Electronic Colloquium on Computational Complexity, Report No. 115 (2018)

Satisfiability and Derandomization for Small Polynomial Threshold Circuits

Valentine Kabanets* Zhenjian Lu†

June 11, 2018

Abstract

A polynomial threshold function (PTF) is defined as the sign of a polynomial $p\colon \{0,1\}^n \to \mathbb{R}$. A PTF circuit is a Boolean circuit whose gates are PTFs. We study the problems of exact and (promise) approximate counting for PTF circuits of constant depth.

- **Satisfiability (#SAT).** We give the first zero-error randomized algorithm faster than exhaustive search that counts the number of satisfying assignments of a given constant-depth circuit with a super-linear number of wires whose gates are s-sparse PTFs, for s almost quadratic in the input size of the circuit; here a PTF is called s-sparse if its underlying polynomial has at most s monomials. More specifically, we show that, for any large enough constant c, given a depth-d circuit with $(n^{2-1/c})$-sparse PTF gates that has at most $n^{1+\varepsilon_d}$ wires, where ε_d depends only on c and d, the number of satisfying assignments of the circuit can be computed in randomized time $2^{n-n^{\varepsilon_d}}$ with zero error. This generalizes the result by Chen, Santhanam and Srinivasan (CCC, 2016) who gave a SAT algorithm for constant-depth circuits of super-linear wire complexity with linear threshold function (LTF) gates only.
- **Quantified derandomization.** The quantified derandomization problem, introduced by Goldreich and Wigderson (STOC, 2014), asks to compute the majority value of a given Boolean circuit, under the promise that the minority-value inputs to the circuit are very few. We give a quantified derandomization algorithm for constant-depth PTF circuits with a super-linear number of wires that runs in quasi-polynomial time. More specifically, we show that for any sufficiently large constant c, there is an algorithm that, given a degree-Δ PTF circuit C of depth d with n^{1+1/c^d} wires such that C has at most $2^{n^{1-1/c}}$ minority-value inputs, runs in quasi-polynomial time $\exp\left((\log n)^{O(\Delta^2)}\right)$ and determines the majority value of C. (We obtain a similar quantified derandomization result for PTF circuits with n^{Δ}-sparse PTF gates.) This extends the recent result of Tell (STOC, 2018) for constant-depth LTF circuits of super-linear wire complexity.
- **Pseudorandom generators.** We show how the classical Nisan-Wigderson (NW) generator (JCSS, 1994) yields a nontrivial pseudorandom generator for PTF circuits (of unrestricted depth) with sub-linearly many gates. As a corollary, we get a PRG for degree-Δ PTFs with the seed length $\exp\left(\sqrt{\Delta \cdot \log n}\right) \cdot \log^2(1/\epsilon)$.

Keywords: constant-depth circuits, polynomial threshold functions, circuit analysis algorithms, SAT, #SAT, derandomization, quantified derandomization, pseudorandom generators.

*School of Computing Science, Simon Fraser University, Burnaby, BC, Canada; kabanets@sfu.ca
†School of Computing Science, Simon Fraser University, Burnaby, BC, Canada; zla54@sfu.ca

ISSN 1433-8092

Contents

1 Introduction

Satisfiability and derandomization are famous examples of "circuit analysis" problems that, apart from being important algorithmic problems in their own right, are also intimately related to the notoriously difficult problem of proving circuit lower bounds. In this paper, we give several algorithmic results for these problems for the class of Boolean circuits with polynomial-threshold functions (PTFs) as gates. A PTF is a generalization of the standard (weighted) majority function (linear threshold function) to the case of arbitrary polynomials, and is formally defined as the sign of a polynomial $p\colon \{0,1\}^n \to \mathbb{R}$.

We next describe in more detail the satisfiability and derandomization problems that we consider in this paper, as well as the class of PTF circuits for which we consider them. We then state our main results, and discuss our techniques.

1.1 Circuit-SAT

Circuit-SAT asks to determine whether a given Boolean circuit has a satisfying assignment. As a canonical NP-complete problem, it is not believed to have a polynomial-time (or subexponential-time) algorithm. However, it is still very interesting to look for nontrivial algorithms for Circuit-SAT running faster than naive exhaustive search. More specifically, given a circuit of polynomial size on n variables, is there a satisfiability algorithm that runs in time at most $2^n/n^{\omega(1)}$?

It turns out that this task is challenging even for very restricted classes of circuits. The difficulty of obtaining such a SAT algorithm can be partially explained by the work of Williams [Wil10, Wil11] showing that a Circuit-SAT algorithm faster than exhaustive search for a given class of circuits can often be used to prove nontrivial circuit lower bounds against that same class of circuits (given that the class of circuits satisfies some mild conditions). In fact, Williams designed such a Circuit-SAT algorithm for ACC circuits (constant-depth circuits with AND, OR, NOT, and modular counting gates) that runs in time $2^{n-n^{1/\exp(d)}}$ (recently improved to $2^{n-n^{1/\mathsf{poly}(d)}}$ by [CP16]), where d is the depth of the circuit, and then used this algorithm to show that NEXP contains a language that is not computable by any family of polynomial-size constant-depth ACC circuits, a breakthrough result in circuit complexity.

Given the connections between nontrivial Circuit-SAT algorithms and circuit lower bounds, one of the next big goals in circuit complexity is to design such an algorithm for the class of TC^0 circuits, constant-depth circuits with majority gates. Lower bounds against the class of polynomial-size TC^0 circuits is currently one of the most important open problems in complexity.

1.2 Derandomization

A central problem in derandomization is to give an efficient deterministic algorithm for computing the majority value of a given Boolean circuit, under the promise that the fraction of minority-value inputs to the circuit is at most $1/3$. That is, given a circuit that outputs some unknown value $b \in \{0,1\}$ on all but at most $1/3$ fraction of inputs, we need to determine this majority value b, efficiently deterministically.

As for Circuit-SAT, it is also known that a "faster-than-brute-force" algorithm solving the aforementioned derandomization problem for a circuit class $\mathcal{C}$ (satisfying some mild conditions) implies lower bounds against that class $\mathcal{C}$ [Wil10].

Black-Box Derandomization: Pseudorandom generators. One way to solve the derandomization problem for a class $\mathcal{C}$ of circuits is to construct a pseudorandom generator (PRG) for $\mathcal{C}$. A PRG for a class $\mathcal{C}$ of n-input Boolean circuits is an efficiently deterministically computable function G mapping short binary strings (seeds) to longer binary strings so that every $C \in \mathcal{C}$ accepts G's output on a uniformly random seed with about the same probability as that for an actual uniformly random string. More precisely, we say that a generator $G\colon \{0,1\}^r \to \{0,1\}^n$ is *ϵ-fooling* for a class $\mathcal{C}$ of Boolean circuits if for every $C\colon \{0,1\}^n \to \{0,1\}$ from $\mathcal{C}$

$$|\mathbf{Pr}[C(G(x)) = 1] - \mathbf{Pr}[C(y) = 1]| \leq \epsilon,$$

for uniformly random $x \in \{0,1\}^r$ and $y \in \{0,1\}^n$. The parameter r is called the seed length of the PRG. Then given a PRG that fools $\mathcal{C}$, for every $C \in \mathcal{C}$, we can estimate the fraction of accepted inputs to within an additive error ϵ, by trying all possibles seeds. This gives a deterministic algorithm solving the derandomization problem in time approximately 2^r.

Note that a PRG yields *black-box* derandomization in the sense that we do not need to be given as input a circuit $C \in \mathcal{C}$ in order to decide the set of 2^r query points for C; the set of 2^r query points is the same for all circuits in class $\mathcal{C}$.

Quantified derandomization. As standard derandomization appears difficult even for weak circuit classes, one considers relaxations. One relaxation is to assume that a given n-input circuit C outputs an unknown value $b \in \{0,1\}$ on all but "very few" inputs, e.g., $2^n/n^{\omega(1)}$ inputs rather than $2^n/3$ in the case of standard derandomization. Goldreich and Wigderson [GW14] named this a *quantified derandomization problem*. More formally, for a class $\mathcal{C}$ of circuits, and a function $B\colon \mathbb{N} \to \mathbb{N}$, the $(\mathcal{C}, B)$-quantified derandomization problem is the following: given a circuit $C \in \mathcal{C}$ such that C has at most $B(n)$ minority-value inputs in $\{0,1\}^n$, determine the majority value $b \in \{0,1\}$ for C.

It was immediately observed by [GW14] that for "sufficiently powerful" circuit classes (e.g., $\mathsf{AC}^0[\oplus]$, polynomial-size constant-depth circuits with unbounded fan-in AND, OR, parity gates, and negation gates), quantified derandomization is *equivalent* to standard derandomization, as one can perform efficient pseudo-random sampling (via randomness extractors) within the same circuit class. Thus, quantified derandomization may be possible to achieve (given our current knowledge) only for "very weak" circuit classes. [GW14] gave quantified derandomization algorithms for AC^0 (later strengthened by [Tel17a]) and some other classes. Recently, Tell [Tel18] showed that quantified derandomization is also possible for constant-depth LTF circuits of small super-linear wire complexity (and that improving this to slightly higher super-linear wire complexity is as hard as getting nontrivial standard derandomization for the circuit class TC^0, which in turn would imply TC^0 circuit lower bounds).

1.3 PTF circuits

The focus of the present paper is on circuits whose gates are polynomial threshold functions. An n-variate polynomial threshold function (PTF) is defined as the sign $\mathrm{sgn}(p)$ of a multi-linear polynomial $p\colon \{0,1\}^n \to \mathbb{R}$. Here, for $v \in \mathbb{R}$, we define the sign function $\mathrm{sgn}(v)$ to be 1 on $v > 0$, and 0 on $v < 0$. There are two common complexity measures for PTFs: *degree*, which is the degree of p, and *sparsity*, which is the number of monomials in p, where a monomial[1] is of the form

[1]Note that our definition of a monomial is different from the usual definition, where a monomial is a product of some variables (rather than literals, i.e., possibly negated variables). Our definition makes the class of s-sparse PTFs,

$\prod_{i \in S}(x_i \oplus b_i)$ where $S \subseteq [n]$ and $b_i \in \{0,1\}$ for each $i \in S$. We call the PTF s-sparse if $p(x_1, \ldots, x_n)$ is the sum of at most s monomials. PTFs of degree 1 are called linear threshold functions (LTFs). Thus an s-sparse PTF can be equivalently defined as an LTF of at most s terms, where each term is an AND of literals (variables and their negations).

Polynomial threshold circuits are circuits whose gates are PTFs. We will study both circuits with low-degree PTF gates and circuits with sparse PTF gates. We call a circuit degree-Δ PTF circuit if its gates are degree-Δ PTFs. Similarly, a circuit is called s-sparse PTF circuit if its gates are s-sparse PTFs. We note that when discussing circuits, the word "sparse" is often used to describe circuits with a small number of wires (recall that the number of wires is the sum of fan-ins over all gates of the circuit). To avoid ambiguity, we clarify that in this paper the word "sparse" always refers to PTFs. For example, a sub-quadratically sparse PTF circuit means a circuit with gates that are sub-quadratically sparse PTFs (i.e., PTFs that have a sub-quadratic number of monomials).

1.4 Our results

Circuit-SAT for sub-quadratically sparse PTF circuits with $n^{1+\varepsilon}$ wires. PTFs are very powerful even for small sparsity. For example, s-sparse PTFs can encode MAX-SAT with s clauses and exponential weights, a problem known how to solve nontrivially only for a sub-quadratic number of clauses. Therefore, a nontrivial SAT algorithm for PTFs of quadratic sparsity would break the current barrier of solving MAX-SAT with exponential weights. In fact, since a polynomial of degree-2 has at most a quadratic number of monomials, such an algorithm would also give a nontrivial SAT algorithm for degree-2 PTFs, which is currently unknown[2].

We give the first nontrivial #SAT algorithms (counting the number of satisfying assignments of a given circuit) for the class of constant-depth circuits with PTF gates, where the PTF circuit has small super-linear wire complexity (defined as the sum of fan-ins over all gates of the circuit) and each PTF gate has sub-quadratic sparsity. Our main result is the following.

Theorem 1.1 (#SAT algorithm for sub-quadratically sparse PTF circuits)**.** *There is a constant $b_1 > 1$ such that, for every $c \geq b_1$ and $d > 0$, there is a zero-error randomized algorithm that counts the number of satisfying assignments of any given depth-d, n-variate circuit with*

- *$(n^{2-1/c})$-sparse PTF gates, and*
- *at most $n^{1+\varepsilon_d}$ wires.*

The running time of this #SAT algorithm is at most $2^{n-n^{\varepsilon_d}}$, where $\varepsilon_d = c^{-3^d}$.

We also get an algorithm with better parameters if we further assume that the sparse PTF gates in the circuit have low degree. Let $\mathcal{G}_{\Delta,c}$ denote the class of Boolean functions where each function can be computed as an LTF of at most $n^{2-1/(c\cdot\Delta^2)}$ arbitrary Δ-variate Boolean functions.

for fixed sparsity s, much more expressive. For example, the polynomial $p(x_1, \ldots, x_n) = \prod_{i=1}^{n}(x_i \oplus 1) = \prod_{i=1}^{n}(1 - x_i)$ has sparsity 1 by our definition, but sparsity 2^n by the usual definition.

[2]Sakai, Seto, Tamaki and Teruyama [SSTT16] recently reported a faster-than-brute-force algorithm for MAX-k-SAT for any constant k with arbitrary weights (which implies a satisfiability algorithm for degree-k PTFs). However, their algorithm is conditional in that it relies on an assumption that one can *efficiently* reduce the weights of a given n-variate LTF to integral weights of magnitude at most $2^{O(n \log n)}$. While it is known that such small weights exist for every LTF [MTT61], it is currently not known how to find them efficiently.

Theorem 1.2 (#SAT algorithm for sub-quadratically sparse PTF circuits with low-degree)**.** *There exists a constant $b_2 > 1$ such that, for every $d, \Delta > 0$ and $c \geq b_2$, there is a zero-error randomized algorithm that counts the number of satisfying assignments of any given depth-d, n-variate circuit with*

- *gates from $\mathcal{G}_{\Delta,c}$, and*
- *at most $n^{1+\varepsilon_{d,\Delta}}$ wires.*

The running time of this #SAT algorithm is at most $2^{n-n^{\varepsilon_{d,\Delta}}}$, where $\varepsilon_{d,\Delta} = \left(c \cdot \Delta^2\right)^{-d}$.

