Algorithms and Lower Bounds in Circuit Complexity

by

Zhenjian Lu

B.Sc., University of British Columbia, 2013

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2020

Declaration of Committee

Name:	**Zhenjian Lu**
Degree:	**Doctor of Philosophy**
Thesis title:	**Algorithms and Lower Bounds in Circuit Complexity**
Committee:	**Chair:** Binay Bhattacharya Professor, Computing Science
	Andrei Bulatov Co-Supervisor Professor, Computing Science
	Valentine Kabanets Co-Supervisor Professor, Computing Science
	Igor Shinkar Examiner Assistant Professor, Computing Science
	Eric Allender External Examiner Professor, Computer Science Rutgers University

Abstract

Computational complexity theory aims to understand what problems can be efficiently solved by computation. This thesis studies computational complexity in the model of Boolean circuits. Boolean circuits provide a basic mathematical model for computation and play a central role in complexity theory, with important applications in separations of complexity classes, algorithm design, and pseudorandom constructions. In this thesis, we investigate various types of circuit models such as threshold circuits, Boolean formulas, and their extensions, focusing on obtaining complexity-theoretic lower bounds and algorithmic upper bounds for these circuits.

- **Algorithms and lower bounds for generalized threshold circuits.** We extend the study of *linear threshold circuits*, circuits with gates computing linear threshold functions, to the more powerful model of *polynomial threshold circuits* where the gates can compute *polynomial threshold functions*. We obtain hardness and meta-algorithmic results for this circuit model, including strong average-case lower bounds, satisfiability algorithms, and derandomization algorithms for constant-depth polynomial threshold circuits with super-linear wire complexity.

- **Algorithms and lower bounds for enhanced formulas.** We investigate the model of Boolean formulas whose leaf gates can compute complex functions. In particular, we study De Morgan formulas whose leaf gates are functions with "low communication complexity". Such gates can capture a broad class of functions including symmetric functions and polynomial threshold functions. We obtain new and improved results in terms of lower bounds and meta-algorithms (satisfiability, derandomization, and learning) for such enhanced formulas.

- **Circuit lower bounds for MCSP.** We study circuit lower bounds for the Minimum Circuit Size Problem (MCSP), the fundamental problem of deciding whether a given function (in the form of a truth table) can be computed by small circuits. We get new and improved lower bounds for MCSP that nearly match the best-known lower bounds against several well-studied circuit models such as Boolean formulas and constant-depth circuits.

Keywords: circuit lower bounds; satisfiability; pseudorandomness; learning; minimum circuit size problem (MCSP)

Acknowledgements

I would like to thank my supervisor, Valentine Kabanets. It has been an invaluable experience for me to work with and learn from him for the past six years. I would like to thank my co-supervisor, Andrei Bulatov, for his support and understanding throughout my graduate studies. Finally, I would like to thank all my collaborators and the theory group at SFU.

Table of Contents

Chapter 1

Introduction

1.1 Computational complexity

Computational complexity theory aims to understand the amount of computational resources (time, memory, randomness, etc.) needed to solve natural problems. This leads to the study of *complexity classes* that classify problems under the constraints of different resources:

- The class P contains problems that can be solved by a computer whose running time is polynomial-bounded in the input size of the problem.
- By allowing computers to use randomness (the ability to toss random coins), we get *probabilistic* computers, and the corresponding class of polynomial-time solvable problems is called BPP.
- The class NP are problems that can be solved in polynomial-time by a computer with non-determinism (the ability to "guess").
- By allowing computers to utilize quantum bits (qubits), we get *quantum* computers, and the class of polynomial-time solvable problems in such a model is called BQP.

While each of the above complexity classes is interesting by itself as it characterizes many important problems in computing science, the relationships between these complexity classes are unclear, and this forms the most fundamental questions in complexity theory:

- What is the role of non-determinism in computation (P vs. NP)?
- How useful is randomness (P vs. BPP)?
- What is the power of quantum computers in comparison with "classical" computers (BQP vs. $\mathsf{P}, \mathsf{BPP}, \mathsf{NP}$)?

1.2 Circuit complexity

Answering the above questions requires to prove *lower bounds* against *Turing machines*, the mathematical abstraction of modern computers. That is, we need to find a problem (e.g., some problem in NP) that cannot be solved by any Turing machine that runs in less than a certain amount of time (e.g., polynomial time). However, the "unstructured" characteristics of Turing machines (or computer programs) makes them notoriously difficult to analyze in a "non-black-box" manner, and it is known that "black-box" techniques by themselves cannot resolve questions such as P vs. NP (This is known as the *Relativization Barrier* [BGS75]).

Boolean circuits is a computational model that is considered to be mathematically more "structured" than Turing machines and has been intensively studied in complexity theory. A Boolean circuit consists of a collection of *gates*, such as AND, OR, and NOT, which take as input either two bits (for AND or OR) or one bit (for NOT) and output one bit, where bits are values in $\{0, 1\}$. These gates are connected by *wires*, each of which transmits the value from the output of one gate to the input of another gate. A Boolean circuit naturally computes some *Boolean function*; a Boolean function takes a binary input string and outputs one bit. For a Boolean function (representing some computational problem), we are interested in the minimum *size* (measured by either the number of gates or wire) and *depth* (length of the longest path from the output gate to any input variable) of a circuit computing the function.

$\mathsf{P} \neq \mathsf{NP}$ from circuit lower bounds. It can be shown that every problem in P can also be computed by polynomial-size circuits. One way to show $\mathsf{P} \neq \mathsf{NP}$ is to prove a *lower bound for* NP *against polynomial-size circuits*. That is, find a problem in NP that cannot be solved by any polynomial-size circuit.

Derandomization from circuit lower bounds. Proving circuit lower bounds also has consequences for the P vs. BPP question. A beautiful line of research in the area of derandomization shows that proving a certain strong (but plausible) circuit lower bound would imply $\mathsf{P} = \mathsf{BPP}$ [NW94, IW97].

Quantum computing and circuit complexity. For the question of quantum versus classical computers, it is natural to address this question by finding a problem that can be efficiently solved by a quantum computer but lacks efficient circuits (see, e.g., [BGH07, WKST19]). Another connection between quantum computing and circuits complexity can be found in the recent breakthrough result of [RT19], which showed a separation of BQP and PH (Polynomial Hierarchy, a generalization of the class NP) in the "black-box model". The proof of this result crucially made use of analytical results from classical circuit complexity.

1.3 Meta-computational problems

An important line of research in circuit complexity is the study of *meta-computational problems*. Informally, these are problems whose input or output are computational devices (e.g., circuits). Apart from being important algorithmic problems in their own right, meta-computational problems are intimately related to the task of proving circuit lower bounds. For example, "non-trivial" algorithms for such problems often imply circuit lower bounds. Therefore, studying the algorithmic aspects of these problems is beneficial because this allows us to use our knowledge of algorithm design, which is much more extensive than our knowledge of proving lower bounds, to obtain circuit lower bounds. We briefly describe below some well-studied meta-computational problems.

Satisfiability. The (circuit) satisfiability problem (or 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 exhaustive search. More specifically, is there an algorithm that, given a circuit of polynomial-size on n variables, runs in time at most $2^n/n^{\omega(1)}$ and determines whether it has a satisfying assignment?

It turns out that this task is challenging even for very restricted classes of circuits. The difficulty of obtaining such a satisfiability algorithm can be partially explained by the work of Williams [Wil13, Wil14b] 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).

Derandomization. A fundamental problem in complexity theory is to give an efficient deterministic algorithm for computing the majority output 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 and 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 $\mathcal{C}$ [KI04, Wil13].

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 $G\colon \{0,1\}^r \to \{0,1\}^n$ *ε-fools* a class $\mathcal{C}$ of Boolean circuits if for every $C\colon \{0,1\}^n \to \{0,1\}$ from $\mathcal{C}$

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

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 computing the outputs of PRG for all possibles seeds and evaluating C on these outputs. This gives a deterministic algorithm solving the derandomization problem that runs 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 $\mathcal{C}$.

Learning. Roughly speaking, the learning problem is to, given a set of random samples or queries of a target function, output a device that (exactly or approximately) computes the function. The learning problem has received intensive attention in both applied and theoretical research. It is also known that learning algorithms for circuits would imply some kind of circuit lower bounds [FK09, KKO13].

Minimum Circuit Size Problem. The minimum circuit size problem (MCSP) asks if a given truth table[1] represents a function that can be computed by some small-size circuit. MCSP is a fundamental problem in complexity theory. Despite sustained efforts devoted to this problem, its exact complexity remains a mystery. In particular, while it is known that MCSP is in NP, whether MCSP is NP-complete is a widely open question. MCSP also plays an important role in a recent line of research called *hardness magnification*, which states that a weak circuit lower bound for (some variants of) MCSP implies breakthrough results in circuit complexity.

1.4 The frontiers

Proving circuit lower bounds is known to be a difficult task, and this fact was mathematically formulated as the *Natural Proofs Barrier* in circuit complexity [RR97]. Informally, The Natural Proofs Barrier states that if cryptographically secure pseudorandom functions exist (which is a central assumption in cryptography), then for any circuit class that is powerful enough to compute a pseudorandom function, proving lower bounds against that circuit class requires a technique that does not follow some certain "natural" pattern, which is

[1] A truth table is a bit-string that stores the output values of a function for all possible inputs.

presented in almost every currently known method for proving lower bounds. Currently, no super-linear lower bound is known for general circuits.

Threshold circuits. As the general class of polynomial-size Boolean circuits (a.k.a, $\mathsf{P/poly}$) seems well beyond the currently known methods for proving lower bounds, the focus of circuit complexity research has been on various restricted circuit classes. Particularly successful has been the study of constant-depth circuits (which can be thought of as very efficient parallel nonuniform algorithms).

Different natural sets of gates were considered. For the gates AND, OR, and NOT (where AND and OR have unbounded fan-in), the resulting circuit class is AC^0. A milestone in circuit complexity was the proof that the parity function on n bits requires exponential-size AC^0 circuits (with a matching upper bound also known) [Ajt83, FSS84, Yao85, Hås89]. Adding the parity gate to AC^0 circuits, we get the class $\mathsf{AC}^0[2]$ with modulo 2 gates. An exponential lower bound against $\mathsf{AC}^0[2]$ for the n-bit majority function was shown by Razborov [Raz87], and was extended by Smolensky [Smo87] to exponential lower bounds against $\mathsf{AC}^0[p]$, for an arbitrary prime modulus $p > 0$. It is still open (though widely believed) whether the majority function requires exponential size also for the class $\mathsf{AC}^0[m]$ for any composite modulus $m > 1$; significant progress has been recently made by Williams [Wil14b] who showed that a Boolean function computable in nondeterministic exponential time (NEXP) requires superpolynomial-size $\mathsf{AC}^0[m]$ circuits, for any integer modulus $m > 1$.

Adding the majority gate to AC^0 circuits, we get the class TC^0, for which no superpolynomial circuit lower bounds are known for any explicit function (not even for a function in NEXP), despite serious efforts by complexity researchers over the past thirty years. One reason for our inability to prove strong lower bounds against TC^0 stems from the fact that TC^0 is a powerful circuit class, capable of computing many interesting and useful functions: addition, multiplication, division, and sorting (see [Raz92] and the references therein). Surprisingly, every function computable by a polynomial-size $\mathsf{AC}^0[m]$ circuit, for any integer $m > 1$, has an equivalent depth-3 TC^0 circuit of quasipolynomial size [All89, Yao90]. Moreover, TC^0 is conjectured to be capable of computing cryptographically secure pseudorandom function generators [NR04], which, coupled with arguments in [RR97], means that it is highly unlikely that a "natural" lower bound proof method would work against TC^0, as any such "natural" proof would yield an efficient algorithm to break every candidate pseudorandom function generator in TC^0.

As it seems very difficult to prove superpolynomial lower bounds against TC^0, the focus has shifted to proving fixed-polynomial lower bounds (even for a fixed constant depth, say depth 2 or 3). Before discussing these results, let us mention another motivation for studying TC^0. Closely related to the majority function is a Linear Threshold Function (LTF), defined as the sign of a linear (degree 1) polynomial in variables $x_1, \ldots, x_n$; when the variables x_i assume Boolean values, the resulting LTF is a Boolean function. Note that an LTF may have

arbitrarily large coefficients (weights) for the underlying linear polynomial, which makes an LTF provably more powerful than a majority function, or more generally, than an LTF with small (polynomially bounded) weights [MK61]. On the other hand, somewhat surprisingly, an arbitrary LTF can be represented by a polynomial-size depth-2 TC^0 circuit [GHR92].

Linear Threshold Functions (LTFs) and circuits with LTF gates have been studied since at least the 1940s in the context of artificial neural networks [MP43] (see [Ant01] and the references therein). Some of the early lower bounds for LTFs are due to Minsky and Papert [MP69], who showed, for example, that the parity function cannot be computed by any LTF.

For constant-depth LTF circuits, two complexity measures have been considered: the number of gates (excluding the input variables), and the number of wires. The first super-linear wire complexity bound was obtained by Impagliazzo et al. [IPS97], who showed that the n-bit parity function requires depth-d LTF circuits with at least $n^{1+\varepsilon_d}$ wires, where $\varepsilon_d = \exp(-d)$. They also showed that the n-bit parity function requires depth-d LTF circuits with at least $(n/2)^{1/(2(d-1))}$ gates. Recently, these bounds were generalized to average-case (correlation) bounds by Chen et al. [CSS18]. For depth-2 LTF circuits, Kane and Williams [KW16] have recently proved an $n^{3/2}/\mathsf{poly}(\log n)$ gate complexity bound, and $n^{5/2}/\mathsf{poly}(\log n)$ wire complexity bound for an explicit function in P (Andreev's function [And87]). More recently, Alman, Chan and Williams [ACW16] and Tamaki [Tam16] both gave satisfiability algorithms for depth-2 LTF circuits with an almost quadratic number of gates, and obtained lower bounds against these circuits for an explicit function in E^{NP}, using the connection between satisfiability algorithms and circuit lower bounds.

De Morgan formulas Another type of restricted circuits that has been intensively studied is *De Morgan formulas*. A (De Morgan) formula is a binary tree whose internal nodes are labelled by AND or OR gates, and whose leaves are marked with a variable or its negation. The size of a formula is defined to be the number of leaves in the tree. One motivation of studying formulas is that the class of polynomial-size formulas is equivalent to the class of (polynomial-size) circuits with logarithmic depth, which captures the notion of efficient parallel computation. A major open problem in complexity theory is to find a problem in P that cannot be computed by polynomial-size formulas. Such a lower bound against formulas would show that there are problems, despite being efficiently solvable, are inherently serial and have no efficient parallel algorithms. Also, it is worth noting that any function that can be computed by a constant-depth polynomial-size threshold circuit can also be computed by some polynomial-size formula, so the Natural Proof Barrier also applies to polynomial-size formulas.

Despite sustained efforts over the last three decades, the best-known lower bound against De Morgan formulas is only sub-cubic. Namely, there exists a function in P that requires formulas of size at least $n^{3-o(1)}$ [Hås98, Tal14, Tal17a, DM18]. The techniques underlying

these lower bound results have also enabled algorithmic developments. These include a non-trivial satisfiability algorithms for sub-cubic size formulas [Tal15], learning algorithms for size-s formulas in time $n^{O(\sqrt{s})}$ [Rei11b], and a pseudorandom generator with seed length $s^{1/3+o(1)}$ for size-s formulas [IMZ19].

1.5 Contributions and organization of this thesis

This thesis focuses on getting a better understanding of Boolean circuits, by studying complexity lower bounds and meta-algorithms for restricted classes of circuits such as constant-depth threshold circuits, formulas and extensions. After giving some background in Chapter 2, we present our results in the following structure.

1.5.1 Algorithms and lower bounds for polynomial threshold circuits

In Chapter 3 and Chapter 4, we investigate the class of circuits whose gates compute *polynomial threshold functions* (PTFs). A PTF is a generalization of LTF to the case of arbitrary (not necessarily linear) polynomials. We extend the aforementioned results for constant-depth LTF circuits (i.e., lower bounds, satisfiability, and quantified derandomization) to the more powerful model of PTF circuits.

- We prove lower bounds against constant-depth circuits with PTF gates of any degree $1 \leq d \ll \sqrt{\log n / \log\log n}$, generalizing the recent bounds against constant-depth LTF circuits proved by Kane and Williams [KW16] and Chen, Santhanam, and Srinivasan [CSS18]. In particular, we show that there is an n-variate Boolean function $F_n \in \mathsf{P}$ such that every depth-2 circuit with PTF gates of degree $d \geq 1$ that computes F_n must have at least $\left(n^{\frac{3}{2}+\frac{1}{d}}\right) \Big/ (\log n)^{O(d^2)}$ wires. For constant depths greater than 2, we also show average-case lower bounds against such circuits with a super-linear number of wires. These are the first super-linear bounds on the number of wires for circuits with PTF gates.

- 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 there exists a constant $\varepsilon > 0$ such that, given a constant-depth circuit with $(n^{1.99})$-sparse PTF gates that has at most $n^{1+\varepsilon}$ wires, the number of satisfying assignments of the circuit can be computed in randomized time $2^{n-n^{\varepsilon}}$ with zero error. This generalizes the result by

Chen, Santhanam and Srinivasan [CSS18] who gave a SAT algorithm for constant-depth circuits of super-linear wire complexity with LTF gates only.[2]

- The quantified derandomization problem, introduced by Goldreich and Wigderson [GW14], asks to compute the majority output 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 some constant $\varepsilon > 0$, there is an algorithm that, given a constant-depth degree-d PTF circuit C with $n^{1+\varepsilon}$ wires such that C has at most $2^{n^{0.99}}$ minority-value inputs, runs in quasi-polynomial time $\exp\left((\log n)^{O(d^2)}\right)$ and determines the majority value of C. This extends the recent result of Tell [Tel18] for constant-depth LTF circuits of super-linear wire complexity.

- We show a nontrivial pseudorandom generator for PTF circuits (of unrestricted depth) with sub-linearly many gates. As a corollary, we get a PRG for degree-d PTFs with the seed length $\exp\left(\sqrt{d \cdot \log n}\right) \cdot \log^2(1/\varepsilon)$; this gives the first PRG for PTFs with a seed length that is sub-polynomial in both n and $1/\varepsilon$.

The key ingredient in our results is a new structural lemma for low-degree polynomials, which is called *Random Restrictions Lemma*. Using this structural lemma, we also obtain a "derandomized" version of the *Littlewood-Offord theorem*, a type of classical result in additive combinatorics, for low-degree polynomials.

The results in Chapter 3 are from the joint work with Kabanets and Kane [KKL17], and the results in Chapter 4 are from the subsequent work with Kabanets [KL18].

1.5.2 Algorithms and lower bounds for enhanced formulas

While showing lower bounds for large formulas remains a major open problem in circuit complexity, recent research (e.g., [Tal17a, OPS19]) shows that understanding smaller formulas whose leaves are replaced by certain functions would also be very useful.

In Chapter 5, we study the class $\mathsf{FORMULA}[s] \circ \mathcal{G}$ consisting of size-s De Morgan formulas whose leaves are any Boolean functions from a class $\mathcal{G}$. We give *lower bounds* and (SAT, Learning, and PRG) *algorithms* for $\mathsf{FORMULA}[n^{1.99}] \circ \mathcal{G}$, for classes $\mathcal{G}$ of functions with *low communication complexity* (see section 5.2 for definitions). Let $R^{(k)}(\mathcal{G})$ be the maximum k-party number-on-forehead randomized communication complexity of a function in $\mathcal{G}$. Among other results, we show that:

[2]Our satisfiability algorithm for circuits with *sub-quadratically sparse* PTF gates was recently extended by Bajpai el at. [BKK$^+$19] to the case of *polynomially sparse* PTF gates.

- There exists a function in P (the generalized inner product function) that cannot be computed in $\mathsf{FORMULA}[s] \circ \mathcal{G}$ on more than $1/2 + \varepsilon$ fraction of inputs for

$$s = o\left(\frac{n^2}{2^{O(k)} \cdot \left(R^{(k)}(\mathcal{G}) \cdot \log(n/\varepsilon) \cdot \log(1/\varepsilon)\right)^2}\right).$$

  This significantly extends the lower bounds against bipartite formulas obtained by [Tal17b]. As a corollary, we get an (average-case) lower bound against sub-quadratic-size De Morgan formulas whose leaf gates are constant-degree PTFs.

- There is a PRG of seed length $n/2 + O\left(\sqrt{s} \cdot \log(s) \cdot R^{(2)}(\mathcal{G})\right)$ that fools $\mathsf{FORMULA}[s] \circ \mathcal{G}$. For the special case of $\mathsf{FORMULA}[s] \circ \mathsf{LTF}$, we get the better seed length $O(n^{1/2} \cdot s^{1/4} \cdot \log(n) \cdot \log(n/\varepsilon))$, where ε is the error of the PRG. In particular, this provides the first non-trivial PRG (with seed length $o(n)$) for intersections of n half-spaces in the regime where $\varepsilon \leq 1/n$, complementing a recent result of [OST19].

- There exists a randomized (2^{n-t})-time satisfiability counting algorithm for $\mathsf{FORMULA}[s] \circ \mathcal{G}$, where

$$t = \left(\frac{n}{\sqrt{s} \cdot \mathsf{polylog}(s) \cdot R^{(2)}(\mathcal{G})}\right)^{1/2}.$$

  In particular, this implies a nontrivial satisfiability algorithm for sub-quadratic-size formulas whose leaf gates are LTFs.

- The Minimum Circuit Size Problem is not in $\mathsf{FORMULA}[n^{1.99}] \circ \mathsf{XOR}$; thereby making progress on hardness magnification, in connection with results from [OPS19, CJW19]. On the algorithmic side, we show that the concept class $\mathsf{FORMULA}[n^{1.99}] \circ \mathsf{XOR}$ can be PAC-learned in time $2^{O(n/\log n)}$.

This chapter is a joint work with Kabanets, Koroth, Myrisiotis, and Oliveira [KKL+20].

1.5.3 Circuit lower bounds for MCSP

In Chapter 6, we improve several circuit lower bounds for MCSP, using pseudorandom generators (PRGs) that are local; a PRG is called *local* if its output bit strings, when viewed as the truth table of a Boolean function, can be computed by a Boolean circuit of small size. We get new and improved lower bounds for MCSP that almost match the best-known lower bounds against several circuit models. Specifically, we show that computing MCSP, on functions with a truth table of length N, requires

- $N^{3-o(1)}$-size De Morgan formulas, improving the recent $N^{2-o(1)}$ lower bound by Hirahara and Santhanam [HS17],

- $N^{2-o(1)}$-size formulas over an arbitrary basis or general branching programs (no non-trivial lower bound was known for MCSP against these models), and
- $2^{\Omega\left(N^{1/(d+1.01)}\right)}$-size depth-$d$ AC^0 circuits, improving the (implicit, in their work) exponential size lower bound by Allender et al. [ABK$^+$06].

The AC^0 lower bound stated above matches the best-known AC^0 lower bound (for the parity function) up to a small *additive* constant in the depth. Also, for the special case of depth-2 circuits (i.e., CNFs or DNFs), we get an optimal lower bound of $2^{\Omega(N)}$ for MCSP.

The results in Chapter 6 were presented in the joint work with Cheraghchi, Kabanets, and Myrisiotis [CKLM20].

Chapter 2

Preliminaries

In this chapter, we introduce some terminologies, as well as some basic definitions and results that will be useful for understanding this thesis.

2.1 Boolean functions and polynomials

A n-variate *Boolean function* maps n bits to one bit. A common way to represent a bit is to use 0 and 1. For this Boolean domain, i.e. $\{0,1\}$, we think of a Boolean function as $f\colon \{0,1\}^n \to \{0,1\}$. Another useful way to represent a Boolean domain is to use $\{-1,1\}$. To convert from $\{0,1\}$ to $\{-1,1\}$, we map 0 to 1 and 1 to -1. That is, for a bit $x \in \{0,1\}$, its corresponding bit y in $\{-1,1\}$ is $y = 1-2x$. Note that we can also convert from $\{-1,1\}$ to $\{0,1\}$ using $x = (1-2y)/2$.

In this thesis, we may use different Boolean domains in showing different results for the simplicity of presentation. This will be specified in corresponding sections. For the present chapter, we will use $\{0,1\}$ as the Boolean domain. All the definitions and results stated can be easily adapted to the $\{-1,1\}$ domain in a natural way.

We will also consider *polynomials* (over the reals) that take inputs from the Boolean cube $\{0,1\}^n$ (or $\{-1,1\}^n$) and output real numbers. In this case, we treat such a polynomial as a *multilinear* polynomial $p\colon \{0,1\}^n \to \mathbb{R}$ viewed as the sum of (multilinear) monomials:

$$p(x_1, x_2, \ldots, x_n) = \sum_{S \subseteq [n]} \hat{p}(S) \cdot \prod_{i \in S} x_i,$$

where $\hat{p}(S) \in \mathbb{R}$ for every $S \subseteq [n]$. The *degree* of p is the size of the largest set $S \subseteq [n]$ such that $\hat{p}(S) \neq 0$.

2.2 Circuit classes

2.2.1 Boolean circuits

Definition 2.1 (Boolean circuits)**.** *A Boolean circuit on n variables is a directed acyclic graph such that*

- *there are n input node with in-degree 0;*
- *there is one output node with out-degree 1;*
- *all other nodes (henceforth,* gates*) are labeled by Boolean operations $\{\text{AND}, \text{OR}, \text{NOT}\}$; The* AND *and* OR *gates have in-degree (henceforth,* fan-in*) 2 and the* NOT *gates have fan-in 1.*

The size *of the circuit is the number of gates and the* depth *is the length of the longest path from the output node to any input node.*

We often think of the gates of a circuit as layered, where the gates at a layer take inputs only from gates at previous layers, so the bottom layer consists of gates depending on only the inputs and the top layer consists of the output gate.

An n-variate Boolean circuit naturally computes some Boolean function $f\colon \{0,1\}^n \to \{0,1\}$. To solve a *decision problem* (or decide a *language*) which is defined by a function $F\colon \{0,1\}^* \to \{0,1\}$ on every input length, we can use a *circuit family*. A circuit family is a collection of circuits $\{C_n\}_{n\in\mathbb{N}}$ where each C_n is an n-variate Boolean circuit. We say that the circuit family $\{C_n\}_{n\in\mathbb{N}}$ computes F if for every fixed input length n, $C_n(x) = F(x)$ for every $x \in \{0,1\}^n$. Note that for a circuit family, we can have different circuits for different input lengths. Therefore, circuits are thought as a *non-uniform* computation model, as opposed to the Turing machine model, which is uniform because we have a single machine for all input lengths.

It is easy to see that every Boolean function on n bits can be computed by some circuit of size at most 2^n. In fact, it can be shown that every n-bit function has a circuit of size $O(2^n/n)$. On the other hand, it was also observed that almost all functions are hard in the sense that they require circuits of size $\Omega(2^n/n)$ [Sha49].

