6}] \circ \mathsf{THR}$ circuit D on the $(n - n^\delta)$-bit input z.[5]

Let $\mathsf{Amp}_{2n^\delta}(z)$ be the modulus amplification polynomial with $O(2n^\delta)$ degree such that if $z \equiv 0 \pmod 2$, then $\mathsf{Amp}_{2n^\delta}(z) \equiv 0 \pmod{2^{2n^\delta}}$, and if $z \equiv 1 \pmod 2$, then $\mathsf{Amp}_{2n^\delta}(z) \equiv 1 \pmod{2^{2n^\delta}}$. Let $P\colon \{0,1\}^m \to \{0,1\}$ be the $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/3}]$ gate at the top of the circuit. We can also consider the corresponding $\mathbb{Z}$-polynomial $P_{\mathbb{Z}}$ over $\mathbb{Z}$, such that $P_{\mathbb{Z}}(u) = P(u) \pmod 2$ for $u \in \{0,1\}^m$.

Then we can construct the following circuit $D\colon \{0,1\}^{n-n^\delta} \to \mathbb{Z}$ such that for any $z \in \{0,1\}^{n-n^\delta}$,

$$D(z) = \sum_{y \in \{0,1\}^{n^\delta}} \mathsf{Amp}_{2n^\delta}(P_{\mathbb{Z}}(T(y; z))).$$

Note that $D(z) \mod 2^{2n^\delta}$ would give us the number of $y \in \{0,1\}^{n^\delta}$ such that $C(y; z) = 1$. This means that D is a $\mathsf{POLY}_{\mathbb{Z}}[n^{1-\varepsilon/6}] \circ \mathsf{THR}$ polynomial over $n - n^\delta$ bits. An algorithm from Williams can be used to compute its truth-table in essentially 2^{n-n^δ} time, and this would be enough to solve the $\#\mathsf{SAT}$ problem for C.

[4] $\mathsf{POLY}_{\mathbb{F}_2}[d]$ means polynomials over $\mathbb{F}_2$ of degree $\leq d$.

[5] In the batch-evaluation problem, given an input circuit D with m-bit input, the goal is to evaluate D on all possible 2^m many inputs.

Combining with random restriction? Reviewing the above, for each $z \in \{0,1\}^{n-n^\delta}$, we solved the #SAT problem for the circuit $C(\cdot; z)$ on the n^δ-bit input y. Using modulus amplifying polynomial, this can be converted to the batch-evaluation problem for the circuit $D(z)$ above on $n - n^\delta$-bit inputs.

This reminds us of a classical idea in complexity theory: random restrictions [Sub61]. In particular, if we think about y as the "alive" variables, and the z as the variables to be fixed randomly during the restriction, then following [KW16], we can show that if we pick the positions of y randomly (i.e., we pick a random subset $I \in \binom{[n]}{n^\delta}$ and let y be the bits from I), then with probability $1 - n^{-\varepsilon/3}$ over the choice of z, there are only $n^{2-\varepsilon/2}$ many *non-constant* gates in $T_1, \ldots, T_m$.

That is, for most fixed z, we have that $C(y; z)$, as a function in $y \in \{0,1\}^{n^\delta}$, is indeed an $\mathsf{SYM} \circ \mathsf{THR}$ circuit of only $n^{2-\varepsilon/2}$ size. One may hope this observation can be combined with prior techniques to solve the CAPP problem for C.

The obvious difficulty. The obvious difficulty is that for different $z \in \{0,1\}^{n-n^\delta}$, the resulting $\mathsf{SYM} \circ \mathsf{THR}$ circuit is different, and there could be 2^{n-n^δ} many such circuits; this is clearly too many to handle efficiently in our algorithm.

However, the key observation of our work is that although there are many possible sub-quadratic-size $\mathsf{SYM} \circ \mathsf{THR}$ circuits for different z, by opening up the construction in [ACW16, AW15], there are only quasi-polynomially distinct "types" of such circuits, and these can be used to design a non-trivial algorithm.

2.2 Our New Algorithm for $\oplus_2 \circ \mathsf{SYM} \circ \mathsf{THR}$ Circuits

Let $I \in \binom{[n]}{n^\delta}$ be a subset such that for a $1 - n^{-\varepsilon/3}$ fraction of $z \in \{0,1\}^{n-n^\delta}$, there are only $n^{2-\varepsilon/2}$ many non-constant gates in $T_1(\cdot; z), \ldots, T_m(\cdot; z)$. There is a simple deterministic algorithm that runs in $2^{0.9n}$-time to find such an I. So in the following we will assume such an I is fixed.

Moreover, we say a $z \in \{0,1\}^{n-n^\delta}$ is *good* if there are only $n^{2-\varepsilon/2}$ many non-constant gates in $T_1(\cdot; z), \ldots, T_m(\cdot; z)$. Note that given z, it is easy to check whether a THR gate $T_i(\cdot; z)$ is constant or not in polynomial time. So we can also check whether z is good or not in polynomial time.

In the following, we will aim to solve the CAPP problem for the circuits $C(\cdot; z)$, but only for the good z's. Our algorithm would just "give up" on the bad z's. As a CAPP algorithm for the circuit C over $\{0,1\}^n$, this would incur an additive error of $1/n^{\varepsilon/3}$, which is acceptable.

For different good $z \in \{0,1\}^{n-n^\delta}$, the resulting $\mathsf{SYM} \circ \mathsf{THR}$ circuit $C(\cdot; z)$ is different, so it may seem that for each of them we will have to compute a different $\mathbb{F}_2$-probabilistic polynomial, ending up with 2^{n-n^δ} many different $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\Omega(\varepsilon)}] \circ \mathsf{THR}$ circuits to handle, which is clearly too much.

Our crucial observation is that, by opening up the construction in [ACW16, AW15], there are only quasi-polynomially many $\mathbb{F}_2$-probabilistic polynomials that one has to consider, even if there are 2^{n-n^δ} many good z's.

Now, for each good z, note that the range of $T(y; z)$ is contained in a subcube X_z of dimension at most $n^{2-\varepsilon/2}$. This motivates us to approximate the symmetric gate S by a low-degree polynomial on the subcube X_z.

More specifically, let $X \subseteq \{0,1\}^m$ be a subcube of dimension $n^{2-\varepsilon/2}$. That is, $n^{2-\varepsilon/2}$ many bits from $x \in X$ can be arbitrarily chosen from 0 or 1, and other bits are fixed. Let r be the random bit string in the construction.[6] There are $n^{10 \log n}$ many $\mathbb{F}_2$ polynomials $\{P_{r,a}\}_{a \in [n^{10 \log n}]}$ of

[6] We choose $n^{\varepsilon/100}$ random bits instead of $\mathsf{polylog}(m)$ as we can afford it, but both choices would give the same end result.

degree $n^{1-\varepsilon/6}$ and an "advice" function that takes as input the description of X and r and outputs $a(X,r) \in [n^{10\log n}]$, such that for every X of dimension $n^{2-\varepsilon/2}$ and every $x \in X$, we have

$$\Pr_r\left[S(x) = P_{r,a(X,r)}(x))\right] \geq 1 - 1/n^{\omega(1)}. \tag{1}$$

Since there are very few possible r, we can simply enumerate all of them with a small overhead. Thus, in the following, we will just pretend that r is fixed and assume for simplicity of exposition that the error in Equation (1) is 0. Then, there is a list of polynomials $\{P_a\}_{a\in[n^{10\log n}]}$ such that for every $x \in X$, we have

$$P_{a(X)}(x) = S(x).$$

Since for each good z, the range of $T(y;z)$ is contained in a subcube X_z of dimension $n^{2-\varepsilon/2}$, we get that $P_{a(X_z)} \circ T(\cdot;z)$ will be the same as $S \circ T(\cdot;z)$.

Using the same algorithm as in prior work, for every $a \in [n^{10\log n}]$, we can solve the #SAT problem for the circuit $P_a \circ T(\cdot;z)$, for every $z \in \{0,1\}^{n-n^\delta}$ in essentially 2^{n-n^δ} time, and store it in a lookup table. Then, for each good z, it suffices to simply look at the result for $P_{a(X_z)} \circ T(\cdot;z)$ in the lookup table. This would be enough to solve the CAPP problem for SYM $\circ$ THR.

Note that above we assumed that the list of polynomials, as well as the advice function $a(X,r)$, can be computed in $2^{n-n^{\Omega(\varepsilon)}}$ time. We will discuss how to do that in detail later.

Generalization to $\oplus_2 \circ$ SYM $\circ$ THR circuits. To generalize to $\oplus_2 \circ$ SYM $\circ$ THR circuits, we still use $T_1,\ldots,T_m$ as the list of bottom THR gates, but now we have two top SYM gates $S_1, S_2 \colon \{0,1\}^m \to \{0,1\}$.

We can get two lists of polynomials $\{P_a\}_{a\in[n^{10\log n}]}$ and $\{Q_b\}_{b\in[n^{10\log n}]}$ such that for every $x \in X$, we have $P_{a(X)}(x) = S_1(x)$ and $Q_{b(X)}(x) = S_2(x)$, where X is a subcube of dimension $n^{2-\varepsilon/2}$.

Now, for each good z, we have that $(P_{a(X_z)} \oplus Q_{b(X_z)}) \circ T(\cdot;z)$ is the same as $(S_1 \oplus S_2) \circ T(\cdot;z)$.

To solve the CAPP problem for $\oplus_2 \circ$ SYM $\circ$ THR circuits, for every $a, b \in [n^{10\log n}]$, we can solve the #SAT problem for the circuit $(P_a \oplus Q_b) \circ T(\cdot;z)$, for every $z \in \{0,1\}^{n-n^\delta}$ in essentially 2^{n-n^δ} time. This enumeration of (a,b) incurs only a minor overhead to the runtime. Then, for each good z, it suffices to simply look up the result for $(P_{a(X_z)} \oplus Q_{b(X_z)}) \circ T(\cdot;z)$. This suffices to solve the CAPP problem for $\oplus_2 \circ$ SYM $\circ$ THR circuits.

Generalization to THR $\circ$ THR circuits. To generalize to THR $\circ$ THR circuits, the key idea is to notice that the top THR gate can be written as an approximate linear sum of $\mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ$ SYM gates; we can essentially use the same algorithm above, but now with $O(1)$ many SYM gates instead of two.

2.3 List approximating $\mathbb{F}_2$-probabilistic polynomials for SYM gates

Finally, we explain how to construct the $\mathbb{F}_2$-probabilistic polynomials $\{P_{r,a}\}_{r,a}$. We first review the construction of $\mathbb{F}_2$-probabilistic polynomials for SYM gates from [ACW16, AW15]. To construct an $\mathbb{F}_2$-probabilistic polynomial for a SYM gate, it suffices to construct an $\mathbb{F}_2$-probabilistic polynomial for the following function 1hotSUM$\colon \{0,1\}^m \to \{0,1\}^{m+1}$. Here, 1hotSUM takes m input bits, computes their sum s, and outputs $z_0, z_1, \ldots, z_m$ defined by $z_i = 1$ if and only if $i = s$ (that is, the output is the one-hot vector encoding of the sum s).

Our goal is to construct a family of $\mathbb{F}_2$-polynomials $\{P_r^{(m)} \colon \mathbb{F}_2^m \to \mathbb{F}_2^{m+1}\}$ such that for every $x \in \{0,1\}^m$, with probability at least $1 - 1/n^{\omega(1)}$ over the choice of r, we have $P_r^{(m)}(x) =$ 1hotSUM(x). For $i \in \{0,\ldots,m\}$, we use $P_r^{(m)}[i]$ to denote the i-th coordinate of $P_r^{(m)}(x)$.

The $\mathbb{F}_2$-probabilistic polynomial from [AW15]. Alman and Williams [AW15] does so recursively. For simplicity assume that m is a power of 2. Suppose we already have the construction $P^{(m/2)}$ for $\mathsf{1hotSUM}$ on $m/2$ bits. Then we can construct $P^{(m)}$ for $\mathsf{1hotSUM}$ on m bits as follows:

1. Let $x \in \{0,1\}^m$ be the input.
2. Sample a random subset $K \subseteq [m]$ of size $m/2$ and apply $P^{(m/2)}(x_K)$ to know how many 1's are in x_K. Since K is (pseudo-)randomly chosen, if there are t 1's in x_K, with probability $1 - n^{-\omega(1)}$ the number of 1's in x should belong to the range $2t \pm \sqrt{m}\log m$.
3. Now, note that for every $t' \in \{0,1,\ldots,m\}$ and $s \in \{0,1,\ldots,m\}$, we can interpolate an $O(\sqrt{m}\log m)$-degree polynomial $E_{s,t'}$, such that for every $u \in t' \pm \sqrt{m}\log m$, we have that $E_{s,t'}(u) = 1$ if $u = s$ and $E_{s,t}(u) = 0$ otherwise. ($E_{s,t'}$ is all zero if $s \notin t' \pm \sqrt{m}\log m$.)

   We can then define

$$P^{(m)}[s](x) = \sum_{t=0}^{m/2} P^{(m/2)}[t](x_K) \cdot E_{s,2t}\left(\sum_{i=1}^{m} x_i\right).$$

Note that the above works because with high probability over the randomness, $P^{(m/2)}$ computes $\mathsf{1hotSUM}$ on x_K, and the number of 1's in the whole input x falls in the interval $2t \pm \sqrt{m}\log m$, where t is the exact number of 1's in the input x_K. The original construction from [AW15] picks the subset K uniformly at random, which was later derandomized by [ACW16] using $O(\log n)$-wise independent distribution.