Quantified derandomization for PTF circuits with $n^{1+\varepsilon}$ wires in quasi-polynomial time.

Theorem 1.3 (Quantified derandomization for low-degree (or sparse) PTF circuits)**.** *For any constant $c \geq 122$ and any $\Delta, d > 0$ such that $\Delta \ll \sqrt{\log n/(c^d \cdot \log\log n)}$, let $\mathcal{C} = \mathcal{C}(n, d, \Delta, c)$ be the class of n-variate, depth d PTF circuits with*

- *degree-Δ PTF gates (or n^{Δ/c^d}-sparse PTF gates), and*
- *at most n^{1+1/c^d} wires.*

The $\left(\mathcal{C}, 2^{n^{1-7/\sqrt{c}}}\right)$-quantified derandomization problem is solvable in time $2^{(\log n)^{O\left(\Delta^2\right)}}$.

PRG for PTF circuits with few gates.

Theorem 1.4 (PRG for PTF Circuits)**.** *There exists a constant $E > 0$ such that the following holds. For any positive integers α and Δ, let $\mathcal{C} = \mathcal{C}(n, \alpha, \Delta)$ be the class of degree-Δ PTF circuits on n inputs with at most $s = n^{\frac{1}{\alpha+1}} / \left(E \cdot 5^{\alpha\cdot\Delta} \cdot \log^2(n) \cdot \log(n/\epsilon)\right)$ gates. There exists a $\mathsf{poly}(n)$-time computable PRG $G\colon \{0,1\}^r \to \{0,1\}^n$ ϵ-fooling $\mathcal{C}$, where the seed length is $r = n^{2/(\alpha+1)}$.*

We get the following PRG for a single PTF (by setting α appropriately).

Corollary 1.5 (PRG for PTFs)**.** *There exists a PRG $G\colon \{0,1\}^r \to \{0,1\}^n$, computable in deterministic time $\mathsf{poly}(n)$, that ϵ-fools degree-Δ PTFs on n variables with the seed length*

$$r = \exp\left(O\left(\sqrt{\Delta \cdot \log n}\right)\right) \cdot \log^2(1/\epsilon).$$

1.5 Our techniques

A common way to analyze constant-depth circuits is to apply (random) restrictions, getting some depth reduction, and iterate, until the resulting circuit becomes very simple. Our #SAT algorithms and quantified derandomization algorithms for constant-depth PTF circuits also follow this approach, mainly relying on the ideas of [CSS16] for depth reduction, and [KKL17] for (pseudo-) random restrictions for PTFs. We give more details next.

Satisfiability for small PTF circuits. To get our Circuit-SAT algorithm, we generalize the analysis of the Circuit-SAT algorithm for small LTF circuits in [CSS16]. An oversimplified description is as follows. We show that for a depth-d circuit with a slightly super-linear number of wires, whose gates are sparse PTFs, there exists a shallow decision tree such that, for most of the leaves, the circuit restricted to that leaf can be "approximated" by some depth-$(d-1)$ circuit. Then we recursively apply a Circuit-SAT algorithm to depth-$(d-1)$ circuits. However, to actually implement this idea, we need three ingredients.

1. First, we need a base-case algorithm. In the case of LTF circuits, the base case is a conjunction of LTFs, and there is a known algorithm by Williams [Wil14] for such circuits. In contrast, in our case, the base case is a conjunction of sparse PTFs. Using the polynomial method in circuit complexity, we are able to design a Circuit-SAT algorithm for such circuits.

2. Secondly, to construct the decision tree, we need a random restriction lemma showing that, under a random restriction, a gate in the circuit is likely to be close to constant. In the case of LTFs, Chen et al. [CSS16] proved such a random restriction lemma for LTFs. Here, we show such a restriction lemma for sparse PTFs using a restriction lemma for low-degree PTFs in [KKL17].

3. Finally, since the restricted circuit under a leaf is only "approximated" by some circuit of lower depth, we need to handle the inputs where these two circuits disagree. This issue can be handled if we can enumerate the set of inputs where a gate evaluates to its minority value. As shown in [CSS16], there is an efficient way to do this for functions whose satisfiability can be decided in polynomial time, such as LTFs. However, we cannot apply this for sparse PTFs since there is no known polynomial-time SAT algorithm for sparse PTFs. We overcome this issue for sparse PTFs by reducing to the case of LTFs, using some ideas from Chen and Santhanam [CS15].

Quantified derandomization for small PTF circuits. Our quantified derandomization algorithm at the high level follows the approach of [GW14]. Given a circuit C with at most B minority-value inputs, the idea is to come up with a restriction ρ such that ρ leaves a large number of variables unrestricted, say n', and that C_ρ is very close to some simple function $\tilde{C}$ (say they agree on all but at most $1/6$ fraction of inputs). Then the number of minority-value inputs for $\tilde{C}$ is at most $B + 2^{n'}/6$. If B is also at most $2^{n'}/6$, then we can determine the required majority value for C by finding the majority value $\tilde{C}$, which is a *simple* function. (This approach is also used by Tell [Tel18] to get a quantified derandomization algorithm for LTF circuits with a slightly super-linear number of wires.)

Let's first consider a depth-2 LTF circuit with few wires. In [CSS16], Chen, Santhanam and Srinivasan proved a random restriction lemma for LTFs, which says that under a random restriction, an LTF is likely to become very close to an explicit constant. Using this result, one gets that under such a random restriction, many of the gates in the bottom layer of the circuit are expected to become close to constants. Since the circuit has only a few wires, one can further fix a small number of variables so that only those gates that are close to constants are left. Finally, by replacing these gate with their majority values, we obtain a single LTF that is close to the original depth-2 circuit.

Such a random restriction lemma was extended to low-degree PTFs in [KKL17], so we can conclude the same for low-degree PTF circuits. One important issue, though, is that the above "depth reduction" argument only holds for *random* restrictions (but with high probability). So to

get quantified derandomization, one will need to consider all possible restrictions. To handle this issue, Tell [Tel18] derandomized the random restriction lemma for LTFs mentioned above so that such a restriction can be sampled using few random bits. As a result, one only needs to consider a much smaller sample space of restrictions.

Now we need to apply the above idea to a depth-d circuit C. It seems that all we need to do is applying the pseudorandom restriction $d-1$ times. While this is true, the analysis is much more subtle. For example, after applying the first pseudorandom restriction ρ_1, we get a new circuit $\tilde{C}$ of depth $(d-1)$ on some n' variables so that it agrees with C_{ρ_1} on all but at most say $2^{n'}/6$ inputs. Now consider a subsequent restrictions ρ'. Note that the final number of unrestricted variable n'' after ρ' is much smaller than n'. Therefore, $(C_{\rho_1})_{\rho'}$ and $\tilde{C}_{\rho'}$ can disagree on all the inputs (since $2^{n'}/6 \gg 2^{n''}$) so $\tilde{C}_{\rho'}$ cannot be used to determine the correct output of $(C_{\rho_1})_{\rho'}$, which is also the correct output of C. It turns out that this issue can be handled if we can say that those bottom layer gates, which become close to constant after applying one step of pseudorandom restriction, will remain close to the *same* constant for the subsequent pseudorandom restriction. Such a "bias preservation lemma" for LTFs is also proved in [Tel18].

Both the pseudorandom restriction lemma for LTFs and the bias preservation lemma for LTFs in [Tel18] are obtained using a PRG for LTFs. One way to extend those results to PTFs is to use a PRG for PTFs. However, unlike LTFs, for which a PRG with a very short seed is known, all known PRGs for PTFs have a large seed length (for small error, which is needed for the argument). In fact, the only PRG that we can use in this case is the one in Corollary 1.5, and it would give a quantified derandomization algorithm running in time $2^{\exp\left(\sqrt{\Delta \cdot \log n}\right)}$. To get quasi-polynomial running time, we use a powerful pseudorandom "block restriction lemma" for PTFs in [KKL17] that uses only a poly-logarithmic number of random bits, and convert it into a form of pseudorandom restriction lemma that fits our needs. Also, we use an observation in [KKL17], which says that a concentrated PTF (see Definition 2.1) is likely to remain concentrated under any random restriction that fixes variables limited-wise independently, to get a similar bias preservation lemma for PTFs.

PRG for small PTF circuits. Our PRG is based on the celebrated Nisan-Wigderson "hardness-based" generator (NW PRG) [NW94]. To fool a class $\mathcal{C}$ of Boolean functions f, the NW PRG construction requires a "hard function" h that cannot be computed correctly on significantly more than a half of all possible inputs by any Boolean function g in a related class $\tilde{\mathcal{C}}$ of "slightly more powerful" functions than those from $\mathcal{C}$. Thus, sufficiently strong average-case lower bounds against the class $\tilde{\mathcal{C}}$ can be used to build a PRG fooling the class $\mathcal{C}$.

In our case, the class $\mathcal{C}$ contains all those n-variate Boolean functions that are computable by constant depth-d circuits with at most $s \ll n$ PTF gates of degree-Δ. Our main observation is that the corresponding class $\tilde{\mathcal{C}}$ (for which we require average-case lower bounds) is the class of Boolean functions computable by constant depth-d circuits with at most s PTF gates of degree $\Delta' = \alpha \cdot \Delta$, for some parameter $\alpha \geq 1$ that we can control (and which will determine the seed size of our PRG). That is, the class $\tilde{\mathcal{C}}$ is the same as $\mathcal{C}$, except for a somewhat higher degree Δ' of the allowed PTF gates.

To illustrate the idea of our analysis of the NW PRG for PTF circuits, we consider the special case of a single n-variate PTF f of degree Δ. That is, $f = \mathrm{sgn}(p(x_1, \ldots, x_n))$ for some degree-Δ multi-linear polynomial $p\colon \{0,1\}^n \to \mathbb{R}$. Suppose that the NW generator based on some "hard" Boolean function h failed to ϵ-fool this PTF f.

First, the standard NW analysis shows that the function $h(z)$ can be computed, with probability

at least $1/2 + \epsilon/n$, by (possibly the negation of) the function

$$g(z) = f(h_1(z), h_2(z), \ldots, h_i(z), b_{i+1}, \ldots, b_n), \tag{1}$$

for some $1 \leq i \leq n$, fixed bits $b_{i+1}, \ldots, b_n$, and Boolean functions $h_1, \ldots, h_i$, where each $h_j(z)$ depends on at most some α bits in z, for a parameter $a \geq 1$ coming from the NW construction (the maximum overlap between pairs of sets in the NW design; see Section 5 for details).

It is well known that every Boolean function on α inputs can be written as a multi-linear polynomial of degree α over the reals. Plugging in these polynomials for the function h_j's in Equation (1), we get that $g(z)$ is a PTF of degree at most $\Delta' = \alpha \cdot \Delta$.

Hence, to ensure that this NW generator based on h is indeed ϵ-fooling for degree-Δ PTFs, we just need h to be such that no PTF of degree-$(\alpha \cdot \Delta)$ can compute $h(z)$ on more than $1/2 + \epsilon/n$ of inputs z. Such hard functions h turn out to be easy to construct. For example, we use the average-case hard function for low-degree PTF circuits due to Nisan [Nis94].

The parameters of our PRG $G\colon \{0,1\}^r \to \{0,1\}^n$ (its error ϵ and seed length r) depend on the strength of the average-case lower bound for the hard function h. To get a short seed r, one needs to maximize the aforementioned parameter α, ideally setting $\alpha = \log n$ (as is the case for a standard application of the NW construction). However, we also need to prove (average-case) lower bounds against PTFs of degree $\alpha \cdot \Delta$, where virtually nothing is known for the degree $\log n$. Thus we are forced to set $\alpha \ll \log n$, which limits the stretch of our PRG to be at most only super-polynomial. On the other hand, for such a small α, our hard function h has exponentially small correlation with degree-$(\alpha \cdot \Delta)$ PTFs, thereby allowing our PRG to have an exponentially small error ϵ.

1.6 Related work and comparison

Circuit Satisfiability. Impagliazzo, Paturi and Schneider [IPS13] gave a Circuit-SAT algorithm for depth-2 LTF circuits with few wires; this result was improved by Chen and Santhanam [CS15]. A Circuit-SAT algorithm for constant-depth LTF circuits with few wires was recently given by Chen, Santhanam and Srinivasan [CSS16]. Alman, Chan and Williams [ACW16] and Tamaki [Tam16] both gave Circuit-SAT algorithms for depth-2 LTF circuits with an almost quadratic number of gates.

The most closely related previous work is by Chen, Santhanam and Srinivasan [CSS16] who gave a Circuit-SAT algorithm for circuits with a super-linear number of wires whose gates are LTFs. In particular, they show that the satisfiability of a depth-d, n-variate circuit with LTF gates and at most $n^{1+\varepsilon_d}$ wires can be solved by a zero-error randomized algorithm in time $2^{n-n^{\varepsilon_d}}$, where $\varepsilon_d = c^{-d}$ for some constant c. Our results extend their algorithm to the more general case of circuits with *sparse PTF* gates. In particular, our algorithm in Theorem 1.2 for $\Delta = 1$ subsumes the Circuit-SAT algorithm for LTFs in [CSS16]. Also note that the sparsity of the PTF gates in our model is almost quadratic in n, which is the input size of the circuit. "Opening up" the PTF gates in the circuit and expressing them as LTFs of terms will result in a (constant-depth) LTF circuit that can have an almost quadratic number of wires, and such a circuit can not be analyzed by the result in [CSS16].

Quantified derandomization. The quantified derandomization problem was first introduced by Goldreich and Wigderson in [GW14], where they obtained a polynomial time algorithm that finds the majority output of a given AC^0 circuit that has at most $2^{n^{0.999}}$ minority-value inputs.

The key tool in their algorithm is a derandomized version of Håstad's switching lemma [Hås89] with logarithmic seed length. In addition, they obtain quantified derandomization results for log-space algorithms and arithmetic circuits. The quantified derandomization algorithm for AC^0 was generalized by Tell [Tel17a] to handle AC^0 circuits with at most $2^{\Omega(n/\log^{d-2} n)}$ minority-value inputs, where d is the depth, with an increase of the running time to $2^{\tilde{O}(\log^3 n)}$. As mentioned above, Tell [Tel18] has recently obtained a quantified derandomization algorithm for depth-d LTF circuits with $n^{1+1/\exp(d)}$ wires with at most $2^{n^{1-1/5d}}$ minority-value inputs, running in time $n^{(\log\log n)^2}$. Our result extends this to low-degree PTF circuits and sparse PTF circuits, at the expense of increasing the running time to quasi-polynomial (for constant degree and polynomial sparsity). For the results on reducing standard derandomization to quantified derandomization, see [GW14, Tel17a, Tel17b, Tel18].