We are interested in the power and limitations of small size circuits. In particular, we define $\mathsf{P/poly}$ to be the class of circuit families that are of polynomial-size, and by abusing notation also the class of decision problems that can be solved by such circuit families.

It is known that every polynomial-time Turing machine can be simulated by polynomial-size circuit families (i.e., $\mathsf{P} \subseteq \mathsf{P/poly}$). An interesting question is whether problems from other complexity classes such as NP can also be computed by such circuit families. A negative answer to this question would show that $\mathsf{P} \neq \mathsf{NP}$, which is a central problem in complexity theory. It is widely believed that $\mathsf{NP} \not\subseteq \mathsf{P/poly}$. The task of showing that some explicit function (e.g., a problem in NP) cannot be computed by circuit families of certain

size is called *proving circuit lower bounds*, which is the main goal of the circuit complexity research.

For the rest of the thesis, when speaking of circuit classes, we will simply say "a class of circuits" instead of "a class of circuit families".

2.2.2 Small-depth circuits

As discussed in Section 1.4, since general circuits are hard to analyze, the focus of current research is on restricted types of circuits. This leads to the study of circuits whose depths are restricted to be small:

- AC^0, the class of constant-depth circuits, where the AND and OR gates have *unbounded* fan-in.
- NC^k, the class of $O\left(\log^k n\right)$-depth circuits, where the AND and OR gates have *bounded* fan-in.

Another motivation for studying small-depth circuits is that they capture the notion of efficient parallel computation, as we can evaluate the outputs of all the gates at the same layer simultaneously given the outputs of the gates at the previous layer, so the depth corresponds to the running time of such a parallel algorithm.

Particular success has been found in proving lower bound against AC^0 and extensions such as $\mathsf{AC}^0[p]$ (obtained by adding MOD_p gates, where $p > 1$ is a prime) and $\mathsf{AC}^0[m]$ (by adding MOD_m gates, where $m > 1$ is a constant). The frontier in this line of research is to prove lower bounds against TC^0, the class of constant-depth circuits with (unbounded fan-in) MAJORITY gates (see section 1.4).

Another major open question in circuit complexity is to prove circuit lower bounds against NC^1. This task is tightly connected to the study of small formulas, which we discuss below.

2.2.3 Boolean formulas

Formulas are a special type of circuits that can be represented as a tree.

Definition 2.2 (Formulas)**.** *An n-variate* formula over a basis $\mathcal{B}$ *is a directed rooted tree; its non-leaf vertices (henceforth,* internal gates*) take labels from $\mathcal{B}$ and its leaves (henceforth,* variable gates*) take labels from the set of variables $\{x_1, \ldots, x_n\}$. Each internal gate has fan-in* 2*. The* size *of a De Morgan formula is the number of its leaf gates.*

A De Morgan formula *is a formula over the basis* $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\}$.

It is easy to see that a formula is a circuit whose non-input gates have fan-out 1. In fact, it was observed that the class of polynomial-size formulas is exactly the class of logarithmic-depth circuits (note that logarithmic-depth circuits can have only a polynomial number of gates).

Theorem 2.3 ([Spi71])**.** *A function $f\colon \{0,1\}^n \to \{0,1\}$ can be computed by a polynomial size formula if and only if it can be computed by a circuit of depth $O(\log n)$.*

As discussed in the previous chapter, for De Morgan formulas, the size for which we know how to analyze is only sub-cubic after decades of effort (see section 1.4). For formulas over arbitrary basis, this size is sub-quadratic.

2.3 Pseudorandomness

Pseudorandomness studies how to construct objects that "look random" using little or no randomness.

Pseudorandom distributions and generators

For a distribution $\mathcal{D}$ (over some sample space), we write "$x \sim \mathcal{D}$" to mean that x is *sampled* according to $\mathcal{D}$. For the Boolean cube $\{0,1\}^n$ (resp. $\{-1,1\}^n$), we will often write "$x \sim \{0,1\}^n$" to mean x is sampled uniformly at random from $\{0,1\}^n$.

For a class of Boolean functions $\mathcal{F}$, a pseudorandom distribution $\mathcal{D}$ against $\mathcal{F}$ is a distribution such that every function f in $\mathcal{F}$ "behaves" similarly under $\mathcal{D}$ and the uniform distribution, in the sense that the probability that f outputs 1 on $\mathcal{D}$ is about the same as that of the uniform distribution.

Definition 2.4 (Pseudorandom distributions)**.** *Let $\mathcal{F}$ be a class of Boolean functions, and $0 < \varepsilon < 1$. Let $\mathcal{D}$ be a distribution over $\{0,1\}^n$. We say that $\mathcal{D}$ ε-fools $\mathcal{F}$ if, for every function $f \in \mathcal{F}$, it is the case that*

$$\left| \mathop{\mathbf{Pr}}_{z \sim \mathcal{D}}[f(z) = 1] - \mathop{\mathbf{Pr}}_{x \sim \{0,1\}^n}[f(x) = 1] \right| \leq \varepsilon.$$

Note that for the above definition, we can also write

$$\left| \mathop{\mathbf{E}}_{z \sim \mathcal{D}}[f(z)] - \mathop{\mathbf{E}}_{x \sim \{0,1\}^n}[f(x)] \right| \leq \varepsilon.$$

A pseudorandom generator (PRG) against a class of functions $\mathcal{F}$ is a deterministic procedure G mapping short Boolean strings (seeds) to longer Boolean strings, so that G's output "looks random" to every function in $\mathcal{F}$. In other words, a PRG is a sampling procedure for some pseudorandom distribution.

Definition 2.5 (Pseudorandom generators)**.** *Let $G\colon \{0,1\}^\ell \to \{0,1\}^n$ be a function, $\mathcal{F}$ be a class of Boolean functions, and $0 < \varepsilon < 1$. We say that G is a* pseudorandom generator of seed length ℓ that ε-fools $\mathcal{F}$ *if, for every function $f \in \mathcal{F}$, it is the case that*

$$\left| \mathop{\mathbf{E}}_{z \sim \{0,1\}^\ell}[f(G(z))] - \mathop{\mathbf{E}}_{x \sim \{0,1\}^n}[f(x)] \right| \leq \varepsilon.$$

A pseudorandom generator outputting n bits is called explicit *if it can be computed in* $\mathsf{poly}(n)$ *time.*

All PRGs described in the rest of this thesis are explicit, so we will omit the word "explicit" when stating these PRGs.

Basic pseudorandom constructions

For some targeted class of functions, we want to construct pseudorandom generators with short seed length (or equivalently a pseudorandom distribution that can be sampled using few random bits). We describe below some basic pseudorandom constructions that are useful for constructing PRGs.

A multidimensional distribution is called *k-wise independent* if any k coordinates of the distribution are uniformly distributed.

Definition 2.6 (k-wise independence)**.** *A distribution X over $[m]^n$ is called* k-wise independent *if, for any $1 \leq i_1 < \cdots < i_k \leq n$ and every $b_1, \ldots, b_k \in [m]$, we have*

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

Theorem 2.7 (see [Vad12, Proposition 3.33])**.** *For any positive integers n, m, k, there exists a k-wise independent distribution over $[m]^n$ that can be sampled in $\mathsf{poly}(n, m, k)$ time using $k \cdot \max\{\log n, \log m\}$ random bits.*

Definition 2.8 (Small-bias distributions)**.** *A distribution X over $\{0,1\}^n$ is called* ε-biased *if, for every nonempty subset $S \subset [n]$, we have*

$$\left|\mathbf{Pr}[\oplus_{i \in S} X_i = 1]\right| \leq \frac{1}{2} + \frac{\varepsilon}{2}.$$

Since for every non-constant XOR function[1] f, we have $\mathbf{E}_{x \sim \{0,1\}^n}[f(x)] = 1/2$, we have that ε-biased distributions $(\varepsilon/2)$-fool the class XOR functions.

Theorem 2.9 (see [AGHP92])**.** *For any positive integer n and any $0 < \varepsilon < 1$, there exists a ε-biased distribution over $\{0,1\}^n$ that can be sampled in $\mathsf{poly}(n)$ time using $O(\log(n/\varepsilon))$ random bits.*

2.4 Random restrictions

An important tool in circuits complexity is the method of random restrictions. A restriction for a n-variate Boolean function f, usually denoted as $\rho \in \{0, 1, *\}^n$, specifies a way of fixing the values of some subset of variables for f. That is, if ρ_i is $*$, we leave the i-th variable

[1] An XOR function computes the parity of some fixed set of input bits.

unrestricted and otherwise fix its value to be $\rho_i \in \{0,1\}$. We denote by f_ρ the restricted function after the variables are restricted according to ρ, and denote by $\rho^{-1}(*)$ the set of unrestricted variables. A random restriction is then a distribution over restrictions. We will often view sampling a random restriction as a two-step process: The first step is selecting (in some random manner) a subset of unrestricted variables (also called the "star" or "$*$" variables) and the second step is fixing (in some random manner) the values of all the other variables. Then, a random restriction over n variables can also be specified by a pair $(\sigma, \beta) \in \{0,1\}^n \times \{0,1\}^n$, where σ (as a characteristic string) specifies the set of unrestricted variables, and β specifies the values for fixing the restricted variables.

We say that a random restriction (or random selection) is p-regular if each variable is left unrestricted with probability p. One way to generate a p-regular random restriction is to leave each variable, independently, unrestricted with probability p, and otherwise assign to it a 0 or a 1, uniformly at random. Such a random restriction is called a *(truly) p-random restriction*. Note that to sample such a restriction, we can first pick a string in $\{0,1\}^{n \cdot \log(1/p)} \cong [1/p]^n$ to specify the selection of the unrestricted variables, where a coordinate is unrestricted if and only if all of its corresponding $\log(1/p)$ bits are 0, and then a string in $\{0,1\}^n$ to specify the values assigned to each of the restricted variables. So sampling a restriction in this way requires $n \cdot \log(1/p) + n$ random bits. We can also generate a restriction in a pseudorandom manner, which may use fewer random bits. For example, one way to do this is to use a limited-independence distribution, so that each variable is set to be unrestricted with probability p, and any k of the variables are independent. Note that such a "pseudorandom selection" can be obtained using a k-wise independent distribution on $[1/p]^n$. Also, we can let each variable be assigned a 0 or a 1 uniformly at random in a way such that any k of the variables are independent; this again can be done using a k-wise independent distribution on $\{0,1\}^n$.

Finally, note that we can also get a restriction by combining a sequence of restrictions $\rho_1, \ldots, \rho_t$, in a natural way, namely by applying the sub-restrictions one by one. In this case, we write the final restriction as $\rho_1 \circ \cdots \circ \rho_t$.

2.5 Notation

For a positive integer n, we denote the set $\{1, \ldots, n\}$ by $[n]$. We use $\widetilde{O}(\cdot)$ (and $\widetilde{\Omega}(\cdot)$) to hide polylogarithmic factors. That is, for any $f\colon \mathbb{N} \to \mathbb{N}$, we have that $\widetilde{O}(f(n)) = O(f(n) \cdot \mathsf{polylog}(f(n)))$.

We will sometime abuse the notation of a circuit class and use it to mean the "type" of a circuit. For example, when we say "AC^0 circuits of depth d and size s", we mean circuits of depth d and size s, where the AND and OR gates have *unbounded* fan-in.

Chapter 3

Random Restrictions for Polynomials and Lower Bounds for Polynomial Threshold Circuits

3.1 Background and results

Random restrictions of Boolean functions play an important role in the circuit complexity research. One way to prove that a certain Boolean function h is not computable by a class $\mathcal{C}$ of Boolean circuits is to show that (1) every Boolean function $f \in \mathcal{C}$ becomes "simplified" after a random restriction (which randomly fixes some random subset of variables of f), and (2) the function h "remains hard" after a random restriction. This strategy has been successfully applied to prove circuit lower bounds for explicit Boolean functions against (i) De Morgan formulas [Sub61, And87], (ii) AC^0 circuits [Ajt83, FSS84, Yao85, Hås89], and (iii) constant-depth circuits with LTF (linear threshold function) gates [IPS97, CSS18, KW16]. In most of these results, the notion of "simplified" means that a restricted function is (almost) a constant function; the hard function h is the parity function, which is the ultimate example of a function that cannot be made (close to) constant under any restriction that leaves enough variables unrestricted.

Lower bounds against constant-depth circuits with PTF gates. One of the original motivations for the present work was to extend the lower bounds of [CSS18, KW16] to the class of constant-depth circuits consisting of general Polynomial Threshold Function (PTF) gates. Recall that a degree-d PTF is defined to be the sign of a multilinear degree-d polynomial over the reals. Constant-depth PTF circuits are quite powerful. Every n-variate Boolean function computable by a polynomial-size $\mathsf{AC}^0[m]$ circuit (constant-depth circuits with AND, OR, and MOD_m gates), for any integer $m > 1$, has an equivalent depth-2 circuit of quasipolynomial size with d-degree PTF gates, for $d \leq \mathsf{poly}(\log n)$ [All89, Yao90]. Moreover, every Boolean function computable by a polynomial-size AC^0 circuit can be well approximated by a single PTF gate of degree $d \leq \mathsf{poly}(\log n)$ [LMN93].

We prove circuit lower bounds against constant-depth PTF circuits of super-constant degree d as long as $d \ll \sqrt{\log n / \log \log n}$. This is close to the best possible given the current knowledge, as virtually nothing is known for PTFs of degree bigger than $\log n$. We state our circuit lower bounds below in Section 3.1.1. Our main technical tool is a restriction lemma for PTFs, which we discuss next.

Restriction Lemmas. Let $f(x_1, \ldots, x_n)$ be a Boolean function assuming the values $\{1, -1\}$ over the Boolean cube $\{-1, 1\}^n$. For a parameter $0 < r < 1$, a random r-restriction is defined to leave each variable free (unrestricted) with probability r, and, otherwise (with probability $1 - r$), fixing the variable to either 1 or -1 uniformly at random. For a parameter $0 < \delta < 1$, we say that a Boolean function g is δ-close to constant if, for some $v \in \{-1, 1\}$, we have $g(x) = v$ for all but at most δ fraction of Boolean inputs x. We show that a PTF f of degree d is likely to become δ-close to constant after being hit with a random r-restriction: the probability that f fails to become δ-close to constant is at most $\sqrt{r} \cdot (\log r^{-1} \cdot \log \delta^{-1})^{O(d^2)}$.

This Restriction Lemma for PTFs is sufficient to derive the aforementioned lower bounds against constant-depth PTF circuits. Moreover, it can also be used to re-derive (as an immediate corollary) the optimal-exponent average sensitivity bound for degree-d PTFs due to Kane [Kan14]. But perhaps more interestingly, this Restriction Lemma for PTFs is a consequence of a more general Restriction Lemma for degree-d polynomials, which has other applications.

First, we generalize our PTF Restriction Lemma to the case of "structured" random restrictions. Suppose that the variables of a given Boolean function $f(x_1, \ldots, x_n)$ are partitioned into m disjoint subsets (blocks) of variables. Given such a block partition, a random block restriction is defined as follows: pick a uniformly random block $\ell \in [m]$, and assign each variable outside the chosen block ℓ a uniformly random value in $\{-1, 1\}$. We show that a degree-d PTF f with an arbitrary block partition into m blocks is likely to become δ-close to constant after being hit with a random block restriction: the probability that f fails to become δ-close to constant is at most $m^{-1/2} \cdot (\log m \cdot \log \delta^{-1})^{O(d^2)}$. It is not hard to see that a PTF Restriction Lemma for r-random restrictions is a corollary of the m-block Restriction Lemma when $m = 1/r$.

The PTF Block Restriction Lemma mentioned above is a consequence of the following Block Restriction Lemma for polynomials. If a multilinear degree-d polynomial with a given block partition into m blocks is hit with a random block restriction, it is likely to become "concentrated" in the sense that its standard deviation becomes quite small relative to its expectation. It can be shown that if a polynomial is concentrated, then the sign function of this polynomial (i.e., the corresponding PTF) is close to a constant function. Thus, the PTF Block Restriction Lemma follows. In addition, this structural property of a polyno-

mial becoming "concentrated" under random block restrictions is also useful for proving Littlewood-Offord type anticoncentration results for polynomials.

Littlewood-Offord type anticoncentration bounds. Let p be an arbitrary n-variate degree-d multilinear polynomial containing at least t disjoint maximal monomials (i.e., not contained in other monomials), each with a coefficient at least 1 in magnitude. Meka et al. [MNV16] showed that it is unlikely that, for a random $x \in \{-1,1\}^n$, the value $p(x)$ will fall in the interval $[0,1]$: the probability that $p(x) \in [0,1]$ is at most $(1/\sqrt{t}) \cdot (\log t)^{O(d \log d)} \cdot \exp(d^2 \log d)$. We re-derive this result, in a simple way, from our Block Restriction Lemma for polynomials. Moreover, we also prove a *derandomized* version of this bound, which we discuss next.

Derandomized Block Restriction Lemma and derandomized Littlewood-Offord. A random block restriction chooses a uniformly random block, and then assigns uniformly random values to all variables outside that block. Our Block Restriction Lemma says that such a random block restriction is likely to make an n-variate degree-d multilinear polynomial "concentrated". A derandomized version of this lemma would say that a similar conclusion is true for block restrictions that can be sampled with significantly fewer random bits. We prove such a derandomized version for pseudorandom m-block restrictions that are sampled using about $(\log m)^{O(d^2)} \cdot \log n$ random bits.

We then use this derandomized version of the Block Restriction Lemma to obtain derandomizations of the Littlewood-Offord type bounds. We show that there is an efficient pseudorandom generator for sampling inputs $x \in \{-1,1\}^n$, using significantly fewer than n random bits, such that the following holds. For any degree-d multilinear polynomial p with many degree-d monomials that have large coefficients, it is unlikely that $p(x) \in [0,1]$ for these pseudorandom inputs $x \in \{-1,1\}^n$. No derandomized versions of the Littlewood-Offord type anticoncentration bounds were previously known.

Next we provide more details about our main results and our proof techniques.

3.1.1 Results

Lower bounds for constant-depth PTF circuits. We generalize the lower bounds of [KW16] and [CSS18] to the case of constant-depth circuits with PTF gates of degree $d \geq 1$. For the case of $d = 1$, our results match those obtained in [KW16, CSS18]. For $d > 1$, these appear to be the first super-linear wire complexity lower bounds against constant-depth circuits with degree-d PTF gates.

The following generalizes the lower bounds against LTF circuits of depth 2 of [KW16].

Theorem 3.1. *There is an n-variate Boolean function $F_n \in \mathsf{P}$ such that every depth-2 circuit with PTF gates of degree $d \geq 1$ that computes F_n must have at least $\left(n^{\frac{1}{2}+\frac{1}{d}}\right) \cdot (\log n)^{-O(d^2)}$ gates, and at least $\left(n^{\frac{3}{2}+\frac{1}{d}}\right) \cdot (\log n)^{-O(d^2)}$ wires.*

We also generalize to PTF gates (and somewhat strengthen) a lower bound of [KW16] for depth-3 circuits that have the Majority gate at the top, with depth-2 LTF circuits feeding in.

Theorem 3.2. *There is a polynomial-time computable Boolean function B such that the following holds. For any $\frac{1}{\log n} \ll \varepsilon < 1$, let C be a majority vote of depth-2 circuits with degree-d PTF gates such that the top majority gate has fan-in at most 2^{n^ε} and the total fain of the gates on the bottom layer at most $w = \left(n^{\frac{3}{2}+\frac{1}{d}}\right) \cdot (n^\varepsilon \cdot \log n)^{-c \cdot d^2}$, where c is a constant. Then C cannot compute B.*

For Boolean functions $f, g\colon \{-1,1\}^n \to \{-1,1\}$, define the correlation between f and g as

$$\mathbf{Corr}(f,g) = \left| \mathop{\mathbf{E}}_{x \sim \{-1,1\}^n} [f(x) \cdot g(x)] \right|.$$

Let PARITY_n denote the n-input parity function. We generalize the correlation bounds of [CSS18], getting the following.

Theorem 3.3. *For any $D \geq 1$ and $1 \leq d \ll \sqrt{\log n / \log\log n}$, let C be any depth-D circuit on n inputs with degree-d PTF gates, of wire complexity at most $n^{1+\varepsilon_D}$, where $\varepsilon_D = B^{-(2D-1)}$, for some constant $B > 0$. Then we have*

$$\mathbf{Corr}(C, \mathsf{PARITY}_n) \leq O\left(n^{-\varepsilon_D}\right).$$

Theorem 3.4. *There is an n-variate Boolean function $G_n \in \mathsf{P}$ such that the following holds. For any $D \geq 1$ and $1 \leq d \ll (\log n / \log\log n)^{1/(2D-1)}$, let C be any depth-D circuit on n inputs with degree-d PTF gates, of wire complexity at most $n^{1+\mu_{D,d}}$, where $\mu_{D,d} = (E \cdot d)^{-(2D-1)}$, for some constant $E > 0$. Then we have*

$$\mathbf{Corr}(C, G_n) \leq \exp(-n^{\mu_{D,d}/2}).$$

Restriction lemmas for polynomials. Our main tool is the following structural lemma showing that a PTF is likely to become an almost constant function after being hit with a random restriction. Below, we denote by $\rho \sim R_r$ the process of picking a random restriction ρ that leaves a variable free with probability r, and otherwise fixes uniformly at random to 1 or -1. We denote by f_ρ the function f restricted by ρ. We say that a Boolean function f is δ-close to constant if, for some value $v \in \{-1,1\}$, we have $f(x) = v$ for all but at most δ fraction of Boolean inputs x.

Lemma 3.5 (PTF Restriction Lemma)**.** *For any PTF $f(x) = \mathrm{sgn}(p(x))$ of degree $d \geq 1$, and any $0 < \delta$, $r \leq 1/16$, we have*

$$\Pr_{\rho \sim R_r} [f_\rho \text{ is not } \delta\text{-close to constant}] \leq \sqrt{r} \cdot (\log r^{-1} \cdot \log \delta^{-1})^{O(d^2)}.$$

We note that the bound $r^{1/2}$ in this lemma has an optimal exponent.

The above lemma is a consequence of a more general result, the *Block Restriction Lemma*, which deals with certain structured restrictions that we define next. Suppose the variables of a given function are arbitrarily partitioned into m blocks. For the given block partitioning, a random *block restriction* $\rho \sim B_m$ is defined by picking a block $\ell \in [m]$ uniformly at random, and assigning each variable outside block ℓ the value 1 or -1 uniformly at random. We show that, for an arbitrary partitioning of input variables into m blocks, the probability that a degree-d PTF is not δ-close to constant, after being hit with a random block restriction $\rho \sim B_m$, is at most the same as the bound in the PTF Restriction Lemma above, with $r = 1/m$.

Lemma 3.6 (Block Restriction Lemma: Simplified version)**.** *For any PTF $f(x) = \mathrm{sgn}(p(x))$ of degree $d \geq 1$, any $m \geq 16$, and any $0 < \delta \leq 1/16$, we have*

$$\Pr_{\rho \sim B_m}[f_\rho \text{ is not } \delta\text{-close to constant}] \leq m^{-1/2} \cdot (\log m \cdot \log \delta^{-1})^{O(d^2)}.$$

Note that a standard random restriction $\rho \sim R_r$ can be obtained by first randomly partitioning the input variables into $m = 1/r$ blocks, and then applying a random block restriction from B_m. So the Block Restriction Lemma implies the PTF Restriction Lemma.

Our actual Block Restriction Lemma (Lemma 3.37) shows something even stronger. If a degree d multilinear polynomial p is hit with a random block restriction, it becomes "concentrated around the expectation" in the sense that its standard deviation becomes quite small relative to its expectation (in particular, implying that the restriction of the PTF $\mathrm{sgn}(p)$ is close to constant).

Other applications. Apart from the aforementioned circuit lower bound applications, our Block Restriction Lemma also immediately implies two other results. We get the average sensitivity bound on degree-d PTFs, with an optimal exponent, first shown by Kane [Kan14] in the context of the Gotsman-Linial conjecture [GL94]; see Theorem 3.61 below.

We also get the following Littlewood-Offord type anticoncentration bound for degree-d multilinear polynomials, due to Meka et al. [MNV16], which is an extension of the classical Littlewood-Offord result for linear polynomials [LO43, Erd45].

Theorem 3.7 ([MNV16])**.** *For any real interval I, and any n-variate degree-d multilinear polynomial p such that there exists a set of t disjoint monomials in p, each of which is maximal (i.e., not contained by any other monomials) and has coefficient at least $|I|$ in magnitude, we have*

$$\mathbf{Pr}[p(A) \in I] \leq t^{-1/2} \cdot (\log t)^{O(d \log d)} \cdot 2^{O\left(d^2 \log d\right)},$$

where A is the uniform distribution over $\{-1, 1\}^n$.

Derandomization. We prove a derandomized version of the Block Restriction Lemma mentioned above.

Theorem 3.8 (Derandomized Block Restriction Lemma: Simplified version)**.** *For any $0 < \delta \leq 1/16$ and $0 < \zeta < 1$, there is a polynomial-time algorithm for sampling block restrictions $\rho \in B_m$, for any $m \geq 16$, that uses at most $m^{\zeta} \cdot \log n$ random bits, so that the following holds. For any n-variate degree-d PTF f whose variables are partitioned into m blocks, we have*

$$\mathop{\mathbf{Pr}}_{\rho}[f_\rho \text{ is not } \delta\text{-close to constant}] \leq m^{-1/2} \cdot \left(\log m \cdot \log \delta^{-1}\right)^{O\left(\zeta^{-1} \cdot d^2\right)}.$$

Our actual version of this lemma (see Theorem 3.64) shows that a degree-d polynomial p is likely to become "concentrated" under a pseudorandom block restriction. This in turn is used to prove the following derandomized versions of Theorem 3.7.

Theorem 3.9. *For any positive integers t and d, and $0 < \zeta < 1$, there exists a distribution D on $\{-1,1\}^n$, samplable in $\mathsf{poly}(n)$ time using $t^{\zeta/d} \cdot \log n$ random bits, such that the following holds. For any real interval I, and any n-variate degree-d multilinear polynomial p that has at least t disjoint degree-d monomials with coefficient at least $|I|$ in magnitude, we have*

$$\mathbf{Pr}[p(D) \in I] \leq t^{-\frac{1}{2d}} \cdot (\log t)^{O\left(\zeta^{-1} \cdot d^2\right)}.$$

Theorem 3.10. *For any positive integers t and d, and $0 < \zeta < 1$, there exists a distribution D on $\{-1,1\}^n$, samplable in $\mathsf{poly}(n)$ time using $t^{\zeta} \cdot \log n$ random bits, such that the following holds. For any real interval I, and any n-variate degree-d multilinear polynomial p with at least $t \cdot n^{d-1}$ degree-d monomials whose coefficients are at least $|I|$ in magnitude, we have*

$$\mathbf{Pr}[p(D) \in I] \leq t^{-\frac{1}{2}} \cdot (\log t)^{O\left(\zeta^{-1} \cdot d^2\right)}.$$

Note that it is possible to use PRGs for PTFs directly to get a derandomized Littlewood-Offord anticoncentration bound. However, using the best currently known PRGs for PTFs, such a derandomization will have large error and seed length. In particular, for dense polynomials with $t = n^{1-o(1)}$, Theorem 3.10 achieves error less than $1/n^{0.49}$ with the seed size at most polylogarithmic in n; such parameters are beyond reach of the best available PRGs for PTFs.