Adapting the analysis to subcubes. Now, let X be an $\ell = n^{2-\varepsilon/2}$ dimensional subcube of $\{0,1\}^m$. We proceed as above to construct the probabilistic polynomial $P^{(m)}$ that computes $\mathsf{1hotSUM}$ for $x \in X$. Let $J \in \binom{[m]}{\ell}$ be the set of coordinates that is free in X.

1. Let $x \in X$ be the input.
2. Sample a random subset $K \subseteq [m]$ of size $m/2$ and apply $P^{(m/2)}(x_K)$ to know how many 1's are in x_K. Suppose it is t. Note that $x_{[m]\setminus J}$ is always constant (same for all $x \in X$), so our goal is indeed to obtain an estimate of the number of 1's in x_J; for which we can compute the number of 1's in $x_{K\cap J}$, which is $t - ||x_{K\setminus J}||_1$, where note that $x_{K\setminus J}$ is also always constant.

   Now the estimation error only comes from estimating $\|x_J\|_1$ using $2\|x_{K\cap J}\|_1$, where $|J| = \ell$. So our new range is

$$t' \pm \sqrt{\ell}\log n \quad \text{where } t' := 2(t - ||x_{K\setminus J}||_1) + ||x_{[m]\setminus J}||_1$$

   Note that t' depends on t and $-2||x_{K\setminus J}||_1 + ||x_{\bar{J}}||_1$, where the latter is a number in $[-m, m]$ that only depends on K (which is generated from the randomness r) and the subcube X. This number will be part of the advice.

   For any $x \in X$, since K is (pseudo-)randomly chosen, if there are t 1's in x_K, then with probability $1 - n^{-\omega(1)}$ the number of 1's in x should belong to $t' \pm \sqrt{\ell}\log n$.
3. So, for every $t' \in \{0,1,\ldots,m\}$, given $-2||x_{K\setminus J}||_1 + ||x_{[m]\setminus J}||_1$ and $s \in \{0,1,\ldots,m\}$, we can interpolate an $O(\sqrt{\ell}\log n)$-degree polynomial $E_{s,t}$, such that for every $u \in t' \pm \sqrt{\ell}\log n$, $E_{s,t}(u) = 1$ if $u = s$ and $E_{s,t}(u) = 0$ otherwise. ($E_{s,t}$ is all zero if $s \notin t' \pm \sqrt{\ell}\log n$.)

We can then define

$$P^{(m)}[s](x) = \sum_{t=0}^{m/2} P^{(m/2)}[t](x_K) \cdot E_{s,2t-2||x_{K\setminus J}||_1+||x_{\bar{J}}||_1}\left(\sum_{i=1}^{m} x_i\right).$$

The argument would work similarly as before. The only difference is that now the range is shifted by a number $-2||x_{K\setminus J}||_1 + ||x_{\bar{J}}||_1$ that depends on K and the subcube X. We would encode this number to be part of the advice $a(X, r)$, where r denotes the randomness used to sample the different K's. Note that the advice can be easily calculated from K and a concise description of X, specifying whether each coordinate is fixed to 0, 1, or is free. Since there are $O(\log n)$ layers of recursion, the list of shifts above for one particular subcube X has only $(2m+1)^{O(\log n)}$ options and can be encoded as an integer $a(X, r) \in [n^{10 \log n}]$.

3 Preliminaries

We first define some standard notation. For any mathematical statement α, let $\mathbb{I}[\alpha]$ be the indicator that α holds, i.e., it equals 1 if α holds and 0 otherwise. Define the L_1-distance of two vectors $(a_1, \cdots, a_d)$ and $(b_1, \cdots, b_d)$ by $\sum_{i=1}^{d} |a_i - b_i|$.

3.1 Gates and Circuit Classes

We first define the following gates we will use.

Definition 3.1. (Gates) *We define the following gates (on input $(x_1, x_2, \cdots, x_m) \in \{0,1\}^m$):*

- **1hotSUM:** *This gate has $m+1$ output wires, the i-th wire ($i = 0, 1, \cdots, m$) outputs 1 iff there are exactly i 1-s in $x_1, \cdots, x_m$.*
- **THR:** *This gate has parameters $w_1, w_2, \cdots, w_m, t \in \mathbb{R}$, and outputs*

  $$\mathbb{I}[w_1x_1 + w_2x_2 + \cdots + w_mx_m \geq t].$$

  (Replacing $\geq$ by $>$, $<$ or $\leq$ gives the same definition.)
- **ETHR:** *This gate has parameters $w_1, w_2, \cdots, w_m, t \in \mathbb{R}$, and outputs*

  $$\mathbb{I}[w_1x_1 + w_2x_2 + \cdots + w_mx_m = t].$$

- **SYM:** *This gate outputs a bit that only depends on $x_1 + \cdots + x_m$.*
- **POLY$_{\mathbb{F}_2}[d]$:** *This gate evaluates some m-variable polynomial over $\mathbb{F}_2$ with degree at most d.*
- **DOR:** *It is guaranteed that $x_1, \cdots, x_m$ is either all 0 or contains exactly one 1. This gate outputs 1 iff there is exactly one 1.*
- **GapAND$_\delta$:** *$0 < \delta < 1$ is some parameter. It is guaranteed that $x_1, \cdots, x_m$ is either all 1 (in which case it outputs 1), or contains at most δ fraction of 1's (in which case it outputs 0).*
- **$\wedge, \vee, \oplus$:** *Multi fan-in AND, OR, XOR gates*
- **$\wedge_k, \vee_k, \oplus_k$:** *AND, OR, XOR gates with fan-in k.*

- $\widetilde{\mathsf{SUM}}$: *This gate has two disjoint intervals I_0, I_1 as parameters, and it is guaranteed that the number of 1-s in the input is in $I_0 \cup I_1$. $\widetilde{\mathsf{SUM}}$ outputs 0 if the number of 1-s in the input is in I_0, and 1 if in I_1.*

Lemma 3.2 ([MTT61])**.** *For any* THR *gate on m input bits, there is an equivalent* THR *gate where all parameters are integers in range $\pm m^m$.*

Definition 3.3. (Constant Depth Circuits) *For any gate types or circuit classes $\mathscr{G}_1, \mathscr{G}_2, \cdots, \mathscr{G}_d$, denote $\mathscr{G}_1 \circ \mathscr{G}_2 \circ \cdots \circ \mathscr{G}_d$ by circuits with $\mathscr{G}_1, \mathscr{G}_2, \cdots, \mathscr{G}_d$ gates on the $1, 2, \cdots, d$-th layers, where layer 1 is the layer closest to the output.*

3.2 Structure Lemmata for Threshold Circuits

Lemma 3.4. *For circuit classes $\mathscr{A}, \mathscr{B}$. We denote $\mathscr{A} \subseteq \mathscr{B}$ by: there is a polynomial time deterministic algorithm that, given an $\mathscr{A}$-circuit C_1 with n input bits and $\mathrm{poly}(n)$ size, outputs an equivalent $\mathscr{B}$-circuit C_2. (Therefore, C_2's size is at most $\mathrm{poly}(n)$.) We have:*

1. $\wedge_{\mathrm{poly}(n)} \circ \mathsf{ETHR} \subseteq \mathsf{ETHR}$*; [HP10]*
2. $\mathsf{THR} \subseteq \mathsf{DOR} \circ \mathsf{ETHR}$*; [HP10, CW19]*
3. *For any constant $c > 0$, $\mathsf{ETHR} \subseteq \mathsf{GapAND}_{1/n^c} \circ \mathsf{SYM}$; [CW19]*
4. *For any $\varepsilon > 0$, $\mathsf{SYM} \subseteq \mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM}$, where the* SYM *gate on the left hand side may have $\mathrm{poly}(n)$ many wires to each input gate, but on the right hand side, each* 1hotSUM *gate has at most n^ε wires to each input bit, and there are at most $O(1)$ many* 1hotSUM *gates. (Here $O(1)$ may depend on ε and the exponent of n in* SYM*'s size.) [ACW16]*

For Item 4, the construction in Theorem 7.1 of [ACW16] has those properties, although not explicitly stated. We give a proof in Appendix A.1 for completeness.

3.3 Circuit Analysis Problems

Definition 3.5. (#SAT) *For any circuit class $\mathscr{A}$ (with binary output), $\mathscr{A}$-#SAT is the following problem: given an $\mathscr{A}$-circuit C on n input bits, we need to output $\Pr_{x\sim\{0,1\}^n}[C(x) = 1]$.*

Definition 3.6. (CAPP) *For any circuit class $\mathscr{A}$ (with binary output) and a real number $0 < \delta < 1$, $\mathscr{A}$-CAPP_δ is the following problem: given an $\mathscr{A}$-circuit C on n input bits, we need to estimate $\Pr_{x\sim\{0,1\}^n}[C(x) = 1]$ with additive error $\leq \delta$.*

Definition 3.7. (L_1-CAPP for circuits with top 1hotSUM gates) *For any circuit class $\mathscr{A}$ with a* 1hotSUM *gate at the top and a real number $0 < \delta < 1$, $\mathscr{A}$-L_1-CAPP_δ is the following problem: given an $\mathscr{A}$-circuit C on n input bits, suppose C's top* 1hotSUM *gate has m inputs and $m+1$ outputs, we need to estimate the vector*

$$\left(\Pr_{x\sim\{0,1\}^n}[C(x) = \boldsymbol{e}_0], \Pr_{x\sim\{0,1\}^n}[C(x) = \boldsymbol{e}_1], \cdots, \Pr_{x\sim\{0,1\}^n}[C(x) = \boldsymbol{e}_m]\right)$$

with additive error $\leq \delta$ in L_1-distance. Here $C(x) = \boldsymbol{e}_k$ means only the k-th output of $C(x)$ is 1, i.e., the top 1hotSUM *gate has k 1's in its inputs.*

We also need the following connection between algorithms and circuit lower bounds [CW19, BW24].

Lemma 3.8 ([BW24, Theorem 3.7, Section 3.3]). *Let $\alpha > 1$. If the* CAPP *with error $o(1)$ for $O(n^\alpha)$-size $\oplus_2 \circ \mathsf{THR} \circ \mathsf{THR}$ circuit can be solved in $2^n/n^{\omega(1)}$ time, then E^{NP} does not have n^α-size $\mathsf{THR} \circ \mathsf{THR}$ circuits.*

The same also holds if we replace $\mathsf{THR} \circ \mathsf{THR}$ by $\mathsf{SYM} \circ \mathsf{THR}$.

Proof Sketch. Theorem 3.7 of [BW24] and its generalization in Section 3.3 require that (i) n^α-size $\mathsf{THR} \circ \mathsf{THR}$ can compute parity (which is true for $\alpha \geq 1$) and (ii) a CAPP algorithm for $O(n^\alpha)$-size $\mathsf{OR}_3 \circ \mathsf{THR} \circ \mathsf{THR}$. The second requirement can be relaxed to $\oplus_2 \circ \mathsf{THR} \circ \mathsf{THR}$ by using the 2-query PCPP specified in [CW19, Lemma 25] instead of the 3-query PCPP used in [BW24].

The argument above also holds for $\mathsf{SYM} \circ \mathsf{THR}$. □

3.3.1 Known Circuit Analysis Algorithms

Lemma 3.9 (Folklore). *There is a deterministic $2^{n/2} \cdot \mathrm{poly}(n)$-time algorithm for* THR-#SAT.

We provide a proof of Lemma 3.9 in Appendix A.2 for completeness.

Lemma 3.10 ([Wil14]). *The truth table of an n-input $2^{n/100}$-size $\mathsf{1hotSUM} \circ \mathsf{ETHR}$ or $\mathsf{1hotSUM} \circ \mathsf{THR}$ circuit can be printed in deterministic $2^n \cdot \mathrm{poly}(n)$ time.*

3.4 Chernoff Bound for k-wise Independent Variables

Lemma 3.11 ([SSS95]). *For any n k-wise independent random variables $X_1, \cdots, X_n \in [0,1]$, denote $X := X_1 + X_2 + \cdots + X_n$ and $\mu := \mathbb{E}[X]$, then for any $\delta > 0$, we have*

$$\Pr\left[|X - \mu| \geq \delta\mu\right] \leq e^{-\Omega\left(\min\{k, \delta\mu, \delta^2\mu\}\right)}.$$

3.5 Modulus Amplification Polynomials

Lemma 3.12 ([BT91]). *For any positive integer k, there exists a degree $2k-1$ polynomial $\mathsf{Amp}_k(x) \in \mathbb{Z}[x]$ such that for any integer $m > 1$ and any integer x, if $x \equiv 0 \pmod m$, then $\mathsf{Amp}_k(x) \equiv 0 \pmod{m^k}$; if $x \equiv 1 \pmod m$, then $\mathsf{Amp}_k(x) \equiv 1 \pmod{m^k}$. Moreover, there is a deterministic $\mathrm{poly}(k)$-time algorithm that, on input k, outputs Amp_k.*

4 Simplification of THR Gates Under Random Restrictions

From now on, let $\varepsilon \in (0, 10^{-9})$ be a global constant.