PRGs. There has been a long sequence of works on constructing PRGs (of varying strength) for various sub-classes of $\mathsf{P/poly}$. Among these known PRG constructions, some are NW-style "hardness-based" generators, while others are *ad hoc* constructions (often using such standard pseudorandomness tools as hashing, limited-wise independence, expander graphs, etc.) The previous PRGs for PTFs due to [MZ13, Kan12] are of the latter kind. The construction uses hashing and limited-wise independence. The analysis is quite involved, and depends on a number of analytic tools for polynomials (concentration and anti-concentration results, the invariance principle, hypercontractivity, regularization, etc.). In contrast, our PRG for PTFs (of Corollary 1.5) is the NW-style construction, whose analysis is simple, assuming an average-case lower bound for an appropriate class of functions.

For constant degree PTFs and constant error ϵ, the PRG of [MZ13, Kan12] has exponential stretch (mapping a seed of length $O(\log n)$ to an n-bit string fooling n-input PTFs). However, these PRGs has polynomial dependence in the error $1/\epsilon$ and cannot handle small error. Our PRG cannot achieve such exponentially long stretch for constant error, but it can achieve even exponentially small error ϵ with a nontrivial (sub-linear) seed size, which is impossible for the PRGs of [MZ13, Kan12].

In their work studying correlation bounds for AC^0 circuits with few symmetric gates [LS11], Lovett and Srinivasan obtained an average-case hard function for constant depth poly-size AC^0 circuits with few LTF gates and used it to construct a PRG fooling such circuits with polynomial stretch and exponentially small error, also based on the generic construction of Nisan and Wigderson. Since a PTF can be viewed as a depth-2 circuit computing an LTF of ANDs, such a PRG also fools small PTF circuits. While the PRG in [LS11] can fool a more general model, which is constant-depth AC^0 circuits augmented with LTF gates, it can have only polynomial seed stretch and the circuit can have only constant depth. Our work here focuses on circuits with only PTF gates. Our PRG can have sub-polynomial seed length and it can fool PTF circuits regardless of the depth as long as the number of gates is small. In particular, our PRG for a single PTF with sub-polynomial seed length (Corollary 1.5) can be used to construct a PRG for degree-2 PTFs with a seed length that is logarithmic in the input size and *sub-polynomial* in the error (see [KR18]).

Threshold circuits. It is well known that the class of constant-depth polynomial-size TC^0 circuits is equivalent to the class of constant-depth polynomial-size circuits with LTF gates [GHR92]. LTF circuits have been intensively studied in complexity theory. PTF circuits have been previously studied for lower bounds [Nis94, KKL17]. Threshold circuits are also studied as a model of artificial

neural networks [MP43] (see also [Ant01]), where a threshold gate is also called a neuron.

Remainder of the paper. We give the necessary background in Section 2. We describe our satisfiability algorithms (of Theorem 1.1 and Theorem 1.2) in Section 3. We prove our quantified derandomization results (Theorem 1.3) in Section 4, and our PRG result (Theorem 1.4) in Section 5. We conclude with some open problems in Section 6.

2 Preliminaries

2.1 Notation

For an positive integer n, let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, we define the *majority value* of f to be the bit value $b \in \{0,1\}$ that maximizes the quantity $\mathbf{Pr}_{x\sim\{0,1\}^n}[f(x) = b]$, and we call $1 - b$ the *minority value*.

We say that two Boolean functions f and g are δ-*close* if $\mathbf{Pr}_x[f(x) \neq g(x)] \leq \delta$. We say that a function f is δ-close to an *explicit* constant if f is δ-close to some constant function and such a constant function can be efficiently determined from f.

We will often view an s-sparse PTF as an LTF of at most s AND gates. It is well known that every LTF on m variables has a canonical representation, where the coefficients are integers of magnitude at most $2^{O(m \log m)}$ [MTT61]. Therefore, every s-sparse PTF is equivalent to some s-sparse PTF whose coefficients are integers of magnitude at most $2^{O(s \log s)}$. Without loss of generality, for a circuit with s-sparse PTF gates, we assume the coefficients of all gates have bit complexity $\mathsf{poly}(s)$.

2.2 Random restrictions

A random restriction is a process that randomly fixes the values of a subset of variables. We will often view a random restriction as a two step process: the first step is selecting (in some random manner) a subset of unrestricted variables and the second step is fix (in some random manner) the values of all the other variables. Depending on different contexts, we will consider different types of random restrictions based on how the unrestricted variables are picked and how the restricted variables are fixed.

Truly random restriction. The first type is the *(truly) r-random restriction*, for a parameter $0 < r < 1$. It is the process that leaves each variable, independently, free with probability r, and otherwise assigns it 0 or 1 uniformly at random.

Pseudorandom restriction using limited-wise independence. We can also pick and fix variables in a pseudorandom manner. One way to do this is to use a limited-wise independent distribution. For integers $n, m > 0$, a distribution X on $[m]^n$ is called k-wise independent if any k coordinates of X are uniformly distributed. That is, for any $1 \leq i_1, \ldots, i_k \leq n$ and every $b_1, \ldots, b_k \in [m]$, we have

$$\mathbf{Pr}[X_{i_1} = b_1, \ldots, X_{i_k} = b_k] = m^{-k}.$$

A k-wise independent distribution over $[m]^n$ can be constructed using $k \cdot \log n$ random bits for $m \leq n$ (see, e.g., [Vad12]).

For a random restriction ρ, we say that ρ picks the unrestricted variables k-wise independently, each with probability r, if each variable is set to be unrestricted by ρ with probability r, and any k of the variables are independent. Note that this process can be done using a k-wise independent distribution over $[1/r]^n$, where n is the number of variables. Also, we say that ρ fixes the variables k-wise independently if each variable is assigned 0 or 1 uniformly at random by ρ and any k of the variables are independent.

Random block restriction. A random block restriction picks the set of unrestricted variables by picking a block from some arbitrary predetermined partition of variables. More formally, an *m-block random restriction* for a function is the following process: given an arbitrary partitioning of input variables into m disjoint blocks, a random m-block restriction picks a uniformly random block $\ell \in [m]$ and fixes all variable outside the chosen block ℓ to 0 or 1 according to some distribution. Note that we can use a random block restriction to simulate the first two types of (pseudo-)random restrictions above. For example, to simulate a truly r-random block restriction, we first randomly partition the variables into $m = 1/r$ disjoint blocks, where each variable is assigned to block $i \in [m]$, independently, with probability $1/m$. Then we apply an m-random block restriction based on the partition in the previous step, by fixing the variables outside the selected block uniformly at random. Similarly, to simulate a pseudorandom restriction using limited-wise independence, we can partition (hash) the variables into disjoint blocks limited-wise independently, and apply a random block restriction where we fix the variables also using a limited-wise independent distribution.

2.3 Useful tools for analyzing PTFs

Definition 2.1 (δ-concentrated PTFs)**.** *Let $p\colon \{0,1\}^n \to \mathbb{R}$ be a degree-Δ multi-linear polynomial and $f = \operatorname{sgn}(p)$. For parameters $0 < \delta \le 1/2$ and $\lambda \ge 1$, we call p (and f) (δ, λ)-*concentrated *if*

$$\mathbf{Var}[p] \le (162 \cdot \log(1/\delta))^{-\lambda \cdot \Delta} \cdot \mathbf{Exp}\left[p^2\right],$$

where **Exp** *and* **Var** *denote the* expectation *and the* variance*, respectively, under the uniform distribution over $\{0,1\}^n$. We refer to $(\delta, 1)$-concentrated polynomials as δ-concentrated.*

A useful property of concentrated PTFs is that they are close to an explicit constant.

Lemma 2.2 (Concentrated implies close to constant)**.** *For any $0 < \delta \le 1/2$, if a PTF $f = \operatorname{sgn}(p)$ is δ-concentrated, then f is δ-close to the constant function* $\operatorname{sgn}(\mathbf{Exp}[p])$.

For a multi-linear polynomial $p\colon \{0,1\}^n \to \mathbb{R}$, it easy to see that $\mathbf{Exp}[p] = p(1/2, \ldots, 1/2)$, and so the expectation of p can be *computed efficiently* given access to p (either the evaluation oracle for p, or the full description of p via its coefficients). Thus, the constant function $\operatorname{sgn}(\mathbf{Exp}[p])$ from Lemma 2.2 is efficiently computable for a given polynomial p.

The following is a random restriction lemma for PTFs which says that a low-degree PTF is likely to become concentrated under a (truly) random block restriction.

Lemma 2.3 (Random block restriction lemma [KKL17] [3])**.** *For any $0 < \delta < 1$ and any positive*

[3] The original result in [KKL17] was stated for PTFs and polynomials for $\{1,-1\}$ domain. It is easy to see that it also holds for $\{0,1\}$ domain. This is because for every multi-linear polynomial $p\colon \{0,1\} \to \mathbb{R}$, there is the unique polynomial $p'\colon \{1,-1\} \to \mathbb{R}$ of the *same degree* such that, for any $x \in \{0,1\}^n$ there is an unique $y \in \{1,-1\}^n$ (that maps the 0's of x to 1 and the 1's to -1) such that $p(x) = p'(y)$. More precisely, for $p(x) = \sum_{S \subseteq [n]} c_S \cdot \Pi_{i \in S} x_i$, we get $p'(y) = \sum_{S \subseteq [n]} c_S \cdot \Pi_{i \in S}(1 - y_i)/2$. It is easy to see that the expectation (variance) of p' over $\{1,-1\}^n$ is the same as that of p over $\{0,1\}^n$.

integers m, λ, let $\mathcal{B}_m$ be a m-block random restriction that fixes variables uniformly at random. Then for degree-Δ PTF f whose variables are partitioned into m blocks, we have

$$\mathbf{Pr}_{\rho\sim\mathcal{B}_m}\left[f_\rho \text{ is not } (\delta,\lambda)\text{-concentrated}\right] \le m^{-1/2}\cdot(\log m\cdot\log(1/\delta))^{O\left(\lambda\cdot\Delta^2\right)}.$$

There is also a derandomized version of the above random block restriction lemma.

Lemma 2.4 (Pseudorandom block restriction lemma [KKL17])**.** *For any $0<\delta,\gamma<1$ and any positive integers m,λ, there is a polynomial-time algorithm for sampling a m-block random restriction $\mathcal{B}'_m$, that uses at most $m^\gamma\cdot\log n$ random bits, so that the following holds. For any n-variate degree-Δ PTF f whose variables are partitioned into m blocks, we have*

$$\mathbf{Pr}_{\rho\sim\mathcal{B}'_m}\left[f_\rho \text{ is not } (\delta,\lambda)\text{-concentrated}\right] \le m^{-1/2}\cdot(\log m\cdot\log(1/\delta))^{O(\lambda\cdot\Delta^2/\gamma)}.$$

Moreover, $\mathcal{B}'_m$ fixes the variables $(192\cdot\Delta\cdot\log(1/\delta))$-wise independently.

The following lemma says that a sparse PTF is likely to become a low-degree PTF under a mild (pseudo-)random restriction.

Lemma 2.5 (Degree reduction lemma)**.** *For any positive integer D, any $(\log n)^{-1}\ll\alpha<1$, let $\mathcal{R}$ be a random restriction such that*

- *$\mathcal{R}$ picks the unrestricted variables D-wise independently, each with probability $n^{-\alpha}$.*
- *$\mathcal{R}$ fixes the variables using a $(D\cdot\alpha\cdot\log n)$-wise independent distribution.*

Then for any s-sparse PTF f on n variables, we have

$$\mathbf{Pr}_{\rho\sim\mathcal{R}}[\deg(f_\rho)>D]\le s\cdot n^{-D\cdot\alpha/2}.$$

Proof. Let M be any monomial in f.

Suppose $|M|\le D\cdot\alpha\cdot\log n$. Then

$$\begin{aligned}
\mathbf{Pr}_{\rho\sim\mathcal{R}}[|M_\rho|\ge D] &\le \sum_{S\subseteq M:|S|=D}\mathbf{Pr}_\rho[\text{all variable in } S \text{ are set unrestricted by } \rho]\\
&= \sum_{S\subseteq M:|S|=D}\left(n^{-\alpha}\right)^D\\
&= \binom{|M|}{D}\cdot\left(n^{-\alpha}\right)^D\\
&\le \binom{D\cdot\alpha\cdot\log n}{D}\left(n^{-\alpha}\right)^D\\
&\le (e\cdot\alpha\cdot\log n)^D\cdot\left(n^{-\alpha}\right)^D\\
&\le \left(n^{\alpha/2}\cdot n^{-\alpha}\right)^D\\
&= n^{-D\cdot\alpha/2}.
\end{aligned}$$

In above, we use the fact that $\mathcal{R}$ picks the set of unrestricted variables D-wise independently, each with probability $n^{-\alpha}$.

Now suppose $|M| \geq D \cdot \alpha \cdot \log n$. Let S be any subset of M such that $|S| = D \cdot \alpha \cdot \log n$. Then

$$\begin{aligned}
\mathbf{Pr}_{\rho \sim \mathcal{R}}[|M_\rho| \geq D] &\leq \mathbf{Pr}_\rho[M_\rho \neq 0] \\
&= \mathbf{Pr}_\rho[\text{no variable in } S \text{ is set to 0 by } \rho] \\
&= \left(1 - \frac{1 - n^{-\alpha}}{2}\right)^{|S|} \\
&\leq \left(\frac{2}{3}\right)^{D \cdot \alpha \cdot \log n} \\
&\leq n^{-D \cdot \alpha/2}.
\end{aligned}$$

In above, we use the fact that $\mathcal{R}$ fixes the variables using a $(D \cdot \alpha \cdot \log n)$-wise independent distribution.

The lemma then follows by applying the union bound over all s monomials. □

3 #SAT algorithm for PTF circuits

In this section, we present our counting (#Circuit-SAT) algorithm in Theorem 1.1 for circuits with sparse PTF gates without any degree restriction on the monomials. We start with some useful tools.

Definition 3.1 (Probabilistic Polynomials)**.** *We say that a function $f\colon \{0,1\}^n \to \{0,1\}$ has an ϵ-error probabilistic polynomial of degree d if there is a distribution $\mathbf{P}$ of polynomials $p(x_1, \ldots, x_n) \in \mathbb{F}_2[x_1, \ldots, x_n]$ such that, for any $x \in \{0,1\}^n$, $\mathbf{Pr}_{p \sim \mathbf{P}}[f(x) \neq p(x)] \leq \epsilon$. We will call the distribution $\mathbf{P}$ a* probabilistic polynomial*.*

There are known constructions of probabilistic polynomials for LTFs and AND/OR functions that use few random bits.