3.1.2 Related work

Random restrictions. The concept of random restrictions for Boolean functions was introduced by Subbotovskaya [Sub61], who applied it to show that the n-bit parity function requires De Morgan formulas[1] of size $\Omega(n^{1.5})$ (later improved to the optimal bound

[1]De Morgan formulas are built using AND, OR, and NOT gates.

$\Omega(n^2)$ by [Khr71]). Andreev [And87] combined random restrictions with a counting argument to show a stronger lower bound against De Morgan formulas for a function in P (resulting in the $n^3/\mathsf{poly}(\log n)$ bound, when using Håstad's improved restriction lemma for De Morgan formulas [Hås98]). Random restrictions were also used for showing the aforementioned exponential lower bounds against AC^0 circuits computing the parity function [Ajt83, FSS84, Yao85, Hås89], as well as for the lower bounds against constant-depth LTF circuits by [IPS97, CSS18, KW16]. A common feature in all of these lower bound proofs is a structural result showing how "easy" Boolean functions (of appropriately small formula or circuit complexity) become much "simpler" (e.g., become almost constant) after being hit with random restrictions. In contrast, the parity function is the ultimate "restriction-resistant" function that does not simplify under random restrictions, but rather stays the parity function (albeit on a smaller number of variables).

Two classical examples of restriction lemmas are the Shrinkage Lemma for De Morgan formulas [Sub61, IN93, PZ93, Hås98, Tal14], and Håstad's Switching Lemma for AC^0 circuits [Hås89] (see also [RST15, Hås16]). The Shrinkage Lemma says that a De Morgan formula of size s is expected to shrink to size about $r^2 \cdot s$, after being hit with a random restriction $\rho \sim R_r$ that leaves each variable free with probability r, and otherwise fixes the variable to a uniform bit. Håstad's Switching Lemma says that any given k-CNF formula (the conjunction of clauses of size at most k each) is very likely to become expressible as a k-DNF (the disjunction of size-k terms), after being hit with a random restriction $\rho \sim R_r$ for $r = O(1/k)$. By repeatedly applying this Switching Lemma to a given AC^0 circuit (of not too large size), level by level, we can merge the adjacent levels, thereby collapsing the original circuit to depth at most 2. Once the original AC^0 circuit is thus "simplified", one can argue directly that the new circuit is too weak to compute the restriction of the original function (e.g., the parity function).

A similar strategy was used by Impagliazzo et al. [IPS97] to show that the parity function is hard for constant-depth LTF circuits. The main technical result of [IPS97] shows that a depth d LTF circuit (of not too large size) can be reduced to a depth $d-1$ LTF circuit by fixing not too many input variables. This is argued by showing that there exists a particular restriction of input variables, chosen adaptively, that will make all LTF gates at the bottom level of the circuit to be constants (or depend on at most one input). To extend the worst-case lower bounds of [IPS97] to the average-case correlation bounds, Chen et al. [CSS18] extended the restriction lemma of [IPS97] to the setting of truly random, non-adaptive restrictions. So too did Kane and Williams [KW16] to get a lower bound against depth-2 LTF circuits for Andreev's function; their restriction lemma is for certain "block-structured" random restrictions, as required by Andreev's original argument.

Our restriction lemma, Lemma 3.5, can be used to re-derive the same lower bounds (up to polylogarithmic factors) for LTF circuits as in [CSS18, KW16]. Moreover, it extends these lower bounds to the case of PTF gates of any degree $d \ll \sqrt{\log n / \log\log n}$.

Comparison with [KW16], [CSS18] and [Nis94]. Kane and Williams [KW16] prove an LTF Restriction Lemma with similar parameters to our PTF Restriction Lemma (for $d = 1$), for certain random block restrictions and for the case of the restricted LTF becoming a constant function (rather than close to constant). For the proof, they rely on the Littlewood-Offord lemma from additive combinatorics [LO43, Erd45]. It is not clear how to extend such a proof to the case of higher degree PTFs.

Chen et al. [CSS18] obtain a quantitatively weaker version of the LTF Restriction Lemma, using proof techniques similar to ours, but with worse parameters. We get better (almost optimal) parameters for both LTFs and higher degree PTFs, by using the more refined proof techniques developed in [Kan14].

Nisan [Nis94] obtains an almost linear $\Omega_d(n^{1-o(1)})$ gate complexity lower bound against circuits with degree-d PTF gates of any depth, using the techniques from communication complexity, which are quite different from those in [KW16, CSS18] and this work. It is not clear how such techniques can be used to obtain super-linear lower bounds for wire or gate complexity. For degree $d = 1$, our lower bound result for depth-2 PTF circuits recovers the super-linear $n^{1.5-o(1)}$ gate complexity lower bounds against depth-2 LTF circuits first shown in [KW16]. For higher degrees, this result cannot give super-linear lower bounds and does not match Nisan's lower bounds. On the other hand, both our results for depth-2 and higher constant depth PTF circuits give a super-linear wire complexity lower bound, which is not implied by [Nis94] or prior works.

Lower bounds against TC^0. For depth-2 circuits with majority gates (equivalently, LTF gates with polynomially small weights), Hajnal et al. [HMP$^+$93] showed an exponential size lower bound for the Inner Product modulo 2 (IP2) function. For the parity function, Paturi and Saks [PS94] showed a nearly optimal $\tilde{\Omega}(n)$ gate complexity lower bound against depth-2 majority circuits. This was extended by Siu et al. [SRK94] to depth-D such circuits, showing that n-bit Parity requires at least $\tilde{\Omega}(D \cdot n^{1/(D-1)})$ gates; they also showed a matching upper bound of $O(D \cdot n^{1/(D-1)})$ gates.

Goldmann et al. [GHR92] (improving upon [SB91]) proved a surprising result that any general LTF circuit of constant depth D has an equivalent majority circuit of polynomially related size and depth $D + 1$. Thus any superpolynomial lower bound against majority circuits of constant depth D would immediately yield a superpolynomial lower bounds against general LTF circuits of depth $D - 1$. (This connection may explain the lack of any strong lower bounds even for depth-3 majority circuits.) Allender and Koucký [AK10] show that proving superpolynomial circuit lower bounds against TC^0 circuits (for an NC^1-complete function) is equivalent to proving super-linear, $n^{1+\varepsilon}$, lower bounds for every depth $D \geq 2$, where $\varepsilon > 0$ is independent of the depth D.

PTFs. PTFs have also been studied in the context of learning [STT12, DOSW11, DSTW14], pseudorandomness [DGJ+10, DKN10, Kan11, Kan12, MZ13, Kan14, Kan15], approximate counting [DDS14, DS14], and extremal combinatorics [Sak93, GL94, OS08, DRST14, Kan14].

3.1.3 Techniques

Block Restriction Lemma. The proof of our Block Restriction Lemma (Lemma 3.6) relies on the techniques in [Kan14]. An oversimplified proof sketch is as follows. We first show that if a degree-d multilinear polynomial is not "concentrated" (i.e., has the standard deviation much larger that the expectation), then it is expected to have a relatively large directional derivative compared to its actual value. We then use anticoncentration bounds for polynomials to argue that it is unlikely that a random restriction of a degree-d multilinear polynomial will have such a property.

One issue is that strong enough anticoncentration bounds for polynomials (e.g., the Carbery-Wright bound [CW01], or the bound from [Kan14] that we will actually use) are true only under the Gaussian measure rather than the uniform distribution over the Boolean cube. To use these anticoncentration results, we thus need to move from the Bernoulli distribution to the Gaussian distribution over the inputs of polynomials. Such change of the probability measure is possible thanks to the celebrated Invariance Principle of [MOO10]. It applies to "regular" polynomials only, but fortunately there is a "regularity lemma" of [DSTW14] (or a variant from [Kan14]) that allows one to reduce the analysis of arbitrary polynomials to the case of regular ones, at a small cost.

The next problem is that the Invariance Principle incurs significant (and unavoidable) losses that have a bad dependence on the degree d of the polynomial in question. To mitigate such losses, we apply a random block restriction ρ in a series of few steps, viewing ρ as a composition of t restrictions $\rho_1 \circ \rho_2 \circ \cdots \circ \rho_t$ (for not too large value $t \geq 1$), where each ρ_i is on relatively small number of blocks m_i. This allows us to ensure that the loss from the Invariance Principle at each step i is "absorbed" by the parameter m_i.

Thus we get a recursive proof, where in each step we apply the regularity lemma, the Invariance Principle, and the anticoncentration bound. Carrying out such a proof directly, we get a weak version of the Block Restriction Lemma (see Lemma 3.34). By a more careful recursive analysis (using a "soft" measure of "non-concentration" for polynomials), we get the stronger version stated in Lemma 3.37.

Derandomized Block Restriction Lemma. To prove a derandomized version of the Block Restriction Lemma (Theorem 3.8), we first observe that a block restriction ρ makes a given degree-d polynomial "concentrated" if and only if a certain PTF of degree $2d$ evaluates to -1 on ρ. Thus finding a good-restriction ρ is reduced to the task of fooling degree-$2d$ PTFs. For the latter, we can use known constructions of pseudorandom generators (PRGs) for PTFs, e.g., the construction due to Meka and Zuckerman [MZ13]. Unfortunately, the

parameters of the known PRGs for PTFs are far from optimal. Using such a PRG in a single step would yield a derandomized Block Restriction Lemma with very poor parameters. Instead, we use a recursive strategy similar to the recursive proof of Lemma 3.6 above. We build a pseudorandom block restriction in a sequence of steps, where in each single step we are facing a block partition on a relatively small number of blocks, and so can afford the relatively poor parameters of the PRG construction from [MZ13].

Derandomized Littlewood-Offord. To prove Theorem 3.9 and Theorem 3.10, we first argue that bounded-wise independent hash functions can be used to produce a block partition of input variables such that, with high probability, every polynomial p satisfying the assumptions of these theorems will contain within each block a high-degree monomial with a large coefficient. Once we have a good partition, we can use our derandomized Block Restriction Lemma to generate the required pseudorandom inputs x for the polynomial p so that $p(x)$ is unlikely to be contained within a small interval.

Organization of this chapter. We give the necessary background in Section 3.2. In Section 3.3, we prove a simpler Block Restriction Lemma as a warm-up. In Section 3.4, we prove another Block Restriction Lemma that achieves the optimal exponent in the parameter m (the number of blocks). It is a weaker version of our final Block Restriction Lemma, which illustrates our proof techniques. The stronger version is then proved in Section 3.5. In Section 3.6, we give our applications of the Block Restriction Lemma: we prove Theorems 3.1–3.4; re-derive Kane's average sensitivity bound for degree-d PTFs in Section 3.6.4; and show a Littlewood-Offord type anticoncentration bound for degree-d multilinear polynomials in Section 3.6.5. We prove our derandomized restriction lemma and derandomized Littlewood-Offord type anticoncentration bounds in Section 3.7. Section 3.8 contains some open problems.

3.2 Preliminaries

Here we present some definitions and results on polynomials and polynomial threshold functions, and introduce some basic notions in the analysis of Boolean functions. For more details on these (and related) topics, the reader is referred to [O'D14].

Notation

Throughout this chapter, we will use $\{-1, 1\}$ as the Boolean domain.

We will denote by X, Y, Z standard multidimensional Gaussian random variables. That is, for dimension n, we have $X \sim N(0,1)^n$, where $X = (X_1, \ldots, X_n)$ and all components $X_i \sim N(0,1)$ are independent Gaussians. Similarly, we denote by A, B, C multidimensional Bernoulli variables, where, for dimension n, $A = (A_1, \ldots, A_n)$, and all components $A_i \sim$

$\{-1,1\}$ are independent fair coin flips. Occasionally, for results that hold for both Gaussian and Bernoulli distributions, we use I, J to denote distributions that may be either standard n-dimensional Gaussian, or Bernoulli distributions.

Boolean functions and polynomial threshold functions

We think of an n-variate Boolean function f as $f\colon \{-1,1\}^n \to \{-1,1\}$. For two Boolean functions $f, g\colon \{-1,1\}^n \to \{-1,1\}$ and a parameter $\delta \in [0,1]$, we say that f and g are *δ-close* if $\mathbf{Pr}_{x\sim\{-1,1\}^n}[f(x) \neq g(x)] \leq \delta$. We say that a Boolean function f is δ-close to constant if there is a constant function v, where $v = -1$ or $v = 1$, such that f and v are δ-close.

Definition 3.11. *A* degree-d polynomial threshold function (PTF) *is a function $f\colon \{-1,1\}^n \to \{-1,1\}$ of the form $f = \mathrm{sgn}(p)$, where $p\colon \mathbb{R}^n \to \mathbb{R}$ is a multilinear polynomial of degree at most d, and $\mathrm{sgn}\colon \mathbb{R} \to \{-1,1\}$ is the sign function defined to be 1 on all positive inputs, and -1 on all negative inputs and on 0.*[2]

Concentration and anticoncentration for polynomials

Definition 3.12 (L^t norm)**.** *For $f\colon \mathbb{R}^n \to \mathbb{R}$ and a real number $t \geq 1$, the Gaussian (Bernoulli) L^t norm of f is defined as*

$$\|f\|_t = \left(\mathbf{E}[|f(I)|^t]\right)^{1/t},$$

where I is an n-dimensional Gaussian (Bernoulli) random variable.

It is easy to see that the Gaussian and Bernoulli L^2 norms are the same for any multilinear polynomial p, i.e.,

$$\mathbf{E}[|p(X)|^2] = \mathbf{E}[|p(A)|^2].$$

We denote by $\|p\|_2$ the L^2 norm of a multilinear polynomial p under Gaussian (or Bernoulli) distribution. For multilinear polynomials p, we also have

$$\mathbf{E}[p(X)] = \mathbf{E}[p(A)].$$

Hence the variance of p is the same under Gaussian and Bernoulli measures, and we will denote this variance by $\mathbf{Var}[p]$.

The hypercontractivity results of [Bon70] relate the L^t norm of a polynomial to its L^2 norm, both for Gaussian and Bernoulli measures. For multilinear d-degree polynomials p,

[2]Without loss of generality, we may assume for every PTF $f = \mathrm{sgn}(p)$ that $p(x) \neq 0$ for all $x \in \{-1,1\}^n$. The reason is that we may always change p to a new polynomial $\tilde{p} = p + \eta$, for a small constant $\eta \in \mathbb{R}$, so that, for every $x \in \{-1,1\}^n$, we have both $\mathrm{sgn}(p(x)) = \mathrm{sgn}(\tilde{p}(x))$ and $\tilde{p}(x) \neq 0$.

the relevant hypercontractive inequality is

$$\|p\|_t \leq (t-1)^{d/2} \cdot \|p\|_2 , \tag{3.1}$$

where the L^t norm on the left-hand side may be either Gaussian or Bernoulli.

The following strong concentration bound for polynomials is an immediate consequence of Equation (3.1) and the Markov inequality.

Theorem 3.13 (Concentration bound)**.** *For every d-degree multilinear polynomial $p\colon \mathbb{R}^n \to \mathbb{R}$, and for every $K \geq 2^d$, we have*

$$\mathbf{Pr}\left[|p(I)| \geq K \cdot \|p\|_2\right] \leq \exp\left(-(1/4) \cdot K^{2/d}\right),$$

where I is an n-dimensional Gaussian or Bernoulli random variable.

The following weak anticoncentration result for polynomials is also an immediate consequence of Equation (3.1) (for $t = 4$) and the Paley-Zygmund inequality (applied to p^2).

Theorem 3.14 (Weak anticoncentration bound)**.** *For every d-degree multilinear polynomial $p\colon \mathbb{R}^n \to \mathbb{R}$, we have*

$$\mathbf{Pr}\left[|p(I)| \geq (1/2) \cdot \|p\|_2\right] \geq (1/2) \cdot 9^{-d},$$

where I is an n-dimensional Gaussian or Bernoulli random variable.

A stronger anticoncentration result for polynomial with respect to the Gaussian measure, due to Carbery and Wright [CW01], shows that $|p(X)|$ is likely to exceed $\varepsilon \cdot \|p\|_2$.

Theorem 3.15 (Anticoncentration bound [CW01])**.** *For any non-zero degree-d polynomial p and any $\varepsilon > 0$, we have*

$$\mathbf{Pr}\left[|p(X)| \leq \varepsilon \cdot \|p\|_2\right] = O(d \cdot \varepsilon^{1/d}).$$

The anticoncentration bound above has poor dependence on the degree d. For an improved dependence on d, we use a version of the anticoncentration result due to Kane [Kan14], where $|p(X)|$ is compared to the directional derivative of p, rather than the norm of p.

Definition 3.16 (Directional derivative)**.** *For an n-variate function $g(x_1, \ldots, x_n)$ from $\mathbb{R}^n$ to $\mathbb{R}$, and $u, v \in \mathbb{R}^n$, the* directional derivative of g at u in the direction v*, denoted $\mathsf{D}_v\, g(u)$, is defined as*

$$\mathsf{D}_v\, g(u) = v \cdot \nabla g(u),$$

where

$$\nabla g = \left(\frac{\partial g}{\partial x_1}, \ldots, \frac{\partial g}{\partial x_n}\right)$$

is the gradient of g, and "$\cdot$" denotes the usual inner product of vectors.

Theorem 3.17 (Strong anticoncentration bound [Kan14])**.** *For any non-zero polynomial p of degree d and any $\varepsilon > 0$, we have*

$$\mathbf{Pr}\left[|p(X)| \leq \varepsilon \cdot |\mathsf{D}_Y\, p(X)|\right] = O\left(d^2 \cdot \varepsilon\right).$$

This strong anticoncentration bound will be useful to us thanks to the following (easily provable) identity:

$$\mathbf{E}\left[|\mathsf{D}_J\, p(I)|^2\right] = \mathbf{E}\left[\|\nabla p(I)\|_2^2\right], \tag{3.2}$$

where I and J are either independent Gaussians, or independent Bernoulli distributions. In turn, the quantity on the right-hand of Equation (3.2) can be related to the variance $\mathbf{Var}[p]$, via the notion of *influence*, to be discussed in the next subsection.

We conclude this subsection with the following useful relationship between the directional derivative and the gradient.

Lemma 3.18. *For any non-negative integer k and any degree-d multilinear n-variate polynomial p such that $p(x) \neq 0$ for all $x \in \{-1,1\}^n$, we have*

$$\mathop{\mathbf{E}}_{A,B}\left[\min\left\{k, \frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right\}\right] \leq \mathop{\mathbf{E}}_{A}\left[\min\left\{k, \frac{\|\nabla p(A)\|_2^2}{|p(A)|^2}\right\}\right], \tag{3.3}$$

$$\mathop{\mathbf{E}}_{A}\left[\min\left\{k, \frac{\|\nabla p(A)\|_2^2}{|p(A)|^2}\right\}\right] \leq 72 \cdot \mathop{\mathbf{E}}_{A,B}\left[\min\left\{k, \frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right\}\right]. \tag{3.4}$$

Proof. Note that, for any random variable Z, we have $\mathbf{E}[\min\{k, Z\}] \leq k$ and $\mathbf{E}[\min\{k, Z\}] \leq \mathbf{E}[Z]$. Thus,

$$\mathop{\mathbf{E}}_{A,B}\left[\min\left\{k, \frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right\}\right] \leq \mathop{\mathbf{E}}_{A}\left[\min\left\{k, \mathop{\mathbf{E}}_{B}\left[\frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right]\right\}\right],$$

which implies Equation (3.3).

We now prove Equation (3.4). For a fixed A, let

$$q_A(B) = B \cdot \frac{\nabla p(A)}{|p(A)|} = \frac{\mathsf{D}_B\, p(A)}{|p(A)|}.$$

Note that q_A is a degree-1 polynomial in the variables B with $\|q_A\|_2^2 = \frac{\|\nabla p(A)\|_2^2}{|p(A)|^2}$. Then by Theorem 3.14, we have for every A that

$$\Pr_B\left[\frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2} \geq (1/4) \cdot \frac{\|\nabla p(A)\|_2^2}{|p(A)|^2}\right] = \Pr_B\left[|q_A(B)|^2 \geq (1/4) \cdot \|q_A\|_2^2\right] \geq 1/18. \tag{3.5}$$

Now let

$$X = X(A) = \frac{\|\nabla p(A)\|_2^2}{|p(A)|^2},$$

and

$$Y = Y(A,B) = \frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}.$$

By Equation (3.5), $Y \geq X/4$ with probability at least $1/18$. Next we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{A,B}\left[\min\{k, Y\}\right] &\geq \frac{1}{18} \cdot \mathop{\mathbf{E}}_{A}\left[\mathop{\mathbf{E}}_{B}\left[\min\{k, Y\} \mid Y \geq X/4\right]\right] \\
&\geq \frac{1}{18} \cdot \mathop{\mathbf{E}}_{A}\left[\mathop{\mathbf{E}}_{B}\left[\min\{k, X/4\} \mid Y \geq X/4\right]\right] \\
&\geq \frac{1}{72} \cdot \mathop{\mathbf{E}}_{A}\left[\min\{k, X\}\right],
\end{aligned}$$

where in the last step we dropped the expectation over B since X does not depend on B. □

Invariance principle for polynomials

For a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ and $i \in [n]$, the *influence of coordinate i on f*, denoted $\mathbf{Inf}_i[f]$, is defined as

$$\mathbf{Inf}_i[f] = \mathop{\mathbf{Pr}}_{x \sim \{-1,1\}^n}[f(x) \neq f(x^{\oplus i})],$$

where $x^{\oplus i}$ is x with the ith coordinate x_i replaced with $-x_i$. The *total influence* (also known as average sensitivity) of $f\colon \{-1,1\}^n \to \{-1,1\}$, denoted $\mathbf{Inf}[f]$, is defined as

$$\mathbf{Inf}[f] = \sum_{i=1}^{n} \mathbf{Inf}_i[f].$$

For a function $f\colon \{-1,1\}^n \to \mathbb{R}$, the definition of influence becomes:

$$\mathbf{Inf}_i[f] = \frac{1}{4} \cdot \mathop{\mathbf{E}}_{x \sim \{-1,1\}^n}\left[\left|f(x) - f(x^{\oplus i})\right|^2\right].$$

For the case of multilinear polynomials $p\colon \mathbb{R}^n \to \mathbb{R}$, it can be equivalently expressed as follows:

$$\mathbf{Inf}_i[p] = \left\|\frac{\partial p}{\partial x_i}\right\|_2^2, \tag{3.6}$$

yielding

$$\mathbf{Inf}[p] = \mathbf{E}\left[\|\nabla p(A)\|_2^2\right]. \tag{3.7}$$

The following is a well-known fact about influence; we sketch the proof for completeness.

Theorem 3.19. *For every d-degree multilinear polynomial $p\colon \mathbb{R}^n \to \mathbb{R}$, we have*

$$\mathbf{Var}[p] \leq \mathbf{Inf}[p] \leq d \cdot \mathbf{Var}[p].$$

Proof. For a multilinear polynomial $p(x_1, \ldots, x_n)$ and a set $S \subseteq [n]$, denote by $\hat{p}(S)$ the coefficient of p at the monomial $\prod_{i \in S} x_i$ (the Fourier coefficient of p at S). The proof is obtained from the following (easily verifiable) identities: $\mathbf{Inf}_i[p] = \sum_{S \ni i} \hat{p}(S)^2$, and $\mathbf{Var}[p] = \sum_{\emptyset \neq S \subseteq [n]} \hat{p}(S)^2$. □

Definition 3.20 (τ-regular)**.** *We say that a polynomial p is τ-regular if for all i.*

$$\mathbf{Inf}_i[p] \leq \tau \cdot \mathbf{Var}[p].$$

A polynomial threshold function $f(x) = \mathrm{sgn}(p(x))$ is ε-regular if p is τ-regular.

The following result shows that, for every PTF f, there exists a partitioning of the Boolean cube $\{-1, 1\}^n$ into few sub-cubes so that, on most of these sub-cubes, the PTF f restricted to the sub-cube is either regular or has small variance relative to its L^2 norm.

Theorem 3.21 ([Kan14])**.** *For all $1/4 > \tau, \delta, \varepsilon > 0$ and $\gamma > 0$, every degree-d multilinear polynomial p can be expressed as a decision tree of depth at most*

$$\tau^{-1} \cdot (d \cdot \log \tau^{-1} \cdot \log \delta^{-1})^{O(\gamma \cdot d)} \cdot \log \varepsilon^{-1},$$

so that, with probability at least $1 - \varepsilon$, a random leaf ω (reached from the root of the tree by branching uniformly at random at each internal node) defines a restricted polynomial p_ω (obtained from p by setting the variables on the branch leading to ω to the values specified by the branch) such that the polynomial $p_\omega(y)$ either is τ-regular or satisfies $\mathbf{Var}[p_\omega] \leq (\log \delta^{-1})^{-\gamma \cdot d} \cdot \|p_\omega\|_2^2$.

The following is a version of the Invariance Principle of Mossel et al. [MOO10] in the form that will be convenient for us.

Theorem 3.22 (Invariance principle [Kan14])**.** *Let p and q be two polynomials such that for some $\tau > 0$, $\mathbf{Inf}_i[p], \mathbf{Inf}_i[q] \leq \tau$ for all i and that $\|p + q\|_2, \|p - q\|_2 \geq 1$. Then*

$$\mathbf{Pr}[|p(A)| \leq |q(A)|] = \mathbf{Pr}[|p(X)| \leq |q(Y)|] + O\left(d \cdot \tau^{-1/8d}\right).$$

Corollary 3.23. *For any d-degree τ-regular non-constant multilinear polynomial p, and any $\varepsilon > 0$, we have*

$$\mathbf{Pr}[|p(A)| \leq \varepsilon \cdot |\mathsf{D}_B\, p(A)|] = O\left(d^2 \cdot \varepsilon + d \cdot \tau^{-1/8d}\right).$$

Proof. The idea is to apply the Invariance Principle of Theorem 3.22, and then the strong anticoncentration bound of Theorem 3.17. To this end, we need to argue that the assumptions of these two theorems are satisfied. First, we normalize our polynomial p so that the new polynomial has all influences at most τ.