Lemma 4.1 ([KW16]). *Let $n > k$ be positive integers. Let $f\colon \{0,1\}^n \to \{0,1\}$ be a linear threshold function. Then for a random restriction $\rho \in \{0,1,*\}^n$, which has exactly k many $*$-s, with probability $\geq 1 - O\left(k/\sqrt{n}\right)$, $f|_\rho$ is a constant function.*

We note that [KW16] proved a slightly different version of Lemma 4.1, where they handle any equi-partition of $[n]$ into k subsets and pick one $*$ from each subset uniformly at random. Their statement easily implies ours, as one can get the uniform distribution over sets of size k out n by first randomly taking an equi-partition of $[n]$ to k non-empty parts and then picking one element in each part. We include an alternative proof of Lemma 4.1 in Appendix A.3 for completeness.

Next, we split the circuit's n input bits into two parts, $I \subset [n]$ of size $k = n^{\varepsilon/10}$ and $[n] \setminus I$ of size $n - k$. Based on this partition, we can arrange the 2^n possible input strings to the circuit in a $2^k \times 2^{n-k}$ matrix, where the row and column determine the value for x_I and $x_{[n]\setminus I}$ respectively. We then use Lemma 4.1 to argue that for a typical choice of I, for most columns, many THR gates become constant on all inputs in this column.

Definition 4.2. *For positive integers $n \geq k$, and a set $I \in \binom{[n]}{k}$, (we call it a "partition",) split any $x \in \{0,1\}^n$ into two parts: $x_I \in \{0,1\}^I$ and $x_{[n]\setminus I} \in \{0,1\}^{[n]\setminus I}$, and we denote x by $(x_I, x_{[n]\setminus I})$. We partition $\{0,1\}^n$ to 2^{n-k} columns:*

$$\mathsf{Column}_z^{(I)} := \big\{(y,z)\colon y \in \{0,1\}^I\big\}, \quad \text{for } z \in \{0,1\}^{[n]\setminus I}.$$

Definition 4.3. *Given a collection of $m \leq O(n^{2.5-\varepsilon})$ linear threshold gates*

$$\mathcal{C} = \big\{T_i\colon \{0,1\}^n \to \{0,1\}\big\}_{i\in[m]},$$

and a partition $I \in \binom{[n]}{n^{\varepsilon/10}}$. For any $z \in \{0,1\}^{[n]\setminus I}$, we say $\mathsf{Column}_z^{(I)}$ is $\mathcal{C}$-good, if at most $n^{2-\varepsilon/2}$ gates in $\mathcal{C}$ are non-constant on $\mathsf{Column}_z^{(I)}$.

We show that it is easy to tell in each column, whether each THR gate is constant 0 or 1 or non-constant.

Theorem 4.4. *There is a deterministic $O\Big(2^{n-n^{\varepsilon/10}} \cdot \mathrm{poly}(n)\Big)$-time algorithm that, given a collection of $m \leq O\big(n^{2.5-\varepsilon}\big)$ linear threshold gates*

$$\mathcal{C} = \big\{T_i\colon \{0,1\}^n \to \{0,1\}\big\}_{i\in[m]}$$

and a partition $I \in \binom{[n]}{n^{\varepsilon/10}}$, outputs for each $z \in \{0,1\}^{[n]\setminus I}$ and $i \in [m]$, whether T_i is always 0 or always 1 or non-constant on $\mathsf{Column}_z^{(I)}$.

Proof. For each $z \in \{0,1\}^{[n]\setminus I}$ and $i \in [m]$, suppose $T_i(x) = \mathbb{I}\left[\sum_{j=1}^n w_{i,j}x_j \geq t_j\right]$, then we can determine the maximum and minimum of $\sum_{j=1}^n w_{i,j}x_j$ on $\mathsf{Column}_z^{(I)}$ by setting each x_j ($j \in I$) to be 0 or 1 based on the sign of $w_{i,j}$. And thus figure out whether T_i is always 0 or always 1 or non-constant on $\mathsf{Column}_z^{(I)}$. This can be computed in $2^{n-n^{\varepsilon/10}} \cdot \mathrm{poly}(n)$ time. □

Below we present a deterministic algorithm for finding a partition where most columns are good.

Theorem 4.5. *There is a deterministic $2^{n/2+o(n)}$-time algorithm that, given a collection of $m \leq O(n^{2.5-\varepsilon})$ linear threshold gates*

$$\mathcal{C} = \big\{T_i\colon \{0,1\}^n \to \{0,1\}\big\}_{i\in[m]},$$

outputs a partition $I_0 \in \binom{[n]}{n^{\varepsilon/10}}$ such that at least $(1 - 1/n^{\varepsilon/3})$ fraction of the $\mathsf{Column}_z^{(I_0)}$-s ($z \in \{0,1\}^{[n]\setminus I_0}$) are $\mathcal{C}$-good.

Proof. **The existence of a good I_0.** By Lemma 4.1, for each $i \in [m]$, we have

$$\Pr_{I,z}\Big[T_i \text{ is non-constant on } \mathsf{Column}_z^{(I)}\Big] \leq 1/n^{1/2-\varepsilon/8},$$

where $\Pr_{I,z}$ samples a random $I \in \binom{[n]}{n^{\varepsilon/10}}$ and a random $z \sim \{0,1\}^{[n]\setminus I}$. Thus,

$$\Pr_{i\sim[m]}\Pr_{I,z}\Big[T_i \text{ is non-constant on } \mathsf{Column}_z^{(I)}\Big] \leq 1/n^{1/2-\varepsilon/8}. \tag{2}$$

Define, for each $I \in \binom{[n]}{n^{\varepsilon/10}}$ and $i \in [m]$,

$$a_{I,i} := \#\left\{z \in \{0,1\}^{[n]\setminus I} \colon T_i \text{ is non-constant on } \mathsf{Column}_z^{(I)}\right\}. \tag{3}$$

In this terminology, Equation (2) is equivalent to

$$\mathop{\mathbb{E}}_{i\sim[m]} \mathop{\mathbb{E}}_{I} [a_{I,i}] \le 2^{n-n^{\varepsilon/10}}/n^{1/2-\varepsilon/8}.$$

So there exists an I_0 such that

$$\mathop{\mathbb{E}}_{i\sim[m]} [a_{I_0,i}] \le 2^{n-n^{\varepsilon/10}}/n^{1/2-\varepsilon/8}.$$

Next, we show that such I_0 satisfies the requirement. By the definition of $a_{I_0,i}$, we have

$$\Pr_{i\sim[m]} \Pr_{z\sim\{0,1\}^{[n]\setminus I_0}} \left[T_i \text{ is non-constant on } \mathsf{Column}_z^{(I_0)}\right] \le 1/n^{1/2-\varepsilon/8},$$

and therefore,

$$\begin{aligned}
&\Pr_{z\sim\{0,1\}^{[n]\setminus I_0}} \left[\#\left\{i \in [m] \colon T_i \text{ is non-constant on } \mathsf{Column}_z^{(I_0)}\right\} > n^{2-\varepsilon/2}\right] \\
&< \left(m/n^{2-\varepsilon/2}\right) \cdot \left(1/n^{1/2-\varepsilon/8}\right) \\
&\le \left(O\left(n^{2.5-\varepsilon}\right)/n^{2-\varepsilon/2}\right) \cdot \left(1/n^{1/2-\varepsilon/8}\right) \\
&< 1/n^{\varepsilon/3}.
\end{aligned}$$

Finding a good I_0. According to the proof above we only need to compute all $a_{I,i}$-s for $I \in \binom{[n]}{n^{\varepsilon/10}}$, $i \in [m]$. Then we can output an I_0 such that $\mathbb{E}_{i\sim[m]}[a_{I_0,i}] \le 2^{n-n^{\varepsilon/10}}/n^{1/2-\varepsilon/8}$, which satisfies the requirement.

For each $i \in [m]$, suppose

$$T_i(x_1, x_2, \cdots, x_n) = \mathbb{I}\left[\sum_{j=1}^{n} w_{i,j}x_j \ge t_i\right].$$

Then for each $I \in \binom{[n]}{n^{\varepsilon/10}}$ and $i \in [m]$, we define two THR circuits, $A_{I,i}(z)$ and $A'_{I,i}(z)$, that capture whether T_i always outputs 0 or 1 on $\mathsf{Column}_z^{(I)}$, respectively. Namely,

$$\begin{aligned}
A_{I,i}(z) &:= \mathbb{I}\left[\sum_{j\in[n]\setminus I} w_{i,j}z_j < t_i - \sum_{j\in I} \max\{w_{i,j}, 0\}\right], \\
A'_{I,i}(z) &:= \mathbb{I}\left[\sum_{j\in[n]\setminus I} w_{i,j}z_j \ge t_i - \sum_{j\in I} \min\{w_{i,j}, 0\}\right].
\end{aligned}$$

Thus

$$1 - A_{I,i}(z) - A'_{I,i}(z) = \mathbb{I}[T_i \text{ is non-constant on } \mathsf{Column}_z^{(I)}].$$

Therefore, we can apply the THR-#SAT algorithm to $A_{I,i}, A'_{I,i}$ and get

$$a_{I,i} = \sum_{z \in \{0,1\}^{[n]\setminus I}} (1 - A_{I,i}(z) - A'_{I,i}(z)).$$

By Lemma 3.9, the total time for computing all $a_{I,i}$-s is $O\Big(\binom{n}{n^{\varepsilon/10}} \cdot m \cdot 2^{(n-n^{\varepsilon/10})/2} \cdot \mathrm{poly}(n)\Big)$ which is at most $2^{n/2+o(n)}$. □

5 Approximating 1hotSUM Gates with Low-Degree Polynomials and Applications

5.1 Main Lemma

We follow the work of Alman and Williams [AW15], but adapt it to our setting, where $m = O(n^{2.5-\varepsilon})$ (as opposed to subquadratic in n) and we list approximate a 1hotSUM gate on any subcube $X \subset \{0,1\}^m$ of dimension $\leq n^{2-\varepsilon/2}$.

Lemma 5.1 (List-Approximating SUM with Low-Degree Polynomials over All Subcubes of Dimension $n^{2-\varepsilon/2}$)**.** *Let $m = O(n^{2.5-\varepsilon})$ and n be sufficiently large. There exists a collection of multi-output polynomials*

$$\left\{P_{r,a} \in (\mathbb{F}_2[x_1, x_2, \cdots, x_m])^{m+1}\right\}_{r \in \left[2^{n^{\varepsilon/100}}\right],\ a \in \left[n^{10 \log n}\right]}$$

which attempts to list approximate a 1hotSUM *gate on m inputs under all subcubes of dimension $\leq n^{2-\varepsilon/2}$. (Note that each $P_{r,a}$ outputs $m+1$ bits, similar to* 1hotSUM*.) More precisely, we have the following guarantees:*

1. ***Low Degree Efficiently Computable Polynomials.*** *$\deg P_{r,a} \leq n^{1-\varepsilon/6}$ for any r, a and there exists an algorithm that given n, m, outputs $\{P_{r,a}\}$ in deterministic time $O\Big(2^{n^{1-\varepsilon/6}}\Big)$.*

2. ***List Approximators over Large Subcubes.*** *For any subcube $X \subset \{0,1\}^m$ of dimension $\leq n^{2-\varepsilon/2}$ and $r \in \left[2^{n^{\varepsilon/100}}\right]$ there exists an $a(X,r) \in [n^{10 \log n}]$ such that: for each $x \in X$,*

$$\Pr_{r \sim \left[2^{n^{\varepsilon/100}}\right]}[\mathsf{1hotSUM}(x) = P_{r,a(X,r)}(x)] \geq 1 - 1/2^{n^{\varepsilon/500}}.$$

Moreover, $a(X,r)$ can be computed in deterministic polynomial time given X and r.

Proof.
Part 1. *We first construct the collection of polynomials $\{P_{r,a}\}$.*

- We use r to sample a collection of $2 \log n + 1$ many layers $\mathsf{Layer}^{(0)}, \mathsf{Layer}^{(1)}, \cdots, \mathsf{Layer}^{(2 \log n)}$, where $[m] = \mathsf{Layer}^{(0)} \supseteq \mathsf{Layer}^{(1)} \supseteq \cdots \supseteq \mathsf{Layer}^{(2 \log n)}$. The layers are sampled as follows:
  - The first layer is $\mathsf{Layer}^{(0)} := [m]$;
  - For $k \in \{0, 1, \ldots, 2 \log n - 1\}$, $\mathsf{Layer}^{(k+1)}$ is a pseudo-random subset of $\mathsf{Layer}^{(k)}$ where each element is selected with probability $1/2$ and the choices are $n^{\varepsilon/200}$-wise independent.
- We use a to encode a sequence of integers $(a_0, a_1, \cdots, a_{2 \log n - 1})$, where for each i, $|a_i| \leq n^{2.5}$.

Next, we explain how to construct $P_{r,a}$ for a fixed pair (r, a). We do it by getting low-degree approximating polynomials for $\mathsf{1hotSUM}$ over all layers, starting from the last layer and going backwards.