Theorem 3.2 (Randomness-efficient probabilistic polynomials for LTFs [Sri13, Tam16])**.** *For any $0 < \epsilon < 1/2$ and LTF $f\colon \{0,1\}^n \to \{0,1\}$, f has a ϵ-error probabilistic polynomial $\mathbf{P}$ of degree at most $d = O\left(\sqrt{n \cdot \log(1/\epsilon)} \cdot \log^5 n\right)$. Moreover, $\mathbf{P}$ is samplable in time $\binom{n}{d} \cdot \mathsf{poly}(n)$ using $O\left(\log^2(n/\epsilon)\right)$ random bits.*

Theorem 3.3 (Randomness-efficient probabilistic polynomials for AND/OR [Raz87, CW16])**.** *For any $0 < \epsilon < 1/2$, AND/OR on n variables has a ϵ-error probabilistic polynomial $\mathbf{P}$ of degree at most $d = O\left(\log(1/\epsilon)\right)$. Moreover, $\mathbf{P}$ is samplable in time $\binom{n}{d} \cdot \mathsf{poly}(n)$ using $O\left(\log n \cdot \log(1/\epsilon)\right)$ random bits.*

We need the following useful tools for analyzing polynomials.

Lemma 3.4 (Fast multi-point polynomial evaluation [Yat37, Wil11])**.** *Let p be a n-variate polynomial given as a sum of monomials. Then p can be evaluated on all points in $\{0,1\}^n$ in time $2^n \cdot \mathsf{poly}(n)$.*

Theorem 3.5 (Toda's polynomials [Tod91, BT94])**.** *For any integer $\ell \geq 0$, there exists an explicit polynomial F_ℓ of degree $2\ell - 1$ such that the following holds*

1. *if* $y = 0 \mod 2$, *then* $F_\ell(y) = 0 \mod 2^\ell$.

2. *if* $y = 1 \mod 2$, *then* $F_\ell(y) = 1 \mod 2^\ell$.

We also need the followings.

Proposition 3.6 (see Section 4.1 of [CS15])**.** *Let $\phi_1, \ldots, \phi_s$ be a sequence of terms whose literals are from a set of n variables, where $s \geq n$. There exists a decision tree with at most $2^{n-\Omega(n^2/s)}$ leaves such that restricted to each leaf of the tree, ϕ_i contains at most 1 literal, for all $i \in [s]$.*

Proposition 3.7 (Proposition 5.2 of [CSS16])**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be an LTF with coefficients of bit complexity $\mathsf{poly}(n)$ and let S be the set of inputs on which f evaluates to 0 (or 1). Then S can be enumerated in time $|S| \cdot \mathsf{poly}(n)$*

3.1 Conjunction of sparse PTFs

First, we give a #Circuit-SAT algorithm for conjunctions of sparse PTFs, which is needed for our main algorithm as the base case. The algorithm is based on the framework for designing satisfiability algorithms developed by Williams [Wil11, Wil14]. The idea is to transform a given constant-depth circuit into a low-degree probabilistic polynomial and solve satisfiability by evaluating the polynomial on all points in a faster-than-brute-force manner. Applying this idea naively, we get a randomized SAT algorithm that makes error. Such a base-case algorithm would result in the final SAT algorithm for PTF circuits that also makes error. However, using some derandomization ideas similar to those in [CW16, Tam16], we are able to obtain a *deterministic* base-case algorithm that can count the number of satisfying assignments. This allows us to make our final SAT algorithm for PTF circuits to be *zero-error* randomized algorithm that *counts* the number of satisfying assignments.

Lemma 3.8. *There exists a deterministic algorithm that counts the number of satisfying assignments of every n-variate circuit C that is a conjunction of k s-sparse PTF gates, where $s \geq n$, such that the algorithm runs in time at most*

$$2^{n-\left(\frac{n}{\sqrt{s}\cdot(\log s)^{O(1)}\cdot\log^2 k}\right)^{1/2}}.$$

The following lemma says that a conjunction of sparse PTFs has low degree probabilistic polynomial that can be constructed using few random bits.

Lemma 3.9. *For any $0 < \epsilon < 1/2$ and n-variate circuit C that is a conjunction of k s-sparse PTF gates, C has an ϵ-error probabilistic polynomial $\mathbf{P}$ of degree $d = \sqrt{s} \cdot (\log s)^{O(1)} \cdot \log k \cdot \log(1/\epsilon)$. Moreover, $\mathbf{P}$ is samplable in time $\binom{n}{d} \cdot \mathsf{poly}(n)$ using $(\log s)^{O(1)} \cdot \log k \cdot \log(1/\epsilon)$ random bits.*

Proof. Let $f_1, \ldots, f_k$ be the s-sparse PTFs at the bottom (closest to the inputs) layer of C. We first consider the probabilistic polynomial for a single sparse PTF.

Claim 3.10. *Each f_i, $i \in [k]$, has a ϵ-error probabilistic polynomial of degree at most $d = \sqrt{s} \cdot (\log s)^{O(1)} \cdot \log(1/\epsilon)$ that is samplable in time $\binom{n}{d} \cdot \mathsf{poly}(n)$ using $(\log s)^{O(1)} \cdot \log(1/\epsilon)$ random bits.*

Proof of claim. We view f_i as an LTF of s AND gates. We first represent each of the AND gates at the bottom by a $\frac{1}{10 \cdot s}$-error probabilistic polynomial of degree $O(\log s)$, using Theorem 3.3. This takes $O(\log n \cdot \log s)$ random bits for a single AND gate. Then we represent the LTF gate (on s variables) with a $\frac{1}{10}$-error probabilistic polynomial of degree $O\left(\sqrt{s} \cdot \log^5 s\right)$, using Theorem 3.2. This takes $O\left(\log^2 s\right)$ random bits. By composing the probabilistic polynomial for the bottom ANDs with the probabilistic polynomial for the top LTF, we obtain, by the union bound, a $\frac{1}{5}$-error probabilistic polynomial of degree $\sqrt{s} \cdot (\log s)^{O(1)}$ for f_i. Note that since we are taking the union bound over the bottom AND gates here, we can use the same random bits to construct the probabilistic polynomials for all the AND gates. Finally, we sample $O(\log(1/\epsilon))$ independent copies of such polynomials for f_i and take the majority. Note that the majority function on t variables can be computed by a polynomial of degree t. By a standard concentration bound, we conclude that each f_i has an ϵ-error probabilistic polynomial of degree at most $\sqrt{s} \cdot (\log s)^{O(1)} \cdot \log(1/\epsilon)$. □

By Claim 3.10, we now can represent each f_i in C with a $\frac{1}{10 \cdot k}$-error probabilistic polynomial of degree $\sqrt{s} \cdot (\log s)^{O(1)} \cdot \log k$. This takes $(\log s)^{O(1)} \cdot \log k$ for a single f_i. Also, we represent the top AND gate of C by a $\frac{1}{10}$-error probabilistic polynomial of constant degree. We then obtain a $\frac{1}{5}$-error probabilistic polynomial for C of degree $\sqrt{s} \cdot (\log s)^{O(1)} \cdot \log k$. Again we use the same random bits for the f_i's as we are taking the union bound. Finally, by standard error reduction as described above, we get an ϵ-error probabilistic polynomial for C of degree $\sqrt{s} \cdot (\log s)^{O(1)} \cdot \log k \cdot \log(1/\epsilon)$. □

We are now ready to describe our #Circuit-SAT algorithm for conjunctions of sparse PTFs.

Proof of Lemma 3.8. Consider a subset of n' variables and a partial assignment $a \in \{0,1\}^{n'}$. Denote by C_a the restricted circuit where the values of these n' variables are fixed to a. We enumerate all such partial assignments in $\{0,1\}^{n'}$ and obtain a list of $r = 2^{n'}$ restricted circuits $C_{a_1}, \ldots, C_{a_r}$. Let

$$Q(x) = \sum_{i \in [r]} C_{a_i}(x).$$

Note that the number of satisfying assignments of C is equal to

$$\sum_{x \in \{0,1\}^{n-n'}} Q(x).$$

Now let $\mathbf{P^i}$ be an $\frac{1}{3r}$-error probabilistic polynomial of C_{a_i} of degree $\sqrt{s} \cdot (\log s)^{O(1)} \cdot \log k \cdot \log(r)$ given by Lemma 3.9. Consider the following quantity:

$$R^i(x) = \frac{\sum_{p \in \mathbf{P^i}} (p(x) \bmod 2)}{|\mathbf{P^i}|}.$$

Here we view $p \in \mathbf{P^i}$ as a polynomial with integer coefficients rather than a polynomial over $\mathbb{F}_2$. Since $\mathbf{P^i}$ is an $\frac{1}{3r}$-error probabilistic polynomial of C_{a_i}, we have

$$R^i(x) = C_{a_i}(x) \pm \frac{1}{3r}.$$

Then, we get

$$\sum_{i \in [r]} R^i(x) = Q(x) \pm \frac{1}{3}.$$

Therefore, to compute $Q(x)$, which is an integer, it suffices to compute $\sum_{i\in[r]} R^i(x)$. First, note that for all $i \in [r]$,

$$|\mathbf{P^i}| = M = 2^{(\log s)^{O(1)}\cdot \log k \cdot \log r}.$$

Let $\ell = (\log s)^{O(1)} \cdot \log k \cdot \log r$ so that $2^\ell \geq r \cdot M$. Let F be a degree $2\ell - 2$ polynomial given by Theorem 3.5. Then

$$\sum_{i\in[r]} R^i(x) = \frac{\sum_{i\in[r]} \sum_{p\in\mathbf{P^i}} \left(F(p(x)) \bmod 2^\ell\right)}{M} = \frac{\left(\sum_{i\in[r]} \sum_{p\in\mathbf{P^i}} F(p(x))\right) \bmod 2^\ell}{M}.$$

Now let

$$Q'(x) = \sum_{i\in[r]} \sum_{p\in\mathbf{P^i}} F(p(x)).$$

Note that Q' is a polynomial of degree at most

$$\ell \cdot \sqrt{s} \cdot (\log s)^{O(1)} \cdot \log k \cdot \log(r) \leq (n')^2 \cdot \sqrt{s} \cdot (\log s)^{O(1)} \cdot \log^2 k.$$

Moreover, Q' can be explicitly computed in time $2^{(n')^2\cdot\sqrt{s}\cdot(\log s)^{O(1)}\cdot\log^2 k}$. We then do fast multi-point evaluation (Lemma 3.4) of Q' to obtain the values $Q'(x)$ for each $x \in \{0,1\}^{n-n'}$. Note that we can recover $\sum_{i\in[r]} R^i(x)$, hence also $Q(x)$, from $Q'(x)$. We then sum over all x's in $\{0,1\}^{n-n'}$ to obtain the number of satisfying assignments of C. In total, the running time is at most

$$\tilde{O}\left(2^{n-n'} + 2^{(n')^2\cdot\sqrt{s}\cdot(\log s)^{O(1)}\cdot\log^2 k}\right).$$

Setting

$$n' = \left(\frac{n}{3\cdot\sqrt{s}\cdot(\log s)^{O(1)}\cdot\log^2 k}\right)^{1/2}$$

completes the proof. □

3.2 Depth reduction for sparse PTF circuits with few wires

In this section, we show how to use the random restriction lemma for PTFs (Lemma 2.3) to simplify a sparse PTF circuit with few wires.

Lemma 3.11. *For any integer $d \geq 2$ and any $(\log n)^{-1} \ll \varepsilon < 1$, let*

- *$\beta = E \cdot \varepsilon$, where E is some constant,*
- *$s \leq n^{O(1)}$,*
- *$\delta = \exp\left(-n^{\Omega(\beta^3)}\right)$,*
- *C be any depth-d, n-variate, s-sparse PTF circuit with at most $w = n^{1+\varepsilon}$ wires.*

Then there exists a decision tree T of depth $n - n^{1-\beta}$ such that, for a random leaf σ of T, with probability at least $1 - \exp(-n^\varepsilon)$, we have the following: C_σ is a depth-d circuit of wire complexity at most w such that its bottom layer has at most n gates that are δ-close to an explicit constant and at most n^β gates that are not δ-close to an explicit constant. Moreover, such a tree can be constructed in zero-error randomized time $\tilde{O}\left(2^{n-n^{1-\beta}}\right)$.

We need the following which is implicit in [CSS16].

Proposition 3.12 (see Section 4.1.1 of [CSS16])**.** *For any integer $d \geq 2$ and any $0 < \varepsilon < 1$, let*

- *$\beta = E \cdot \varepsilon$, where E is some constant,*
- *$r = n^{-\beta/2}$,*
- *$\mathcal{G}$ be any class of Boolean functions such that for each $g \in \mathcal{G}$,*

$$\mathbf{Pr}_{\rho \sim \mathcal{R}_r}[g_\rho \text{ is not } \delta\text{-close to an explicit constant}] \leq r^{\Omega(1)},$$

  where $\mathcal{R}_r$ denotes the truly r-random restriction,

- *C be any depth-d, n-variate circuit with gates from $\mathcal{G}$ and at most $w = n^{1+\varepsilon}$ wires.*

Then there exists a decision tree T of depth $n - n^{1-2\beta}$ such that, for a random leaf σ of T, with probability at least $1 - \exp(-n^{\varepsilon})$, we have the following: C_σ is a depth-d circuit of wire complexity at most w such that its bottom layer has at most n gates that are δ-close to constant and at most n^{β} gates that are not δ-close to constant. Moreover, such a tree can be constructed in zero-error randomized time $\tilde{O}\left(2^{n-n^{1-2\beta}}\right)$.