For the given multilinear polynomial $p(x_1, \ldots, x_n)$, define

$$q(x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{i=1}^{n} y_i \cdot \frac{\partial p}{\partial x_i}, \tag{3.8}$$

which is easily seen to be a multilinear polynomial of degree at most d. Let $\sigma = \sqrt{\mathbf{Var}[p]}$. Since p is a non-constant function, we have $\sigma \neq 0$. Define the normalized polynomial $p' = p/\sigma$. We have $\mathbf{Var}[p'] = 1$, and, by the definition of influence in Equation (3.6), we also have for all $i \in [n]$ that

$$\begin{aligned} \mathbf{Inf}_i[p'] &= \mathbf{Inf}_i[p]/\mathbf{Var}[p] \\ &\leq \tau, \end{aligned} \tag{3.9}$$

since $\mathbf{Inf}_i[p] \leq \tau \cdot \mathbf{Var}[p]$ for all $i \in [n]$ by assumption. By the linearity of differentiation, we also get that $q' = q/\sigma$ is the directional derivative of p'. Thus our task is reduced to upper-bounding the probability

$$\mathbf{Pr}[|p'(A)| \leq \varepsilon \cdot |\mathsf{D}_B\, p'(A)|]. \tag{3.10}$$

To apply the Invariance Principle of Theorem 3.22 to Equation (3.10), we need to upper-bound the influences of q'. For every $i \in [n]$, we get from Equations (3.8) and (3.9) that

$$\begin{aligned} \left\| \frac{\partial q'}{\partial y_i} \right\|_2^2 &= \left\| \frac{\partial p'}{\partial x_i} \right\|_2^2 \\ &= \mathbf{Inf}_i[p'] \\ &\leq \tau. \end{aligned}$$

We also get

$$\begin{aligned} \left\| \frac{\partial q'}{\partial x_i} \right\|_2^2 &= \mathbf{E}\left[\left| \mathsf{D}_B\, \frac{\partial p'}{\partial x_i}(A) \right|^2 \right] \\ &= \mathbf{E}\left[\left\| \nabla \frac{\partial p'}{\partial x_i}(A) \right\|_2^2 \right] && \text{(by Equation (3.2))} \\ &= \mathbf{Inf}\left[\frac{\partial p'}{\partial x_i} \right] && \text{(by Equation (3.7))} \\ &\leq d \cdot \mathbf{Var}\left[\frac{\partial p'}{\partial x_i} \right] && \text{(by Theorem 3.19)} \\ &\leq d \cdot \left\| \frac{\partial p'}{\partial x_i} \right\|_2^2 \\ &= d \cdot \mathbf{Inf}_i[p'] && \text{(by Equation (3.6))} \\ &\leq d\tau. && \text{(by Equation (3.9))} \end{aligned}$$

Thus all of the influences of p' and q' are at most $d\tau$.

Finally, for every $\lambda \in \mathbb{R}$, we have for independent n-dimensional standard Gaussians A and B that

$$\begin{aligned}\|p' + \lambda \cdot q'\|_2^2 &= \mathbf{E}\left[|p'(A) + \lambda \cdot q'(A,B)|^2\right] \\ &= \mathbf{E}[|p'(A)|^2] + \lambda^2 \cdot \mathbf{E}[|q'(A,B)|^2] + (2\lambda) \cdot \mathbf{E}[p'(A) \cdot q'(A,B)]. \end{aligned} \tag{3.11}$$

By Equation (3.8), we get that $\mathbf{E}[p'(A) \cdot q'(A,B)] = 0$. Hence, we get from Equation (3.11) that $\|p' + \lambda \cdot q'\|_2^2 \geq \|p'\|_2^2 \geq \mathbf{Var}[p'] = 1$.

Thus p' and q' satisfy all assumptions of Theorem 3.22, with the influences bounded by $d\tau$. Applying Theorem 3.22 and then Theorem 3.17, we get the required upper bound on the probability in Equation (3.10), concluding the proof. □

Random block restrictions and concentrated polynomials

Definition 3.24 (Random block restriction)**.** *Suppose the variables of a polynomial are arbitrarily partitioned into m blocks. A random block restriction ρ is obtained by the following process:*

1. *Uniformly at random pick a block $\ell \in [m]$.*
2. *Assign each variable that is outside the chosen block ℓ a uniformly random value in $\{-1, 1\}$, independently.*

We use B_m to denote the distribution over all possible restrictions ρ generated by the above process.

We need the following notion of "concentration" for polynomials.

Definition 3.25 (δ-concentrated polynomials)**.** *Let p be a degree-d multilinear polynomials and $f = \mathrm{sgn}(p)$. For a universal constant $L = 192$, and parameters $0 < \delta \leq 1/2$ and $\gamma > 0$, we call p (and f) (δ, γ)-*concentrated *if*

$$\mathbf{Var}[p] \leq \left(L \cdot \log \delta^{-1}\right)^{-\gamma \cdot d} \cdot \|p\|_2^2 .$$

We refer to $(\delta, 1)$-concentrated polynomials as δ-concentrated.

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

Lemma 3.26. *For every degree PTF $f = \mathrm{sgn}(p)$ and every $0 < \delta \leq 1/2$, if p is δ-concentrated, then f is δ^2-close to constant.*

Proof. Let $p' = p - \mu$, where $\mu = \mathbf{E}[p(A)]$, and let $\nu = (L \cdot \log \delta^{-1})^d$ for a constant $L > 0$ to be determined. Since p is δ-concentrated and $\|p\|_2^2 = \mu^2 + \mathbf{Var}[p]$, we get

$$\mu^2 \geq (\nu - 1) \cdot \mathbf{Var}[p] \geq \frac{\nu}{4} \cdot \mathbf{Var}[p],$$

for $L \geq 4/3$. Thus we have

$$|\mu| \geq \frac{\sqrt{\nu}}{2} \cdot \|p'\|_2 . \tag{3.12}$$

Note that for all points $x \in \{-1,1\}^n$ where $|p'(x)| < |\mu|$, we have $\mathrm{sgn}(p(x)) = \mathrm{sgn}(\mu)$. Therefore,

$$\begin{aligned}
\mathbf{Pr}[\mathrm{sgn}(p(A)) \neq \mathrm{sgn}(\mu)] &\leq \mathbf{Pr}[|p'(x)| \geq |\mu|] \\
&\leq \mathbf{Pr}\left[|p'(x)| \geq \frac{\sqrt{\nu}}{2} \cdot \|p'\|_2\right] && \text{(by Equation (3.12))} \\
&\leq \delta^2, && \text{(by Theorem 3.13)}
\end{aligned}$$

where the last inequality holds if we choose $L \geq 32$. □

3.3 Block Restriction Lemma: A simple bound

As a warm-up, we first prove a simpler bound on the probability that under random block restrictions, a degree-d multilinear polynomial does not become concentrated.

Lemma 3.27 (Block Restriction Lemma: Simple Bound)**.** *For any degree-d multilinear polynomial p, and any $m \geq 16$, $\gamma \geq 1$, $0 < \delta \leq 1/16$, we have*

$$\mathop{\mathbf{Pr}}_{\rho \sim B_m}[p_\rho \text{ is not } (\delta,\gamma)\text{-concentrated}] \leq m^{-\frac{1}{8d+1}} \cdot \left(d \cdot \log m \cdot \log \delta^{-1}\right)^{O(\gamma \cdot d)} + 2\delta.$$

3.3.1 Regularization

In this section, we show that it suffices to consider only regular polynomials. We start with the following definition.

Definition 3.28. *Let $\mathcal{P}(d,m,\delta,\gamma)$ be the supremum, over all degree-d multilinear polynomials p and all possible partitions of the variables into m blocks, of the probabilities*

$$\mathop{\mathbf{Pr}}_{\rho \sim B_m}[p_\rho \text{ is not } (\delta,\gamma)\text{-concentrated}] .$$

Let $\mathcal{P}_{\mathrm{reg}}(d,m,\delta,\gamma,\tau)$ be the same as $\mathcal{P}$ but only for τ-regular polynomials. We will use $\mathcal{P}(d,m,\delta)$ (resp. $\mathcal{P}_{\mathrm{reg}}(d,m,\delta,\tau)$) for $\mathcal{P}(d,m,\delta,1)$ (resp. $\mathcal{P}_{\mathrm{reg}}(d,m,\delta,1,\tau)$).

Claim 3.29. *For any $m \geq 1/16$, $\gamma > 0$, and $0 < \delta, \tau \leq 1/4$, we have*

$$\mathcal{P}(d,m,\delta,\gamma) \leq \mathcal{P}_{\mathrm{reg}}(d,m,\delta,\gamma,\tau) + \frac{1}{m} \cdot \tau^{-1} \cdot \left(d \cdot \log \tau^{-1} \cdot \log \delta^{-1}\right)^{O(\gamma \cdot d)} + 2\delta.$$

Proof. Let p be a degree-d multilinear polynomial with its variables petitioned into m blocks. By Theorem 3.21, there exists a decision tree of depth at most

$$H = \tau^{-1} \cdot (d \cdot \log \tau^{-1} \cdot \log \delta^{-1})^{O(\gamma \cdot d)}$$

such that for a random leaf ω of the tree, the restricted polynomial p_ω (obtained from p by fixing the variables on the branch leading to ω, as specified by the branch) is either τ-regular or $(\delta, \gamma + 1)$-concentrated, with probability at least $1 - \delta$. The given decision tree partitions the Boolean cube $\{-1, 1\}^n$ into disjoint regions (sub-cubes) according to the partial restrictions labelling the branches of the tree. Let us call a random restriction ρ *partition-respecting* if it is consistent with some partial restriction labelling one of the branches of the decision tree (i.e., the restriction does not select a block containing any of the variables appearing on the branch, and the assignment to those variables agrees with their corresponding values on the branch). We claim that the probability that a random restriction $\rho \sim B_m$ is *not* partition-respecting is at most H/m.

Indeed, first note that choosing a random restriction $\rho \sim B_m$ is equivalent to first picking a uniformly random assignment to all variables, and then un-assigning the variables in a uniformly random block $i \in [m]$. Picking a uniformly random assignment to all variables is equivalent to picking a random branch in our decision tree (setting some of the variables to constants), and then randomly assigning the remaining variables (not appearing on the branch). For each fixed variable on the branch, the probability that its corresponding block is chosen when we pick a uniformly random block $i \in [m]$ is $1/m$. It follows by the union bound that the overall probability that a random block $i \in [m]$ contains some variable from the given branch is at most H/m.

Thus, at the expense of the additive error term

$$\frac{H}{m} = \frac{1}{m} \cdot \tau^{-1} \cdot \left(d \cdot \log \tau^{-1} \cdot \log \delta^{-1}\right)^{O(\gamma \cdot d)},$$

it suffices to upper-bound the probability

$$\mathop{\mathbf{Pr}}_{\rho \sim B_m} [p_\rho \text{ is not } (\delta, \gamma)\text{-concentrated}]$$

only for partition-respecting restrictions ρ. This probability can be expressed as the expectation over random leaves ω of the decision tree for f of the probability

$$\mathop{\mathbf{Pr}}_{\rho' \sim B_m} [(p_\omega)_{\rho'} \text{ is not } (\delta, \gamma)\text{-concentrated}],$$

where f_ω is the restriction of f to the leaf ω, and ρ' is a random restriction on the variables of f_ω (those not fixed by the branch leading to ω).

Finally, as p_ω is neither τ-regular nor $(\delta, \gamma+1)$-concentrated with probability at most δ over random leaves ω, and a polynomial that is $(\delta, 2)$-concentrated will stay $(\delta, 1)$-concentrated with probability at least $1-\delta$ by Lemma 3.35, we get that

$$\mathcal{P}(d, m, \delta, \gamma) \leq \mathcal{P}_{\mathrm{reg}}(d, m, \delta, \gamma, \tau) + \frac{1}{m} \cdot \tau^{-1} \cdot \left(d \cdot \log \tau^{-1} \cdot \log \delta^{-1}\right)^{O(\gamma \cdot d)} + 2\delta,$$

as required. □

3.3.2 Proof of the simple bound

Given Claim 3.29, it remains to upper-bound the quantity $\mathcal{P}_{\mathrm{reg}}(d, m, \delta, \gamma, \tau)$. Here we show the following.

Lemma 3.30. *There is a constant $c > 0$ such that, for any $m \geq 1/16$, $\gamma \geq 1$, and $0 < \delta, \tau \leq 1/4$, we have*

$$\mathcal{P}_{\mathrm{reg}}(d, m, \delta, \gamma, \tau) \leq O\left(d^2\right) \cdot (c \cdot \log \delta^{-1})^{\gamma \cdot d} \cdot \left(m^{-1/2} + \tau^{1/(8d)}\right).$$

First we prove the following property of non-concentrated polynomials.

Lemma 3.31. *For any $0 < \delta \leq 1/4$ and $\gamma \geq 1$, if a degree-d multilinear polynomial p is not (δ, γ)-concentrated, then*

$$\mathbf{Pr}\left[|\mathsf{D}_B\, p(A)|^2 \geq (1/16) \cdot \left((9L) \cdot \log \delta^{-1}\right)^{-\gamma \cdot d} \cdot |p(A)|^2\right] \geq (1/4) \cdot 9^{-d},$$

where $L > 0$ is the constant from Definition 3.25.

Proof. For the given multilinear polynomial $p(x_1, \ldots, x_n)$, define

$$q(x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{i=1}^{n} y_i \cdot \frac{\partial p}{\partial x_i},$$

which is easily seen to be a multilinear polynomial of degree at most d. Applying Theorem 3.14 to q, we get that

$$\mathbf{Pr}\left[|q(C)|^2 \geq (1/4) \cdot \|q\|_2^2\right] \geq (1/2) \cdot 9^{-d}. \tag{3.13}$$

Next we relate $\|q\|_2^2$ to $\mathbf{Var}[p]$ as follows:

$$\begin{aligned}
\|q\|_2^2 &= \mathbf{E}\left[|\mathsf{D}_B\, p(A)|^2\right] \\
&= \mathbf{E}\left[\|\nabla p(A)\|_2^2\right] && \text{(by Equation (3.2))} \\
&= \mathbf{Inf}[p] && \text{(by Equation (3.7))} \\
&\geq \mathbf{Var}[p]. && \text{(by Theorem 3.19)}
\end{aligned}$$

Together with Equation (3.13), this implies that

$$\mathbf{Pr}\left[|\mathsf{D}_B\, p(A)|^2 \geq (1/4)\cdot \mathbf{Var}[p]\right] \geq (1/2)\cdot 9^{-d}. \tag{3.14}$$

Applying the Markov inequality to p^2, we get

$$\mathbf{Pr}\left[|p(A)|^2 \geq (4\cdot 9^d)\cdot \|p\|_2^2\right] \leq (1/4)\cdot 9^{-d}. \tag{3.15}$$

We conclude from Equations (3.14) and (3.15) that, with probability at least $(1/4)\cdot 9^{-d}$, we have both

$$|\mathsf{D}_B\, p(A)|^2 \geq (1/4)\cdot \mathbf{Var}[p] \tag{3.16}$$

and

$$|p(A)|^2 \leq (4\cdot 9^d)\cdot \|p\|_2^2\,. \tag{3.17}$$

As p is assumed to be (δ,γ)-concentrated, we also have

$$\mathbf{Var}[p] \geq \left(L\cdot \log \delta^{-1}\right)^{-\gamma\cdot d}\cdot \|p\|_2^2\,. \tag{3.18}$$

Combining Equations (3.16) to (3.18) yields the required claim. □

Definition 3.32. *For p a non-zero polynomial, we define*

$$\alpha(p) :\, = \mathbf{E}\left[\min\left\{1, \frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right\}\right].$$

By Lemma 3.31, we get the following.

Corollary 3.33. *There is a constant $c>0$ such that, for any $0<\delta\leq 1/4$ and $\gamma\geq 1$, if a degree-d multilinear polynomial p is not (δ,γ)-concentrated, then*

$$\alpha(p) \geq \left(c\cdot \log \delta^{-1}\right)^{-\gamma\cdot d}.$$

We are now ready to prove Lemma 3.30.

Proof of Lemma 3.30. For a block, ℓ, we let A_ℓ denote the random assignment to the variables in ℓ and let $A_{\bar{\ell}}$ denote the random assignment to the variables that are not in ℓ. Then

by the definition of random block restriction, we have

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{\rho\sim B_m}[p_\rho \text{ is not } (\delta,\gamma)\text{-concentrated}] &= \frac{1}{m}\cdot\sum_{\ell}\mathop{\mathbf{Pr}}_{A_{\bar{\ell}}}[f_{A_{\bar{\ell}}} \text{ is not } (\delta,\gamma)\text{-concentrated}] \\
&\le \frac{1}{m}\cdot\sum_{\ell}\mathop{\mathbf{Pr}}_{A_{\bar{\ell}}}\left[\left(c\cdot\log\delta^{-1}\right)^{\gamma\cdot d}\cdot\alpha(p_{A_{\bar{\ell}}})\ge 1\right] \\
&\qquad\qquad\text{(by Corollary 3.33)} \\
&\le \frac{1}{m}\cdot\sum_{\ell}\mathop{\mathbf{E}}_{A_{\bar{\ell}}}\left[\left(c\cdot\log\delta^{-1}\right)^{\gamma\cdot d}\cdot\alpha(p_{A_{\bar{\ell}}})\right] \\
&= \frac{1}{m}\cdot\left(c\cdot\log\delta^{-1}\right)^{\gamma\cdot d}\cdot\sum_{\ell}\mathop{\mathbf{E}}_{A_{\bar{\ell}}}\left[\alpha\left(p_{A_{\bar{\ell}}}\right)\right]. \qquad (3.19)
\end{aligned}$$

We upper-bound the sum $\sum_\ell \mathbf{E}_{A_{\bar{\ell}}}\left[\alpha\left(p_{A_{\bar{\ell}}}\right)\right]$ in Equation (3.19) as follows:

$$\begin{aligned}
\sum_{\ell}\mathbf{E}\left[\min\left\{1,\frac{|\mathsf{D}_B\, p_{A_{\bar{\ell}}}(A_\ell)|^2}{|p_{A_{\bar{\ell}}}(A_\ell)|^2}\right\}\right] &\le \sum_{\ell}\mathbf{E}\left[\min\left\{1,\frac{\left\|\nabla p_{A_{\bar{\ell}}}(A_\ell)\right\|^2}{|p_{A_{\bar{\ell}}}(A_\ell)|^2}\right\}\right] \qquad \text{(by Lemma 3.18)} \\
&\le \mathbf{E}\left[\min\left\{m,\frac{\|\nabla p(A)\|^2}{|p(A)|^2}\right\}\right] \\
&\le 72\cdot\mathbf{E}\left[\min\left\{m,\frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right\}\right] \qquad \text{(by Lemma 3.18)}
\end{aligned}$$

If p is a constant function, then its directional derivative is always 0, and hence the expectation above becomes 0. Otherwise, for a non-constant p, we upper-bound this expectation by

$$\begin{aligned}
&\mathbf{E}\left[\min\left\{1,\frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right\}\right] + \int_1^m \mathbf{Pr}\left[|p(A)|\le t^{-1/2}\cdot|\mathsf{D}_B\, p(A)|\right]dt \\
&\le 1+\int_1^m O\left(d^2t^{-1/2}+d\tau^{1/(8d)}\right)dt \qquad \text{(by Corollary 3.23)} \\
&\le O(d^2)\cdot\left(\sqrt{m}+m\cdot\tau^{1/(8d)}\right). \qquad (3.20)
\end{aligned}$$

Combining Equations (3.19) and (3.20), we conclude that

$$\mathop{\mathbf{Pr}}_{\rho\sim B_m}[p_\rho \text{ is not } (\delta,\gamma)\text{-concentrated}] \le O\left(d^2\right)\cdot(c\cdot\log\delta^{-1})^{\gamma\cdot d}\cdot\left(m^{-1/2}+\tau^{1/(8d)}\right),$$

as required. □

We can now finish the proof of Lemma 3.27.

Proof of Lemma 3.27. Combining Claim 3.29 and Lemma 3.30, we get

$$\mathcal{P}(d,m,\delta,\gamma) \le O\left(d^2\right)\cdot(c\cdot\log\delta^{-1})^{\gamma\cdot d}\cdot\left(m^{-1/2}+\tau^{1/(8d)}\right) + 2\delta + \frac{1}{m}\cdot\tau^{-1}\cdot\left(d\cdot\log\tau^{-1}\cdot\log\delta^{-1}\right)^{O(\gamma\cdot d)}.$$

Setting $\tau = m^{-\frac{8d}{8d+1}}$, we get the desired bound. □

3.4 Block Restriction Lemma with optimal exponent: Weak version

The bound in Lemma 3.27 has an undesirable dependence on the degree d in the exponent of the interested parameter m. In this section, we prove a better bound that achieves the optimal exponent in m. To illustrate the ideas of the proof techniques, we first prove the following weaker version.

Lemma 3.34 (Block Restriction Lemma: Weak Version)**.** *For any degree-d multilinear polynomial p, and any $m \ge 16$, $\gamma \ge 1$, and $0 < \delta \le 1/16$, we have*

$$\Pr_{\rho\sim B_m}[p_\rho \text{ is not } (\delta,\gamma)\text{-concentrated}] \le \left(m^{-1/2}+\delta\right)\cdot O\left(d\cdot\log m\cdot\log\delta^{-1}\right)^{O\left(\gamma\cdot d^2\cdot\log\log m\right)}.$$

Our road map for the proof is as follows. We set up a recurrence (in Section 3.4.1), reducing the analysis of random restrictions from B_m to that of random restrictions from $B_{m/b}$, for a parameter $b > 0$. Solving this recurrence (in Section 3.4.2) will conclude the proof of Lemma 3.34.

The reason for the recursive analysis is to be able to control the error coming from an application of the invariance principle, Theorem 3.22. That error (of the form $\tau^{1/(8d)}$) has an undesirable dependence on the degree d in the exponent, which would be overwhelming if we try to apply a random block restriction $\rho \sim B_m$ in a single step, as in the proof of Lemma 3.27 in Section 3.3. However, by viewing ρ as a two-step process, where we first apply a random block restriction $\rho_1 \sim B_b$, and then apply another random block restriction $\rho_2 \sim B_{m/b}$, we only need to ensure that the error coming from the invariance principle is small relative to the value of $1/b$ (more precisely, $b^{-1/2}$). By choosing b so that $b^{-1/2}$ is equal to $\tau^{1/(8d)}$, we ensure that the error from the invariance principle is not overwhelming when we reduce from the case of B_m to the case of $B_{m/b}$. Then we repeat this recursive process enough times to get the final bound.

For simplicity, we only prove Lemma 3.34 for $\gamma = 1$. It is easy to modify the proof for any γ.

3.4.1 Setting up the recurrence

By Claim 3.29, we have

$$\begin{aligned}\Pr_{\rho\sim B_m}[p_\rho \text{ is not } \delta\text{-concentrated}] \leq \Pr_{\rho\sim B_m}[q_\rho \text{ is not } \delta\text{-concentrated}] \\ + \frac{1}{m}\cdot\tau^{-1}\cdot\left(d\cdot\log\tau^{-1}\cdot\log\delta^{-1}\right)^{O(d)} + 2\delta,\end{aligned}$$

where q is some τ-regular polynomial of degree at most d.

We now upper bound

$$\Pr_{\rho\sim B_m}[q_\rho \text{ is not } \delta\text{-concentrated}].$$

Consider the following equivalent way of choosing a random block restriction $\rho \sim B_m$. Let $0 < b \leq m$ be an integer parameter.

1. Partition the m blocks of variables of p into b disjoint super-blocks, where each super-block has m/b blocks.
2. Uniformly at random pick a super-block $\ell \in [b]$, and assign each variable that is outside the chosen super-block ℓ a uniformly random value in $\{-1, 1\}$, independently.
3. Uniformly at random pick a block within super-block ℓ, and assign each variable that is outside the chosen block a uniformly random value in $\{-1, 1\}$, independently.

To avoid some technicalities due to divisibility that can be overcome easily by adding dummy blocks, we assume here that m is divisible by b.

Note that step 2 above is an application of random block restriction on b blocks, and step 3 is an application of random block restriction on m/b blocks. Then we have

$$\Pr_{\rho\sim B_m}[q_\rho \text{ is not } \delta\text{-concentrated}] = \Pr_{\rho_1\sim B_b,\rho_2\sim B_{m/b}}[(q_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-concentrated}].$$

Let $E(\rho_1)$ denote the random event that q_{ρ_1} is $(\delta, 2)$-concentrated. By conditioning on this event, we get that the probability above equals

$$\begin{aligned}&\Pr_{\rho_1,\rho_2}[(q_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-concentrated} \mid E(\rho_1)]\cdot\Pr_{\rho_1}[E(\rho_1)] \\ &+ \Pr_{\rho_1,\rho_2}[(q_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-concentrated} \mid \neg E(\rho_1)]\cdot\Pr_{\rho_1}[\neg E(\rho_1)]. \qquad (3.21)\end{aligned}$$

The first summand in Equation (3.21) contains the quantity

$$\Pr_{\rho_1,\rho_2}[(q_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-concentrated} \mid q_{\rho_1} \text{ is } (\delta, 2)\text{-concentrated}]. \qquad (3.22)$$

To bound this quantity, we use the following which says a concentrated polynomial is likely to remain concentrated under random block restrictions.

Lemma 3.35. *For any $m > 0$, if a degree-d multilinear polynomial p is $(\delta, \gamma{+}1)$-concentrated, then*

$$\Pr_{\rho \sim B_m}[p_\rho \text{ is not } (\delta,\gamma)\text{-concentrated}] \leq \delta.$$

Proof. Fix an arbitrary block $\ell \in [m]$. Let S be the set of variables in block ℓ and $\bar{S}$ be the set of variables outside block ℓ (i.e., S is the set of unrestricted variables and $\bar{S}$ is the set of restricted variables). Then we can write

$$p(A_S, A_{\bar{S}}) = q(A_S, A_{\bar{S}}) + r(A_{\bar{S}}) + \mu,$$

where r contains all the monomials in p that only depend on variables in $\bar{S}$, and $\mu = \mathbf{E}[p(A)]$. Also, for $\rho \in \{-1,1\}^{|\bar{S}|}$, let $\mu'(\rho) = r(\rho) + \mu$, and define $Q(\rho) = \|q(A_S, \rho)\|_2^2 = \mathbf{Var}[p_\rho]$. It can be shown (see, e.g., [DSTW14, proof of Lemma 3.8]) that Q is a degree-$2d$ polynomial with

$$\|Q\|_2 \leq 3^d \cdot \sum_{i \in S} \mathbf{Inf}_i[p].$$

Thus, we have

$$\mathbf{Inf}[p] \geq \frac{1}{3^d} \cdot \|Q\|_2 . \tag{3.23}$$

Now let $\nu = \left(L \cdot \log \delta^{-1}\right)^d$, where $L > 0$ is the constant from Definition 3.25. Then we want to show

$$\Pr_\rho \left[\mathbf{Var}[p_\rho] \geq \nu^{-\gamma} \cdot \|p_\rho\|_2^2\right] \leq \delta. \tag{3.24}$$

Note that to show Equation (3.24), it suffices to show

$$\Pr_\rho \left[Q(\rho) \geq \nu^{-\gamma} \cdot |\mu'(\rho)|^2\right] \leq \delta. \tag{3.25}$$

We first prove the following claim.