For the last layer, with very high probability, there are at most $O(n^{1-\varepsilon})$ variables in it. (Actually, there are only $m/n^2 = O(n^{0.5-\varepsilon})$ many variables in expectation.) So there exists an exact degree $O(n^{1-\varepsilon})$ polynomial that computes

$$\mathsf{1hotSUM}\left(x_i \colon i \in \mathsf{Layer}^{(2\log n)}\right).$$

Next we want to do the same for all layers. The idea is to iteratively go backwards from layer $2\log n$ to layer 0 and compute a good approximation of $\mathsf{1hotSUM}\left(x_i \colon i \in \mathsf{Layer}^{(k)}\right)$ given a good approximation of $\mathsf{1hotSUM}\left(x_i \colon i \in \mathsf{Layer}^{(k+1)}\right)$.

Recall that each element in $\mathsf{Layer}^{(k+1)}$ is a random subset of $\mathsf{Layer}^{(k)}$, and each element in $\mathsf{Layer}^{(k)}$ has probability $1/2$ to be in $\mathsf{Layer}^{(k+1)}$. Thus, we expect $2\sum_{i\in\mathsf{Layer}^{(k+1)}} x_i$ to be similar to $\sum_{i\in\mathsf{Layer}^{(k)}} x_i$, with an additive error of roughly $O\left(\sqrt{\left|\mathsf{Layer}^{(k)}\right|}\right)$ due to standard deviation. This error will typically be much larger than n for small k. We note though that on any large subcube $X \subset \{0,1\}^m$ of dimension $\le n^{2-\varepsilon/2}$ that have coordinates $J \subseteq [m]$ non-constant, and any $x \in X$, the difference $2\sum_{i\in\mathsf{Layer}^{(k+1)}} x_i - \sum_{i\in\mathsf{Layer}^{(k)}} x_i$ comes from two sources:

- The difference among the fixed coordinates (i.e., those in $[m] \setminus J$) which only depends on X. This difference can be typically $O\left(\sqrt{\left|\mathsf{Layer}^{(k)}\right|}\right)$ which is larger than n for small k,
- The difference among the coordinates in J, which depends on the choice of x in X, but has mean 0 and much smaller standard deviation $O\left(\sqrt{\left|J \cap \mathsf{Layer}^{(k)}\right|}\right) \le O\left(\sqrt{|J|}\right) \le O(n^{1-\varepsilon/4})$.

The idea is that for every subcube X there will be a sequence of advices $a_0, \cdots, a_{2\log n-1}$ that will capture the differences of the first kind. Next, we explain how to construct an approximating polynomial $P_{r,a}$ for $\mathsf{1hotSUM}$ given the randomness r and the advice $a = (a_0, \ldots, a_{2\log n-1})$.

Defining Helper "Indicator in Interval" Low-Degree Polynomials $E_{k,t,s}$. For each $k \in \{0, 1, \cdots, 2\log n\}$, $t \in \{0, 1, \cdots, m\}$, and $s \in [t - n^{1-\varepsilon/5}, t + n^{1-\varepsilon/5}] \cap [0, m]$, we define the polynomial $E_{k,t,s} \in \mathbb{F}_2[x_1, \cdots, x_m]$ to be the interpolating polynomial that satisfies

$$E_{k,t,s}(x_1, \cdots, x_m) = 1 \quad \text{if} \sum_{i\in\mathsf{Layer}^{(k)}} x_i = s,$$
$$E_{k,t,s}(x_1, \cdots, x_m) = 0 \quad \text{if} \sum_{i\in\mathsf{Layer}^{(k)}} x_i \in [t - n^{1-\varepsilon/5}, t + n^{1-\varepsilon/5}] \setminus \{s\}.$$

i.e., when $\{x_i\}_{i\in\mathsf{Layer}^{(k)}}$ contains $t \pm n^{1-\varepsilon/5}$ many 1-s, $E_{k,t,s}$ is an indicator whether there are exactly s many 1-s (the polynomial may be arbitrary for other x's). Note that $\deg(E_{k,t,s}) \le 2n^{1-\varepsilon/5}$.

Defining Polynomials $A_{k,s}$ that Attempt to Compute $\mathsf{1hotSUM}$ on Each Layer. For each $k = 2\log n, \cdots, 1, 0$ (i.e., in backward order) and $0 \le s \le m$, define the polynomial $A_{k,s} \in \mathbb{F}_2[x_1, \cdots, x_m]$ as follows:

- For $k = 2\log n$, for each $0 \le s \le m$, let $A_{k,s} := E_{k,0,s}$. (For $s > n^{1-\varepsilon/5}$, set $A_{k,s} := 0$.) If $\{x_i\}_{i\in\mathsf{Layer}^{(2\log n)}}$ contains $\le n^{1-\varepsilon/5}$ many 1-s, (which happens with very high probability, since even $\mathbb{E}\left[\left|\mathsf{Layer}^{(2\log n)}\right|\right] \le \sqrt{n}$,) then $A_{k,s}$ is an indicator whether there are exactly s 1-s in $\{x_i\}_{i\in\mathsf{Layer}^{(2\log n)}}$;
- For each $k = 2\log n - 1, \cdots, 1, 0$, for each $0 \le s \le m$, let

$$A_{k,s} := \sum_t E_{k,2t+a_k,s} A_{k+1,t},$$

where $\sum_t$ takes the sum over all t-s such that $E_{k,2t+a_k,s}$ and $A_{k+1,t}$ are well-defined. Observe that when $E_{k,2t+a_k,s}$ is well-defined it equals 1 if and only if $\sum_{i\in\mathsf{Layer}^{(k)}} x_i = s$.

The crucial observation is for each k, $A_{k,s}(x)$ is an indicator whether there are exactly s 1-s in $\{x_i\}_{i\in\mathsf{Layer}^{(k)}}$ (simultaneously for all s-s) as long as the following conditions hold for all $k' \in \{k, k+1, \cdots, 2\log n\}$:

$$\left| \sum_{i\in\mathsf{Layer}^{(k')}} x_i - 2 \sum_{i\in\mathsf{Layer}^{(k'+1)}} x_i - a_{k'} \right| \le n^{1-\varepsilon/5}. \tag{4}$$

(For $k' = 2\log n$, view $\mathsf{Layer}^{(2\log n+1)} := \varnothing$ and $a_{2\log n} := 0$.)

In particular, if this is the case when $k = 0$ then $(A_{0,0}(x), \cdots, A_{0,m}(x)) = \mathsf{1hotSUM}(x)$.

Observe that for every k and s the degree of $A_{k,s}$ is at most $(2\log n - k + 1)\cdot 2n^{1-\varepsilon/5}$.

Finally, let

$$P_{r,a} := (A_{0,0}, A_{0,1}, \cdots, A_{0,m}).$$

By the definition of $P_{r,a}$, we have $\deg P_{r,a} \le 2n^{1-\varepsilon/5}\cdot(2\log n + 1) \le n^{1-\varepsilon/6}$. Also note that the description of each $P_{r,a}$ can be computed in $O\left(2^{n^{1-\varepsilon/5}\cdot\mathsf{polylog}(n)}\right)$ time, and then by enumerating all (r,a)-s, we can compute $\{P_{r,a}\}$ in $O\left(2^{n^{1-\varepsilon/6}}\right)$ time.

Part 2. *For each subcube $X \subset \{0,1\}^m$ of dimension $\le n^{2-\varepsilon/2}$ and $r \in \left[2^{n^{\varepsilon/100}}\right]$, we show how to pick an $a(X,r)$ such that: for each $x \in X$ $P_{r,a(X,r)}(x) = \mathsf{1hotSUM}(x)$ with high probability over r. We also show that $a(X,r)$ can be computed in deterministic polynomial time (given a concise description of X).*

Picking $a(X,r)$-s. Fix a subcube $X \subset \{0,1\}^m$ of dimension $\le n^{2-\varepsilon/2}$ and $r \in \left[2^{n^{\varepsilon/100}}\right]$. Consider by $J \subseteq [m]$ the set of indices such that the subcube X is non-constant on. Then, $|J| = \mathsf{dim}(X) \le n^{2-\varepsilon/2}$. Let $x = (x_1, x_2 \cdots, x_m) \in X$ be any fixed point in X. For each $0 \le k \le 2\log n - 1$ let

$$a_k := \sum_{i\in\mathsf{Layer}^{(k)}:\, i\notin J} x_i - 2 \sum_{i\in\mathsf{Layer}^{(k+1)}:\, i\notin J} x_i.$$

Since for $i \notin J$, x_i is fixed in X, we know a only depends on X, r, and is independent of x. Observe that $a_k(X,r)$ is computable in deterministic polynomial time, assuming that the subcube X is given in a concise description of its fixed coordinates and their values (e.g., X can be described as a string in $\rho = \{0,1,\star\}^m$ where $\rho_i \in \{0,1\}$ if all $x \in X$ satisfy $x_i = \rho_i$ and $\rho_i = \star$ otherwise.)

Proof of Approximation. We prove that for any fixed $x \in X$,

$$\Pr_{r\sim\left[2^{n^{\varepsilon/100}}\right]}\left[\mathsf{1hotSUM}(x) = P_{r,a(X,r)}(x)\right] \geq 1 - 1/2^{n^{\varepsilon/500}}. \tag{5}$$

Note that if

$$\left|\sum_{i\in\mathsf{Layer}^{(k)}} x_i - 2\sum_{i\in\mathsf{Layer}^{(k+1)}} x_i - a_k\right| \leq n^{1-\varepsilon/5}$$

for all $0 \leq k \leq 2\log n$, then $\mathsf{1hotSUM}(x) = P_{r,a(X,r)}(x)$. Therefore, by the definition of a_k-s, a sufficient condition for $\mathsf{1hotSUM}(x) = P_{r,a(X,r)}(x)$ is:

$$\left|\sum_{i\in\mathsf{Layer}^{(k)}:\, i\in J} x_i - 2\sum_{i\in\mathsf{Layer}^{(k+1)}:\, i\in J} x_i\right| \leq n^{1-\varepsilon/5}, \qquad \text{for all } 0 \leq k \leq 2\log n.$$

Note that for each $1 \leq k \leq 2\log n$, the distribution of $\mathsf{Layer}^{(k)}$ is equivalent to: for each element in $i \in [m]$, let $\mathsf{Layer}^{(k)}$ contains i with probability $1/2^k$, and the decision for all i-s are $n^{\varepsilon/200}$-wise independent. So, by Lemma 3.11, for any fixed $x \in X$ and J of size at most $n^{2-\varepsilon/2}$, we have

$$\begin{aligned}
&\Pr_{r\sim\left[2^{n^{\varepsilon/100}}\right]}\left[\left|\sum_{i\in\mathsf{Layer}^{(k)}:\, i\in J} x_i - \frac{1}{2^k}\sum_{i\in J} x_i\right| \leq n^{1-\varepsilon/5}/3\right] \\
&= \Pr_{r\sim\left[2^{n^{\varepsilon/100}}\right]}\left[\left|\sum_{i\in J} x_i\cdot\mathbb{I}[i\in\mathsf{Layer}^{(k)}] - \frac{1}{2^k}\sum_{i\in J} x_i\right| \leq n^{1-\varepsilon/5}/3\right] \\
&\geq 1 - \exp\left(-\Omega\left(\min\left\{n^{\varepsilon/200}, n^{1-\varepsilon/5}/3, \frac{(n^{1-\varepsilon/5}/3)^2}{\frac{1}{2^k}\sum_{i\in J} x_i}\right\}\right)\right) && \text{(Lemma 3.11)} \\
&\geq 1 - 1/2^{n^{\varepsilon/400}}.
\end{aligned}$$

Therefore, with probability $\geq 1 - 1/2^{n^{\varepsilon/500}}$ over r, for all $0 \leq k \leq 2\log n$, we have

$$\left|\sum_{i\in\mathsf{Layer}^{(k)}:\, i\in J} x_i - \frac{1}{2^k}\sum_{i\in J} x_i\right| \leq n^{1-\varepsilon/5}/3,$$

and thus for all $0 \leq k \leq 2\log n - 1$,

$$\left|\sum_{i\in\mathsf{Layer}^{(k)}:\, i\in J} x_i - 2\cdot\sum_{i\in\mathsf{Layer}^{(k+1)}:\, i\in J} x_i\right| \leq n^{1-\varepsilon/5},$$

and hence $\mathsf{1hotSUM}(x) = P_{r,a(X,r)}(x)$. So Equation (5) holds. □

5.2 Approximating $\mathsf{1hotSUM}_{n^{2.5-\varepsilon}} \circ \mathsf{THR}$ by low-degree polynomials on almost all good columns

Theorem 5.2. *There is a deterministic $O\left(2^{n-n^{\varepsilon/30}}\right)$ time algorithm that, given a collection of $m = O\left(n^{2.5-\varepsilon}\right)$ linear threshold gates $\mathcal{C} = \left\{T_i\colon \{0,1\}^n \to \{0,1\}\right\}_{i\in[m]}$ and a partition $I \in \binom{[n]}{n^{\varepsilon/10}}$,*

outputs a collection of multi-output polynomials

$$\left\{P_{r,a} \in (\mathbb{F}_2[x_1, x_2, \cdots, x_m])^{m+1}\right\}_{r\in\left[2^{n^{\varepsilon/100}}\right], a\in\left[n^{10\log n}\right]}$$

of degree $n^{1-\varepsilon/6}$*, and outputs for each pair* $(z, r) \in \{0,1\}^{[n]\setminus I} \times \left[2^{n^{\varepsilon/100}}\right]$*, a choice* $a(z, r) \in \left[n^{10\log n}\right]$*, such that: for any* $z \in \{0,1\}^{[n]\setminus I}$ *such that* $\mathsf{Column}_z^{(I)}$ *is* $\mathcal{C}$*-good,*

$$\Pr_{r\sim[2^{n^{\varepsilon/100}}],\, x\sim\mathsf{Column}_z^{(I)}}[\mathsf{1hotSUM}(\mathcal{C}(x)) = P_{r,a(z,r)}(\mathcal{C}(x))] \geq 1 - 1/2^{n^{\varepsilon/500}}.$$

Proof. Using Lemma 5.1, we have a collection of polynomials

$$\left\{P_{r,a} \in (\mathbb{F}_2[x_1, x_2, \cdots, x_m])^{m+1}\right\}_{r\in\left[2^{n^{\varepsilon/100}}\right],\ a\in\left[n^{10\log n}\right]},$$

that list approximate a $\mathsf{1hotSUM}$ gate on any subcube $X \subset \{0,1\}^m$ of dimension $\leq n^{2-\varepsilon/2}$.