Proof of Lemma 3.11. The proof uses Proposition 3.12 and we need to show that a sparse PTF is likely to become close to an explicit constant under a r-random restriction. More specifically, we need to show that for any s-sparse PTF f,

$$\mathbf{Pr}_{\rho \sim \mathcal{R}_r}[f_\rho \text{ is not } \delta\text{-close to an explicit constant}] \leq n^{\Omega(\beta)} = r^{\Omega(1)}.$$

Let

- $r_1 = r_2 = \sqrt{r}$, where $r = n^{-\beta/2}$,
- $\delta = \exp(-n^{\beta^3/c_1})$, where $c_1 < B$ is a sufficiently large constant,
- $D = c_2 \cdot \beta^{-1}$, where $c_2 < c_1$ is a sufficiently large constant,

By Lemma 2.5, we have

$$\mathbf{Pr}_{\rho_1 \sim \mathcal{R}_{r_1}}[\deg(f_{\rho_1}) \geq D] \leq s \cdot n^{-c_2/8} \leq r^{\Omega(1)}. \tag{2}$$

Next, consider any degree-D PTF g and the random restriction $\mathcal{R}_{r_2}$. Note that $\mathcal{R}_{r_2}$ can be sampled equivalently as follows: first randomly partitioning the variables into $m = 1/r_2$ disjoint blocks so that each variable is assigned to each block with probability r_2, and then applying a random m-block restriction, where we fixes the variables outside of the chosen block uniformly at random. Then by Lemma 2.3 and Lemma 2.2, for every partition of variables, the probability that g restricted by a random block restriction is not δ-close to an explicit constant is at most

$$\sqrt{r_2} \cdot (\log(1/r_2) \cdot \log(1/\delta))^{O(D^2)} \leq n^{\Omega(\beta)} = r^{\Omega(1)}. \tag{3}$$

Finally, we have

$$\begin{aligned}
&\mathbf{Pr}_{\rho \sim \mathcal{R}_{r_1 \cdot r_2}}[f_\rho \text{ is not } \delta\text{-close to an explicit constant}] \\
&\leq \mathbf{Pr}_{\rho_1, \rho_2}[(f_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-close to an explicit constant} \mid \deg(f_{\rho_1}) < D] + \mathbf{Pr}_{\rho_1}[\deg(f_{\rho_1}) \geq D] \\
&\leq r^{\Omega(1)}. \qquad \text{(by Equations (2) and (3))}
\end{aligned}$$

□

3.3 Enumerating minority outputs of sparse PTFs

In our main algorithm, we will need to apply the depth reduction lemma (Lemma 3.11) to the circuit to conclude that many of the gates at the bottom layer will become close to constant so that we can replace them with actual constants. This changes the function of the circuit and we need to deal with the inputs where these gates do not evaluate to their majority values. As we will see, we can handle this issue if given a sparse PTF we can find the set of all inputs where it evaluates to its minority value, in a relatively efficient way. Then for the case of sparse PTF, we use Proposition 3.6 to reduce to the case of LTF, where we can perform this task efficiently using Proposition 3.7.

Lemma 3.13. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a s-sparse PTF with coefficients of bit complexity $\mathsf{poly}(n)$, where $s \geq n$ and let S be the set of inputs on which f evaluates to 0 (or 1). Then S can be enumerated in time $\left(2^{n-\Omega(n^2/s)} + |S|\right) \cdot \mathsf{poly}(n)$.*

Proof. We view f as an LTF of s AND gates. By Proposition 3.6, there exists a decision tree for f with at most $2^{n-\Omega(n^2/s)}$ leaves such that Φ restricted to each leaf is an LTF. We then go through each leaf σ and enumerate the set of inputs on which f_σ evaluates to 0. Let S_σ be the size of such set. By Proposition 3.7, this enumeration takes time $S_\sigma \cdot \mathsf{poly}(n)$. The total running time is the time for going through the leaves of the decision tree, which is at most $2^{n-\Omega(n^2/s)}$, and the time to enumerate the set of inputs evaluating to 0, which is at most $\sum_\sigma S_\sigma \cdot \mathsf{poly}(n) \leq |S| \cdot \mathsf{poly}(n)$. □

3.4 Putting it all together

Let $\varepsilon_d = 1/\left(E^{3^{d-1}}\right)$ and $\beta_d = E \cdot \varepsilon_d$, where E is a sufficiently large constant. We show the following.

Theorem 3.14. *For any integer $d \geq 1$, the number of satisfying assignments of a depth-d, n-variate circuit with $(n^{2-10\beta_2})$-sparse PTF gates and at most $n^{(1+\varepsilon_d)}$ wires can be computed by a zero-error randomized algorithm in time $\mathsf{poly}(n) \cdot 2^{n-n^{\Omega\left(\beta_d^3\right)}}$.*

Definition 3.15 (Skew Circuits)**.** *We say that a circuit C is (d,n,t,s)-skew if it is a n-variate circuit that can be expressed as a conjunction of some circuit C' and at most t s-sparse PTFs, where C' is a depth-d circuit with s-sparse PTF gates and has at most $w = n^{1+\varepsilon_d}$ wires. We call C' the skew subcircuit of C.*

Let $\mathcal{T}(d,n,t,s)$ denote the supremum, over all (d,n,t,s)-skew circuits C, of the randomized running time of counting the number of satisfying assignments of C. Throughout this subsection, we will assume that the constant E in the definition of ε_d and β_d is a sufficiently large constant.

The following lemma says that we can reduce the task of counting satisfying assignments of depth-d circuits to that of depth-$(d-1)$ circuits. This is done in a way that is similar to that in [CSS16].

Lemma 3.16. *If $s \leq n^{2-5\beta_d}$, then*

$$\mathcal{T}(d,n,t,s) \leq 2^{n-n^{1-2\beta_d}} \cdot 2^{n^{\beta_d}} \cdot \mathcal{T}\left(d-1, n^{1-2\beta_d}, t+2n, s\right) + \mathsf{poly}(n) \cdot 2^{n-n^{\Omega\left(\beta_d^3\right)}}.$$

Proof. Let C be any (d, n, t, s)-skew circuit, where its skew subcircuit is C'. To count the number of satisfying assignments of C. We first apply Lemma 3.11 to C' to get a decision tree with the claimed property. We then count the number of satisfying assignments at each leaves. For those "bad" leaves for which the conditions in Lemma 3.11 are not satisfied, we will simply do brute force on all $n^{1-2\beta_d}$ variables. The time to perform this is

$$2^{n-n^{1-2\beta_d}} \cdot \exp(-n^{\varepsilon_d}) \cdot 2^{n^{1-2\beta_d}} \leq 2^{n-n^{\varepsilon_d}}. \tag{4}$$

Next, consider a "good" leaf σ that satisfies the conditions in Lemma 3.11. We now describe how to count the number of satisfying assignments of C_σ. We call a gate *imbalanced* if it is δ-close to an explicit constant and *balanced* otherwise. Let $(g_1, \ldots, g_{\ell \leq n})$ be the set of imbalanced gates and $(a_1, \ldots, a_\ell)$ be their majority values. Let $(h_1, \ldots, h_{t \leq n^{\beta_d}})$ be the set of balanced gates.

We first count the number of satisfying assignments of C_σ in the following subset of inputs:

$$S = \{x : \exists\ i \in [\ell] \text{ for which } g_i(x) \neq a_i\}.$$

To do so, for each of the imbalanced gates, we enumerate the set of inputs on which it evaluates to its minority value, and keep those that satisfy the circuit C_σ. By Lemma 3.13, the running time for this is

$$\mathsf{poly}(n) \cdot \left(2^{n^{1-2\beta_d} - \Omega\left(\frac{n^{2\cdot(1-2\beta_d)}}{s}\right)} + 2^{n^{1-2\beta_d}} \cdot \delta \right), \tag{5}$$

where $\delta = \exp\left(-n^{\Omega\left(\beta_d^3\right)}\right)$. Note that for $s \leq n^{2-5\beta_d}$, we have

$$2^{n^{1-2\beta_d} - \Omega\left(\frac{n^{2\cdot(1-2\beta_d)}}{s}\right)} \leq 2^{n^{1-2\beta_d} - \Omega\left(n^{\beta_d}\right)} \leq 2^{n - n^{\Omega\left(\beta_d^3\right)}},$$

so Equation (5) is at most

$$\mathsf{poly}(n) \cdot 2^{n^{1-2\beta_d} - n^{\Omega\left(\beta_d^3\right)}}. \tag{6}$$

We do this for every imbalanced gate and obtain a set of satisfying inputs. In the end we simply take the union of these sets to get the satisfying assignments in S.

Next, we counts the number of satisfying assignments in

$$T = \{0,1\}^n - S.$$

Let $C'_{\sigma,a}$ be the circuit with those imbalanced gates in C'_σ replaced with their majority values (i.e., the values given by $(a_1, \ldots, a_\ell)$). Instead of counting the number of satisfying assignments for the original circuit C_σ, we consider the following circuit:

$$D = C'_{\sigma,a} \wedge \bigwedge_{i:\ a_i = -1} g_i \wedge \bigwedge_{i:\ a_i = 1} \neg g_i.$$

It is easy to see that $D(x) = 0$ for every $x \in S$ and $D(x) = C_\sigma(x)$ for every $x \in T$. We now need to count the number of satisfying assignments of D. We first partition T into 2^t subsets, each of which is indexed by some $b = (b_1, \ldots, b_t) \in \{0,1\}^t$, where the subset T_b given by the index b is

$$T_b = \{x : x \in T,\ h_1(x) = b_1, \ldots,\ h_t(x) = b_t\}.$$

To count the number of satisfying assignments of D in T_b. We consider the following circuit:

$$E_b = D_b \wedge \bigwedge_{i:\, b_i=-1} h_i \wedge \bigwedge_{i:\, b_i=1} \neg h_i,$$

where D_b is the circuit D with the balanced gates replaced by the values $b_1, \ldots, b_t \in \{0,1\}$. Again, we have $E_b(x) = 0$ for every $x \in [n] - T_b$ and $E_b(x) = D(x)$ for every $x \in T_b$. Now our task is reduced to counting the number of satisfying assignments of E_b for each $b \in \{0,1\}^t$. But note that each E_b is a conjunction of some depth-$(d-1)$ circuit (i.e., the skew subcircuit of E_b) and k s-sparse PTFs, where $k = t + n + n^\beta \leq t + 2n$. Also, the skew subcircuit has at most $n^{1+\varepsilon_d}$ wires, and we have

$$n^{1+\varepsilon_d} \leq \left(n^{1-2\beta_d}\right)^{1+\varepsilon_{d-1}}.$$

Therefore, each E_b is a $\left(d-1, n^{1-2\beta_d}, t+2n, s\right)$-skew circuits, and its number of satisfying assignments can be computed in time $\mathcal{T}\left(d-1, n^{1-2\beta_d}, t+2n, s\right)$. Then the total time for counting the number of satisfying assignments of the original circuit C_σ in the subset T is

$$2^t \cdot \mathcal{T}\left(d-1, n^{1-2\beta_d}, t+2n, s\right) \leq 2^{n^{\beta_d}} \cdot \mathcal{T}\left(d-1, n^{1-2\beta_d}, t+2n, s\right). \tag{7}$$

Therefore, by Equation (6) and Equation (7), counting the number of satisfying assignments of C_σ can be done in time

$$\mathsf{poly}(n) \cdot 2^{n^{1-2\beta_d} - n^{\Omega\left(\beta_d^3\right)}} + 2^{n^{\beta_d}} \cdot \mathcal{T}\left(d-1, n^{1-2\beta_d}, t+2n, s\right). \tag{8}$$

There are at most $L = 2^{n-n^{1-2\beta_d}}$ such leaves. Multiplying L by the running time in Equation (8) and combining Equation (4) yields the desired running time. □

Given the recursion in Lemma 3.16, we are now ready to prove Theorem 3.14.

Proof of Theorem 3.14. It suffices to show

$$\mathcal{T}\left(d, n, 0, s = n^{2-10\beta_2}\right) \leq \mathsf{poly}(n) \cdot 2^{n-n^{\Omega\left(\beta_d^3\right)}}. \tag{9}$$

We will iteratively apply Lemma 3.16 until we reach $d = 1$. Then we use the base case algorithm (Lemma 3.8) for the depth 1 case.

Recall that $\beta_i = E/E^{3^{i-1}}$ for $1 \leq i \leq d$. We will always assume E is a sufficiently large constant. Let $n_d = n$ and $n_i = n_{i+1}^{1-2\beta_{i+1}}$. That is, n_i is the number of variables of the circuit after its skew subcircuit has depth i. It is easy to see by induction that for $1 \leq i \leq d$, $\sum_{j=i+1}^{d} \beta_j \leq \beta_i$. Then we have

$$n_i \geq n^{1-2\cdot\sum_{j=i+1}^{d}\beta_j} \geq n^{1-4\beta_{i+1}}. \tag{10}$$

By Lemma 3.16, if $s \leq n^{2-5\cdot\beta_d}$, then

$$\mathcal{T}(d, n, t, s) \leq 2^{n-n_{d-1}} \cdot 2^{n^{\beta_d}} \cdot \mathcal{T}\left(d-1, n_{d-1}, t+2n, s\right) + \mathsf{poly}(n) \cdot 2^{n-n^{\Omega\left(\beta_d^3\right)}} \tag{11}$$

Now note that using Equation (10), we have for all $3 \leq i \leq d$,

$$n_i^{2-5\cdot\beta_i} \geq n^{(1-4\beta_{i+1})(2-5\beta_i)} \geq n^{2-10\beta_2} = s.$$

Using Equation (11), we unwind $\mathcal{T}(d,n,0,s)$ $d-1$ times, and we get

$$\mathcal{T}(d,n,0,s) \leq 2^{n-n_1}\cdot 2^{n_2^{\beta_2}}\cdot \mathcal{T}(1,n_1,k,s)+\sum_{i=3}^{d} 2^{n_i^{\beta_i}}\cdot \mathsf{poly}(n_i)\cdot 2^{n-n_{i-1}^{\Omega\left(\beta_{i-1}^3\right)}}+\mathsf{poly}(n)\cdot 2^{n-n^{\Omega\left(\beta_d^3\right)}}, \quad (12)$$

where $k=\sum_{i=2}^{d} 2n_i \leq 2dn$.

We now upper bound Equation (12). We first upper bound the first summand in Equation (12). By Lemma 3.8 and Equation (10), we have

$$2^{n_2^{\beta_2}}\cdot \mathcal{T}(1,n_1,k,s) \leq 2^{n_2^{\beta_2}}\cdot 2^{n_1-\left(\frac{n_1}{\sqrt{s}\cdot(\log s)^{O(1)}\cdot \log^2 k}\right)^{1/2}} \leq 2^{n_1-n^{\Omega\left(\beta_d^3\right)}},$$

so the first summand is at most $2^{n-n^{\Omega\left(\beta_d^3\right)}}$.