Claim 3.36. *We have*

$$\Pr_\rho \left[|\mu'(\rho)|^2 < \frac{\nu^{\gamma+1}}{8 \cdot d \cdot 3^d} \cdot \|Q\|_2\right] \leq \delta/2.$$

Proof. We have for all ρ,

$$\begin{aligned} |\mu'(\rho)| &= |\mu + r(\rho)| \\ &\geq |\mu| - |r(\rho)| \\ &\geq \sqrt{(\nu^{\gamma+1} - 1) \cdot \mathbf{Var}[p]} - |r(\rho)| \\ &\geq \sqrt{\frac{\nu^{\gamma+1}}{2} \cdot \mathbf{Var}[p]} - |r(\rho)|, \end{aligned} \tag{3.26}$$

where the third line above is by the assumption that p is $(\delta, \gamma+1)$-concentrated. Also, by Theorem 3.13, we have

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{\rho}\left[|r(\rho)| \geq \sqrt{\frac{\nu^{\gamma+1}}{8}} \cdot \|r\|_2\right] &\leq \mathop{\mathbf{Pr}}_{\rho}\left[|r(\rho)| \geq \sqrt{\frac{\nu}{8}} \cdot \|r\|_2\right] \\
&\leq \exp\left(-(1/4) \cdot \left(\frac{\nu}{8}\right)^{1/d}\right) \\
&\leq \delta/2. \qquad (3.27)
\end{aligned}$$

Combining Equation (3.26) and Equation (3.27), we get that, with probability at least $1 - \delta/2$,

$$\begin{aligned}
|\mu'(\rho)| &\geq \sqrt{\frac{\nu^{\gamma+1}}{2} \cdot \mathbf{Var}[p]} - \sqrt{\frac{\nu^{\gamma+1}}{8}} \cdot \|r\|_2 \\
&= \sqrt{\frac{\nu^{\gamma+1}}{8}} \cdot \left(2 \cdot \sqrt{\mathbf{Var}[p]} - \|r\|_2\right) \\
&\geq \sqrt{\frac{\nu^{\gamma+1}}{8} \cdot \mathbf{Var}[p]} \\
&\geq \sqrt{\frac{\nu^{\gamma+1}}{8 \cdot d} \cdot \mathbf{Inf}[p]} && \text{(by Theorem 3.19)} \\
&\geq \sqrt{\frac{\nu^{\gamma+1}}{8 \cdot d \cdot 3^d} \cdot \|Q\|_2}, && \text{(by Equation (3.23))}
\end{aligned}$$

as desired. □

Therefore, we have

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\rho}\left[Q(\rho) \geq \nu^{-\gamma} \cdot |\mu'(\rho)|^2\right] \\
&\leq \mathop{\mathbf{Pr}}_{\rho}\left[Q(\rho) \geq \frac{\nu^{-\gamma} \cdot \nu^{\gamma+1}}{8 \cdot d \cdot 3^d} \cdot \|Q\|_2\right] + \delta/2 && \text{(by Claim 3.36)} \\
&= \mathop{\mathbf{Pr}}_{\rho}\left[Q(\rho) \geq \frac{\nu}{8 \cdot d \cdot 3^d} \cdot \|Q\|_2\right] + \delta/2 \\
&\leq \delta, && \text{(by Theorem 3.13)}
\end{aligned}$$

which completes the proof of Equation (3.24) and hence the lemma. □

Now by Lemma 3.35, the quantity in Equation (3.22) is at most δ. The second summand in Equation (3.21) is the product of two probabilities, the first of which is the same as for the original problem but with the restriction parameter m/b instead of m, and so can be

analyzed inductively. By our arguments above, we get the recurrence:

$$\mathcal{P}(d,m,\delta) \leq \mathcal{P}(d,m/b,\delta)\cdot\mathcal{P}_{\mathrm{reg}}(d,b,\delta,2,\tau)+3\delta+\frac{1}{m}\cdot\tau^{-1}\cdot\left(d\cdot\log\tau^{-1}\cdot\log\delta^{-1}\right)^{O(d)}. \quad (3.28)$$

Therefore, to reduce from $\mathcal{P}(d,m,\delta)$ to $\mathcal{P}(d,m/b,\delta)$, it remains to bound

$$\mathcal{P}_{\mathrm{reg}}(d,b,\delta,2,\tau).$$

However, by Lemma 3.30, we know that

$$\mathcal{P}_{\mathrm{reg}}(d,b,\delta,2,\tau) \leq O\left(d^2\right)\cdot(c\cdot\log\delta^{-1})^{2d}\cdot\left(b^{-1/2}+\tau^{1/(8d)}\right). \quad (3.29)$$

3.4.2 Solving the recurrence

We are now ready to finish the proof of Lemma 3.34.

Proof of Lemma 3.34. Let $b=\lceil m^{1/(8d)}\rceil$ and $\tau=m^{-1/2}$. Note that $b^{-1/2}\leq m^{-1/(16d)}$. Then by Equation (3.28) and Equation (3.29) we get that

$$\begin{aligned}\mathcal{P}(d,m,\delta) \leq\ & m^{-1/2}\cdot\left(d\cdot\log m\cdot\log\delta^{-1}\right)^{O(d)}\\ &+3\delta+O\left(d^2\right)\cdot\left(c\cdot\log\delta^{-1}\right)^{2d}\cdot m^{-1/(16d)}\cdot\mathcal{P}(d,m/b,\delta). \quad (3.30)\end{aligned}$$

We now show the following:

$$\mathcal{P}(d,m,\delta) \leq (m^{-1/2}+\delta)\cdot(d\cdot\log m\cdot\log\delta^{-1})^{16Ed^2\log\log m}, \quad (3.31)$$

where E is a sufficiently large constant.

We proceed by induction on m. The base case is $m\leq 2^d$. In this case, the right hand side of Equation (3.31) is greater than 1 when E is sufficiently large and Equation (3.31) holds trivially. Now suppose Equation (3.31) holds for all smaller values of m. Let $M=d\cdot\log m\cdot\log\delta^{-1}$. By Equation (3.30), we obtain the recurrence

$$\mathcal{P}(d,m,\delta) \leq (m^{-1/2}+\delta)\cdot M^{E\cdot d}+m^{-1/(16d)}\cdot\mathcal{P}(d,m/b,\delta)\cdot M^{E\cdot d}, \quad (3.32)$$

for a sufficiently large constant E. Then by the induction hypothesis, we get

$$\begin{aligned}&m^{-1/(16d)}\cdot\mathcal{P}(d,m/b,\delta)\cdot M^{E\cdot d}\\ &\leq m^{-1/(16d)}\cdot(2\cdot m^{-1/2+1/(16d)}+\delta)\cdot(d\cdot\log(m/b)\cdot\log\delta^{-1})^{16Ed^2\log\log(m/b)}\cdot M^{E\cdot d}\\ &\leq 2\cdot(m^{-1/2}+\delta)\cdot M^{16Ed^2\log\log(m/b)}\cdot M^{E\cdot d}, \quad (3.33)\end{aligned}$$

where the first inequality above uses the fact that $(m/b)^{-1/2} \leq 2 \cdot m^{-1/2+1/(16d)}$ for $m > 2^d$. Combining Equation (3.32) and Equation (3.33), we have

$$\mathcal{P}(d, m, \delta) \leq (m^{-1/2} + \delta) \cdot 3 \cdot M^{16Ed^2 \log\log(m/b)+E\cdot d}.$$

Note that

$$\begin{aligned}\log\log(m/b) &\leq \log\log(m^{1-1/(8d)}) \\ &= \log(1 - 1/(8d)) + \log\log m \\ &\leq -1/(8d) + \log\log m.\end{aligned}$$

Therefore, when E is sufficiently large, we have

$$3 \cdot M^{16Ed^2 \log\log(m/b)+E\cdot d} \leq 3 \cdot M^{16Ed^2 \log\log(m)-E\cdot d} \leq M^{16Ed^2 \log\log(m)},$$

as required. □

3.5 Block Restriction Lemma with optimal exponent: Strong version

In this section, we prove a stronger version of Lemma 3.34. Note that for $d = 1$, the notation $O(d \cdot \log d)$ below should be interpreted as $O(1)$ (rather than 0).

Lemma 3.37 (Block Restriction Lemma: Strong Version)**.** *For any degree-d multilinear polynomial p, any $m \geq 16$, $\gamma \geq 1$, and $0 < \delta \leq 1/16$, we have*

$$\Pr_{\rho \sim B_m}[p_\rho \textit{ is not } (\delta, \gamma)\textit{-concentrated}] \leq (m^{-1/2} + \delta) \cdot (\log m)^{O(\gamma \cdot d \cdot \log d)} \cdot (\log \delta^{-1})^{O(\gamma \cdot d^2)}.$$

We follow a strategy similar to that in the proof of Lemma 3.34, except we do not bound the number of blocks where the polynomial p restricted to that block has its α function (see Definition 3.32) greater than some fixed threshold $\left(c \cdot \log \delta^{-1}\right)^{-d}$. Using such a rigid threshold for declaring a polynomial not δ-concentrated results in significant losses at each iteration of the recursion. To get an improved analysis, we instead keep track of an upper bound on the expected value of the function $\alpha(p)$, throughout the recursion. Such an upper bound provides a soft measure of the likelihood that the current function is still not δ-concentrated (cf. Corollary 3.33).

Thus, our proof of Lemma 3.37 will be as follows. We first argue (in Section 3.5.1) that it suffices to consider regular polynomials. Then we set up a recurrence (in Section 3.5.2), reducing the case of restrictions from B_m to that of restrictions from $B_{m/b}$, for a parameter $b > 0$. Finally, we solve the recurrence (in Section 3.5.3) to conclude the proof of Lemma 3.37.

As in the previous section, we only show for $\gamma = 1$ and note that the proof works for any γ.

3.5.1 Regularization

We shall modify our earlier definition of $\mathcal{P}$ and $\mathcal{P}_{\mathrm{reg}}$, using the function α from Definition 3.32.

Definition 3.38. *Let $\mathcal{P}(d, m, \delta, a)$ be the supremum, over all degree-d polynomials p with $\alpha(p) \leq a$ and all possible partitions of the variables into m blocks, of the probabilities*

$$\Pr_{\rho \sim B_m}[p_\rho \text{ is not } \delta\text{-concentrated}].$$

Let $\mathcal{P}_{\mathrm{reg}}(d, m, \delta, a, \tau)$ be the same as $\mathcal{P}$ but only for τ-regular polynomials.

We show that the analysis of $\mathcal{P}$ can be reduced to that of $\mathcal{P}_{\mathrm{reg}}$.

Lemma 3.39. *For any real $0 < \tau, \delta < 1/4$ and $a > 0$, integer $m > 4$ and $d, b \geq 1$, we have*

$$\mathcal{P}(d, m, \delta, a) \leq \frac{1}{m} \cdot \tau^{-1} \cdot (d \cdot \log \tau^{-1} \cdot \log \delta^{-1})^{O(d)} + 2\delta + \mathop{\mathbf{E}}_{\aleph}[\mathcal{P}_{\mathrm{reg}}(d, m, \delta, \aleph, \tau)],$$

where $\aleph$ is a non-negative random variable with $\mathbf{E}[\aleph] \leq O(a)$.

Proof. Let p be a degree-d multilinear polynomial with its variables petitioned into m blocks and $\alpha(p) \leq a$. Consider the decision tree given by Theorem 3.21 with $\varepsilon = \delta$ and $\gamma = 2$. Note that the depth of this decision tree is

$$H = \tau^{-1} \cdot (d \cdot \log \tau^{-1} \cdot \log \delta^{-1})^{O(d)}.$$

We view ρ as $\rho = (\ell, \lambda)$, where ℓ is the selected block and λ is an uniform restriction to the variables outside block ℓ. For each leaf ω, let R_ω be the set of random restrictions consistent with the branch leading to ω. As observed above, the probability ξ of choosing a restriction from the complement of $\cup_\omega R_\omega$ is at most H/m. We get

$$\Pr_{\rho \sim B_m}[p_\rho \text{ is not } \delta\text{-concentrated}] \leq (1 - \xi) \cdot \Pr_{\rho \in \cup_\omega R_\omega}[p_\rho \text{ is not } \delta\text{-concentrated}] + \xi. \quad (3.34)$$

By conditioning on $\rho \in R_\omega$, we get that $\mathbf{Pr}_{\rho \in \cup_\omega R_\omega}[p_\rho$ is not δ-concentrated$]$ equals to

$$\sum_\omega \Pr_\rho[p_\rho \text{ is not } \delta\text{-concentrated} \mid \rho \in R_\omega] \cdot \mathbf{Pr}[\rho \in R_\omega \mid \rho \in \cup_\omega R_\omega].$$

Note that the probability of choosing $\rho \in R_\omega$ conditioned on $\rho \in \cup_\omega R_\omega$ is $2^{-\ell_\omega} \cdot (1 - \xi)^{-1}$, where ℓ_ω is the length of the branch leading to ω. Hence, the right-hand side of

Equation (3.34) is at most

$$\mathop{\mathbf{E}}_{\omega}[\mathop{\mathbf{Pr}}_{\rho\in R_\omega}[p_\rho \text{ is not } \delta\text{-concentrated}]] + \xi.$$

Each restriction $\rho = (\ell, \lambda) \in R_\omega$ can be viewed as a restriction of the variables on the branch leading to ω (as specified by the branch) plus an uniform restriction λ' to the remaining variables outside block ℓ. So we can express p_ρ as $(p_\omega)_{\rho'}$, where $\rho' = (\ell, \lambda')$.

Note that ρ' is a random block restriction on m blocks, which comes from the set of those restrictions that chose block ℓ outside at most H blocks containing the variables on the branch leading to ω. The set of all such restrictions ρ' has the probability mass at least $1 - H/m$ within the set of all random block restrictions ρ_ω (which pick block ℓ uniformly at random from the set of all m blocks). Therefore, we can upperbound the expression in Equation (3.34) by

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\omega}[\mathop{\mathbf{Pr}}_{\rho'}[(p_\omega)_{\rho'} \text{ is not } \delta\text{-concentrated}]] + \xi \\
&\leq (1 - H/m)^{-1} \cdot \mathop{\mathbf{E}}_{\omega}[\mathop{\mathbf{Pr}}_{\rho_\omega}[(p_\omega)_{\rho_\omega} \text{ is not } \delta\text{-concentrated}]] + \xi \\
&\leq \mathop{\mathbf{E}}_{\omega}[\mathop{\mathbf{Pr}}_{\rho_\omega}[(p_\omega)_{\rho_\omega} \text{ is not } \delta\text{-concentrated}]] + \xi + 2(H/m),
\end{aligned}$$

where the last inequality uses the fact that $(1-x)^{-1} \leq 1 + 2x$ whenever $0 < x \leq 1/2$.

Thus, we have

$$\mathop{\mathbf{Pr}}_{\rho\sim B_m}[p_\rho \text{ is not } \delta\text{-concentrated}] \leq \mathop{\mathbf{E}}_{\omega}\left[\mathop{\mathbf{Pr}}_{\rho_\omega\sim B_m}[(p_\omega)_{\rho_\omega} \text{ is not } \delta\text{-concentrated}]\right] + \frac{3H}{m}. \quad (3.35)$$

Note that a leaf ω can be in one of the three cases.

1. The polynomial restricted by ω is neither τ-regular nor $(\delta, 2)$-concentrated.
2. The polynomial restricted by ω is $(\delta, 2)$-concentrated.
3. The polynomial restricted by ω is not $(\delta, 2)$-concentrated but τ-regular.

Then the contribution from the ω's in case $i \in [3]$ to the expected value in Equation (3.35) is

$$\sum_{\omega \text{ in case } i} \mathop{\mathbf{Pr}}_{\rho_\omega\sim B_m}[(p_\omega)_{\rho_\omega} \text{ is not } \delta\text{-concentrated}] \cdot \mathbf{Pr}[\omega].$$

By Theorem 3.21, the contribution form those ω's in the first case is at most $\varepsilon = \delta$. If ω is in the second case, then by Lemma 3.35, the probability over ρ_ω that $(p_\omega)_{\rho_\omega}$ is not δ-concentrated is at most δ, so those ω's contribute at most δ. Finally, the contribution from those ω's in the third case is at most

$$\mathop{\mathbf{E}}_{\omega}\left[\mathcal{P}_{\mathrm{reg}}(d, m, \delta, \alpha(p_\omega), \tau)\right].$$

Therefore, we have

$$\mathcal{P}(d,m,\delta,a) \leq \frac{1}{m}\cdot\tau^{-1}\cdot(d\cdot\log\tau^{-1}\cdot\log\delta^{-1})^{O(d)}+2\delta+\mathop{\mathbf{E}}_{\omega}[\mathcal{P}_{\mathrm{reg}}(d,m,\delta,\alpha(p_\omega),\tau)].$$

To complete the proof, we need to show that

$$\mathop{\mathbf{E}}_{\omega}[\alpha(p_\omega)] \leq O\left(a\right).$$

We have

$$\begin{aligned}
\mathop{\mathbf{E}}_{\omega}[\alpha(p_\omega)] &= \mathop{\mathbf{E}}_{\omega}\left[\mathop{\mathbf{E}}_{A,B}\left[\min\left\{1,\frac{|\mathsf{D}_B\, p_\omega(A)|^2}{|p_\omega(A)|^2}\right\}\right]\right] \\
&\leq \mathop{\mathbf{E}}_{\omega}\left[\mathop{\mathbf{E}}_{A}\left[\min\left\{1,\frac{\|\nabla p_\omega(A)\|^2}{|p_\omega(A)|^2}\right\}\right]\right] && \text{(by Lemma 3.18)} \\
&\leq \mathbf{E}\left[\min\left\{1,\frac{\|\nabla p(A)\|^2}{|p(A)|^2}\right\}\right] \\
&\leq 72\cdot\mathbf{E}\left[\min\left\{1,\frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right\}\right] && \text{(by Lemma 3.18)} \\
&\leq 72\cdot a,
\end{aligned}$$

as required. □

3.5.2 Setting up the recurrence

We show the following recurrence relation for regular polynomials.

Lemma 3.40. *For any real $0<\tau,\delta<1/4$ and $a>0$, integer $m>4$ and $d,b\geq 1$, we have*

$$\mathcal{P}_{\mathrm{reg}}(d,m,\delta,a,\tau) \leq \mathop{\mathbf{E}}_{\aleph}[\mathcal{P}(d,m/b,\delta,\aleph)],$$

where $\aleph$ is a non-negative random variable with $\mathbf{E}[\aleph]=O\left(d^3ab^{-1/2}+d^4\tau^{1/(8d)}\right)$.

We shall need the following analogue of the function $\alpha(p)$ for the Gaussian case.

Definition 3.41. *For p a non-zero polynomial, we define*

$$\beta(p)\colon = \mathbf{E}\left[\min\left\{1,\frac{|\mathsf{D}_Y\, p(X)|^2}{|p(X)|^2}\right\}\right].$$

The functions $\alpha(p)$ and $\beta(p)$ are related in the following way.

Lemma 3.42. *Let p be a degree-d, τ-regular, non-zero polynomial. Then*

$$\beta(p)=\alpha(p)+O\left(d\cdot\tau^{1/(8d)}\right).$$

Proof. We have

$$\begin{aligned}
\beta(p) &= \mathbf{E}\left[\min\left\{1, \frac{|\mathsf{D}_Y\, p(X)|^2}{|p(X)|^2}\right\}\right] \\
&= \int_0^1 \mathbf{Pr}\left[|p(X)| \le t^{-1/2} \cdot |\mathsf{D}_Y\, p(X)|\right] dt \\
&= \int_0^1 \mathbf{Pr}\left[|p(A)| \le t^{-1/2} \cdot |\mathsf{D}_B\, p(A)|\right] dt + O\left(d \cdot \tau^{1/(8d)}\right) \qquad \text{(by Theorem 3.22)} \\
&= \alpha(p) + O\left(d \cdot \tau^{1/(8d)}\right),
\end{aligned}$$

as required. □

In Lemma 3.40, we want to keep track of $\alpha(p)$ in the recurrence, and so we need a version of the anticoncentration bound that takes $\alpha(p)$ into account. This is achieved by the following version of Theorem 3.17 that takes $\beta(p)$ into account.

Theorem 3.43 ([Kan14])**.** *For any d-degree polynomial p and any $0 < \varepsilon < 1$, we have*

$$\mathbf{Pr}\left[|p(X)| \le \varepsilon \cdot |\mathsf{D}_Y\, p(X)|\right] = O\left(d^3 \beta(p) \varepsilon\right).$$

We get the following.

Corollary 3.44. *For any d-degree τ-regular non-constant multilinear polynomial p, and any $\varepsilon > 0$, we have*

$$\mathbf{Pr}[|p(A)| \le \varepsilon \cdot |\mathsf{D}_B\, p(A)|] = O\left(d^3 \cdot \varepsilon \cdot \beta(p) + d \cdot \tau^{1/(8d)}\right).$$

Proof. The proof is the same as that of Corollary 3.23, with Theorem 3.43 replacing Theorem 3.17. □

We are now ready to prove Lemma 3.40.

Proof of Lemma 3.40. Let p be a τ-regular, degree-d, multilinear polynomial with $\mathbf{Var}[p(x)] = 1$ and $\alpha(p) \le a$. Consider the way of choosing a random block restriction as described in Section 3.4.1. Recall that ρ_1 is a random block restriction on b blocks and ρ_2 is a random block restriction on m/b blocks. Then for any arbitrary partition of the variables into m blocks, we have

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{\rho \sim B_m}[p_\rho \text{ is not } \delta\text{-concentrated}] &= \mathop{\mathbf{Pr}}_{\rho_1 \sim B_b, \rho_2 \sim B_{m/b}}[(p_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-concentrated}] \\
&= \mathop{\mathbf{E}}_{\rho_1}\left[\mathop{\mathbf{Pr}}_{\rho_2}[(p_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-concentrated}]\right] \\
&\le \mathop{\mathbf{E}}_{\rho_1}\left[\mathcal{P}(d, m/b, \delta, \alpha(p_{\rho_1}))\right].
\end{aligned}$$

Therefore, it suffices to show that

$$\mathop{\mathbf{E}}_{\rho_1 \sim B_b}[\alpha(p_{\rho_1})] = O\left(d^3 a b^{-1/2} + d^4 \tau^{1/(8d)}\right). \tag{3.36}$$

Note that

$$\mathop{\mathbf{E}}_{\rho_1 \sim B_b}[\alpha(p_{\rho_1})] = \frac{1}{b} \cdot \sum_{\ell} \mathop{\mathbf{E}}_{A_{\bar{\ell}}}[\alpha(p_{A_{\bar{\ell}}})],$$

where $A_{\bar{\ell}}$ is a random assignment to the variables that are not in block ℓ. From the calculation in Lemma 3.30 (Equation (3.20)), we have

$$\begin{aligned}
\sum_{\ell} \mathop{\mathbf{E}}_{A_{\bar{\ell}}}[\alpha(p_{A_{\bar{\ell}}})] &= \mathbf{E}\left[\min\left\{1, \frac{|\mathsf{D}_B\, p(A)|^2}{|p(A)|^2}\right\}\right] + \int_1^b \mathbf{Pr}\left[|p(A)| \le t^{-1/2} \cdot |\mathsf{D}_B\, p(A)|\right] dt \\
&= \alpha(p) + \int_1^b O\left(d^3 \beta(p) t^{-1/2} + d\tau^{1/(8d)}\right) dt && \text{(by Corollary 3.44)} \\
&= \alpha(p) + O\left(d^3 \beta(p) \sqrt{b} + bd\tau^{1/(8d)}\right) \\
&= O\left(d^3 \alpha(p) \sqrt{b} + bd^4 \tau^{1/(8d)}\right) && \text{(by Lemma 3.42)} \\
&= O\left(d^3 a \sqrt{b} + bd^4 \tau^{1/(8d)}\right),
\end{aligned}$$

as required. □

3.5.3 Solving the recurrence

Since $\alpha(p) \le 1$ by definition, we have $\mathcal{P}(d, m, \delta) = \mathcal{P}(d, m, \delta, 1)$. Thus, to prove Lemma 3.37, it suffices to prove the following stronger result, and apply it with $a = 1$.

Theorem 3.45. *There is a constant $B > 0$ such that, for any $d > 0$, $m \ge 16$, $0 < \delta \le 1/16$ and $0 < a \le 1$, we have*

$$\mathcal{P}(d, m, \delta, a) \le (a \cdot m^{-1/2} + \delta) \cdot (\log m)^{B \cdot d \cdot \log d} \cdot (\log \delta^{-1})^{B \cdot d^2}. \tag{3.37}$$

Proof. First we argue that, for a sufficiently large constant $B > 0$, we may assume that a and m are relatively large.

Claim 3.46. *For a sufficiently large constant $B > 0$, we may assume that both*

$$a \ge (c \cdot \log \delta^{-1})^{-2d}, \tag{3.38}$$

and

$$m^{1/(32 \cdot d)} \ge (c \cdot \log \delta^{-1})^{2d}, \tag{3.39}$$

where $c > 0$ is the constant from Corollary 3.33.

Proof of Claim 3.46. If Equation (3.38) is false, then, by Corollary 3.33, the given polynomial is $(\delta, 2)$-concentrated, and by Lemma 3.35, the probability that its random block restriction is not δ-concentrated is at most δ, and so Equation (3.37) is satisfied. Next, assume Equation (3.38), and suppose that Equation (3.39) is false. Then we get that $a \cdot m^{-1/2} > (\log \delta^{-1})^{-T \cdot d^2}$, for some constant $T > 0$, implying that $a \cdot m^{-1/2} \cdot (\log \delta^{-1})^{B \cdot d^2} > 1$, for $B \geq T + 1$. Hence, the right-hand side of Equation (3.37) is greater than 1 in this case, and so Equation (3.37) holds. □

By Claim 3.46, we can assume for the rest of the proof that Equations (3.38) and (3.39) both hold.

Claim 3.47. *There is a constant $E > 0$ such that, for any $m \geq 16$ and $0 < \delta \leq 1/16$, we have*

$$\mathcal{P}(d, m, \delta, a) \leq \frac{1}{2} \cdot (a \cdot m^{-1/2} + \delta) \cdot (\log m)^{E \cdot d \cdot \log d} \cdot (\log \delta^{-1})^{E \cdot d^2} + \mathop{\mathbf{E}}_{\aleph}[\mathcal{P}(d, m/\lceil m^{1/(16d)} \rceil, \delta, \aleph)], \tag{3.40}$$

where $\aleph$ is a non-negative random variable with $\mathbf{E}[\aleph] = O\left(d^4 \cdot a \cdot m^{-1/(32d)}\right)$.