Denote by $A \subseteq \{0,1\}^{[n]\setminus I}$ the set of all $z \in \{0,1\}^{[n]\setminus I}$ such that $\mathsf{Column}_z^{(I)}$ is $\mathcal{C}$-good. For any good $z \in A$, let $X = X(z) \subset \{0,1\}^m$ be the minimal subcube that contains the support of $\mathcal{C}(\mathsf{Column}_z^{(I)})$. Note that if z is good, then $X(z)$ has dimension $\leq n^{2-\varepsilon/2}$. Let

$$a(z, r) := a(X(z), r).$$

Taking expectation over $x \sim \mathsf{Column}_z^{(I)}$ and applying Lemma 5.1, we get

$$\Pr_{r\sim[2^{n^{\varepsilon/100}}],\, x\sim\mathsf{Column}_z^{(I)}}[\mathsf{1hotSUM}(\mathcal{C}(x)) = P_{r,a(z,r)}(\mathcal{C}(x))] \geq 1 - 1/2^{n^{\varepsilon/500}}. \tag{6}$$

□

5.3 CAPP Algorithm for $\oplus_2 \circ \mathsf{SYM}_{O(n^{2.5-\varepsilon})} \circ \mathsf{THR}$

In this section, we apply Theorem 5.2 to obtain a $\mathsf{CAPP}_{o(1)}$ algorithm for $\oplus_2 \circ \mathsf{SYM}_{O(n^{2.5-\varepsilon})} \circ \mathsf{THR}$ circuits. We need the following lemma:

Lemma 5.3. *There is a deterministic* $O\left(2^{n-n^{\varepsilon/50}}\right)$ *time algorithm that, given an* n*-input* $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/8}] \circ \mathsf{THR}$ *circuit* C *with at most* $\mathrm{poly}(n)$ *different* THR *gates at the bottom, and a partition* $I \in \binom{[n]}{n^{\varepsilon/10}}$*, outputs*

$$\Pr_{x\in\mathsf{Column}_z^{(I)}}[C(x) = 1]$$

for each $z \in \{0,1\}^{[n]\setminus I}$*.*

Lemma 5.3 has been implicitly used in previous works [ACW16]. Nevertheless, here we give a proof for completeness in Appendix A.4.

Theorem 5.4. *There is a deterministic algorithm for* $\oplus_2 \circ \mathsf{SYM}_{O(n^{2.5-\varepsilon})} \circ \mathsf{THR}$*-*$\mathsf{CAPP}_{o(1)}$ *that runs in* $O\left(2^{n-n^{\varepsilon/100}}\right)$ *time.*

Proof. Denote the $\oplus_2 \circ \mathsf{SYM}_{O(n^{2.5-\varepsilon})} \circ \mathsf{THR}$ circuit by

$$C = (f_1 \circ \mathcal{C}_1) \oplus (f_2 \circ \mathcal{C}_2),$$

where

- $f_1 \colon \{0,1\}^{m_1} \to \{0,1\}$ and $f_2 \colon \{0,1\}^{m_2} \to \{0,1\}$ are the SYM gates on the first layer, where $m_1, m_2 \le O\big(n^{2.5-\varepsilon}\big)$;
- $\mathcal{C}_1, \mathcal{C}_2$ are the collection of THR gates under f_1, f_2, respectively. If a THR gate has multiple wires to f_j, then it is counted multiple times in $\mathcal{C}_j$.

By Theorem 4.5, in deterministic $2^{n/2+o(n)}$ time, we can find a partition $I \in \binom{[n]}{n^{\varepsilon/10}}$ such that at least $(1 - 1/n^{\varepsilon/3})$ fraction of the $\mathsf{Column}_z^{(I)}$-s ($z \in \{0,1\}^{[n]\setminus I}$) are $(\mathcal{C}_1 + \mathcal{C}_2)$-good, and hence they are both $\mathcal{C}_1$-good and $\mathcal{C}_2$-good. For $j \in \{1,2\}$, use Theorem 4.4 to compute

$$A_j := \Big\{ z \in \{0,1\}^{[n]\setminus I} \colon \mathsf{Column}_z^{(I)} \text{ is } \mathcal{C}_j\text{-good} \Big\}.$$

Now we apply Theorem 5.2 to $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. In $O\Big(2^{n-n^{\varepsilon/30}}\Big)$ time, we can find two collections of polynomials $\Big\{P_{r,a}^{(1)}\Big\}$, $\Big\{P_{r,a}^{(2)}\Big\}$ of degree $n^{1-\varepsilon/6}$ and choices $a^{(1)}(z,r), a^{(2)}(z,r)$-s for $\mathcal{C}_1, \mathcal{C}_2$ respectively that satisfy: for each $j \in \{1,2\}$ and $z \in A_j$, we have

$$\Pr_{r\sim\left[2^{n^{\varepsilon/100}}\right],\, x\sim\mathsf{Column}_z^{(I)}} \Big[\mathsf{1hotSUM} \circ \mathcal{C}_j(x) = P^{(j)}_{r,a(z,r)} \circ \mathcal{C}_j(x) \Big] \ge 1 - 1/2^{n^{\varepsilon/500}}.$$

For each $j \in \{1,2\}$ and $r \in \Big[2^{n^{\varepsilon/100}}\Big]$ and $a \in \big[n^{10\log n}\big]$, we compute the polynomial

$$Q^{(j)}_{r,a} := \sum_{0\le s\le m_j \colon f_j \text{ outputs 1 on } s \text{ 1-s}} \Big(P^{(j)}_{r,a}\Big)_s.$$

Then for any $z \in A_j$, we have

$$\Pr_{r\sim[2^{n^{\varepsilon/100}}],\, x\sim\mathsf{Column}_z^{(I)}} \Big[f_j \circ \mathcal{C}_j(x) = Q^{(j)}_{r,a^{(j)}(z,r)} \circ \mathcal{C}_j(x) \Big] \ge 1 - 1/2^{n^{\varepsilon/500}} \tag{7}$$

Then, for each tuple $(r, a_1, a_2) \in \Big[2^{n^{\varepsilon/100}}\Big] \times \big[n^{10\log n}\big]^2$, consider the circuit

$$\widetilde{C}_{r,a_1,a_2} := \Big(Q^{(1)}_{r,a_1} \circ \mathcal{C}_1\Big) \oplus \Big(Q^{(2)}_{r,a_2} \circ \mathcal{C}_2\Big),$$

i.e., replace the two SYM gates by $Q^{(1)}_{r,a_1}$ and $Q^{(2)}_{r,a_2}$. Note that $\widetilde{C}_{r,a_1,a_2}$ is a $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/6}] \circ \mathsf{THR}$ circuit, so by Lemma 5.3, in $O\Big(2^{n-n^{\varepsilon/50}}\Big)$ time, we can output

$$\Pr_{x\sim\mathsf{Column}_z^{(I)}} \Big[\widetilde{C}_{r,a_1,a_2}(x) = 1\Big]$$

for all $z \in \{0,1\}^{[n]\setminus I}$. Then, by enumerating over all tuples (r, a_1, a_2), we can output

$$\Pr_{x\sim\mathsf{Column}_z^{(I)}} \Big[\widetilde{C}_{r,a_1,a_2}(x) = 1\Big]$$

for all $(r, a_1, a_2) \in \Big[2^{n^{\varepsilon/100}}\Big] \times \big[n^{10\log n}\big]^2$ and $z \in \{0,1\}^{[n]\setminus I}$ in total time $O\Big(2^{n-n^{\varepsilon/60}}\Big)$. Finally, we output

$$\widetilde{p} := \frac{1}{2^{n-|I|}} \sum_{z\in A_1\cap A_2} \Pr_{\substack{x\sim\mathsf{Column}_z^{(I)},\\ r\sim[2^{n^{\varepsilon/100}}]}} \Big[\widetilde{C}_{r,a^{(1)}(z,r),a^{(2)}(z,r)}(x) = 1\Big].$$

We show that $\widetilde{p}$ is close to C's acceptance probability, which we denote by p.

Applying the union bound on Equation (7), for any $z \in A_1 \cap A_2$,

$$\Pr_{r\sim[2^{n^{\varepsilon/100}}],\, x\sim\mathsf{Column}_z^{(I)}}\left[\forall j \in \{1,2\} : Q^{(j)}_{r,a(z,r)} \circ \mathcal{C}_j(x) = f_j \circ \mathcal{C}_j(x)\right] \geq 1 - 2/2^{n^{\varepsilon/500}} \tag{8}$$

and hence

$$\Pr_{r\sim[2^{n^{\varepsilon/100}}],\, x\sim\mathsf{Column}_z^{(I)}}\left[\widetilde{C}_{r,a^{(1)}(z,r),a^{(2)}(z,r)}(x) = C(x)\right] \geq 1 - 2/2^{n^{\varepsilon/500}} \tag{9}$$

So

$$\left|\widetilde{p} - \frac{1}{2^{n-|I|}} \sum_{z\in A_1\cap A_2} \Pr_{x\sim\mathsf{Column}_z^{(I)}}[C(x)=1]\right| \leq 2/2^{n^{\varepsilon/500}}. \tag{10}$$

Also note that since $|A_1| \geq (1 - 1/n^{\varepsilon/3}) \cdot 2^{n-|I|}$, $|A_2| \geq (1 - 1/n^{\varepsilon/3}) \cdot 2^{n-|I|}$, we have $|A_1 \cap A_2| \geq (1 - 2/n^{\varepsilon/3}) \cdot 2^{n-|I|}$, and thus

$$\begin{aligned}
&\left|p - \frac{1}{2^{n-|I|}} \sum_{z\in A_1\cap A_2} \Pr_{x\sim\mathsf{Column}_z^{(I)}}[C(x)=1]\right| \\
&= \frac{1}{2^{n-|I|}} \sum_{z\in\{0,1\}^{[n]\setminus I}\setminus(A_1\cap A_2)} \Pr_{x\sim\mathsf{Column}_z^{(I)}}[C(x)=1] \\
&\leq 2/n^{\varepsilon/3}.
\end{aligned} \tag{11}$$

Equations (10) and (11) give that $|\widetilde{p} - p| \leq o(1)$, which satisfies the requirement. □

Combining Theorem 5.4 and Lemma 3.8 we get:

Corollary 5.5. *For any constant $\varepsilon > 0$, E^{NP} does not have $n^{2.5-\varepsilon}$-sized $\mathsf{SYM} \circ \mathsf{THR}$ circuits.*

6 CAPP Algorithm for $\oplus_2 \circ \mathsf{THR}_{O(n^{2.5-2\varepsilon})} \circ \mathsf{THR}$

Theorem 6.1. *There is a deterministic algorithm for $\oplus_2 \circ \mathsf{THR}_{O(n^{2.5-2\varepsilon})} \circ \mathsf{THR}$-$\mathsf{CAPP}_{o(1)}$ that runs in $O\left(2^{n-n^{\varepsilon/100}}\right)$ time.*

(Remark: To prove Theorem 1.2 with parameter ε, it suffices to prove Theorem 6.1 with parameter $\varepsilon/2$; we use 2ε in Theorem 6.1 for better readability.)

Proof. Let C be the $\oplus_2 \circ \mathsf{THR}_{O(n^{2.5-2\varepsilon})} \circ \mathsf{THR}$ circuit.

Our proof consists of three parts:

1. We use several items from Lemma 3.4 to transform C into a

   $$\widetilde{\mathsf{SUM}} \circ \mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM} \circ \mathsf{THR}$$

   circuit C^*, where crucially each $\mathsf{POLY}_{\mathbb{F}_2}[O(1)]$ gate has only $O(1)$ many different $\mathsf{1hotSUM}$ gates feeding it.

2. We reduce $\mathsf{CAPP}_{o(1)}$ on C^* to (some variant of) $\mathsf{CAPP}_{o(1)}$ on $\mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM} \circ \mathsf{THR}$ circuits $C^*_1, \ldots, C^*_N$.