We now upper bound the second summand. Note for $3\leq i\leq d$, we have

$$\Omega\left(\beta_{i-1}^3\right)=\Omega\left(E^3/E^{3^i}\right)\geq E\cdot\beta_i.$$

Thus,

$$2^{n_i^{\beta_i}}\cdot 2^{n-n_{i-1}^{\Omega\left(\beta_{i-1}^3\right)}} \leq 2^{n_i^{\beta_i}}\cdot 2^{n-n_{i-1}^{E\cdot\beta_i}} \leq 2^{n_i^{\beta_i}}\cdot 2^{n-n^{E\cdot\beta_i/2}} \leq 2^{n-n^{\Omega\left(\beta_d^3\right)}}. \quad (13)$$

Therefore, the second summand is also bounded by

$$\mathsf{poly}(n)\cdot 2^{n-n^{\Omega\left(\beta_d^3\right)}}.$$

□

3.5 Circuits with gates that are LTFs of few functions

In this section, we describe our #Circuit-SAT algorithm in Theorem 1.2 for circuits whose gates are LTFs of few functions of small arity. The algorithm is similar to that in the previous subsections, and we only provide a sketch here.

We need a different base-case algorithm, as now the base case is a circuit that is a conjunction of gates each of which is an LTF of few arbitrary functions of bounded arity. We need the following #Circuit-SAT algorithm for such circuits.

Lemma 3.17. *Let $\mathcal{G}_{s,\Delta}$ be the class of Boolean functions computable by an LTF of s arbitrary Δ-variate functions, and let C be an n-variate circuit that is a conjunction of k $\mathcal{G}_{s,\Delta}$ gates. There exists a deterministic algorithm that counts the number of satisfying assignments of every such circuit C and runs in time at most*

$$2^{n-\sqrt{n}\cdot\left(s^{1/4}\cdot(\log s)^{O(1)}\cdot(\log k)\cdot\Delta\right)^{-1}}.$$

The proof of the above algorithm is similar to that of Lemma 3.8. We first show that a conjunction of k functions from the class $\mathcal{G}_{s,\Delta}$ has a probabilistic polynomial of degree at most

$$\sqrt{s}\cdot(\log s)^{O(1)}\cdot\log k\cdot\log(1/\epsilon)\cdot\Delta,$$

as in the proof of Lemma 3.9. The only difference is that for a function f from $\mathcal{G}_{s,\Delta}$, viewed as an LTF of functions with bounded arity Δ, we can simply replace those bottom functions with bounded arity Δ with polynomials of degree Δ (instead of using Theorem 3.3), since any function on Δ variables can be computed by a polynomial of degree Δ (over any field).

As for random restriction lemma, we can directly use Lemma 2.3, since every function in $\mathcal{G}_{s,\Delta}$ can be expressed as a degree-Δ PTFs. Following an argument similar to the proof of Lemma 3.11, we get that

Lemma 3.18. *For any integers $\Delta \geq 1$, $d \geq 2$, let*

- $\varepsilon_{d,\Delta} = (E \cdot \Delta)^{-(2d-1)}$ *and* $\beta_{d,\Delta} = E \cdot \Delta^2 \cdot \varepsilon_{d,\Delta}$, *where E is a sufficiently large constant,*
- $\delta = \exp\left(-n^{\Omega(\beta_{d,\Delta}/\Delta^2)}\right) = \exp\left(-n^{\varepsilon_{d,\Delta}}\right)$,
- *C be any depth-d, n-variate, degree-Δ PTF circuit with at most $w = n^{1+\varepsilon}$ wires.*

Then there exists a decision tree T of depth $n - n^{1-2\beta_{d,\Delta}}$ such that, for a random leaf σ of T, with probability at least $1-\delta$, we have the following: C_σ is a depth-d circuit of wire complexity at most w such that its bottom layer has at most n gates that are δ-close to an explicit constant and at most $n^{\beta_{d,\Delta}}$ gates that are not δ-close to an explicit constant. Moreover, such a tree can be constructed in zero-error randomized time $\tilde{O}\left(2^{n-n^{1-2\beta_{d,\Delta}}}\right)$.

Finally, to enumerate the set of inputs where a function from $\mathcal{G}_{s,\Delta}$ evaluates to its minority value, we can use the following.

Proposition 3.19 (see Section 3.2 of [CS15])**.** *Let $\phi_1, \ldots, \phi_s$ be a sequence of arbitrary Δ-constrains whose literals are from a set of n variables, where $S \geq n$. There exists a decision tree with at most $2^{n-\Omega(n^2/(s\cdot\Delta^2))}$ leaves such that restricted to each leaf ϕ_i, $i \in [s]$, contains at most 1 literal.*

Then combining with Proposition 3.6, we have the following which is analogous to Lemma 3.13

Lemma 3.20. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be an LTF of at most s arbitrary Δ-variate functions with coefficients of bit complexity $\mathsf{poly}(n)$, where $s \geq n$, and let S be the set of inputs on which f evaluates to 0 (or 1). Then S can be enumerated in time $\left(2^{n-\Omega(n^2/(s\cdot\Delta^2))} + |S|\right) \cdot \mathsf{poly}(n)$.*

The final algorithm then follows from an argument that is similar to that of Section 3.4 with a different settings of parameters, which now are $\varepsilon_{d,\Delta}$ and $\beta_{d,\Delta}$.

4 Quantified derandomization for PTF circuits

In this section, we prove our quantified derandomization results.

4.1 Pseudorandom restrictions for PTFs

We prove in this subsection some pseudorandom restriction lemmas for both low-degree PTFs and sparse PTFs, which will be used to reduce the depth of a PTF circuit with few wires. We obtain these pseudorandom restriction lemmas from the pseudorandom block restriction lemma (Lemma 2.4). We first show for low-degree PTFs.

Lemma 4.1 (Pseudorandom restriction lemma for low-degree PTFs)**.** *For any constant $c_1 > 0$, any $\alpha_1 < 1$ and any positive integer Δ such that $\Delta \ll \sqrt{\alpha_1 \cdot \log n / \log\log n}$, there is a random restriction $\mathcal{R}^1$ such that the following holds:*

- *$\mathcal{R}^1$ picks the unrestricted variables $(\log n)$-wise independently, each with probability $n^{-\alpha_1}$.*
- *$\mathcal{R}^1$ fixes the variables $(600 \cdot c_1 \cdot \Delta \cdot \log n)$-wise independently.*
- *$\mathcal{R}^1$ can be sampled in polynomial time using $(\log n)^{O(\Delta^2)}$ random bits.*
- *For any degree-Δ PTF f on n variables,*

$$\mathbf{Pr}_{\rho\sim\mathcal{R}^1}[f_\rho \text{ is not } (n^{-c_1}, 3)\text{-concentrated}] \leq n^{-\alpha_1/3}.$$

Proof. We define $\mathcal{R}^1$ by describing the following process of sampling a random restriction from $\mathcal{R}^1$:

1. Hash the variables into n^{α_1} blocks $(\log n)$-wise independently.
2. Apply a $\left(n^{\alpha_1/2}\right)$-block pseudorandom restriction from Lemma 2.4 for degree-Δ PTFs, with parameters
   - $\delta = n^{-c_1}$ and $\lambda = 3$.
   - $\gamma = (c_1 \cdot \Delta^2 \cdot \log\log n)/(\alpha_1 \cdot \log n)$, where c_1 is a sufficiently large constant (note that $\gamma < 1$ for $\Delta \ll \sqrt{\alpha_1 \cdot \log n / \log\log n}$).

We now argue that the random restriction $\mathcal{R}^1$ has the desired properties. For the first item, it is easy to see from the above that $\mathcal{R}^1$ picks the set of unrestricted variables $(\log n)$-wise independently, each with probability $1/n^{-\alpha_1/2}$. The second item follows from Lemma 2.4 that the pseudorandom block restriction fixes the variables $(600 \cdot c_1 \cdot \Delta \cdot \log n)$-wise independently. For the third item, note that to sample from $\mathcal{R}^1$, we need $\mathsf{polylog}(n)$ random bits for its first step, and the number of random bits for its second step is

$$n^{\alpha_1 \cdot \gamma} \cdot \log n \leq (\log n)^{O(\Delta^2)}.$$

Finally, for the last item, note that in the above process of sampling $\mathcal{R}^1$, for any partition into n^{α_1} blocks generated in the first step, by Lemma 2.4, the probability over the restrictions in the second step that the restricted PTF is not $(n^{-c_1}, 3)$-concentrated is at most $n^{-\alpha_1/2} \cdot (\log n)^{O(\Delta^2/\gamma)}$. Thus,

$$\mathbf{Pr}_\rho[f_\rho \text{ is not } (n^{-c_1}, 3)\text{-concentrated}] \leq n^{-\alpha_1/2} \cdot (\log n)^{O(\Delta^2/\gamma)} \leq n^{-\alpha_1/2} \cdot n^{\alpha_1/6} \leq n^{-\alpha_1/3}.$$

□

To obtain a similar pseudorandom restriction lemma for sparse PTFs, we combine the pseudorandom restriction lemma for low-degree PTFs (Lemma 4.1) and the degree reduction lemma (Lemma 2.5) to get the following pseudorandom restriction lemma for sparse PTFs.

Lemma 4.2 (Pseudorandom restriction lemma for sparse PTFs)**.** *For any constant $c_2 > 0$, any $\alpha_2 < 1$ and any positive integer Δ such that $\Delta \ll \sqrt{\alpha_2 \cdot \log n / \log\log n}$, there is a random restriction $\mathcal{R}^2$ such that the following holds:*

- $\mathcal{R}^2$ *picks the unrestricted variables* $(\log n)$*-wise independently, each with probability* $n^{-\alpha_2}$*.*
- $\mathcal{R}^2$ *fixes the variables* $(600 \cdot c_2 \cdot \Delta \cdot \log n)$*-wise independently.*
- $\mathcal{R}^2$ *can be sampled in polynomial time using* $(\log n)^{O(\Delta^2)}$ *random bits.*
- *For any* $n^{\Delta \cdot \alpha_2}$*-sparse PTF* f *on* n *variables,*

$$\mathbf{Pr}_{\rho \sim \mathcal{R}^2}[f_\rho \text{ is not a degree-}(4\Delta) \ (n^{-c_2}, 2)\text{-concentrated PTF}] \leq n^{-\alpha_2/5}.$$

Proof. The idea is first applying a random restriction from Lemma 2.5 to reduce the degree of the sparse PTFs, and then using a random restriction from Lemma 4.1 for low degree PTFs.

Let ρ_1 be a random restriction that picks the set of unrestricted variables $(\log n)$-wise independently, each with probability $n^{-\alpha_2/2}$, and fixes the other variables using a $(600 \cdot c_2 \cdot \Delta \cdot \log n)$-wise independent distribution. Note that, by Lemma 2.5 (with parameters $D = 4\Delta$ and $\alpha = \alpha_2/2$), we have

$$\mathbf{Pr}_{\rho_1}[\deg(f_{\rho_1}) > 4\Delta] \leq n^{-\alpha_2/2}.$$

Let ρ_2 be a random restriction from Lemma 4.1 for degree-4Δ PTFs with parameters $\beta_0 = \alpha_2/2$.

Define $\mathcal{R}^2$ to be the random restriction $\rho_1 \circ \rho_2$. Note that since both ρ_1 and ρ_2 pick the set of unrestricted variables $(\log n)$-wise independently, each with probability $n^{-\alpha_2/2}$, the set of unrestricted variables is picked by $\mathcal{R}^2$ $(\log n)$-wise independently, each with probability $n^{-\alpha_2}$. For the second item, note that ρ_1 can be sampled using $\mathsf{polylog}(n)$ random bits and ρ_2 can be sampled using $(\log n)^{O(\Delta^2)}$ random bits. Therefore, the total number of random bits needed to sample ρ is at most $(\log n)^{O(\Delta^2)}$. Finally, we have

$$\begin{aligned}
&\mathbf{Pr}_{\rho \sim \mathcal{R}}[f_\rho \text{ is not a degree-}(4\Delta) \ (n^{-c_2}, 2)\text{-concentrated PTF}] \\
&= \mathbf{Pr}_{\rho_1, \rho_2}[(f_{\rho_1})_{\rho_2} \text{ is not a degree-}(4\Delta) \ (n^{-c_2}, 2)\text{-concentrated PTF}] \\
&\leq \mathbf{Pr}_{\rho_1, \rho_2}[(f_{\rho_1})_{\rho_2} \text{ is not } (n^{-c_2}, 2)\text{-concentrated} \mid \deg(f_{\rho_1}) \leq 4\Delta] + \mathbf{Pr}_{\rho_1}[\deg(f_{\rho_1}) > 4\Delta] \\
&\leq n^{-\alpha_2/4} \cdot (\log n)^{O(\Delta^2/\gamma)} + n^{-\alpha_2/2} \\
&\leq n^{-\alpha_2/4} \cdot (\log n)^{O(\Delta^2/\gamma)} \\
&\leq n^{-\alpha_2/4} \cdot n^{\alpha_2/20} \\
&= n^{-\alpha_2/5}.
\end{aligned}$$

□

Finally, we will need the following lemma which says that a concentrated PTF is likely to remain concentrated under any random restriction that fixes variables limited-wise independently.

Lemma 4.3 (see Lemma 4.2 and Claim 7.7 of [KKL17])**.** *Let* $f = \mathrm{sgn}(p)$ *be any degree-*Δ *PTF that is* $(\delta, \lambda + 1)$*-concentrated. Let* ρ *be a random restriction that fixes any subset of variables according to some* $(192 \cdot \Delta \cdot \log(1/\delta))$*-wise independent distribution. Then with probability at least* $1 - \delta$ *we have*

- f_ρ *is* (δ, λ)*-concentrated.*
- $\mathrm{sgn}(\mathbf{Exp}(p_\rho)) = \mathrm{sgn}(\mathbf{Exp}(p))$.

The above means that if a PTF is $(\delta, 2)$-concentrated and hence close to some constant. Then the restricted PTF is likely to remain close to the same constant.

4.2 Quantified derandomization for sparse PTF circuits

Let $\mathcal{G}$ be a class of Boolean functions, we say that a circuit C is a $(n, d, w, s, \mathcal{G})$-sparse PTF circuit if

1. C is an n-variate circuit of depth-d with at most w wires and
2. C has s-sparse PTFs as its gates except for the top gate, which is a function from $\mathcal{G}$.

Similarly, we use $(n, d, w, \Delta, \mathcal{G})$-low-degree PTF circuits for the analogous type of circuits whose gates (except for the top gate) are degree-Δ PTFs.