Proof. Let $\tau = m^{-1/2}$ and $b = \lceil m^{1/(16d)} \rceil$. By Lemma 3.39, we get that

$$\mathcal{P}(d, m, \delta, a) \leq m^{-1/2} \cdot (d \cdot \log m \cdot \log \delta^{-1})^{O(d)} + 2\delta + \mathop{\mathbf{E}}_{\aleph_1}[\mathcal{P}_{\mathrm{reg}}(d, m, \delta, \aleph_1, m^{-1/3})], \tag{3.41}$$

for some non-negative random variable $\aleph_1$ with $\mathbf{E}[\aleph_1] \leq O(a)$. By Equation (3.38), we get

$$m^{-1/2} \cdot (d \cdot \log m \cdot \log \delta^{-1})^{O(d)} + 2\delta \leq \frac{1}{2} \cdot (a \cdot m^{-1/2} + \delta) \cdot (\log m)^{E \cdot d \cdot \log d} \cdot (\log \delta^{-1})^{E \cdot d^2}, \tag{3.42}$$

for a sufficiently large constant $E > 0$. Next, by Lemma 3.40, we get

$$\begin{aligned} \mathop{\mathbf{E}}_{\aleph_1}[\mathcal{P}_{\mathrm{reg}}(d, m, \delta, \aleph_1, m^{-1/2})] &\leq \mathop{\mathbf{E}}_{\aleph_1}\left[\mathop{\mathbf{E}}_{\aleph_2}[\mathcal{P}(d, m/b, \delta, \aleph_2)]\right] \\ &\leq \mathop{\mathbf{E}}_{\aleph}[\mathcal{P}(d, m/b, \delta, \aleph)], \end{aligned} \tag{3.43}$$

for some non-negative random variable $\aleph$ with

$$\begin{aligned} \mathbf{E}[\aleph] &= O\left(d^3 \cdot a \cdot b^{-1/2} + d^4 \cdot \tau^{1/(8d)}\right) \\ &\leq O\left(d^3 \cdot a \cdot m^{-1/(32d)} + d^4 \cdot m^{-1/(16d)}\right) \\ &\leq O\left(d^4 \cdot a \cdot m^{-1/(32d)}\right). \qquad \text{(by Equations (3.38) and (3.39))} \end{aligned}$$

Equations (3.42) and (3.43) imply Equation (3.40). □

We now prove Theorem 3.45 by induction on m. We start with the base case $m \leq 2^d$. By Equation (3.38), we only need to consider $a \geq (c \cdot \log \delta^{-1})^{-2d}$. Note that in this case, the

bound in Theorem 3.45 is greater 1 when B is sufficiently large. Now suppose Theorem 3.45 holds for all smaller values of m. Let $M = (\log m)^{Bd\log(d)} \cdot (\log \delta^{-1})^{Bd^2}$ for $B > E$ to be determined, where E is the constant in Claim 3.47. By Claim 3.47, we have

$$\mathcal{P}(d, m, \delta, a) \leq \frac{1}{2} \cdot (a \cdot m^{-1/2} + \delta) \cdot M + \mathop{\mathbf{E}}_{\aleph}[\mathcal{P}(d, m/\lceil m^{1/(16d)} \rceil, \delta, \aleph)],$$

where $\aleph$ is a non-negative random variable with $\mathbf{E}[\aleph] = C \cdot d^4 \cdot a \cdot m^{-1/(32d)}$ and C is some constant. Then by the induction hypothesis, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\aleph}[\mathcal{P}(d, m/\lceil m^{1/(16d)} \rceil, \delta, \aleph)] \\
&\leq \mathop{\mathbf{E}}_{\aleph}\left[(\aleph \cdot 2 \cdot m^{-1/2+1/(32d)} + \delta) \cdot (\log m^{1-1/(16d)})^{Bd\log(d)} \cdot (\log \delta^{-1})^{Bd^2}\right] \\
&\leq 2 \cdot (\mathop{\mathbf{E}}_{\aleph}[\aleph] \cdot m^{-1/2+1/(32d)} + \delta) \cdot (1 - 1/(16d))^{Bd\log(d)} \cdot M \\
&\leq 2 \cdot (C \cdot d^4 \cdot a \cdot m^{-1/(32d)} \cdot m^{-1/2+1/(32d)} + \delta) \cdot (1 - 1/(16d))^{Bd\log(d)} \cdot M \\
&= 2 \cdot C \cdot d^4 \cdot (1 - 1/(16d))^{Bd\log(d)} \cdot (a \cdot m^{-1/2} + \delta) \cdot M \\
&\leq 2 \cdot C \cdot d^4 \cdot e^{-B\log(d)/16} \cdot (a \cdot m^{-1/2} + \delta) \cdot M \\
&\leq \frac{1}{2} \cdot (a \cdot m^{-1/2} + \delta) \cdot M,
\end{aligned}$$

where the last inequality holds if B is sufficiently large. □

Note that the only reason why we have the factor $(\log \delta^{-1})^{O(d^2)}$ rather than $(\log \delta^{-1})^{O(d)}$ in Equation (3.37) is to justify the assumption in Equation (3.39). If we assume this condition explicitly, then we get the following slightly stronger version of the PTF restriction lemma for δ not too small.

Lemma 3.48. *For any degree-d multilinear p, any $m \geq 16$, and any $0 < \delta < 1/16$ such that $m^{1/(64 \cdot d)} \geq (c \cdot \log \delta^{-1})^d$ for the constant $c > 0$ from Corollary 3.33, we have*

$$\Pr_{\rho \sim B_m}[p_\rho \text{ is not } \delta\text{-concentrated}] \leq (m^{-1/2} + \delta) \cdot (\log m)^{O(d\log d)} \cdot (\log \delta^{-1})^{O(d)}.$$

3.6 Applications

3.6.1 Lower bounds for depth-2 circuits with PTF gates

Here we generalize the gate and wire complexity lower bound of [KW16] for Andreev's function against depth-2 circuits with LTF gates, to depth-2 circuits with degree-d PTF gates, for any $d \geq 1$. Our lower bounds match those of [KW16] for the case of $d = 1$ (up to polylogarithmic factors), and extend to any degree $d \ll \sqrt{(\log n)/(\log \log n)}$.

The main technical tool used by [KW16] was a restriction lemma saying, roughly, that an n-variate LTF function hit by some "structured" random restriction that leaves $(\log n)$ variables, will become a constant function except with probability $(\log n)/\sqrt{n}$. This restriction

lemma is then combined with a careful counting argument to show that Andreev's function requires depth-2 LTF circuits with at least $\Omega(n^{1.5}/(\log^3 n))$ gates, and $\Omega(n^{2.5}/(\log^{7/2} n))$ wires.

The restriction lemma of [KW16] is proved using the Littlewood-Offord lemma from additive combinatorics [LO43]. It is not clear how to prove a similar restriction lemma for higher degree $d > 1$ using the same tools. However, we show that our Block Restriction Lemma yields such a generalization for any $1 \leq d \ll \sqrt{(\log n)/(\log\log n)}$. The reason is that we can make the parameter δ in Lemma 3.37 very small compared to the number of unrestricted variables so that, for the restricted function being δ-close to constant is the same as being constant. We start by proving the following restriction lemma for PTFs.

Lemma 3.49. *Let f be any n-variate degree-d PTF. Let $\mathscr{P}$ be a partition of $[n]$ into parts of equal-sized with $|\mathscr{P}| \leq n/16$, and let $\mathscr{R}_{\mathscr{P}}$ be the distribution on restrictions $\rho : [n] \to \{-1, 1, *\}$ that randomly selects one variables from each part of $\mathscr{P}$ and restricts all other variables uniformly at random. Then*

$$\Pr_{\rho\sim\mathscr{R}_{\mathscr{P}}}[f_\rho \text{ is not a constant}] \leq \frac{1}{\sqrt{n}} \cdot (|\mathscr{P}| \cdot \log n)^{O(d^2)}. \tag{3.44}$$

Moreover, if f depends on at most w of its inputs, then

$$\Pr_{\rho\sim\mathscr{R}_{\mathscr{P}}}[f_\rho \text{ is not a univariate function}] \leq \frac{1}{n^{3/2}} \cdot w \cdot (|\mathscr{P}| \cdot \log n)^{O(d^2)}. \tag{3.45}$$

Proof. Consider the following equivalent way of choosing a random restriction $\rho \sim \mathscr{R}_{\mathscr{P}}$.

1. Create $m = \frac{n}{|\mathscr{P}|}$ blocks. For each part in $\mathscr{P}$, randomly assign the $\frac{n}{|\mathscr{P}|}$ variables in the part to the m blocks so that each block takes exactly one of the variables from the part.

2. Apply a random block restriction $\rho' \sim B_m$ based on the partition in the previous step.

By Lemma 3.37 and Lemma 3.26, for any partition into blocks generated in the first step above, the probability over the restrictions in the second step that the restricted PTF is not δ-close to constant is at most

$$\left(\sqrt{\frac{|\mathscr{P}|}{n}} + \delta\right) \cdot \left(\log \frac{n}{|\mathscr{P}|} \cdot \log \delta^{-1}\right)^{O(d^2)}. \tag{3.46}$$

Now let $\delta = \min\left\{2^{-(|\mathscr{P}|+1)}, \sqrt{\frac{|\mathscr{P}|}{n}}\right\}$. In this case, the restricted function, which is on $|\mathscr{P}|$ variables and δ-close to constant, is indeed a constant. Note that for such δ, Equation (3.46) implies Equation (3.44).

Next, for each wire $i \in [w]$, define the following random event E_i: the function f_ρ depends on wire i and on some other wire. Note that if E_i happens, then wire i is assigned

$*$ by ρ which happens with probability $|\mathscr{P}|/n$, and that both $(f_\rho)_{w_i=-1}$ and $(f_\rho)_{w_i=1}$ are non-constant functions. It is not hard to see that given wire i is assigned $*$, the probability that $(f_\rho)_{w_i=-1}$ (or $(f_\rho)_{w_i=1}$) is non-constant is

$$\mathop{\mathbf{E}}_{\rho_1}[\mathop{\mathbf{Pr}}_{\rho_2}[((f_{\rho_1})_{w=-1})_{\rho_2} \text{ is not a constant}], \tag{3.47}$$

where ρ_1 is a random partial assignment to the wires (except wire i) in the part that contains wire i, and ρ_2 is a restriction that randomly selects one variable from each of the rest $|\mathscr{P}|-1$ parts and fixes all other variables uniformly at random. Then the inner probability in Equation (3.47) can be upperbounded by Equation (3.44).

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\rho\sim\mathscr{R}_{\mathscr{P}}}[f_\rho \text{ is not a univariate function}] \\
&= \mathbf{Pr}\left[\vee_{i=1}^{w} E_i\right] \\
&\leq \sum_{i=1}^{w} \mathbf{Pr}[E_i] \\
&\leq w \cdot \frac{|\mathscr{P}|}{n} \cdot 2 \cdot \frac{1}{\sqrt{n}} \cdot ((|\mathscr{P}|-1)\cdot \log n)^{O(d^2)},
\end{aligned}$$

implying Equation (3.45). □

To simplify the presentation, we only argue the worst-case lower bound; a correlation bound as in [KW16] can also be proved in a similar way. We prove a lower bound for the Andreev's function.

Definition 3.50. *Define Andreev's function* $A_n\colon \{-1,1\}^{5n} \to \{-1,1\}$ *as follows:*

$$A_n(x_1,\ldots,x_{4n},y_1,\ldots,y_n) = x_i,$$

where $i \in [4n]$ *is a positive integer uniquely given by the binary string* $z \in \{-1,1\}^{\log 4n}$ *obtained as follows: partition* $[n]$ *into* $\log 4n$ *parts so that each part has* $t = \frac{n}{\log 4n}$ *variables, and the* j*-th part* P_j *is the set* $\{y_{(j-1)\cdot t+k}\colon k=1,\ldots,t\}$*. Then* $z_j = \prod_{y_k\in P_j} y_k$*.*

For simplicity, we assume here that n is divisible by $\log 4n$. We show the following gate and wire lower bounds for A_n against depth-2 circuits with d-degree PTF gates; for $d=1$, these bounds match those of [KW16], up to polylogarithmic factors.

Theorem 3.51 (Lower bounds for depth-2 degree-d PTF circuits)**.** *Every depth-2 circuit on* n *inputs and degree-*d *PTF gates, that computes* A_n *must have at least* $\left(n^{\frac{1}{2}+\frac{1}{d}}\right)\cdot(\log n)^{-O(d^2)}$ *gates, and at least* $\left(n^{\frac{3}{2}+\frac{1}{d}}\right)\cdot(\log n)^{-O(d^2)}$ *wires.*

For the proof of Theorem 3.51, we shall need the following straightforward generalization to degree-d PTFs of the result in [RSO94] about the number of LTFs on s inputs, where each

input is some Boolean function of n variables; the latter result is in turn a generalization of [Win61, Cho61].

Theorem 3.52 ([RSO94])**.** *For any degree-d PTF g on s variables, and any collection of Boolean functions $f_1, \ldots, f_s \colon \{-1,1\}^n \to \{-1,1\}$, the n-variate Boolean function*

$$h(\vec{x}) = g(f_1(\vec{x}), \ldots, f_s(\vec{x}))$$

where $\vec{x} = (x_1, \ldots, x_n) \in \{-1,1\}^n$, can be completely specified using $O(s^d \cdot n)$ bits.

As an immediate consequence of Theorem 3.52, we get the following.

Corollary 3.53. *Every depth-2 circuit on n inputs with s degree-d PTF gates can be completely specified using at most $O\left((s+n)^d \cdot n + (s+n) \cdot n^{d+1}\right)$ bits.*

Now we are ready to complete the proof of Theorem 3.51.

Proof of Theorem 3.51. For an arbitrary $\vec{a} = (a_1, \ldots, a_{4n}) \in \{-1,1\}^{4n}$, let

$$F(y_1, \ldots, y_n) = A_n(a_1, \ldots, a_{4n}, y_1, \ldots, y_n).$$

Towards contradiction, suppose that A_n, and hence also F, is computable by a depth-2 circuit with degree-d PTF gates of wire or gate complexity less than the bounds claimed in the theorem statement (for sufficiently large constants in the $O(d^2)$ exponents of the polylog factors). Let $\mathscr{P}$ be the partition of $[n]$ into $\log 4n$ parts of equal size as specified in Definition 3.50. We then apply a random restriction $\rho \sim \mathscr{R}_{\mathscr{P}}$ to the function $F(y_1, \ldots, y_n)$. Then F_ρ can be used to reconstruct, in the information-theoretic sense (say, in the sense of Kolmogorov complexity) the string $\vec{a}$ (by the definition of A_n). More precisely, to reconstruct $\vec{a}$, it suffices to know the restriction ρ plus the description of some circuit computing F_ρ. The restriction ρ can be described using at most $2n$ bits (by specifying for each $i \in [n]$ whether it is 1, -1, or unrestricted). Next we bound the size of a circuit computing F_ρ, for some ρ satisfying the above condition.

By Lemma 3.49, the expected number of bottom PTF gates of the depth-2 circuit computing $F_\rho(y_1, \ldots, y_n)$ is at most $s_0 = n^{1/d}/(\log n)^{O(d^2)}$ (if either the number of gates or the number of wires of F is small). By the Markov inequality, the probability over $\rho \sim \mathscr{R}_{\mathscr{P}}$ that the actual number s of gates of the circuit for F_ρ is more than $2 \cdot s_0$ is at most $1/2$.

It follows that with probability at least $1/2$, we get a random restriction $\rho \sim \mathscr{R}_{\mathscr{P}}$ such that F_ρ has at most $2 \cdot s_0$ gates. By Corollary 3.53, the circuit for F_ρ is described with at most n bits.

We conclude that every $\vec{a} \in \{-1,1\}^{4n}$ can be described with at most $2n + n = 3n$ bits. However, by a simple counting argument, we know that almost all $4n$-bit strings $\vec{a}$ require the description size strictly greater than $3n$. A contradiction. □

3.6.2 Lower bounds for depth-3 circuits with PTF gates

Here we generalize the lower bound of [KW16] against circuits that are majority votes of depth-2 LTF circuits, to majority votes of depth-2 circuits with degree-d PTF gates. In [KW16], it was shown that there exists a polynomial time function that requires circuits of the form mentioned above with $\Omega\left(n^{2.5}/(\log^{7/2} n)\right)$ wires. Here, we show a lower bound against circuits that can have sub-exponential size as long as the total fan-in of the bottom layer gates is small.

We first define a generalized Andreev function. Recall that a (ζ, L)-list-decodable code is a function $K\colon \{-1,1\}^k \to \{-1,1\}^n$ that maps k-bits messages to n-bits codewords such that for any codeword $y \in \{-1,1\}^n$, there are at most L codewords in the range of K that have relative hamming distance within ζ from y. We will use the following list-decodable code (see, e.g., [CKK+15] for its construction).

Theorem 3.54. *For any given $0 < \varepsilon < 1$, there exists a binary code K mapping $4n$-bit message to a codeword of length 2^{n^ε}, such that K is (ζ, L)-list-decodable for $\zeta = 1/2 - O\left(2^{-n^\varepsilon/4}\right)$ and $L \leq O\left(2^{n^\varepsilon/2}\right)$. Furthermore, there is a polynomial-time algorithm for computing $K(x)$ in position z, for any inputs $x \in \{-1,1\}^{4n}$ and $z \in \{-1,1\}^{n^\varepsilon}$.*

Definition 3.55. *Let $0 < \varepsilon < 0$. Define the function $B_{n,\varepsilon}\colon \{-1,1\}^{5n} \to \{-1,1\}$ as follows:*

$$B_{n,\varepsilon}(x_1, \ldots, x_{4n}, y_1, \ldots, y_n) = K(x)_i,$$

where K is the code from Theorem 3.54, and $i \in [2^{n^\varepsilon}]$ is a positive integer uniquely given by the binary string $z \in \{-1,1\}^{n^\varepsilon}$ obtained as follows: partition $[n]$ into n^{ε/d^2} parts so that each part has $t = \frac{n}{n^\varepsilon}$ variables, and the j-th part P_j is the set $\{y_{(j-1)\cdot t+k}\colon k = 1, \ldots, t\}$. Then $z_j = \prod_{y_k \in P_j} y_k$.

Note that the function above is polynomial-time computable since we can compute $K(x)$ in position i in polynomial time.

We are now ready to prove the lower bound.

Theorem 3.56. *For any $\frac{1}{\log n} \ll \varepsilon < 1$, let C be a majority vote of depth-2 circuits with degree-d PTF gates such that the top majority gate has fanin at most 2^{n^ε} and the total fanin of the gates on the bottom layer at most $w = \left(n^{\frac{3}{2}+\frac{1}{d}}\right) \cdot (n^\varepsilon \cdot \log n)^{-c\cdot d^2}$, where c is a constant. Then C cannot compute $B_{n,\varepsilon}$.*

Proof. Let $a \in \{-1,1\}^{4n}$ be a string with Kolmogorov complexity at least $4n$, and let

$$F(y_1, \ldots, y_n) = B_{n,\varepsilon}(a_1, \ldots, a_{4n}, y_1, \ldots, y_n).$$

Let D be an arbitrary depth-2 circuit in n^ε variables with degree-d PTF gates, of size at most $s_0 = n^{1/d - O\left(\varepsilon \cdot d^2\right)}$. Note that by Corollary 3.53, D can be described with at most n

bits. Let $\mathscr{P}$ be the partition of $[n]$ into n^ε parts of equal size as specified in Definition 3.55. We claim that for any $\rho \sim \mathscr{R}_{\mathscr{P}}$,

$$\mathbf{Corr}(F_\rho, D) < 2^{-n^\varepsilon}.$$

Toward a contradiction, suppose D agrees with $F\rho$ on at least $1/2 + 2^{-n^\varepsilon}$ of the inputs for some ρ. Then we can recover a as follows. We first use the circuit D and the string ρ to compute the corrupted codeword K' such that K' and $K(a)$ have relative hamming distance at most $1/2 - 2^{-n^\varepsilon}$. We then list-decode K' to obtain a list of $L \leq O\left(2^{n^\varepsilon/2}\right)$ codewords, which must contain $K(a)$. Finally, we use an index string of length at most $\log(L)$ to get $K(a)$ from the list of codewords and recover a. This shows that we can use fewer than $4n$ bits to describe a, which contradicts the assumption that a has Kolmogorov complexity at least $4n$.

Next, let C_a be the circuit obtained from C by setting the first $4n$ variables to be a, and let $\rho_0 \sim \mathscr{R}_{\mathscr{P}}$ be a restriction such that $(C_a)_{\rho_0}$ has at most s_0 gates on the bottom layer. The existence of such a restriction is guaranteed by Equation (3.45), when the total fanin of the bottom layer gates is at most w and c is sufficiently large. By the nature of majority function and a simple averaging argument, we know that $(C_a)_{\rho_0}$ must have correlation at least 2^{-n^ε} with one of its sub-circuits, which is a depth-2 circuit in n^ε variables with degree-d PTF gates, of size at most s_0. Thus, we conclude that C_a cannot compute F. □

3.6.3 Lower bounds for constant-depth circuits with PTF gates

Here we extend the wire complexity correlation bounds of [CSS18] for parity and the generalized Andreev's function against constant-depth circuits with LTF gates to constant-depth circuits with degree-d PTF gates, for any $d \geq 1$. We do this by generalizing the structural lemma for LTFs used in [CSS18] to degree-d PTFs.

Lemma 3.57. *For any PTF $f(x) = \mathrm{sgn}(p(x))$ of degree $d \geq 1$, and any $0 < \delta$, $r \leq 1/16$, we have*

$$\Pr_{\rho \sim R_r}[f_\rho \text{ is not } \delta\text{-close to constant}] \leq (\sqrt{r} + \delta) \cdot (\log r^{-1} \cdot \log \delta^{-1})^{O(d^2)}.$$

Proof. Let r_0 be so that $r_0^{-1} = \lfloor r^{-1} \rfloor$. Then we have

$$\Pr_{\rho \sim R_r}[f_\rho \text{ is not } \delta\text{-close to constant}] = \Pr_{\rho_1 \sim R_{r_0}, \rho_2 \sim R_{r/r_0}}[(f_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-close to constant}]. \tag{3.48}$$

Let $E(\rho_1)$ denote the random event that f_{ρ_1} is δ^2-close to constant. Then Equation (3.48) can be expressed as

$$\begin{aligned}&\mathop{\mathbf{Pr}}_{\rho_1,\rho_2}[(f_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-close to constant} \mid \neg E(\rho_1)] \cdot \mathop{\mathbf{Pr}}_{\rho_1}[\neg E(\rho_1)] \\ &+ \mathop{\mathbf{Pr}}_{\rho_1,\rho_2}[(f_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-close to constant} \mid E(\rho_1)] \cdot \mathop{\mathbf{Pr}}_{\rho_1}[E(\rho_1)] \end{aligned} \tag{3.49}$$

By the fact that a function δ^2-close to constant is expected to remain δ^2-close to constant under random restrictions and by Markov's inequality, the second summand in Equation (3.49) is at most δ. We then upperbound the first summand in Equation (3.49) by showing the following

$$\mathop{\mathbf{Pr}}_{\rho_1}[\neg E(\rho_1)] \leq (\sqrt{r} + \delta) \cdot (\log r^{-1} \cdot \log \delta^{-1})^{O(d^2)}.$$

Since $r_0 \leq 2r$, it suffices to show

$$\mathop{\mathbf{Pr}}_{\rho_1 \sim R_{r_0}} \left[f_{\rho_1} \text{ is not } \delta^2\text{-close to constant} \right] \leq (\sqrt{r_0} + \delta) \cdot (\log r_0^{-1} \cdot \log \delta^{-1})^{O(d^2)}. \tag{3.50}$$

Equation (3.50) follows immediately from Lemma 3.37 and Lemma 3.26 by noting the following equivalent way of choosing a random restriction $\rho_1 \sim R_{r_0}$.

1. Randomly partition the variables of f into $m = 1/r_0$ disjoint blocks, where each variable is assigned to block $i \in [m]$, independently, with probability $1/m$.
2. Apply a random block restriction $\rho' \sim B_m$ based on the partition in the previous step.

□

We now state our correlation bounds against constant-depth circuits with PTF gates. Let PARITY_n denote the parity function on n variables, and let $A'_n \in \mathsf{P}$ denote the variant of Andreev's function on $5n$ variables as defined in [CKK+15].[3] For Boolean functions $f, g\colon \{-1,1\}^n \to \{-1,1\}$, recall that the correlation between f and g is

$$\mathbf{Corr}(f,g) = \left| \mathop{\mathbf{E}}_{x \sim \{-1,1\}^n} [f(x) \cdot g(x)] \right|.$$

We get the following correlation bounds.

Theorem 3.58. *For any $D \geq 1$ and $1 \leq d \ll \sqrt{\log n / \log\log n}$, let C be any depth-D circuit on n inputs with degree-d PTF gates, of wire complexity at most $n^{1+\varepsilon_D}$, where*

[3] We have $A'_n(x_1, \ldots, x_{4n}, y_1, \ldots, y_n) = \mathrm{Enc}(x_1, \ldots, x_{4n})_{\mathrm{Ext}(y_1,\ldots,y_n)}$, where $\mathrm{Enc}(\cdot)$ denotes the encoding with a certain error-correcting code, and $\mathrm{Ext}(\cdot)$ is a certain extractor; see [CKK+15] or [CSS18] for more details.

$\varepsilon_D = B^{-(2D-1)}$, *for some constant* $B > 0$. *Then we have*

$$\mathbf{Corr}(C, \mathsf{PARITY}_n) \leq O\left(n^{-\varepsilon_D}\right).$$

Theorem 3.59. *For any* $D \geq 1$ *and* $1 \leq d \ll (\log n / \log\log n)^{1/(2D-1)}$, *let* C *be any depth-*D *circuit on* $5n$ *inputs with degree-*d *PTF gates, of wire complexity at most* $n^{1+\mu_{D,d}}$, *where* $\mu_{D,d} = (E \cdot d)^{-(2D-1)}$, *for some constant* $E > 0$. *Then we have*

$$\mathbf{Corr}(C, A'_n) \leq \exp(-n^{\mu_{D,d}/2}).$$

Remark 3.60. *In Theorem 3.58, the exponent* ε_D *in the correlation bound does not depend on the degree* d *of the PTF gates in the circuit* C, *and stays polynomially small even for super-constant degree* $d \ll \sqrt{\log n / \log\log n}$. *In Theorem 3.59, the correlation bound is exponentially small for a constant degree* d, *and is super-polynomially small for* $d \ll (\log n / \log\log n)^{1/(2D-1)}$.