3. Similarly to Theorem 5.4, we present a deterministic $O(2^{n-n^{\varepsilon/100}})$-time algorithm that outputs, for any $k \in [N]$ and each good column z, an estimate $\widetilde{p}_k(z) \in \mathbb{R}$ that is $1/n^{\omega(1)}$ close to $\Pr_{x\sim\mathsf{Column}_z^{(I)}}[C_k^*(x)=1]$. Finally, taking the average over $\widetilde{p}_k(z)$ for good columns z would give us the desired $\Pr_{x\sim\{0,1\}^n}[C_k^*(x)=1]$ up to additive error $o(1)$.

Part 1. *We use Lemma 3.4 to transform C.*

Apply Item 2 of Lemma 3.4 to C's middle layer, we transform C into an c

$$\oplus_2 \circ \mathsf{DOR} \circ \mathsf{ETHR} \circ \mathsf{THR}$$

circuit C_1, where there are at most $O\big(n^{2.5-2\varepsilon}\big)$ many different THR gates in the bottom layer. Suppose each DOR gate in C_1 has exactly n^c gates. (We can append some constant-0 inputs to each DOR gate to ensure that.) Then apply Item 3 of Lemma 3.4 to C_1's ETHR layer, we transform C_1 into an

$$\oplus_2 \circ \mathsf{DOR} \circ \mathsf{GapAND}_{1/n^{c+2}} \circ \mathsf{SYM} \circ \mathsf{THR}$$

circuit C_2, where there are at most $O\big(n^{2.5-2\varepsilon}\big)$ many different THR gates in the bottom layer. Then apply Item 4 of Lemma 3.4 to C_2's SYM layer, we transform C_2 into an

$$\oplus_2 \circ \mathsf{DOR} \circ \mathsf{GapAND}_{1/n^{c+2}} \circ \mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM} \circ \mathsf{THR}$$

circuit C_3, where:

- There are at most $O\big(n^{2.5-2\varepsilon}\big)$ many different THR gates in the bottom layer;
- For each $\mathsf{POLY}_{\mathbb{F}_2}[O(1)]$ circuit in the 4-th layer, the subcircuit under which has at most $O(1)$ many different $\mathsf{1hotSUM}$ gates (each with $\mathrm{poly}(n)$ output bits);
- Each $\mathsf{1hotSUM}$ gate in the 5-th layer has at most $O\big(n^{2.5-\varepsilon}\big)$ input wires, and moreover, has at most $O(n^{\varepsilon})$ wires to each THR gate in the bottom layer.

Now consider the $\mathsf{DOR} \circ \mathsf{GapAND}_{1/n^{c+2}}$ part in C_3. Recall that each DOR gate has exactly n^c inputs. Next, suppose all $\mathsf{GapAND}_{1/n^{c+2}}$ gates have at most n^d many incoming wires. We transform them so that each will have exactly n^d wires by duplicating wires equitably. Indeed, if a $\mathsf{GapAND}_{1/n^{c+2}}$ gate has s incoming wires, we can duplicate each wire either $\lceil n^d/s \rceil$ or $\lfloor n^d/s \rfloor$ times to get exactly n^d many wires. Under this transformation, the new gate needs to distinguish inputs that are all 1s from those on which at most $(s/n^{c+2}) \cdot \lceil n^d/s \rceil \le 2n^d/n^{c+2} \le n^d/n^{c+1}$ are 1s. This can be done by a $\mathsf{GapAND}_{1/n^{c+1}}$ gate on exactly n^d wires. Let C_4 be the new

$$\oplus_2 \circ \mathsf{DOR} \circ \mathsf{GapAND}_{1/n^{c+1}} \circ \mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM} \circ \mathsf{THR}$$

circuit, where each DOR has exactly n^c input bits and each $\mathsf{GapAND}_{1/n^{c+1}}$ gate has exactly n^d input bits. We view each $\mathsf{DOR} \circ \mathsf{GapAND}_{1/n^{c+1}}$ gate as an n^{c+d}-input circuit, then when it outputs 1, the number of 1-s in its input bits is in the range $[n^d, n^d + n^{d-1}]$, (because the GapAND gate that outputs 1 has n^d input 1-s, and each remaining GapAND gate has at most n^{d-c-1} input 1-s;) when it outputs 0, the number of 1-s in its input bits is in range $[0, n^{d-1}]$, (because each GapAND gate has at most n^{d-c-1} input 1-s.) Therefore, we can replace each $\mathsf{DOR} \circ \mathsf{GapAND}_{1/n^{c+1}}$ by an n^{c+d}-input gate $\widetilde{\mathsf{SUM}}$ gate, which outputs

$$\begin{cases} 0 & \text{if \#1-s in its input is} \in [0, n^{d-1}] \\ 1 & \text{if \#1-s in its input is} \in [n^d, n^d + n^{d-1}]. \end{cases}$$

Let C_5 be the resulting

$$\oplus_2 \circ \widetilde{\mathsf{SUM}} \circ \mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM} \circ \mathsf{THR}$$

circuit.

Below we transform $\oplus_2 \circ \widetilde{\mathsf{SUM}}$ into $\widetilde{\mathsf{SUM}} \circ \{\wedge_2, \neg\}$. For the top $\oplus_2 \circ \widetilde{\mathsf{SUM}}$ circuit on $2n^{c+d}$ inputs, denote its inputs by $x_1, x_2, \cdots, x_{n^{c+d}}, x'_1, x'_2, \cdots, x'_{n^{c+d}}$, where x_i-s are the inputs corresponding to one $\widetilde{\mathsf{SUM}}$ and x_i-s are corresponding to the other one. Let $S := \sum_{i=1}^{n^{c+d}} x_i$, $S' := \sum_{i=1}^{n^{c+d}} x'_i$, and let $f(S, S')$ be $\oplus_2 \circ \widetilde{\mathsf{SUM}}$'s output, then

$$f(S,S') = \begin{cases} 0 & \text{if } S \in [0, n^{d-1}] \text{ and } S' \in [0, n^{d-1}] \\ 1 & \text{if } S \in [0, n^{d-1}] \text{ and } S' \in [n^d, n^d + n^{d-1}] \\ 1 & \text{if } S \in [n^d, n^d + n^{d-1}] \text{ and } S' \in [0, n^{d-1}] \\ 0 & \text{if } S \in [n^d, n^d + n^{d-1}] \text{ and } S' \in [n^d, n^d + n^{d-1}]. \end{cases}$$

Thus

$$f(S,S') = \begin{cases} 0 & \text{if } (S - n^d/2)(S' - n^d/2) \in \left[n^{2d}/4 - 2n^{2d-1}, n^{2d}/4 + 2n^{2d-1}\right] \\ 1 & \text{if } (S - n^d/2)(S' - n^d/2) \in \left[-n^{2d}/4 - 2n^{2d-1}, -n^{2d}/4 + 2n^{2d-1}\right]. \end{cases}$$

Consider the following multiset:

$$\mathcal{M} := \left\{x_i \wedge x'_j \colon i \in [n^{c+d}], j \in [n^{c+d}]\right\} + n^d/2 \cdot \left\{\neg x_i \colon i \in [n^{c+d}]\right\} + n^d/2 \cdot \left\{\neg x'_i \colon i \in [n^{c+d}]\right\},$$

then, if $\mathcal{M}$ contains $n^{c+2d} \pm 2n^{2d-1}$ many 1-s, $\oplus_2 \circ \widetilde{\mathsf{SUM}}$ outputs 0; if $\mathcal{M}$ contains $n^{c+2d} - n^{2d}/2 \pm 2n^{2d-1}$ many 1-s, $\oplus_2 \circ \widetilde{\mathsf{SUM}}$ outputs 1. Therefore, we transform $\oplus_2 \circ \widetilde{\mathsf{SUM}}$ into $\widetilde{\mathsf{SUM}} \circ \{\wedge_2, \neg\}$, where $\mathcal{M}$ gives the bottom $\wedge_2, \neg$ gates, and the top $\widetilde{\mathsf{SUM}}$ gate outputs

$$\begin{cases} 0 & \text{if \#1-s in its input is } \in [n^{c+2d} - 2n^{2d-1}, n^{c+2d} + 2n^{2d-1}] \\ 1 & \text{if \#1-s in its input is } \in [n^{c+2d} - n^{2d}/2 - 2n^{2d-1}, n^{c+2d} - n^{2d}/2 + 2n^{2d-1}]. \end{cases}$$

Finally, the $\wedge_2, \neg$ gates can be merged into the $\mathsf{POLY}_{\mathbb{F}_2}[O(1)]$-s in the next layer. The number of $\mathsf{1hotSUM}$ gates under each $\mathsf{POLY}_{\mathbb{F}_2}[O(1)]$ is doubled.

To summarize, we transform C into a

$$\widetilde{\mathsf{SUM}} \circ \mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM} \circ \mathsf{THR}$$

circuit C^*, where

- There are at most $O\left(n^{2.5-2\varepsilon}\right)$ many different THR gates in the bottom layer;
- The top $\widetilde{\mathsf{SUM}}$ gate has $n^{2c+2d} + n^{c+2d}$ input bits, and outputs

  $$\begin{cases} 0 & \text{if \#1-s in its input is } \in n^{c+2d} \pm 2n^{2d-1} \\ 1 & \text{if \#1-s in its input is } \in n^{c+2d} - n^{2d}/2 \pm 2n^{2d-1}. \end{cases}$$

  (It is guaranteed that #1-s in its input is always in either of these two intervals.)
- For each $\mathsf{POLY}_{\mathbb{F}_2}[O(1)]$ circuit in the second layer, the subcircuit under which has at most $O(1)$ many different $\mathsf{1hotSUM}$ gates;

- Each $\mathsf{1hotSUM}$ gate in the third layer has at most $O\big(n^{2.5-\varepsilon}\big)$ input wires, and moreover, has at most $O(n^{\varepsilon})$ wires to each THR gate in the bottom layer.

Part 2. *We reduce* $\mathsf{CAPP}_{o(1)}$ *on* C^* *to (some variant of)* CAPP *on* $\mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM} \circ \mathsf{THR}$ *circuits.*

Let $\mathcal{C}$ be the collection of n^{ε} copies of all THR gates at the bottom layer of C^*. $|\mathcal{C}| \le O\big(n^{2.5-\varepsilon}\big)$. Using Theorem 4.5, in $2^{n/2+o(n)}$ time, we can find a partition $I \in \binom{[n]}{n^{\varepsilon}/10}$ such that at least $(1 - 1/n^{\varepsilon/3})$ fraction of $\mathsf{Column}_z^{(I)}$-s ($z \in \{0,1\}^{[n]\setminus I}$) are $\mathcal{C}$-good. Denote

$$A := \Big\{ z \in \{0,1\}^{[n]\setminus I} \colon \mathsf{Column}_z^{(I)} \text{ is } \mathcal{C}\text{-good} \Big\}$$

Let $C_1^*, \cdots, C_N^*$ ($N = n^{2c+2d} + n^{c+2d}$) be the $\mathsf{POLY}_{\mathbb{F}_2}[O(1)] \circ \mathsf{1hotSUM} \circ \mathsf{THR}$ subcircuits of C^*. We prove the following claim, which reduces $\mathsf{CAPP}_{o(1)}$ on C^* to some variant of CAPP on C_k^*-s:

Claim 6.1.1. *Assume there is a deterministic* $O\Big(2^{n-n^{\varepsilon/100}}\Big)$*-time algorithm that, for any* $k \in [N]$*, on input* C_k^**, for each* $z \in A$*, outputs an estimate* $\widetilde{p}_k(z) \in \mathbb{R}$ *such that*

$$\left| \widetilde{p}_k(z) - \Pr_{x \sim \mathsf{Column}_z^{(I)}} [C_k^*(x) = 1] \right| \le 1/n^{\omega(1)}.$$

Then there is a deterministic $O\Big(2^{n-n^{\varepsilon/200}}\Big)$*-time algorithm that, on input* C^**, solves* $\mathsf{CAPP}_{o(1)}$ *on* C^**.*

Proof of Claim 6.1.1.

We show that the following simple algorithm works: on input C^*, outputs

$$\widetilde{p} := \frac{1}{2^{n-|I|}} \sum_{z \in A} \frac{n^{c+2d} - \sum_{k \in [N]} \widetilde{p}_k(z)}{n^{2d}/2}.$$

Note that by the definition of the intervals in the top $\widetilde{\mathsf{SUM}}$ gate, for any $x \in \{0,1\}^n$, we have

$$\left| \frac{n^{c+2d} - \sum_{k \in [N]} \mathbb{I}[C_k^*(x) = 1]}{n^{2d}/2} - \mathbb{I}[C^*(x) = 1] \right| \le 4/n.$$

For any $z \in A$, take the expectation over $x \sim \mathsf{Column}_z^{(I)}$, we have

$$\left| \Pr_{x \sim \mathsf{Column}_z^{(I)}} [C^*(x) = 1] - \frac{n^{c+2d} - \sum_{k \in [N]} \Pr_{x \sim \mathsf{Column}_z^{(I)}} [C_k^*(x) = 1]}{n^{2d}/2} \right| \le 4/n,$$

take the sum over all $z \in A$, we get

$$\left| \frac{1}{2^{n-|I|}} \sum_{z \in A} \Pr_{x \sim \mathsf{Column}_z^{(I)}} [C^*(x) = 1] - \frac{1}{2^{n-|I|}} \sum_{z \in A} \frac{n^{c+2d} - \sum_{k \in [N]} \Pr_{x \sim \mathsf{Column}_z^{(I)}} [C_k^*(x) = 1]}{n^{2d}/2} \right| \le 4/n.$$

By the properties of $\widetilde{p}_k(z)$-s, we have

$$\left| \frac{1}{2^{n-|I|}} \sum_{z \in A} \Pr_{x \sim \mathsf{Column}_z^{(I)}} [C^*(x) = 1] - \frac{1}{2^{n-|I|}} \sum_{z \in A} \frac{n^{c+2d} - \sum_{k \in [N]} \widetilde{p}_k(z)}{n^{2d}/2} \right| \le 4/n + 1/n^{\omega(1)},$$

i.e.,

$$\left|\widetilde{p} - \frac{1}{2^{n-|I|}} \sum_{z\in A} \Pr_{x\sim \mathsf{Column}_z^{(I)}}[C^*(x)=1]\right| \le 4/n + 1/n^{\omega(1)}$$

Also note that $|A| \ge (1-1/n^{\varepsilon/3})\cdot 2^{n-|I|}$, so

$$\left|\widetilde{p} - \Pr_{x\sim\{0,1\}^n}[C^*(x)=1]\right| \le 4/n + 1/n^{\omega(1)} + 1/n^{\varepsilon/3} \le o(1).$$

□

So it only remains to show how to compute $\widetilde{p}_k(z)$-s for Claim 6.1.1.