For a class of Boolean functions $\mathcal{G}$, we denote by $\mathrm{Appr}_{n,\epsilon}(\mathcal{G})$ the running time, given an n-variate function g from $\mathcal{G}$, of approximating the acceptance probability of g to within an additive error ϵ.

We first show the following.

Theorem 4.4. *For any constant $E \geq 11$ and any positive integers Δ and d such that $\Delta \ll \sqrt{\varepsilon_d \cdot \log n / \log\log n}$, where $\varepsilon_d = E^{-2(d-1)}$, let $\mathcal{C}$ be the class of $\left(n, d, n^{1+\varepsilon_d}, n^{\Delta\cdot\varepsilon_d}, \mathcal{G}\right)$-sparse PTF circuits. Then the $\left(\mathcal{C}, 2^{n^{1-7/E}}\right)$-quantified derandomization problem is solvable in time $2^{(\log n)^{O\left(\Delta^2\right)}} \cdot \mathrm{Appr}_{n,1/6}(\mathcal{G})$.*

Theorem 4.4 implies Theorem 1.3 for sparse PTF circuits since we can always add a dummy gate (e.g., AND) to the top of a PTF circuits, which only increase the depth by 1.

We will iteratively use the pseudorandom restriction lemma (Lemma 4.2) to reduce the depth of the circuit until the circuit has depth 1. We first show how to do this in one step.

Lemma 4.5. *For any constants $E \geq 11$, $c > 0$, any $\varepsilon \leq 1/(7E)$, and any positive integer Δ such that $\Delta \ll \sqrt{E \cdot \varepsilon \cdot \log n / \log\log n}$, there is a polynomial time algorithm that, given a $\left(n, d, n^{1+\varepsilon}, n^{\Delta\cdot\varepsilon}, \mathcal{G}\right)$-sparse PTF circuit C and a random seed of length $(\log n)^{O\left(\Delta^2\right)}$, outputs the following with probability at least $1 - n^{\varepsilon}$:*

- *A restriction $\rho \in \{0, 1, *\}^n$ that leaves $n' = n^{1-3E\cdot\varepsilon}$ variables unrestricted and that the restricted variables are fixed $(600 \cdot c \cdot \Delta \cdot \log n)$-wise independently.*
- *A $\left(n', d-1, (n')^{1+7E\cdot\varepsilon}, (n')^{2\Delta\cdot\varepsilon}, \mathcal{G}\right)$-sparse PTF circuit $\tilde{C}$ such that for all subsequent random restriction ρ' that fixes the variables in a $(600 \cdot c \cdot \Delta \cdot \log n)$-wise independent manner, with probability $1 - n^{-c}$ over ρ', it holds that $\tilde{C}_{\rho'}$ is n^{-c}-close to $(C_\rho)_{\rho'}$.*

For the second item, we say that the restriction ρ' is good (for $\tilde{C}$ and C_ρ) if it holds that $\tilde{C}_{\rho'}$ is n^{-c}-close to $(C_\rho)_{\rho'}$.

The proof of Lemma 4.5 is similar to that in [Tel18], which is based on the argument in [CSS16], but requires some critical modifications. We sketch the proof below and highlight these modifications.

Proof sketch. Let $\beta = E \cdot \varepsilon$ and $p = n^{-\beta}$. The restriction ρ consists of three sub-restrictions.

ρ_1: Preprocessing. Fix each of the variables with fan-out greater than $2n^{\varepsilon}$ using a $(600 \cdot c \cdot \Delta \cdot \log n)$-wise independent distribution. Since the number of wires is at most $n^{1+\varepsilon}$, it can be easily seen that the number of variables needed to be fixed is at most $n^{1+\varepsilon}/(2n^{\varepsilon}) = n/2$.

ρ_2: Pseudorandom restriction to simplify PTFs. Let ρ_2 be a random restriction from Lemma 4.2 with parameters $\alpha_2 = \beta$ and $c_2 = 2c$. Note that ρ_2 fixes the variables $(600 \cdot c \cdot \Delta \cdot \log n)$-wise independently. Now by Lemma 4.2, after ρ_2, we expect all but at most a fraction of $n^{-\beta/5}$ of the gates in the bottom layer to become $(n^{-2c}, 3)$-concentrated. Moreover, since the number of unrestricted variables is picked in a $(\log n)$-wise independent manner, by a Chernoff-type concentration bound (for k-wise independence), the fan-in of each of the non-concentrated gates (there are only about a fraction of $n^{-\beta/5}$ of such gates) will shrink by a factor of p with high probability, assuming they have large fan-ins. Then we can expect to eliminate all those non-concentrated gates by fixing a small number of variables. As for the gates with small fan-ins, using a simple graph theoretic argument along with the condition given by the preprocessing step, we can also eliminate those gate by fixing a few variables. More precisely, as in [CSS16, Tel18], it can be shown that with probability except $O\left(n^{-\beta/10}\right)$, over the random restriction ρ_2, the following holds: there is a set T of variables such that all the bottom layer gates that are not $(n^{-2c}, 3)$-concentrated can be replaced by constants after fixing the variables in T. The number of unrestricted variables after applying ρ_2 and fixing T is at least $n^{1-3E\cdot\varepsilon}$.

ρ_3: Eliminate non-concentrated gates. We will use a $(600 \cdot c \cdot \Delta \cdot \log n)$-wise independent distribution to fix the variable in the set T described above. Note that the number of unrestricted variables is at least $n' = n^{1-3E\cdot\varepsilon}$. We may further fix additional variables so that the number of unrestricted variables is exactly n'. Although this restriction eliminates all non-concentrated gates in the bottom layer, it may also cause some concentrated gates to become non-concentrated. However, by Lemma 4.3, the probability that each of these gate is not $(n^{-2c}, 2)$-concentrated is at most n^{-2c}. By the union bound, we get with probability all but $n^{-2c} \cdot n^{1+\varepsilon} \leq n^{-c}$, all these gates remain $(n^{-2c}, 2)$-concentrated.

Obtaining $\tilde{C}$. By above, with probability at least $1 - O\left(n^{-\beta/10}\right)$, we have a restriction ρ such that all the bottom layer gates of C_ρ are $(n^{-2c}, 2)$-concentrated and hence close to some associated constants. Let's call these constants V. $\tilde{C}$ is the circuit obtained from C_ρ by replacing those concentrated gates in the bottom with the constants V. Let's argue that $\tilde{C}_{\rho'}$ and $(C_\rho)_{\rho'}$ are n^{-c}-close to each other for any subsequent random restriction ρ' that fixes the variables $(600\cdot c\cdot\Delta\cdot\log n)$-wise independently. Consider such a subsequent random restriction ρ' and the restricted circuit $(C_\rho)_{\rho'}$. By Lemma 4.3, with probability except n^{-2c}, the bottom layer gates of $(C_\rho)_{\rho'}$, which are just the bottom layer gates of C_ρ, are still (n^{-2c})-concentrated. Moreover, they are close to the same constants V. Now by replacing these gates in $(C_\rho)_{\rho'}$ with the constants V, we obtain a circuit C'. By a union bound, C' and $(C_\rho)_{\rho'}$ are (n^{-c})-close to each other. On the other hand, consider the circuit $\tilde{C}$, which is obtained by replacing the concentrated gates in the bottom C_ρ with the constant V. Note that $\tilde{C}_{\rho'} = C'$. Thus, $\tilde{C}_{\rho'}$ and $(C_\rho)_{\rho'}$ are n^{-c}-close to each other. Finally, we need to show that $\tilde{C}$ is a $\left(n', d-1, (n')^{1+4E\cdot\varepsilon}, (n')^{\Delta\cdot E\cdot\varepsilon}, \mathcal{G}\right)$-sparse PTF circuit. As for the number of wires in $\tilde{C}$, note that

$$(n')^{1+7E\cdot\varepsilon} = n^{(1-3E\cdot\varepsilon)\cdot(1+7E\cdot\varepsilon)} \geq n^{1+\varepsilon}.$$

Also, we have

$$(n')^{2\Delta\cdot\varepsilon} = n^{(1-3E\cdot\varepsilon)\cdot 2\Delta\cdot\varepsilon} \geq n^{\Delta\cdot\varepsilon}.$$

□

We are now ready to describe our quantified derandomization algorithm.

Proof of Theorem 4.4. For $1 \le i \le d$, define $\varepsilon_i = E^{-2(i-1)}$. Also define $n_d = n$ and $n_i = n_{i+1}^{1-3E\cdot\varepsilon_{i+1}}$. It is easy to see by induction that for $1 \le i \le d$, $\sum_{j=i+1}^{d} \varepsilon_j \le \varepsilon_i$. Then we have

$$n_i \ge n^{1-3E\cdot\sum_{j=i+1}^{d}\varepsilon_j} \ge n^{1-6E\cdot\varepsilon_{i+1}}. \tag{14}$$

Let ρ be a sequence of d random restriction $\rho_1, \cdots, \rho_d$, such that ρ_i is a random restriction from Lemma 4.5 with parameters ε_i and $c = 1$. We say that the restriction ρ_i is successful if the two items in Lemma 4.5 are satisfied.

We claim the following.

Claim 4.6. *Let $K_d = \sum_{i=2}^{d} n_i^{-1} \le 1/10$. The probability, over ρ, that C_ρ is not (K_d)-close to a $\mathcal{G}$ function is at most $2 \cdot \sum_{i=2}^{d} n_i^{-\varepsilon_i} \le 1/3$.*

Proof of claim. We prove by induction on the depth d. The base case $d = 2$ follows from Lemma 4.5. Now suppose the claim holds for $d-1$, we show that it holds for d. We have

$$\begin{aligned}
&\mathbf{Pr}_\rho[C_\rho \text{ is not } K_d\text{-close to } \mathcal{G}] \\
&\le \mathbf{Pr}_{\rho_1,\rho'=\rho_2,\ldots,\rho_d}[(C_{\rho_1})_{\rho'} \text{ is not } K_d\text{-close to } \mathcal{G} \mid \rho_1 \text{ is successful}] + \mathbf{Pr}_{\rho_1}[\rho_1 \text{ is not successful}] \\
&\le \mathbf{Pr}_{\rho_1,\rho'}[(C_{\rho_1})_{\rho'} \text{ is not } K_d\text{-close to } \mathcal{G} \mid \rho_1 \text{ is successful and } \rho' \text{ is good}] \cdot \mathbf{Pr}_{\rho'}[\rho' \text{ is good}] \\
&\quad + \mathbf{Pr}_{\rho'}[\rho' \text{ is not good}] + n^{-\varepsilon_d} \\
&\le \mathbf{Pr}_{\rho_1,\rho'}[(C_{\rho_1})_{\rho'} \text{ is not } K_d\text{-close to } \mathcal{G} \mid \rho_1 \text{ is successful and } \rho' \text{ is good}] \cdot \mathbf{Pr}_{\rho'}[\rho' \text{ is good}] \\
&\quad + 2 \cdot n^{-\varepsilon_d}.
\end{aligned} \tag{15}$$

Now if ρ_1 is successful and ρ' is good, then by Lemma 4.5, $(C_{\rho_1})_{\rho'}$ is n^{-1}-close to some circuit $\tilde{C}_{\rho'}$, where $\tilde{C}$ is a $\big(n', d-1, (n')^{1+7E\cdot\varepsilon_d}, (n')^{2\Delta\cdot\varepsilon_d}, \mathcal{G}\big)$-sparse PTF circuit, with

- $n' = n^{1-3E\cdot\varepsilon_d} = n_{d-1}$.
- $(n')^{1+7E\cdot\varepsilon_d} \le (n_{d-1})^{1+\varepsilon_{d-1}}$.
- $(n_{d-1})^{2\Delta\cdot\varepsilon_d} \le (n_{d-1})^{\Delta\cdot\varepsilon_{d-1}}$

Also, if $(C_{\rho_1})_{\rho'}$ is n^{-1}-close to $\tilde{C}_{\rho'}$ and $\tilde{C}_{\rho'}$ is (K_{d-1})-close to $\mathcal{G}$, then $(C_{\rho_1})_{\rho'}$ would be K_d-close to $\mathcal{G}$. Therefore, Equation (15) is at most

$$\begin{aligned}
&\mathbf{Pr}_{\rho_1,\rho'}[\tilde{C}_{\rho'} \text{ is not } (K_{d-1})\text{-close to } \mathcal{G} \mid \rho_1 \text{ is successful and } \rho' \text{ is good}] \cdot \mathbf{Pr}_{\rho'}[\rho' \text{ is good}] + 2 \cdot n^{-\varepsilon_d} \\
&\le \mathbf{Pr}_{\rho'}[\tilde{C}_{\rho'} \text{ is not } (K_{d-1})\text{-close to } \mathcal{G} \mid \rho' \text{ is good}] \cdot \mathbf{Pr}_{\rho'}[\rho' \text{ is good}] + 2 \cdot n^{-\varepsilon_d} \\
&\le \mathbf{Pr}_{\rho'}[\tilde{C}_{\rho'} \text{ is not } (K_{d-1})\text{-close } \mathcal{G}] + 2 \cdot n^{-\varepsilon_d}.
\end{aligned}$$

By the induction hypothesis, the above is at most $2 \cdot \sum_{i=2}^{d} n_i^{-\varepsilon_i}$.

□

If the original circuit C has at most $2^{n^{1-7/E}}$ bad inputs, then C_ρ (on $n_1 \ge n^{1-6/E}$ variables) also has at most $2^{n^{1-7/E}}$ bad inputs. Now suppose the restriction ρ is successful that we obtain a single $\mathcal{G}$ function that is $\big(\frac{1}{10}\big)$-close to C_ρ. Then it must accept have at most

$$\frac{1}{10} \cdot 2^{n_1} + 2^{n^{1-7/E}} \le \frac{1}{6} \cdot 2^{n_1}$$

bad inputs. If we now approximate the acceptance probability of this $\mathcal{G}$ function within error $1/6$, we can correctly determine the correct value.

Finally, we can enumerate all the possible seeds and take the majority vote to decide the correct answer. □

4.3 Quantified derandomization for low-degree PTF circuits

Here, we briefly describe the quantified derandomization algorithm for low-degree PTF circuits.

Theorem 4.7. *For any constant $E \geq 11$ and any positive integers Δ and d such that $\Delta \ll \sqrt{\varepsilon_d \cdot \log n / \log\log n}$, where $\varepsilon_d = E^{-2(d-1)}$, let $\mathcal{C}$ be the class of $\left(n, d, n^{1+\varepsilon_d}, \Delta, \mathcal{G}\right)$-low-degree PTF circuits. Then the $\left(\mathcal{C}, 2^{n^{1-7/E}}\right)$-quantified derandomization problem can be solved in time $2^{(\log n)^{O\left(\Delta^2\right)}} \cdot \mathrm{Appr}_{n,1/6}(\mathcal{G})$.*

The above result can be proved in the same way as Theorem 4.4, using the following one-step pseudorandom restriction for low-degree PTF circuits.