The proofs of Theorem 3.58 and Theorem 3.59 are analogous to those in [CSS18] for the case of LTF circuits, with just a couple of changes. The proofs are by induction on the depth D. For the proof of correlation bounds with parity in [CSS18], the base case is the noise sensitivity bound for LTFs due to Peres [Per04]; for the proof of Theorem 3.58, we can use the noise sensitivity bound for degree-d PTFs due to Kane [Kan14]. For the correlation bounds with Andreev's function, the base case in [CSS18] needs an upper bound on the number of distinct LTFs on n variables; we can use the bound for PTFs given by Theorem 3.52. Finally, for the inductive step, [CSS18] use their LTF restriction lemma to show that, under a particular type of random restriction, with high probability, a depth-D circuit with LTF gates will become close to some circuit of depth $D-1$. We can use our PTF Restriction Lemma, Lemma 3.57, with an appropriately small value of δ (for example, $\exp\left(-(r^{-1/(c\cdot d^2)})\right)$, for a sufficiently large constant c).

3.6.4 Influence bound for PTFs

Here we show that Kane's bound on the total influence (average sensitivity) of degree-d PTFs is a corollary of our Block Restriction Lemma.

Theorem 3.61 ([Kan14])**.** *For any* d*-degree PTF* f *on* $n > 1$ *variables, we have*

$$\mathbf{Inf}[f] \leq \sqrt{n} \cdot (\log n)^{O(d \log d)} \cdot 2^{O\left(d^2 \log d\right)}.$$

Proof. We first partition the variables into n blocks so that each block contains exactly one variable. We then apply a variant of our Block Restriction Lemma, Lemma 3.48, with $\delta = 1/n^2$. For

$$d \leq \sqrt{(\log n)/(c' \cdot \log\log n)}, \tag{3.51}$$

for some constant $c' > 0$, the assumption of Lemma 3.48 on the largeness of δ is satisfied. Note that since the restricted function is on one variable, being $(1/n^2)$-close to constant is the same as being constant. Therefore, by Lemma 3.48 and Lemma 3.26, we get

$$\mathop{\mathbf{Pr}}_{\rho\sim B_n}[f_\rho \text{ is not a constant}] \le n^{-1/2}\cdot(\log n)^{O(d\log d)}. \tag{3.52}$$

Also, by the definition of random block restriction, we have

$$\mathop{\mathbf{Pr}}_{\rho\sim B_n}[f_\rho \text{ is not a constant}] = \frac{1}{n}\cdot\sum_{i=1}^{n}\mathop{\mathbf{Pr}}_{A_{\bar{i}}}\left[f_{A_{\bar{i}}} \text{ is not a constant}\right], \tag{3.53}$$

where $A_{\bar{i}}$ is a random assignment to the variables except the i-th variable. Note that for every fixed i,

$$\mathop{\mathbf{Pr}}_{A_{\bar{i}}}\left[f_{A_{\bar{i}}} \text{ is not a constant}\right] = \mathop{\mathbf{Pr}}_{x\sim\{-1,1\}^n}[f(x)\neq f(x^{\oplus i})] = \mathbf{Inf}_i[f]. \tag{3.54}$$

Combining Equation (3.53) and Equation (3.54), we have

$$\sum_{i=1}^{n}\mathbf{Inf}_i[f] = n\cdot\mathop{\mathbf{Pr}}_{\rho\sim B_n}[f_\rho \text{ is not a constant}].$$

Together with Equation (3.52), we get

$$\mathbf{Inf}[f] \le \sqrt{n}\cdot(\log n)^{O(d\log d)}. \tag{3.55}$$

Note that Equation (3.55) holds for small degrees d satisfying Equation (3.51). If we multiply the right-hand side of Equation (3.55) by $2^{O\left(d^2\log d\right)}$, we ensure that the bound on influence holds also for all large d (as then the right-hand side of Equation (3.55) becomes at least n, which is a trivial upper bound on $\mathbf{Inf}[f]$). □

3.6.5 Littlewood-Offord type anticoncentration bounds for polynomials

Here we use our Block Restriction Lemma to drive the following anticoncentration bounds for degree-d multilinear polynomials.

Theorem 3.62 ([MNV16])**.** *For any real interval I, and any degree-d multilinear polynomial p such that there exists a set of t disjoint monomials in p, each of which is maximal (i.e., not contained by any other monomials) and has coefficient at least $|I|$ in magnitude, we have*

$$\mathbf{Pr}[p(A)\in I] \le t^{-1/2}\cdot(\log t)^{O(d\log d)}\cdot 2^{O\left(d^2\log d\right)}.$$

Proof. Our proof is very similar to that of [MNV16], except they used Kane's bound of Theorem 3.61, whereas we use a variant of our Block Restriction Lemma (Lemma 3.48).

Without loss of generality, we can assume I is centered at 0; otherwise, the center of I is c and we can bound the probability that the polynomial $p' = p - c$ takes values within the interval I' centered at 0 with $|I'| = |I|$.

We first partition the variables into t blocks so that p restricted to each block (i.e., the restricted polynomial that only depends on the variables in that block) has at least one monomial with the coefficient at least $|I|$ in magnitude. Consider the following equivalent way of sampling a uniformly random input to p: apply a block restriction based on the partition above and randomly assign 1 or -1 to the variables in the unrestricted block. Then we have that

$$\begin{aligned}\mathbf{Pr}[p(A) \in I] &= \mathop{\mathbf{Pr}}_{\rho \sim B_t, C}[p_\rho(C) \in I] \\ &= \mathop{\mathbf{Pr}}_{\rho, C}[p_\rho(C) \in I \mid p_\rho \text{ is not } \delta\text{-concentrated}] \cdot \mathop{\mathbf{Pr}}_{\rho}[p_\rho \text{ is not } \delta\text{-concentrated}] \\ &\quad + \mathop{\mathbf{Pr}}_{\rho, C}[p_\rho(C) \in I \mid p_\rho \text{ is } \delta\text{-concentrated}] \cdot \mathop{\mathbf{Pr}}_{\rho}[p_\rho \text{ is } \delta\text{-concentrated}], \quad (3.56)\end{aligned}$$

where C is a multidimensional Bernoulli random variable.

Let $\delta = t^{-1/2}$. By Lemma 3.48, we have

$$\mathop{\mathbf{Pr}}_{\rho \sim B_t}[p_\rho \text{ is not } \delta\text{-concentrated}] \leq t^{-1/2} \cdot (\log t)^{O(d \log d)} \cdot 2^{O\left(d^2 \log d\right)}. \quad (3.57)$$

Note that we multiply by the factor $2^{O\left(d^2 \log d\right)}$ on the right-hand side of Equation (3.57) so that it holds for all degrees. This bounds the first summand of Equation (3.56). To bound the second summand of Equation (3.56), we use the following observation from a preliminary version of [MNV16].

Claim 3.63. *For any real interval I centered at 0, and any δ-concentrated degree-d multilinear polynomial q that has at least one monomial with coefficient greater than $|I|$ in magnitude, we have*

$$\mathbf{Pr}[q(A) \in I] \leq \delta.$$

Proof. Let $q = q' + \mu$ where $\mu = \mathbf{E}[q(A)]$, and let $\nu = (L \cdot \log \delta^{-1})^d$ where $L > 0$ is the constant from Definition 3.25. Since q is δ-concentrated and has at least one monomial with coefficient greater than $|I|$ in magnitude, we have

$$|\mu|^2 \geq (\nu - 1) \cdot \mathbf{Var}[q] \geq (\nu - 1) \cdot |I|^2 \geq 4 \cdot |I|^2.$$

Now since $|\mu| \geq 2 \cdot |I|$, we note that for all points $x \in \{-1, 1\}^n$ where $q(x) \in I$, it must be the case that $|q'(x)| \geq |\mu| - |I|$. Also, we have

$$|\mu| - |I| \geq \frac{|\mu|}{2} \geq \frac{\sqrt{(\nu - 1) \cdot \mathbf{Var}[q]}}{2} \geq \frac{\sqrt{\nu \cdot \mathbf{Var}[q]}}{4} = \frac{\sqrt{\nu}}{4} \cdot \|q'\|_2 . \quad (3.58)$$

As a result,

$$\begin{aligned}\mathbf{Pr}[q(A) \in I] &\leq \mathbf{Pr}\left[|q'(A)| \geq |\mu| - |I|\right] \\ &\leq \mathbf{Pr}\left[|q'(A)| \geq \frac{\sqrt{\nu}}{4} \cdot \|q'\|_2\right] && \text{(by Equation (3.58))} \\ &\leq \delta. && \text{(by Theorem 3.13)}\end{aligned}$$

□

By Claim 3.63, we get

$$\mathop{\mathbf{Pr}}_{\rho,C}[p_\rho(C) \in I \mid p_\rho \text{ is } \delta\text{-concentrated}] \leq \delta,$$

which bounds the second summand of Equation (3.56). This completes the proof. □

3.7 Derandomization

3.7.1 Derandomized Block Restriction Lemma

In this subsection, we show how to derandomize our Block Restriction Lemma, by giving an algorithm for sampling pseudorandom block restrictions (using significantly fewer random bits) so that the probability a given degree-d polynomial is not concentrated under such a pseudorandom block restriction is about the same as that for true random block restrictions. Our pseudorandom block restriction will pick a uniformly random block, and then fix the variables in the remaining blocks in a pseudorandom fashion (using few truly random bits).

Theorem 3.64 (Derandomized Block Restriction Lemma)**.** *For any $0 < \delta \leq 1/16$ and $0 < \zeta < 1$, there is a polynomial-time algorithm for sampling block restrictions $\rho \in B_m$, for any $m \geq 16$, that uses at most $m^\zeta \cdot \log n$ random bits, so that the following holds. For any n-variate degree-d multilinear polynomial p whose variables are partitioned into m blocks, we have*

$$\mathop{\mathbf{Pr}}_{\rho}\left[p_\rho \textit{ is not } \delta\textit{-concentrated}\right] \leq \left(m^{-1/2} + \delta\right) \cdot (\log m)^{O(\zeta^{-1} \cdot d \cdot \log d)} \cdot (\log \delta^{-1})^{O(\zeta^{-1} \cdot d^2)}.$$

We first define our pseudorandom block restrictions that yields Theorem 3.64. We start with some notations. Let D be a distribution on $\{-1,1\}^n$. Let S be a set of K coordinates and let ω be an assignment for the coordinates in S. We define D^ω to be the distribution on the remaining $n - K$ unfixed coordinates such that for any $a \in \{-1,1\}^{n-K}$,

$$\mathbf{Pr}[D^\omega = a] = \mathbf{Pr}[D_{[n]-S} = a \mid D_S = \omega].$$

We will refer to D^ω as *the distribution D conditioned on fixing S to ω*.

The main idea of our pseudorandom block restriction is to fix the variables using the output of a pseudorandom generator (PRG) for PTFs. Recall that a function $G : \{-1,1\}^s \to \{-1,1\}^n$ is a *PRG of seed length s that ε-fools PTFs of degree d* if, for any degree-d PTF f, we have

$$\left| \mathop{\mathbf{Pr}}_{z \sim \{-1,1\}^s}[f(G(z)) = -1] - \mathop{\mathbf{Pr}}_{x \sim \{-1,1\}^n}[f(x) = -1] \right| \leq \varepsilon.$$

Definition 3.65. *Suppose the variables of a polynomial are arbitrarily partitioned into m blocks. We call a random block restriction ρ (m,ε)-fooling if it selects a block uniformly at random and fixes all variables outside the selected block using some distribution D that ε-fools PTFs of degree $2d$ (in the appropriate number of variables). Moreover, we call such a random block restriction (m,ε,K)-fooling if D ε-fools PTFs of degree $2d$ even conditioned on fixing at most K coordinates and is $\left(192 \cdot d \cdot \log \delta^{-1} + K\right)$-wise independent.*

We will use the construction of PRGs for PTFs due to Meka and Zuckerman. First recall that a multidimensional distribution on $\{-1,1\}^n$ is called *k-wise independent* if any k coordinates of the distribution are independent. A family of hash functions $\mathcal{H} = \{h \colon [n] \to [\ell]\}$ is called k-wise independent if for any $(x_1, \ldots, x_k) \in [n]^k$, where $x_1, \ldots, x_k$ are distinct, and any $(y_1, \ldots, y_k) \in [\ell]^k$, we have

$$\mathop{\mathbf{Pr}}_{h \sim \mathcal{H}}[h(x_1) = y_1 \wedge \cdots \wedge h(x_k) = y_k] = 1/\ell^k.$$

There exist k-wise independent distributions that can be sampled in $\mathsf{poly}(n,k)$ time using $O\left(k \cdot \log n\right)$ random bits, and there exist k-wise independent hash families $\mathcal{H}$ such that a random $h \in \mathcal{H}$ can be sampled using $O\left(k \cdot \log(n \cdot \ell)\right)$ bits (see, e.g., [Vad12]).

The generator in the following theorem views its random seed as a tuple of $\ell + 1$ disjoint random strings (for a certain parameter $\ell \geq 1$), and uses the first string to sample a hash function h, and the remaining ℓ strings to get ℓ samples from a k-wise independent distribution.

Theorem 3.66 ([MZ13])**.** *For $0 < \varepsilon < 1$, let $\ell = 2^{O(d)} \cdot \log^2(\varepsilon^{-1}) \cdot \varepsilon^{-(4d+1)}$. Let $G : \{-1,1\}^s \to \{-1,1\}^n$ be the following process of generating an assignment for n coordinates using $s = 2^{O(d)} \cdot (\log n) \cdot \varepsilon^{-(8d+3)}$ random bits:*

1. *Partition the n coordinates into ℓ buckets using a function $h \colon [n] \to [\ell]$ randomly picked from a 2-wise independent hash family.*
2. *For each bucket, generate a $(\ell + 4d)$-wise independent distribution for the coordinates in that bucket.*

Then G is a PRG that ε-fools n-variate PTFs of degree d.

Lemma 3.67. *For $0 < \varepsilon < 1$, there exists a (m,ε,K)-fooling random block restriction that is samplable using $s = 2^{O(d)} \cdot \left(\varepsilon^{-(16d+3)} + \varepsilon^{-(8d+2)} \cdot (K + \log \delta^{-1})\right) \cdot \log n$ random bits.*

Proof. Let $\ell = 2^{O(d)} \cdot \log^2(\varepsilon^{-1}) \cdot \varepsilon^{-(8d+1)}$. Consider the distribution D sampled as follows:

1. Partition the n coordinates into ℓ buckets using a function $h\colon [n] \to [\ell]$ randomly picked from a 2-wise independent hash family.

2. For each bucket, generate a $\left(\ell + 4d + K + 192 \cdot d \cdot \log \delta^{-1}\right)$-wise independent distribution for the coordinates in that bucket.

Note that by Theorem 3.66, D ε-fools PTFs of degree $2d$ even conditioned on fixing at most K coordinates. This is because D has sufficient bounded independence for each bucket even conditioned on fixing K coordinates. Also, D is $\left(192 \cdot d \cdot \log \delta^{-1} + K\right)$-wise independent. Note that D is samplable using s random bits. We then define our (m, ε, K)-fooling random block restriction as the restriction that randomly selects a block and fixes all variables outside the selected block using D. Finally, note that the number of random bits needed to select a block is at most $\log n$. □

While it is possible to use a (m, ε, K)-fooling random block restriction to obtain a derandomized block restriction lemma, it requires large seed length to get small error if we do this in one shot. To deal with this issue, we use a sequence of pseudorandom block restrictions, so that, in each step, we only set the error parameter to match the probability that a random block restriction does not make the polynomial concentrated in the current step. Consider a random block restriction defined as follows. Let $m, \kappa \geq 16$.

1. Partition the m blocks of variables of p into $b = m^{1/\kappa d}$ disjoint super-blocks, where each super-block has m/b blocks.

2. Apply a $(b, \varepsilon = b^{-1}, K)$-fooling random block restriction on the b super-blocks.

3. Repeat the above two steps for the remaining blocks with m replaced by m/b until there is at most 2^d blocks, in which case we randomly choose a single block and fix the other variables using a $\left(192 \cdot d \cdot \log \delta^{-1} + K\right)$-wise independent distribution.

If at any round the blocks cannot be partitioned evenly into super-blocks, we can divide them so that the sizes of any two super-blocks differ by at most 1. We then select a super-block with probability proportional to its size. This makes sure that a block is selected uniformly at random. To avoid some technicalities that can be overcome easily, we assume here that at each round, the blocks can be partitioned evenly into super-blocks. Note that a random block restriction ρ generated as above can be decomposed into a sequence of sub-restrictions $\rho_1, \ldots, \rho_t, \rho_{t+1}$, where $t = O\left(\kappa \cdot d \cdot \log \log m\right)$ so that there are at most 2^d blocks remaining after $\rho_1, \ldots, \rho_t$, and each ρ_i, except the last one, is a (b_i, b_i^{-1}, K)-fooling random block restriction with $b_i = m^{(1-1/\kappa d)^{i-1}/\kappa d}$. Also, the last restriction fixes the variables using some $\left(192 \cdot d \cdot \log \delta^{-1} + K\right)$-wise independent distribution. We call such a random block restriction (m, κ, K)-*good*. By the above, we have

Lemma 3.68. *There exists a (m, κ, K)-good block restriction that is samplable using*

$$2^{O(d)} \cdot \left(m^{(19/\kappa)} + m^{(10/\kappa)} \cdot (K + \log \delta^{-1})\right) \cdot \log n \cdot \kappa \cdot \log\log m$$

random bits.

To prove Theorem 3.64, we will show that the argument in Section 3.5 still goes through if we replace a truly random block restriction with our pseudorandom block restriction described above. We will need versions of the key lemmas in Section 3.5 for our pseudorandom block restrictions. In particular, we need a version of Lemma 3.35 for k-wise independent distributions. We first show a version of Theorem 3.13 for k-wise independent distributions. The following can be proved in the same way as Theorem 3.13.

Claim 3.69. *For any degree-d multilinear polynomial $p\colon \mathbb{R}^n \to \mathbb{R}$, any $T \geq 2^d$, and any $\left(\frac{d \cdot T^{2/d}}{2}\right)$-wise independent distribution D on $\{-1,1\}^n$, we have*

$$\mathbf{Pr}\left[|p(D)| \geq T \cdot \|p\|_2\right] \leq \exp\left(-(1/4) \cdot T^{2/d}\right).$$

Proof. Let $W = \frac{T^{2/d}}{2}$. By Markov's inequality, we have

$$\mathbf{Pr}\left[|p(D)| \geq T \cdot \|p\|_2\right] = \mathbf{Pr}\left[|p(D)|^W \geq (T \cdot \|p\|_2)^W\right] \leq \frac{\mathbf{E}[|p(D)|^W]}{(T \cdot \|p\|_2)^W}. \tag{3.59}$$

Since D is a $(d \cdot W)$-wise independent distribution on $\{-1,1\}^n$, we get, using Equation (3.1), that

$$\mathbf{E}[|p(D)|^W] = \|p\|_W^W \leq \left((W-1)^{d/2} \cdot \|p\|_2\right)^W \leq \left(W^{d/2} \cdot \|p\|_2\right)^W. \tag{3.60}$$

Combining Equations (3.59) and (3.60), we get

$$\mathbf{Pr}\left[|p(D)|^W \geq (T \cdot \|p\|_2)^W\right] \leq \left(\frac{W^{d/2}}{T}\right)^W \leq \exp\left(-(1/4) \cdot T^{2/d}\right),$$

as required. □

Claim 3.70. *For any degree-d multilinear polynomial p that is $(\delta, \gamma+1)$-concentrated, let ρ be a random block restriction for p that picks a uniformly random block and assigns the variables outside the block using a $(192 \cdot d \cdot \log \delta^{-1})$-wise independent distribution. Then we have that*

$$\mathbf{Pr}_{\rho}[p_\rho \text{ is not } (\delta, \gamma)\text{-concentrated}] \leq \delta.$$

Proof. The proof is the same as that of Lemma 3.35, with Claim 3.69 replacing Theorem 3.13. □

Next, we show two recurrence relations that are similar to Lemma 3.39 and Lemma 3.40, but with respect to our pseudorandom block restrictions.

Definition 3.71. *Let $\mathcal{Q}(d, m, \delta, \kappa, K, a)$ be the supremum, over all degree-d polynomials p with $\alpha(p) \le a$, all possible partitions of the variables into m blocks, and all (m, κ, K)-good random block restrictions ρ, of the probabilities*

$$\Pr_{\rho}[p_\rho \text{ is not } \delta\text{-concentrated}].$$

Let $\mathcal{Q}_{\mathrm{reg}}(d, m, \delta, \kappa, K, a, \tau)$ be the same as $\mathcal{P}$ but only for τ-regular polynomials. For simplicity, we will omit some parameters when they are clear in the context. In particular, we will use $\mathcal{Q}(m, K, a)$ (resp. $\mathcal{Q}_{\mathrm{reg}}(m, K, a, \tau)$) for $\mathcal{Q}(d, m, \delta, \kappa, K, a)$ (resp. $\mathcal{Q}_{\mathrm{reg}}(d, m, \delta, \kappa, K, \tau)$).

Lemma 3.72. *For any $0 < \tau, \delta < 1/4$ and $a > 0$, $m > 4$, $\kappa \ge 16$, $d > 1$, and $K \ge H$, where $H = \tau^{-1} \cdot (d \cdot \log \tau^{-1} \cdot \log \delta^{-1})^{O(d)}$, we have*

$$\mathcal{Q}(m, K, a) \le \frac{1}{m} \cdot \tau^{-1} \cdot (d \cdot \log \tau^{-1} \cdot \log \delta^{-1})^{O(d)} + 2\delta + \mathop{\mathbf{E}}_{\aleph}[\mathcal{Q}_{\mathrm{reg}}(m, K - H, \aleph, \tau)],$$

where $\aleph$ is a non-negative random variable with $\mathbf{E}[\aleph] \le O(a)$.

Proof. The proof is very similar to that of Lemma 3.39, but has a few critical differences. For clarity, we provide details for this proof. Let p be a degree-d multilinear polynomial whose variables are partitioned into m blocks. Let ρ be a (m, κ, K)-good random block restriction. Consider the decision tree given by Theorem 3.21 with $\varepsilon = \delta$ and $\gamma = 2$. Note that the depth of this decision tree is H. Since a block is chosen uniformly at random, the probability that ρ is not consistent with any branch of the decision tree is at most H/m.

Next we show the following.

Claim 3.73.

$$\Pr_{\rho}[p_\rho \text{ is not } \delta\text{-concentrated}] \le \mathop{\mathbf{E}}_{\omega}\left[\Pr_{\rho_\omega}[(p_\omega)_{\rho_\omega} \text{ is not } \delta\text{-concentrated}]\right] + 3(H/m),$$

where the expectation is over random leaves ω of the decision tree, p_ω is the restriction of p obtained by fixing the variables on the branch leading to ω as specified by the branch, and ρ_ω is an $(m, \kappa, K - H)$-good random block restriction.

Proof. We view ρ as $\rho = (\ell, \lambda)$, where ℓ is the selected block and λ is an assignment to the variables outside block ℓ. We can view the distribution of λ as a sequence of distributions, each ε-fooling PTFs of degree $2d$ even conditioned on fixing at most K coordinates, and being $(192 \cdot d \cdot \log \delta^{-1} + K)$-wise independent.

For each leaf ω, let R_ω be the set of (m, κ, K)-good random restrictions consistent with the branch leading to ω. As observed above, the probability ξ of choosing a restriction from the complement of $\cup_\omega R_\omega$ is at most H/m. We get

$$\Pr_{\rho}[p_\rho \text{ is not } \delta\text{-concentrated}] \le (1 - \xi) \cdot \Pr_{\rho \in \cup_\omega R_\omega}[p_\rho \text{ is not } \delta\text{-concentrated}] + \xi. \tag{3.61}$$

By conditioning on $\rho \in R_\omega$, we get that $\mathbf{Pr}_{\rho \in \cup_\omega R_\omega}[p_\rho \text{ is not } \delta\text{-concentrated}]$ equals to

$$\sum_\omega \mathop{\mathbf{Pr}}_\rho[p_\rho \text{ is not } \delta\text{-concentrated} \mid \rho \in R_\omega] \cdot \mathbf{Pr}[\rho \in R_\omega \mid \rho \in \cup_\omega R_\omega].$$

As ρ is K-wise independent and $K \geq H$, the probability of choosing $\rho \in R_\omega$ conditioned on $\rho \in \cup_\omega R_\omega$ is $2^{-\ell_\omega} \cdot (1-\xi)^{-1}$, where ℓ_ω is the length of the branch leading to ω. Hence, the right-hand side of Equation (3.61) is at most

$$\mathop{\mathbf{E}}_\omega[\mathop{\mathbf{Pr}}_{\rho \in R_\omega}[p_\rho \text{ is not } \delta\text{-concentrated}]] + \xi. \tag{3.62}$$

Each (m, κ, K)-good restriction $\rho = (\ell, \lambda) \in R_\omega$ can be viewed as a restriction of the variables on the branch leading to ω (as specified by the branch) plus a restriction λ' to the remaining variables outside block ℓ. So we can express p_ρ as $(p_\omega)_{\rho'}$, where $\rho' = (\ell, \lambda')$.

Note that ρ' is a $(m, \kappa, K-H)$-good restriction, which comes from the set of those $(m, \kappa, K-H)$-good restrictions that chose block ℓ outside at most H blocks containing the variables on the branch leading to ω. The set of all such restrictions ρ' has the probability mass at least $1 - H/m$ within the set of all $(m, \kappa, K-H)$-good restrictions ρ_ω (which pick block ℓ uniformly at random from the set of all m blocks). Therefore, we can upperbound the expression in Equation (3.62) by

$$\begin{aligned}
&\mathop{\mathbf{E}}_\omega[\mathop{\mathbf{Pr}}_{\rho'}[(p_\omega)_{\rho'} \text{ is not } \delta\text{-concentrated}]] + \xi \\
&\leq (1 - H/m)^{-1} \cdot \mathop{\mathbf{E}}_\omega[\mathop{\mathbf{Pr}}_{\rho_\omega}[(p_\omega)_{\rho_\omega} \text{ is not } \delta\text{-concentrated}]] + \xi \\
&\leq \mathop{\mathbf{E}}_\omega[\mathop{\mathbf{Pr}}_{\rho_\omega}[(p_\omega)_{\rho_\omega} \text{ is not } \delta\text{-concentrated}]] + \xi + 2(H/m),
\end{aligned}$$

where the last inequality uses the fact that $(1-x)^{-1} \leq 1 + 2x$ whenever $0 < x \leq 1/2$. The claim follows. □

The proof can now proceed in the same way as that of Lemma 3.39, but instead of using Lemma 3.35 there, we use Claim 3.70 and the fact that ρ_ω fixes the variables $(192 \cdot d \cdot \log \delta^{-1})$-wise independently. □

Lemma 3.74. *For any real $0 < \tau, \delta < 1/4$ and $a > 0$, $m > 4$, $\kappa \geq 16$, $K \geq 0$ and $d > 1$, we have*

$$\mathcal{Q}_{\mathrm{reg}}(m, K, a, \tau) \leq \mathop{\mathbf{E}}_{\aleph}[\mathcal{Q}(m^{1-1/\kappa d}, K, \aleph)],$$

where $\aleph$ is a non-negative random variable with $\mathbf{E}[\aleph] = O\left(d^3 a m^{-1/2\kappa d} + d^4 \tau^{1/(8d)} + m^{-1/\kappa d}\right)$.