Part 3. *We present a deterministic $O\left(2^{n-n^{\varepsilon/100}}\right)$-time algorithm that, for any $k\in[N]$, on input C_k^*, for each $z\in A$, outputs an estimate $\widetilde{p}_k(z)\in\mathbb{R}$ such that*

$$\left|\widetilde{p}_k(z) - \Pr_{x\sim\mathsf{Column}_z^{(I)}}[C_k^*(x)=1]\right| \le 1/n^{\omega(1)}.$$

This step is very similar to Theorem 5.4.

Let g be C_k^*'s top $\mathsf{POLY}_{\mathbb{F}_2}[O(1)]$ circuit. Let $f_1, f_2, \cdots, f_t$ ($t\le O(1)$) be the $\mathsf{1hotSUM}$ gates in C_k^*'s middle layer. For each $j\in[t]$, let $\mathcal{C}_j$ be the collection of THR gates connected to f_j. (If a gate has multiple wires to f_j, it is counted multiple times in $\mathcal{C}_j$.) Since f_j has at most n^ε wires to each bottom THR gates, we have $\mathcal{C}_j \le \mathcal{C}$. Let

$$A_j := \left\{z\in\{0,1\}^{[n]\setminus I} \colon \mathsf{Column}_z^{(I)} \text{ is } \mathcal{C}_j\text{-good}\right\}.$$

Then $A_j \supseteq A$.

Apply Theorem 5.2 to $f_1,\cdots,f_t$ respectively, we get for each $j\in[t]$, polynomials

$$\left\{P_{r,a}^{(j)}\right\}_{r\in\left[2^{n^{\varepsilon/100}}\right], a\in\left[n^{10\log n}\right]}$$

and choices $a^{(j)}(z,r)$ ($z\in A_j, r\in\left[2^{n^{\varepsilon/100}}\right]$), such that for any $z\in A_j$,

$$\Pr_{r\sim\left[2^{n^{\varepsilon/100}}\right],\, x\sim\mathsf{Column}_z^{(I)}}\left[f_j\circ\mathcal{C}_j(x) = P^{(j)}_{r,a^{(j)}(z,r)}\circ\mathcal{C}_j(x)\right] \ge 1 - 1/2^{n^{\varepsilon/500}} \tag{12}$$

Now for each tuple $(r,a_1,a_2,\cdots,a_t)\in\left[2^{n^{\varepsilon/100}}\right]\times\left[n^{10\log n}\right]^t$, consider the circuit

$$D_{r,a_1,a_2,\cdots,a_t} := g\left(P^{(1)}_{r,a_1}\circ\mathcal{C}_1, P^{(2)}_{r,a_2}\circ\mathcal{C}_2, \cdots, P^{(t)}_{r,a_t}\circ\mathcal{C}_t\right).$$

i.e., in C_k^*, replace each f_j by $P^{(j)}_{r,a_j}$. Note that $D_{r,a_1,a_2,\cdots,a_t}$ is a $\mathsf{POLY}_{\mathbb{F}_2}[O\left(n^{1-\varepsilon/6}\right)]\circ\mathsf{THR}$ circuit, so by Lemma 5.3, in $O\left(2^{n-n^{\varepsilon/50}}\right)$ time, we can output

$$\Pr_{x\sim\mathsf{Column}_z^{(I)}}[D_{r,a_1,a_2,\cdots,a_t}(x)=1]$$

for all $z \in \{0,1\}^{[n]\setminus I}$. Then enumerate over tuples $(r, a_1, a_2, \cdots, a_t)$, (there are $2^{n^{\varepsilon/100}} \cdot n^{10t\log n}$ many tuples,) we can output

$$\Pr_{x\sim \mathsf{Column}_z^{(I)}}[D_{r,a_1,a_2,\cdots,a_t}(x)=1]$$

for all $(r, a_1, a_2, \cdots, a_t) \in \left[2^{n^{\varepsilon/100}}\right] \times \left[n^{10\log n}\right]^t$ and $z \in \{0,1\}^{[n]\setminus I}$. This can be done in $O\left(2^{n-n^{\varepsilon/60}}\right)$ time in total.

Finally, for each $z \in A$, we output

$$\widetilde{p}_k(z) := \Pr_{r\sim\left[2^{n^{\varepsilon/100}}\right],\, x\sim \mathsf{Column}_z^{(I)}}\left[D_{r,a^{(1)}(z,r),a^{(2)}(z,r),\cdots,a^{(t)}(z,r)}(x)=1\right]$$

We show that $\widetilde{p}_k(z)$ satisfies the requirement. Consider a fixed $z \in \{0,1\}^{[n]\setminus I}$. Applying a union bound on all $j \in [t]$ to Equation (12) gives

$$\Pr_{r\sim\left[2^{n^{\varepsilon/100}}\right],\, x\sim \mathsf{Column}_z^{(I)}}\left[\forall j\in[t] : f_j\circ\mathcal{C}_j(x) = P^{(j)}_{r,a^{(j)}(z,r)}\circ\mathcal{C}_j(x)\right] \geq 1 - t/2^{n^{\varepsilon/500}} \tag{13}$$

Note that for any $r \in \left[2^{n^{\varepsilon/100}}\right]$ and $x \in \mathsf{Column}_z^{(I)}$, if $f_j \circ \mathcal{C}_j(x) = P^{(j)}_{r,a^{(j)}(z,r)} \circ \mathcal{C}_j(x)$ holds for all $j \in [t]$, then $C^*_k(x) = D_{r,a^{(1)}(z,r),\ldots,a^{(t)}(z,r)}(x)$. So for each $z \in A$,

$$\Pr_{r\in\left[2^{n^{\varepsilon/100}}\right],\, x\sim \mathsf{Column}_z^{(I)}}\left[C^*_k(x) = D_{r,a^{(1)}(z,r),a^{(2)}(z,r),\cdots,a^{(t)}(z,r)}(x)\right] \geq 1 - t/2^{n^{\varepsilon/500}},$$

so

$$\left|\Pr_{x\sim\mathsf{Column}_z^{(I)}}[C^*_k(x)=1] - \widetilde{p}_k(z)\right| \leq t/2^{n^{\varepsilon/500}} \leq 1/n^{\omega(1)}.$$

All the above can be done in $O\left(2^{n-n^{\varepsilon/100}}\right)$ time.

□

Combining with Lemma 3.8, we get:

Corollary 6.2. *For any constant $\varepsilon > 0$, E^{NP} does not have $\mathsf{THR}\circ\mathsf{THR}$ circuits with $\leq O\left(n^{2.5-\varepsilon}\right)$ gates.*

A Appendix

A.1 Proof of Item 4 of Lemma 3.4

Lemma A.1. *For any $\varepsilon > 0$, $\mathsf{SYM} \subseteq \mathsf{POLY}_{\mathbb{F}_2}[O(1)]\circ\mathsf{1hotSUM}$, where the SYM gate on the left hand side may have $\mathrm{poly}(n)$ many wires to each input gate, but on the right hand side, each $\mathsf{1hotSUM}$ gate has at most n^ε wires to each input bit, and there are at most $O(1)$ many $\mathsf{1hotSUM}$ gates. (Here $O(1)$ may depend on ε and the exponent of n in SYM's size.)*

Proof. Denote the SYM circuit by:

$$C(x) = f\left(\sum_{i=1}^n w_i x_i\right).$$

Here, each w_i is a nonnegative integer not more than $\mathrm{poly}(n)$, which is the number of wires to the input bit x_i. f is a function $f\colon \{0,1,\cdots,w_1+\cdots+w_n\}\to\{0,1\}$. We write each w_i as an integer in base n^ε, with $c+1=O(1)$ digits:

$$w_i=\sum_{j=0}^{c} v_{i,j}\cdot n^{\varepsilon j}.$$

Here, c is a constant that depends on ε and the exponent of n in SYM's size, and $0\le v_{i,j}<n^\varepsilon$.

Then we have

$$\sum_{i\in[n]} w_i x_i=\sum_{j=0}^{c} n^{\varepsilon j}\cdot\left(\sum_{i\in[n]} v_{i,j}x_i\right).$$

For each $0\le j\le c$, let G_j be a $\mathsf{1hotSUM}$ gate with $v_{i,j}$ (which is $\le n^\varepsilon$) wires to x_i for each $i\in[n]$. Recall that for any $0\le k\le\sum_{i\in[n]} v_{i,j}$, the k-th output bit of $G_j(x)$ (denote it by $(G_j(x))_k$) is 1 iff $\sum_{i\in[n]} v_{i,j}x_i=k$. Note that we have

$$C(x)=f\left(\sum_{j=0}^{c} n^{\varepsilon j}\cdot\left(\sum_{i\in[n]} v_{i,j}x_i\right)\right).$$

Writing the right hand side as a lookup table on inputs $\sum_{i\in[n]} v_{i,j}x_i$ $(0\le j\le c)$, we get

$$\begin{aligned}
C(x)&=\sum_{\substack{0\le k_0\le\sum_{i\in[n]} v_{i,0}\\ 0\le k_1\le\sum_{i\in[n]} v_{i,1}\\ \vdots\\ 0\le k_c\le\sum_{i\in[n]} v_{i,c}}} f\left(\sum_{j=0}^{c} n^{\varepsilon j}k_j\right)\cdot\mathbb{I}\left[k_j=\sum_{i\in[n]} v_{i,j}x_i\ (\forall 0\le j\le c)\right]\\
&=\sum_{\substack{0\le k_0\le\sum_{i\in[n]} v_{i,0}\\ 0\le k_1\le\sum_{i\in[n]} v_{i,1}\\ \vdots\\ 0\le k_c\le\sum_{i\in[n]} v_{i,c}}} f\left(\sum_{j=0}^{c} n^{\varepsilon j}k_j\right)\cdot(G_0(x))_{k_0}(G_1(x))_{k_1}\cdots(G_c(x))_{k_c}.
\end{aligned}$$

So C can be expressed as a degree-$(c+1)$ polynomial over $\mathbb{F}_2$ on the $\mathsf{1hotSUM}$ gates $G_0,G_1,\cdots,G_c$. Since $c\le O(1)$ and each $\sum_{i\in[n]} v_{i,j}\le\mathrm{poly}(n)$, the polynomial contains $\le\mathrm{poly}(n)$ many monomials. □

A.2 Proof of Lemma 3.9

Lemma A.2. *There is a deterministic $2^{n/2}\cdot\mathrm{poly}(n)$-time algorithm for THR-$\#\mathsf{SAT}$.*

Proof. We divide an input string $x=(x_1,\cdots,x_n)$ into two parts: $y=(x_1,x_2,\cdots,x_{n/2})$ and $z=(x_{n/2+1},x_{n/2+2},\cdots,x_n)$. We can write the THR gate as

$$\mathbb{I}\left[A(y)+B(z)\ge C\right],$$

where A,B are linear functions, and C is a constant.

Our $2^{n/2}\cdot\mathrm{poly}(n)$-time algorithm works as follows: we first compute the values of all $A(y)$-s ($y\in\{0,1\}^{n/2}$) and $B(z)$-s ($z\in\{0,1\}^{n/2}$). We sort all $B(z)$-s, and then for each y, use binary search to determine for how many z-s, we have $A(y)+B(z)\ge C$. Take the sum over all y-s we get the result for THR-$\#\mathsf{SAT}$. The algorithm runs in time $2^{n/2}\cdot\mathrm{poly}(n)$. □

A.3 Proof of Lemma 4.1

Lemma A.3. *Let $n > k$ be positive integers. Let $f\colon \{0,1\}^n \to \{0,1\}$ be a linear threshold function. Then for a random restriction $\rho \in \{0,1,*\}^n$ which has exactly k many $*$-s, with probability $\geq 1 - O(k/\sqrt{n})$, $f|_\rho$ is a constant function.*

In fact, Lemma A.3 can be generalized to unate functions:

Lemma A.3.1. *Let $n > k$ be positive integers. Let $f\colon \{0,1\}^n \to \{0,1\}$ be a unate function. Then for a random restriction $\rho \in \{0,1,*\}^n$ which has exactly k many $*$-s, the probability that $f|_\rho$ is a constant function is $\geq 1 - O(k/\sqrt{n})$.*

Proof of Lemma A.3.1. We only need to consider the case where $k \leq \sqrt{n}$.