Lemma 4.8. *For any constants $E \geq 11$, $c > 0$, any $\varepsilon \leq 1/(7E)$, and any positive integer Δ such that $\Delta \ll \sqrt{E \cdot \varepsilon \cdot \log n / \log\log n}$, there is a polynomial time algorithm that, given a $\left(n, d, n^{1+\varepsilon}, \Delta, \mathcal{G}\right)$-low-degree PTF circuit C and a random seed of length $(\log n)^{O\left(\Delta^2\right)}$, outputs the following with probability at least $1 - n^{\varepsilon}$:*

- *A restriction $\rho \in \{0, 1, *\}^n$ that leaves $n' = n^{1-3E\cdot\varepsilon}$ variables unrestricted and that the restricted variables are fixed $(600 \cdot c \cdot \Delta \cdot \log n)$-wise independently.*
- *A $\left(n', d-1, (n')^{1+7E\cdot\varepsilon}, \Delta, \mathcal{G}\right)$-sparse PTF circuit $\tilde{C}$ such that for all subsequent random restriction ρ' that fixes the variables in a $(600 \cdot c \cdot \Delta \cdot \log n)$-wise independent manner, with probability $1 - n^{-c}$ over ρ', it holds that $\tilde{C}_{\rho'}$ is n^{-c}-close to $(C_\rho)_{\rho'}$.*

The proof of the above lemma is similar to Lemma 4.5 for the sparse PTF gate case, but uses Lemma 4.1 instead of Lemma 4.2. Given Lemma 4.8, we can prove Theorem 1.3 in the same way as proving Theorem 1.3 in the previous section.

5 PRG for PTF circuits

In this section, we present our NW-style PRG for low-degree PTF circuits with few gates.

Theorem 5.1. *There exists a constant $E > 0$ such that for any positive integers α, Δ and any degree-Δ PTF circuit C on n variables with at most $s = n^{\frac{1}{\alpha+1}} \cdot \left(E \cdot 5^{\alpha\cdot\Delta} \cdot \log^2(n) \cdot \log(n/\epsilon)\right)^{-1}$ gates, there exists a $\mathsf{poly}(n)$-time computable PRG $G\colon \{0,1\}^r \to \{0,1\}^n$ ϵ-fooling C, with the seed length $r = n^{2/(\alpha+1)}$.*

We first need a (average-case) hard function for such circuits.

Theorem 5.2 ([Nis94])**.** *There exists a constant $E > 0$ such that for any degree $\Delta \geq 1$, there exists a polynomial-time computable function $f\colon \{0,1\}^n \to \{0,1\}$ such that for any error parameter ϵ and any n-variate degree-Δ PTF circuit C with at most $n \cdot \left(E \cdot 5^{\Delta} \cdot \log^2(n) \cdot \log(1/\epsilon)\right)^{-1}$ gates, we have*

$$\mathbf{Pr}_{x\sim\{0,1\}^n}\left[C(x) = f(x)\right] \leq \frac{1}{2} + \epsilon.$$

Next we apply the Nisan-Wigderson construction to the hard function of Theorem 5.2. We will use the following (standard) combinatorial designs.

Claim 5.3 (NW Designs [NW94])**.** *For any positive integers n, α, there exists an efficiently computable family of sets $S_1, \ldots, S_n$ such that*

- $S_i \subset [r]$, $\forall i \in [n]$, *where* $r = n^{2/(\alpha+1)}$,
- $|S_i| = \ell = n^{1/(\alpha+1)}$, $\forall i \in [n]$, *and*
- $|S_i \cap S_j| \leq \alpha$, $\forall i, j \in [n]$ *such that* $i \neq j$.

Proof. We view the set $[r]$ as the set of pairs $\mathbb{F}_\ell \times \mathbb{F}_\ell$, for a finite field $\mathbb{F}_\ell$ of size ℓ. Let $e_1, \ldots, e_\ell$ be the elements in $\mathbb{F}_\ell$, and $p_1, \ldots, p_n$ all univariate degree-α polynomials over $\mathbb{F}_\ell$. For each $i \in [n]$, define $S_i = \{(e_1, p_i(a_1)), \ldots, (e_\ell, p_i(e_\ell))\}$. The third condition follows from the fact that a non-zero univariate polynomial of degree α has at most α roots. □

Proof Theorem 5.1. For $\ell = n^{1/(\alpha+1)}$, let $f\colon \{0,1\}^\ell \to \{0,1\}$ be the hard function for degree-$(\alpha \cdot \Delta)$ PTF circuits from Theorem 5.2. By Theorem 5.2 and assuming E is a sufficiently large constant, we have that for any degree-$(\alpha \cdot \Delta)$ PTF circuit D on ℓ variables of size at most s,

$$\mathbf{Pr}_{z \sim \{0,1\}^\ell}[D(z) = f(z)] \leq \frac{1}{2} + \epsilon/n. \tag{16}$$

Let $S_1, \ldots, S_n$ be the sets from Claim 5.3. Define the generator $G_{\alpha,\Delta}\colon \{0,1\}^r \to \{0,1\}^n$ as follows:

$$G_{\alpha,\Delta}(y) = f(y|_{S_1}), \ldots, f(y|_{S_n}),$$

where, for $i \in [n]$, $y|_{S_i}$ denotes the substring of y indexed by the set S_i.

Toward a contradiction, suppose

$$|\mathbf{Pr}_{x \sim \{0,1\}^n}[C(x) = 1] - \mathbf{Pr}_{y \sim \{0,1\}^r}[C(G_{\alpha,\Delta}(y)) = 1]| > \epsilon. \tag{17}$$

By a standard argument via "reduction from distinguishing to predicting" as in [NW94], Equation (17) implies that there exist an $i \in [n]$, and bits $b_{i+1}, \ldots, b_n \in \{0,1\}$, such that

$$\mathbf{Pr}_{z \sim \{0,1\}^\ell}[C'(h_1(z), \ldots, h_i(z), b_{i+1}, \ldots, b_n) = f(z)] > 1/2 + \epsilon/n, \tag{18}$$

where

- $C' = C$ or $C' = \neg C$, and
- $h_1, \ldots, h_i$ are Boolean functions such that each depends on at most α bits of its input z.

First, note that each gate in C' is always a PTF of degree at most Δ. Next, observe that every Boolean function that depends on at most α variables can be computed by a multi-linear polynomial of degree at most α over the reals. Replacing our functions $h_1, \ldots, h_i$ with such degree α polynomials $p_1, \ldots, p_i$ inside C', we get

$$C'(p_1(z), \ldots, p_i(z), b_{i+1}, \ldots, b_n).$$

Now we can we merge the polynomials p_i's into every PTF gate in the circuit that reads from them. This yields a new circuit with *exactly the same* number of gates, and of degree at most $\alpha \cdot \Delta$. Denote this new circuit by C''. Note that C'' is a degree-$(\alpha \cdot \Delta)$ PTF circuit on ℓ variables of size at most s. By Equation (18), this PTF circuit C'' computes the function f with probability greater than $1/2 + \epsilon/n$, contradicting Equation (16). □

Next, we show how to obtain the PRG in Corollary 1.5 for PTFs from the result in Theorem 5.1.

Proof of Corollary 1.5. In order to make sure our PRG in Theorem 5.1 fools any degree-Δ PTF, we need to choose a value for α so that the size s in Theorem 5.1 is at least 1, where

$$s = n^{\frac{1}{\alpha+1}} / \left(E \cdot 5^{\alpha\cdot\Delta} \cdot \log^2(n) \cdot \log(n/\epsilon)\right).$$

Let's pick α so that the seed length of our PRG in Theorem 5.1 is

$$r = n^{2/(\alpha+1)} = 2^{L\cdot\sqrt{\Delta\cdot\log n}} \cdot \log^2(1/\epsilon), \tag{19}$$

where $L > 0$ is a some sufficiently large constant. Then the numerator of s is

$$n^{\frac{1}{\alpha+1}} = r^{1/2} = 2^{\frac{L}{2}\cdot\sqrt{\Delta\cdot\log n}} \cdot \log(1/\epsilon). \tag{20}$$

On the other hand, Equation (19) implies that

$$\alpha = \frac{2\log n}{L\cdot(\sqrt{\Delta\cdot\log n + 2\log\log(1/\epsilon)})} - 1 \leq \frac{2}{L}\cdot\sqrt{\log n/\Delta}.$$

Plugging the above into the denominator of s, we get

$$\left(E \cdot 5^{\alpha\cdot\Delta} \cdot \log^2(n) \cdot \log(n/\epsilon)\right) \leq E \cdot 2^{\frac{6}{L}\cdot\sqrt{\Delta\cdot\log n}} \cdot \log^2 n \cdot \log(n/\epsilon),$$

which is less than the numerator of s given in Equation (20). □

6 Open problems

An important open problem is to get a nontrivial Circuit-SAT algorithm for circuits with degree-2 PTF gates. Our algorithm only works for the case where the PTF gates have a sub-quadratic number of monomials, so it does not work for the degree-2 case in general. Such an algorithm is not known even for a single degree-2 PTF. Another interesting open problem is to derandomize our zero-error randomized algorithms to get deterministic #Circuit-SAT algorithms of similar time complexity.

Can we get any nontrivial standard derandomization for constant-depth PTF (LTF) circuits of small wire complexity? For PRGs, can we get a nontrivial PRG for depth-2 LTF circuits with a super-linear number of gates?

Acknowledgements. We thank Suguru Tamaki for suggesting to us to consider sparse PTFs in the case of satisfiability, and for clarifying the MAX-k-SAT algorithm in [SSTT16] for us.

References

[ACW16] Josh Alman, Timothy M. Chan, and R. Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *FOCS*, pages 467–476, 2016. 8

[Ant01] Martin Anthony. *Discrete Mathematics of Neural Networks: Selected Topics.* SIAM monographs on discrete mathematics and applications. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2001. 10

[BT94] Richard Beigel and Jun Tarui. On ACC. *Computational Complexity*, 4:350–366, 1994. 13

[CP16] Shiteng Chen and Periklis A. Papakonstantinou. Depth-reduction for composites. In *FOCS*, pages 99–108, 2016. 2

[CS15] Ruiwen Chen and Rahul Santhanam. Improved algorithms for sparse MAX-SAT and MAX-k-CSP. In *SAT*, pages 33–45, 2015. 6, 8, 14, 22

[CSS16] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-case lower bounds and satisfiability algorithms for small threshold circuits. In *CCC*, pages 1:1–1:35, 2016. 5, 6, 8, 14, 17, 18, 25, 26

[CW16] Timothy M. Chan and Ryan Williams. Deterministic APSP, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. In *SODA*, pages 1246–1255, 2016. 13, 14

[GHR92] Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992. 9

[GW14] Oded Goldreich and Avi Wigderson. On derandomizing algorithms that err extremely rarely. In *STOC*, pages 109–118, 2014. 3, 6, 8, 9

[Hås89] Johan Håstad. Almost optimal lower bounds for small depth circuits. In S. Micali, editor, *Randomness and Computation*, pages 143–170, Greenwich, Connecticut, 1989. Advances in Computing Research, vol. 5, JAI Press. 9

[IPS13] Russell Impagliazzo, Ramamohan Paturi, and Stefan Schneider. A satisfiability algorithm for sparse depth two threshold circuits. In *FOCS*, pages 479–488, 2013. 8

[Kan12] Daniel M. Kane. A structure theorem for poorly anticoncentrated gaussian chaoses and applications to the study of polynomial threshold functions. In *FOCS*, pages 91–100, 2012. 9

[KKL17] Valentine Kabanets, Daniel M. Kane, and Zhenjian Lu. A polynomial restriction lemma with applications. In *STOC*, pages 615–628, 2017. 5, 6, 7, 9, 11, 12, 24

[KR18] Daniel Kane and Sankeerth Rao. A PRG for Boolean PTF of degree 2 with seed length subpolynomial in ϵ and logarithmic in n. In *CCC*, 2018. 9

[LS11] Shachar Lovett and Srikanth Srinivasan. Correlation bounds for poly-size AC^0 circuits with $n^{1-o(1)}$ symmetric gates. In *APPROX/RANDOM*, pages 640–651, 2011. 9

[MP43] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. 10

[MTT61] Saburo Muroga, Iwao Toda, and Satoru Takasu. Theory of majority decision elements. *Journal of the Franklin Institute*, 271:376–418, 1961. 4, 10

[MZ13] Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. *SIAM J. Comput.*, 42(3):1275–1301, 2013. 9

[Nis94] Noam Nisan. The communication complexity of threshold gates. In *Proceedings of Combinatorics, Paul Erdős is Eighty*, pages 301–315, 1994. 8, 9, 28

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. 7, 29

[Raz87] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *MATHNASUSSR: Mathematical Notes of the Academy of Sciences of the USSR*, 41, 1987. 13

[Sri13] Srikanth Srinivasan. On improved degree lower bounds for polynomial approximation. In *FSTTCS*, pages 201–212, 2013. 13

[SSTT16] Takayuki Sakai, Kazuhisa Seto, Suguru Tamaki, and Junichi Teruyama. Bounded depth circuits with weighted symmetric gates: Satisfiability, lower bounds and compression. In *MFCS*, pages 82:1–82:16, 2016. 4, 30

[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)*, 23:100, 2016. 8, 13, 14

[Tel17a] Roei Tell. Improved bounds for quantified derandomization of constant-depth circuits and polynomials. In *CCC*, pages 13:1–13:48, 2017. 3, 9

[Tel17b] Roei Tell. A note on the limitations of two black-box techniques in quantified derandomization. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:187, 2017. 9

[Tel18] Roei Tell. Quantified derandomization of linear threshold circuits. In *STOC*, 2018. 3, 6, 7, 9, 25, 26

[Tod91] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. 13

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012. 10

[Wil10] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. In *STOC*, pages 231–240, 2010. 2

[Wil11] Ryan Williams. Non-uniform ACC circuit lower bounds. In *CCC*, pages 115–125, 2011. 2, 13, 14

[Wil14] Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *STOC*, pages 194–202, 2014. 6, 14

[Yat37] Frank Yates. *The design and analysis of factorial experiments*. Technical Communication no. 35 of the Commonwealth Bureau of Soils. 1937. 13

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il