We assumed that f is unate, but without loss of generality, we can assume f is monotone, since we can flip some input bits to make f monotone, not affecting the probability that $f|_\rho$ is a constant function. We define:

- Let P be the set of all restriction $\rho \in \{0,1,*\}^n$ which has exactly k many $*$-s. $|P| = \binom{n}{k} \cdot 2^{n-k}$.
- Let Π be the set of all shortest paths from 0^n to 1^n in the n-dimensional hypercube. $|\Pi| = n!$.
- Let

$$E := \big\{(\rho, \pi) \in P \times \Pi\colon f(\rho(0^n)) = 0,\ f(\rho(1^n)) = 1, \text{ and both } \rho(0^n), \rho(1^n) \text{ are in } \pi\big\}.$$

(We remark that $\rho(0^n), \rho(1^n)$ are strings obtained by replacing all $*$-s in ρ by 0-s, 1-s.)

We use two different ways to estimate $|E|$. On one hand, for each $\rho \in P$, if $f|_\rho$ is a constant, then there does not exist $\pi \in \Pi$ such that $(\rho, \pi) \in E$. If $f|_\rho$ is non-constant, suppose ρ has j 0-s and $(n-k-j)$ 1-s, then the number of π-s such that $(\rho, \pi) \in E$ is $k! \cdot j! \cdot (n-k-j)! = n!/(\binom{n}{k} \cdot \binom{n-k}{j})$, which is minimized at $j = \lfloor (n-k)/2 \rfloor$. Therefore,

$$|E| \geq \frac{n!}{\binom{n}{k} \cdot \binom{n-k}{\lfloor (n-k)/2 \rfloor}} \cdot \#\{\rho \in P\colon f|_\rho \text{ is non-constant}\}. \tag{14}$$

On the other hand, we prove that for each $\pi \in \Pi$, there are at most k ρ-s such that $(\rho, \pi) \in E$. Suppose all points on π are $0^n = v_0 \to v_1 \to \cdots \to v_n = 1^n$. Pick the q such that $f(v_q) = 0$ and $f(v_{q+1}) = 1$. Then for any ρ such that $(\rho, \pi) \in E$, $\rho(0^n)$ and $\rho(1^n)$ are two points on π with distance k, and they are on different sides of $v_q \to v_{q+1}$. So, there are at most k such choices of $(\rho(0^n), \rho(1^n))$ in π. Also note that ρ is uniquely determined by $(\rho(0^n), \rho(1^n))$, so there are at most k choices for ρ such that $(\rho, \pi) \in E$. Therefore,

$$|E| \leq k \cdot |\Pi| = k \cdot n!. \tag{15}$$

Combining Equations (14) and (15), we have

$$\begin{aligned} \frac{\#\{\rho \in P\colon f|_\rho \text{ is non-constant}\}}{|P|} &\leq \frac{\binom{n}{k} \cdot \binom{n-k}{\lfloor (n-k)/2 \rfloor}}{n!} \cdot \frac{|E|}{|P|} \\ &\leq \frac{\binom{n}{k} \cdot \binom{n-k}{\lfloor (n-k)/2 \rfloor}}{n!} \cdot \frac{k \cdot n!}{\binom{n}{k} \cdot 2^{n-k}} \end{aligned}$$

$$
\begin{aligned}
&= \frac{k \cdot \binom{n-k}{\lfloor (n-k)/2 \rfloor}}{2^{n-k}} \\
&\leq O\left(k/\sqrt{n-k}\right) \\
&\leq O\left(k/\sqrt{n}\right).
\end{aligned}
$$

The last step is because $k \leq \sqrt{n}$. So for a random restriction $\rho \in P$, with probability $\geq 1 - O\left(k/\sqrt{n}\right)$, $f|_\rho$ is a constant function. □

Proof of Lemma A.3. Note that linear threshold functions are unate, so this is just a subcase of Lemma A.3.1. □

A.4 Proof of Lemma 5.3

Lemma A.4. *There is a deterministic* $O\left(2^{n-n^{\varepsilon/50}}\right)$ *time algorithm that, given an* n*-input* $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/8}] \circ \mathsf{THR}$ *circuit* C *with at most* $\mathrm{poly}(n)$ *different* THR *gates at the bottom, and a partition* $I \in \binom{[n]}{n^{\varepsilon/10}}$*, outputs*

$$\Pr_{x \in \mathsf{Column}_z^{(I)}}[C(x) = 1]$$

for each $z \in \{0,1\}^{[n]\setminus I}$.

Proof.

Suppose C's bottom THR gates are $T_1, T_2, \cdots, T_m$, and its top $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/8}]$ gate is P.

By Item 2 of Lemma 3.4, we can transform each T_i into a disjoint or of ETHR gates: $T_i = \sum_{j=1}^{M_i} T_{i,j}$, where $M_i \leq \mathrm{poly}(n)$. Therefore, we can transform C into a $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/8}] \circ \mathsf{ETHR}$ circuit C_1, whose bottom ETHR gates are $T_{i,j}$-s, and its top POLY gate is

$$P_1\left(\{T_{i,j}\}\right) := P\left(\sum_{j=1}^{M_1} T_{1,j}, \sum_{j=1}^{M_2} T_{2,j}, \cdots, \sum_{j=1}^{M_m} T_{m,j}\right).$$

Note that C_1's top $\mathsf{POLY}_{\mathbb{F}_2}[n^{1-\varepsilon/8}]$ gate can be represented as $\oplus \circ \wedge$, where the fan-in of the top $\oplus$ gate is $\leq \binom{m}{n^{1-\varepsilon/8}} \leq 2^{n^{1-\varepsilon/8} \cdot O(\log n)}$, and by Item 1 of Lemma 3.4, the $\wedge$ gates and the bottom ETHR gates can be transformed into ETHR gates. Therefore, C_1 can be transformed into a $\oplus \circ \mathsf{ETHR}$ circuit C_2, where the top $\oplus$ gate has $\leq 2^{n^{1-\varepsilon/8} \cdot O(\log n)}$ fan-in.

Suppose C_2's bottom ETHR gates are $E_1, E_2, \cdots, E_N$ ($N \leq 2^{n^{1-\varepsilon/8} \cdot O(\log n)}$). Consider the modulus amplification polynomial $\mathsf{Amp}_{n^{\varepsilon/9}}$ (Lemma 3.12). Consider

$$C_3(z) := \sum_{y \in \{0,1\}^I} \mathsf{Amp}_{n^{\varepsilon/9}}\left(E_1(y;z) + E_2(y;z) + \cdots + E_N(y;z)\right) \ (z \in \{0,1\}^{[n]\setminus I}).$$

C_3 is a degree-$2n^{\varepsilon/9}$ polynomial over $E_i(y;\cdot)$-s ($i \in [N]$, $y \in \{0,1\}^I$). We replace the polynomial by $\mathsf{1hotSUM} \circ \wedge$, and transform the $\wedge$ and bottom ETHR-s into ETHR. Thus, C_3 can be computed by a $\mathsf{1hotSUM} \circ \mathsf{ETHR}$ circuit, where the top $\mathsf{1hotSUM}$ gate has fan-in

$$\leq \binom{N \cdot 2^{|I|}}{2n^{\varepsilon/9}} \leq 2^{o(n)}.$$

By Lemma 3.10, C_3's truth table can be computed in time $O\left(2^{n-n^{\varepsilon/20}}\right)$.

Note that for each $y \in \{0,1\}^I$ and $z \in \{0,1\}^{[n]\setminus I}$, we have

$$C(y;z) = (E_1(y;z) + E_2(y;z) + \cdots + E_N(y;z)) \bmod 2,$$

so

$$\mathsf{Amp}_{n^{\varepsilon/9}}\left(E_1(y;z) + E_2(y;z) + \cdots + E_N(y;z)\right) \equiv C(y;z) \pmod{2^{n^{\varepsilon/9}}},$$

and thus

$$C_3(z) \bmod 2^{n^{\varepsilon/9}} = \sum_{y\in\{0,1\}^{|I|}} C(y;z).$$

So we can compute

$$\Pr_{x\sim\mathsf{Column}_z^{(I)}}[C(x)=1] = \frac{C_3(z) \bmod 2^{n^{\varepsilon/9}}}{2^{n^{\varepsilon/10}}}.$$

The algorithm runs in $O\left(2^{n-n^{\varepsilon/50}}\right)$ time. □

References

[Aar16] Scott Aaronson. P =? NP. In John Forbes Nash Jr. and Michael Th. Rassias, editors, *Open Problems in Mathematics*, pages 1–122. Springer, 2016.

[ACW16] Josh Alman, Timothy M. Chan, and R. Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In Irit Dinur, editor, *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. IEEE Computer Society, 2016.

[Ajt83] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Ann. Pure Appl. Log.*, 24(1):1–48, 1983.

[AW15] Josh Alman and Ryan Williams. Probabilistic polynomials and hamming nearest neighbors. In *FOCS*, pages 136–150. IEEE Computer Society, 2015.

[BKK+22] Swapnam Bajpai, Vaibhav Krishan, Deepanshu Kush, Nutan Limaye, and Srikanth Srinivasan. A #sat algorithm for small constant-depth circuits with PTF gates. *Algorithmica*, 84(4):1132–1162, 2022.

[BT91] Richard Beigel and Jun Tarui. On ACC. In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 783–792. IEEE Computer Society, 1991.

[BW24] Gabriel Bathie and R. Ryan Williams. Towards stronger depth lower bounds. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 10:1–10:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[CLW20] Lijie Chen, Xin Lyu, and R. Ryan Williams. Almost-everywhere circuit lower bounds from non-trivial derandomization. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1–12. IEEE, 2020.

[CR20] Lijie Chen and Hanlin Ren. Strong average-case lower bounds from non-trivial derandomization. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1327–1334. ACM, 2020.

[CSS16] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-Case Lower Bounds and Satisfiability Algorithms for Small Threshold Circuits. In *31st Conference on Computational Complexity (CCC 2016)*, pages 1:1–1:35, 2016.

[CW19] Lijie Chen and R. Ryan Williams. Stronger connections between circuit analysis and circuit lower bounds, via pcps of proximity. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPIcs*, pages 19:1–19:43. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Syst. Theory*, 17(1):13–27, 1984.

[Hås89] Johan Håstad. Almost optimal lower bounds for small depth circuits. *Adv. Comput. Res.*, 5:143–170, 1989. Randomness and Computation (S. Micali, ed.).

[HHTT21] Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. Fooling constant-depth threshold circuits (extended abstract). In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 104–115. IEEE, 2021.

[HP10] 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, USA, June 9-12, 2010*, pages 270–279. IEEE Computer Society, 2010.

[IPS13] Russell Impagliazzo, Ramamohan Paturi, and Stefan Schneider. A Satisfiability Algorithm for Sparse Depth Two Threshold Circuits. In *54th IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 479–488, 2013.

[KL18] Valentine Kabanets and Zhenjian Lu. Satisfiability and derandomization for small polynomial threshold circuits. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 46:1–46:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[KW16] Daniel M. Kane and R. 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 Symposium on Theory of Computing (STOC 2016)*, pages 633–643, 2016.

[MTT61] S. Muroga, I. Toda, and S. Takasu. Theory of majority decision elements. *Journal of the Franklin Institute*, 271:376–418, 1961.

[MW18] Cody Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *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. ACM, 2018.

[Raz87] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987)*, pages 77–82. ACM, 1987.

[SSS95] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-hoeffding bounds for applications with limited independence. *SIAM J. Discret. Math.*, 8(2):223–250, 1995.

[SSTT15] Takayuki Sakai, Kazuhisa Seto, Suguru Tamaki, and Junichi Teruyama. A satisfiability algorithm for depth-2 circuits with a symmetric gate at the top and AND gates at the bottom. *Electron. Colloquium Comput. Complex.*, TR15-136, 2015.

[SSTT16] Takayuki Sakai, Kazuhisa Seto, Suguru Tamaki, and Junichi Teruyama. Bounded depth circuits with weighted symmetric gates: Satisfiability, lower bounds and compression. *Electron. Colloquium Comput. Complex.*, TR16-099, 2016.

[Sub61] Bella Abramovna Subbotovskaya. Realizations of linear functions by formulas using +,.,-. *Soviet Mathematics Doklady*, 2:110–112, 1961.

[Tam16] 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 TR16-100), 2016.

[Tel18] Roei Tell. Quantified Derandomization of Linear Threshold Circuits. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018)*, pages 855–865, 2018.

[Wil10] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 231–240. ACM, 2010.

[Wil11] Ryan Williams. Non-uniform ACC circuit lower bounds. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 115–125. IEEE Computer Society, 2011.

[Wil14] Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 194–202. ACM, 2014.

[Wil18] R. Ryan Williams. New Algorithms and Lower Bounds for Circuits With Linear Threshold Gates. *Theory of Computing*, 14(1):1–25, 2018.

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1985)*, pages 1–10. IEEE Computer Society, 1985.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il