Proof. For a (m, κ, K)-good random block restriction ρ, we can decompose it into two restrictions ρ_1 and ρ', where ρ_1 is a $(b = m^{1/\kappa d}, \varepsilon = m^{-1/\kappa d}, K)$-fooling random block

restriction and ρ' is a $(m^{1-1/\kappa d}, \kappa, K)$-good random block restriction. Then

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\rho}[p_\rho \text{ is not } \delta\text{-concentrated}] &= \mathop{\mathbf{E}}_{\rho_1}\left[\mathop{\mathbf{Pr}}_{\rho'}[(p_{\rho_1})_{\rho'} \text{ is not } \delta\text{-concentrated}]\right] \\ &\leq \mathop{\mathbf{E}}_{\rho_1}\left[\mathcal{Q}(m^{1-1/\kappa d}, K, \alpha(p_{\rho_1}))\right].\end{aligned}$$

Therefore, we need to show

$$\mathop{\mathbf{E}}_{\rho_1}[\alpha(p_{\rho_1})] = O\left(d^3 a b^{-1/2} + d^4 \tau^{1/(8d)} + \varepsilon\right).$$

From Equation (3.36), for a truly random block restriction $\sigma \sim B_b$, we have

$$\mathop{\mathbf{E}}_{\sigma}[\alpha(p_\sigma)] = O\left(d^3 a b^{-1/2} + d^4 \tau^{1/(8d)}\right).$$

Thus, it suffices to show that

$$|\mathop{\mathbf{E}}_{\sigma}[\alpha(p_\sigma)] - \mathop{\mathbf{E}}_{\rho_1}[\alpha(p_{\rho_1})]| \leq \varepsilon.$$

Consider a degree-d multilinear polynomial p whose variables are partitioned into m blocks. Fixed a block ℓ. Let A_ℓ be an assignment to the variables in block ℓ, B be a vector of dimension the same as the number of variables in block ℓ, and t be an arbitrary number. Define $T_{\ell,A_\ell,B,t}$ to be the Boolean function on input D such that $T_{\ell,A_\ell,B,t}(D) = -1$ if and only if

$$|p_D(A_\ell)|^2 \leq t \cdot |\mathsf{D}_B\, p_D(A_\ell)|^2.$$

It is easy to see that $T_{\ell,A_\ell,B,t}$ is a PTF of degree at most $2d$.

Now for a block, ℓ, we let A_ℓ denote the random assignment to the variables in ℓ and let $A_{\bar{\ell}}$ denote the random assignment to the variables that are not in ℓ. Let D be the

distribution by which ρ_1 fixes the variables. Note that D ε-fools PTF of degree $2d$. We have

$$\begin{aligned}
\mathop{\mathbf{E}}_{\sigma}[\alpha(p_\sigma)] &= \frac{1}{b} \cdot \sum_{\ell \in [b]} \mathop{\mathbf{E}}_{A_{\bar{\ell}}}[\alpha(p_{A_{\bar{\ell}}})] \\
&= \frac{1}{b} \cdot \sum_{\ell} \mathbf{E}\left[\min\left\{1, \frac{|\mathsf{D}_B\, p_{A_{\bar{\ell}}}(A_\ell)|^2}{|p_{A_{\bar{\ell}}}(A_\ell)|^2}\right\}\right] \\
&= \frac{1}{b} \cdot \sum_{\ell} \int_0^1 \mathbf{Pr}\left[|p_D(A_\ell)|^2 \le t \cdot |\mathsf{D}_B\, p_D(A_\ell)|^2\right] dt \\
&= \frac{1}{b} \cdot \sum_{\ell} \int_0^1 \mathbf{Pr}\left[T_{\ell,A_\ell,B,t}(A_{\bar{\ell}})\right] dt \\
&\le \frac{1}{b} \cdot \sum_{\ell} \int_0^1 \left(\mathbf{Pr}\left[T_{\ell,A_\ell,B,t}(D)\right] + \varepsilon\right) dt \\
&\le \left(\frac{1}{b} \cdot \sum_{\ell} \int_0^1 \mathbf{Pr}\left[T_{\ell,A_\ell,B,t}(D)\right] dt\right) + \varepsilon \\
&= \mathop{\mathbf{E}}_{\rho_1}[\alpha(p_{\rho_1})] + \varepsilon.
\end{aligned}$$

The other direction can be shown similarly. □

We are now ready to prove the following result, which, together with our construction of (m, κ, K)-good random block restrictions in Lemma 3.68, will imply Theorem 3.64

Theorem 3.75. *There exist constants $B, C > 0$ such that, for any $d > 0$, $m, \kappa \ge 16$, $0 < \delta \le 1/16$, $0 < a \le 1$, and $K \ge m^{8/\kappa} \cdot (d \cdot \log m \cdot \log \delta^{-1})^{C \cdot d}$, we have*

$$\mathcal{Q}(d, m, \delta, \kappa, K, a) \le (a \cdot m^{-1/2} + \delta) \cdot (\log m)^{B\kappa \cdot d \cdot \log d} \cdot (\log \delta^{-1})^{B\kappa \cdot d^2}. \tag{3.63}$$

Proof. The proof is similar to that of Theorem 3.45. As in the proof of Theorem 3.45, we can assume, for a sufficiently large constant $B > 0$, that both

$$a \ge (c \cdot \log \delta^{-1})^{-2d}, \tag{3.64}$$

and

$$m^{1/(2\kappa \cdot d)} \ge (c \cdot \log \delta^{-1})^{2d}, \tag{3.65}$$

where $c > 0$ is the constant from Corollary 3.33. Note that if Equation (3.64) if false. Then by Corollary 3.33, the polynomial is $(\delta, 2)$-concentrated and by Claim 3.70 such a polynomial will remain δ-concentrated under restrictions that fix variables $(192 \cdot d \cdot \log \delta^{-1})$-wise independently except with probability at most δ.

Now let $\tau = m^{-8/\kappa}$ and $H = \tau^{-1} \cdot (d \cdot \log \tau^{-1} \cdot \log \delta^{-1})^{O(d)}$. Note that $K \ge H$ when C is sufficiently large. Then combining Lemma 3.72 and Lemma 3.74, and proceeding as in the

proof of Claim 3.47, we have, for a sufficiently large constant E,

$$\begin{aligned}\mathcal{Q}(m, K, a) \leq (a \cdot m^{-1/2} + \delta) \cdot (\log m)^{E \cdot \kappa \cdot d \cdot \log d} \cdot (\log \delta^{-1})^{E \cdot \kappa \cdot d^2} \\ + \mathop{\mathbf{E}}_{\aleph}[\mathcal{Q}(m^{1-1/(\kappa d)}, K - H, \aleph)],\end{aligned} \tag{3.66}$$

where $\aleph$ is a non-negative random variable with $\mathbf{E}[\aleph] = O\left(d^4 \cdot a \cdot m^{-1/(2\kappa d)}\right)$.

To solve the recurrence relation given by Equation (3.66), we again use induction on m. The base case is $m \leq 2^d$. As $a \geq (c \cdot \log \delta^{-1})^{-2d}$, in this case the right hand side of Equation (3.63) is greater than 1 for when B is sufficiently large. Now suppose Theorem 3.45 holds for all smaller values of m. Let $m_1 = m^{1-1/(\kappa d)}$. Then when C is sufficiently large, we have

$$K - H \geq m_1^{8/\kappa} \cdot (d \cdot \log m_1 \cdot \log \delta^{-1})^{C \cdot d}.$$

Therefore, we can apply the induction hypothesis on

$$\mathcal{Q}(m^{1-1/(\kappa d)}, K - H, \aleph)$$

in Equation (3.66). After applying the induction hypothesis and proceeding as in the proof of Theorem 3.45, we can complete the induction step and hence the proof. □

By Lemma 3.68 and Theorem 3.75, there exist constants $B, C > 0$ and a pseudorandom block restriction samplable using

$$m^{(19/\kappa)} \cdot \log n \cdot \kappa \cdot (d \cdot \log m \cdot \log \delta^{-1})^{C \cdot d} \tag{3.67}$$

random bits such that for any degree-d multilinear polynomial p,

$$\mathop{\mathbf{Pr}}_{\rho}[p_\rho \text{ is not } \delta\text{-concentrated}] \leq \left(m^{-1/2} + \delta\right) \cdot (\log m)^{B \cdot \kappa \cdot d \cdot \log d} \cdot (\log \delta^{-1})^{B \cdot \kappa \cdot d^2}. \tag{3.68}$$

Note that we can assume without loss of generality that both

$$\kappa \leq \frac{\log m}{d \cdot \log d \cdot \log \log m}$$

and

$$(d \cdot \log m \cdot \log \delta^{-1})^{C \cdot d} \leq m^{1/\kappa}.$$

Otherwise, the right hand side of the Equation (3.68) is greater than 1 when B is sufficiently large. Then Equation (3.67) is at most

$$m^{O(1/\kappa)} \cdot \log n.$$

By changing the parameter κ, we obtain Theorem 3.64.

3.7.2 Derandomized Littlewood-Offord type anticoncentration bounds

Here we show two versions of derandomized anticoncentration bounds for degree-d multilinear polynomials. We first show the following.

Theorem 3.76. *For any positive integers t and d, and $0 < \zeta < 1$, there exists a distribution D on $\{-1,1\}^n$, samplable in $\mathsf{poly}(n)$ time using $t^{\zeta/d} \cdot \log n$ random bits, such that the following holds. For any real interval I, and any n-variate degree-d multilinear polynomial p that has at least t disjoint degree-d monomials with coefficient at least $|I|$ in magnitude, we have*

$$\mathbf{Pr}\left[p(D) \in I\right] \leq t^{-\frac{1}{2d}} \cdot (\log t)^{O\left(\zeta^{-1} \cdot d^2\right)}.$$

Let us call a degree-d monomial *good* if its coefficient is at least $|I|$ in magnitude, and say that a set of variables *contains* a monomial if every variable in the monomial is in the set. Also, let us call a partition of variables into blocks *good* if every block contains at least one good monomial. From the analysis in Theorem 3.62, it is easy to see that if we are explicitly given a degree-d polynomial with at least t disjoint good monomials, then we can obtain a good partition with t blocks and use our derandomized Block Restriction Lemma to generate the inputs so that the polynomial will take value inside the interval with probability at most about $t^{-1/2}$.

However, we want to derandomize obliviously, without knowing the structure of the polynomial. The idea is to partition the variables randomly, using bounded-independent hashing, so that we get a good partition with high probability. To show that bounded-independent hashing will produce a good partition with high probability, we need the following version of Chernoff bounds for bounded-independent random variables.

Theorem 3.77 ([SSS95])**.** *Let $\varepsilon \leq 1$. If X is the sum of k-wise independent random variables taking values in $[0,1]$, and $\mu = \mathbf{E}[X]$ such that $k \leq \lfloor \varepsilon^2 \mu e^{-1/2} \rfloor$, then*

$$\mathbf{Pr}[|X - \mu| > \varepsilon\mu] < \exp(-\lfloor k/2 \rfloor).$$

We now show the following.

Lemma 3.78. *Let p be a n-variate degree-d multilinear polynomial with at least t disjoint good monomials. If the variables are partitioned into $m = t^{1/d}/\log^{2/d}(t)$ blocks using a random hash function from a $(Cd \log t)$-wise independent hash family, where $C > 0$ is some constant, then the probability that the partition is not good is at most $1/t$.*

Proof. Fix a block ℓ. Let $m_1, \ldots, m_t$ be the t disjoint good monomials. For $i = 1, \ldots, t$, let X_i be the indicator random variable for the event that ℓ contains m_i, using a $(Cd \log t)$-wise independent hashing (i.e., X_i is 1 if every variable in m_i is hashed to ℓ, and 0 otherwise). Note that $\mathbf{Pr}[X_i = 1] = 1/m^d$ for every i, and $X_1, \ldots, X_t$ are $(C \log t)$-wise independent.

Let $X = X_1 + \cdots + X_t$ and $\mu = \mathbf{E}[X] = t/m^d = \log^2 t$. By Theorem 3.77, we have

$$\mathbf{Pr}[X = 0] \leq \mathbf{Pr}[|X - \mu| > (1/2)\mu] \leq \exp(-\lfloor (C \log t)/2 \rfloor) \leq 1/t^2,$$

where the last inequality holds if C is sufficiently large. Taking the union bound over the m blocks, we conclude that probability that there exists one block that does not contain any good monomial is at most $1/t$. □

We will also need the following version of Claim 3.63 for bounded-independent distributions, whose proof is the same as Claim 3.63, with Claim 3.69 replacing Theorem 3.13.

Claim 3.79. *For any real interval I centered at 0, any δ-concentrated degree-d multilinear polynomial q that has at least one monomial with coefficient greater than $|I|$ in magnitude, and any $(192 \cdot d \cdot \log \delta^{-1})$-wise independent distribution D on $\{-1, 1\}^n$, we have*

$$\mathbf{Pr}[q(D) \in I] \leq \delta.$$

Proof of Theorem 3.76. Consider the following process of sampling from D.

1. Partition the variables of p into $m = t^{1/d}/\log^{2/d}(t)$ blocks using $(Cd \log t)$-wise independent hashing, where C is the constant from Lemma 3.78.
2. Apply the derandomized Block Restriction Lemma (Theorem 3.64) based on the partition in the previous step with $\delta = m^{-1/2}$.
3. Fix the variables in the last block (i.e., the unrestricted block after applying random block restriction in the previous step) using a $(192 \cdot d \cdot \log \delta^{-1})$-wise independent distribution.

The amount of random bits used in the first step is $O(d \cdot \log t \cdot \log n)$, and the amount of random bits needed in the second step is at most

$$m^\zeta \cdot \log n \leq t^{\zeta/d} \cdot \log n.$$

The last step only needs $O(d \cdot \log \delta^{-1} \cdot \log n)$ random bits. Therefore, the total amount of random bits needed for the above process is at most $O\left(t^{\zeta/d} \cdot \log n\right)$.

We now show the correctness. By Lemma 3.78, the probability that the partition obtained in the first step is not good is at most $1/t$. Given that the partition in the first step is good, any restricted polynomial after the second step will have at least one good monomial, and by Theorem 3.64 the probability that the restricted polynomial is not δ-concentrated is at most

$$\left(m^{-1/2} + \delta\right) \cdot \left(\log m \cdot \log \delta^{-1}\right)^{O\left(\zeta^{-1} \cdot d^2\right)}.$$

Finally, given that the restricted polynomial in the second step has at least one good monomial and is δ-concentrated, the probability that it falls inside the interval I after taking an input from a $(192 \cdot d \cdot \log \delta^{-1})$-wise independent distribution is at most δ by Claim 3.79. Therefore, by noting $\delta = m^{-1/2}$, the probability that an input obtained in the above process makes p fall inside I is at most

$$1/t + \left(m^{-1/2}\right) \cdot (\log m)^{O\left(\zeta^{-1} \cdot d^2\right)} + m^{-1/2} \leq t^{-1/(2d)} \cdot (\log t)^{O\left(\zeta^{-1} \cdot d^2\right)}.$$

This completes the proof. □

Next, we show another derandomized anticoncentration bound that is quantitatively better when the polynomials are dense (i.e., have many good monomials).

Theorem 3.80. *For any positive integers t and d, and $0 < \zeta < 1$, there exists a distribution D on $\{-1,1\}^n$, samplable in $\mathsf{poly}(n)$ time using $t^{\zeta} \cdot \log n$ random bits, such that the following holds. For any real interval I, and any n-variate degree-d multilinear polynomial p with at least $t \cdot n^{d-1}$ degree-d monomials whose coefficients are at least $|I|$ in magnitude, we have*

$$\mathbf{Pr}\left[p(D) \in I\right] \leq t^{-\frac{1}{2}} \cdot (\log t)^{O\left(\zeta^{-1} \cdot d^2\right)}.$$

Remark 3.81. *For dense polynomials with $t = n^{1-o(1)}$ and any $\varepsilon > (C \cdot d^2 \cdot \log\log n)/\log n$, where C is some constant, setting*

$$\zeta = (C \cdot d^2 \cdot \log\log n)/(\varepsilon \cdot \log n),$$

we get that the bound in Theorem 3.80 is at most $n^{-1/2+o(1)+\varepsilon}$, matching the bound in Theorem 3.62 up to the $n^{o(1)+\varepsilon}$ factor, and that the seed length is at most $(\log n)^{O\left(\varepsilon^{-1} \cdot d^2\right)}$. Such a short seed is beyond reach of the naive derandomization using the PRG from Theorem 3.66, when the error is inverse-polynomially small.

To show Theorem 3.80, we a gain use bounded-independent hashing to partition the variables.

Lemma 3.82. *Let p be a n-variate degree-d multilinear polynomial with at least $t \cdot n^{d-1}$ good monomials. If the variables are partitioned into $m = t/(C \log t)$ blocks using a random hash function from a $(Cd \log t)$-wise independent hash family, where C is a constant, then the probability that the partition is not good is at most $1/t$.*

Proof. We first consider using full randomness to partition the variables. It will be convenient to view a set of variables as an n-bit characteristic string, where a coordinate i is 1 if the ith variable is in the set, and 0 otherwise. For a set S, we will also use S to denote its characteristics string. Let $K = (C/2) \log t$, and let p be so that $1 - (1-p)^K = 1/m$.

Consider a random set U that picks each variable independently with probability p. Note that U can be viewed as a random n-bit string such that each coordinate is 1 with probability p. Also, given U, we can compute the number of good monomials contained in U, using a degree-d polynomial, which is simply the sum of the $t \cdot n^{d-1}$ monomials of p. Let q denote this polynomial and let $\mu = \mathbf{E}[q(U)]$. Now define $r(U) = (q(U) - \mu)^2$. Note that r is a polynomial of degree at most $2d$ and, given our value of p, we have

$$\mu^2 > 2 \cdot \mathbf{E}[r(U)]. \tag{3.69}$$

Also, if U does not contain any good monomial, then $r(U) = \mu^2$.

Let $U_1, \ldots, U_K$ be K independent random sets, where each U_i picks each variable independently with probability p. Let T be a random set that picks each variable independently with probability $1/m = 1 - (1-p)^K$. Note that $U_1 \cup \cdots \cup U_K$ and T have the same distribution. Given a set T, consider the following way of picking a tuple of K random subsets $V^T = V_1^T, \ldots, V_K^T$ of T: pick V^T from the distribution of $U_1, \ldots, U_K$, conditioned on $U_1 \cup \cdots \cup U_K = T$. Now define f as

$$f(T) = \mathop{\mathbf{E}}_{V^T} \left[\prod_{i=1}^{K} r(V_i^T) \right].$$

Note that f can be written as a polynomial of degree at most $2dK$. To see this, consider the following equivalent way of picking V^{T_0} for some T_0: for each variable that appears in T_0, we assign it to each of the K subsets with probability p, conditioned on at least one subset containing the variable. Now consider picking a tuple of K random sets $W = W_1, \ldots, W_K$ in the above way, with $T_0 = \{1, \ldots, n\}$. Then it is easy to see that, for any given set T, $W \cap T = W_1 \cap T, \ldots, W_1 \cap T$ (i.e., after we pick W we remove all the variables that are not in T) and V^T have the same distribution. Therefore, we have

$$f(T) = \mathop{\mathbf{E}}_{W} \left[\prod_{i=1}^{K} r(W_i \cap T) \right],$$

which is clearly a polynomial of degree at most $2dK$ since r is of degree at most $2d$. Note that since each V_i^T is a subset of T, if T does not contain any good monomial, then

$$f(T) = \mu^{2K}. \tag{3.70}$$

Also, by the definition of the distribution for V^T, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{T}[f(T)] &= \mathop{\mathbf{E}}_{T}\left[\mathop{\mathbf{E}}_{V^T}\left[\prod_{i=1}^{K} r(V_i^T)\right]\right] \\
&= \mathop{\mathbf{E}}_{U_1,\ldots,U_K}\left[\prod_{i=1}^{K} r(U_i)\right] \\
&= \prod_{i=1}^{K} \mathbf{E}\left[r(U_i)\right] \\
&< \mu^{2K}/2^K, \qquad (3.71)
\end{aligned}$$

where the last inequality is by Equation (3.69).

Now consider partitioning the n variables into m blocks, using a $(Cd\log t)$-wise independent hash family $\mathcal{H}$. Let D be a random n-bits string such that the coordinates are $(2dK)$-wise independent and each coordinate is 1 with probability $1/m$. Let T be a random set that takes each variable independently with probability $1/m$. Then, for a block $\ell \in [m]$, we have

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{h\sim\mathcal{H}}[\ell \text{ does not contain any good monomial under } h] \\
&\leq \mathop{\mathbf{Pr}}_{D}[f(D) = \mu^{2K}] && \text{(by Equation (3.70))} \\
&\leq \mathop{\mathbf{Pr}}_{D}\left[f(D) > 2^K \cdot \mathop{\mathbf{E}}_{T}[f(T)]\right] && \text{(by Equation (3.71))} \\
&= \mathop{\mathbf{Pr}}_{D}\left[f(D) > 2^K \cdot \mathop{\mathbf{E}}_{D}[f(D)]\right] \\
&\leq 2^{-K} \\
&\leq t^{-C/2},
\end{aligned}$$

where the forth line above is by the fact that f is a polynomial of degree at most $2dK = Cd\log t$ and that D is $(Cd\log t)$-wise independent, and the second last line is by Markov's inequality. Finally, by the union bound over the m blocks, and for C sufficiently large, we get that the probability that there exists one block that does not contain any good monomial is at most $1/t$. □

Given Lemma 3.82, Theorem 3.80 is now proved in the same way as Theorem 3.76.

3.8 Open problems

We proved a restriction lemma for PTFs of degree $d \geq 1$, and used it to derive new lower bounds against constant-depth circuits with PTF gates. What are other applications of the (derandomized) PTF Restriction Lemma? For example, can it be used to get a

PRG for constant-depth PTF circuits? Finally, what are the applications of derandomized Littlewood-Offord type anticoncentration bounds?

Chapter 4

Satisfiability and Derandomization for Small Polynomial Threshold Circuits

4.1 Background and results

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 chapter, we give several algorithmic results for these problems for the class of Boolean circuits with polynomial threshold functions (PTFs) as gates.

We next describe in more detail the the class of PTF circuits for which we consider in this work. We then state our main results, and discuss our techniques.

PTF circuits. The focus of the present work is on circuits whose gates are polynomial threshold functions. Recall that 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 $\prod_{i \in S}(x_i \oplus b_i)$ 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).

[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, 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.

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 a degree-Δ PTF circuit if its gates are degree-Δ PTFs. Similarly, a circuit is called an 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 thesis 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).

4.1.1 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 4.1 (#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 $2^{n-n^{\varepsilon_d}} \cdot \mathsf{poly}(n)$, where $\varepsilon_d = c^{-3^d}$.

[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.

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.

Theorem 4.2 (#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 $2^{n-n^{\varepsilon_{d,\Delta}}} \cdot \mathsf{poly}(n)$, where $\varepsilon_{d,\Delta} = (c\cdot\Delta^2)^{-d}$.

Quantified derandomization for PTF circuits with $n^{1+\varepsilon}$ wires in quasi-polynomial time. 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 [Tel17b]) 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).

We give a quantified derandomization algorithms for the class of PTF circuits with super-linear wire complexity.

Theorem 4.3 (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. Finally, we construct a nontrivial pseudo-random generator (PRG) for PTF circuits (of unrestricted depth) with sub-linearly many gates

Theorem 4.4 (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 $s = n^{\frac{1}{\alpha+1}}/\left(E \cdot 5^{\alpha\cdot\Delta} \cdot \log^2(n) \cdot \log(n/\varepsilon)\right)$ 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 4.5 (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*

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

4.1.2 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 [CSS18] 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 [CSS18]. 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 [Wil14a] 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. [CSS18] 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 [CSS18], 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 [CS15a].

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 [CSS18], 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 gates 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 variables 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 4.5, 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 4.6) 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 $1/2 + \varepsilon/n$, by (possibly the negation of) the function

$$g(z) = f(h_1(z), h_2(z), \ldots, h_i(z), b_{i+1}, \ldots, b_n), \tag{4.1}$$

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 4.5 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 (4.1), 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 $1/2 + \varepsilon/n$ 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.

4.1.3 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 [CS15a]. A Circuit-SAT algorithm for constant-depth LTF circuits with few wires was recently given by Chen, Santhanam and Srinivasan [CSS18]. Alman, Chan and Williams [ACW16] showed a fast satisfiability algorithm for $\mathsf{ACC}^0 \circ \mathrm{LTF} \circ \mathrm{LTF}$ circuits with sub-quadratic number of LTF gates on the bottom layer and sub-exponential number of gates on the other layers, and hence obtained lower bounds against these circuits for an explicit function in E^{NP}, using the connection between satisfiability algorithms and circuit lower bounds due to Williams [Wil13, Wil14b]. Also, Tamaki [Tam16] has recently showed a fast satisfiability algorithm and lower bounds for depth-2 LTF circuits with sub-quadratic number of gates.

The most closely related previous work is by Chen, Santhanam and Srinivasan [CSS18] 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 4.2 for $\Delta = 1$ subsumes the Circuit-SAT algorithm for LTFs in [CSS18]. 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 [CSS18].

Recently, building on the work of [CSS18, KL18], Bajpai et al. [BKK$^+$19] obtained a similar #SAT algorithm, but for constant-depth circuits with a super-linear number of wires whose gates are *constant-degree* PTFs. Such an algorithm can also be adapted to the case where the PTF gates are *polynomially sparse*, and hence subsumes our #SAT algorithm for *sub-quadratically sparse* PTF circuits.

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 [Tel17b] 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 spar-

sity). For the results on reducing standard derandomization to quantified derandomization, see [GW14, Tel17b, Tel17a, 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 4.5) 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 have polynomial dependence in the error $1/\varepsilon$ 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 4.5) 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.

Organization of this chapter. We give the necessary background in Section 4.2. We describe our satisfiability algorithms (of Theorem 4.1 and Theorem 4.2) in Section 4.3. We prove our quantified derandomization results (Theorem 4.3) in Section 4.4, and our PRG result (Theorem 4.4) in Section 4.5. We conclude with some open problems in Section 4.6.

4.2 Preliminaries

Notation

Throughout this chapter, we will use $\{0,1\}$ as the Boolean domain.

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 $\mathsf{poly}(s)$.

Useful tools for analyzing PTFs

We review some definitions and useful results from Chapter 3.

Definition 4.6 (δ-concentrated PTFs)**.** *Let $p\colon \{0,1\}^n \to \mathbb{R}$ be a degree-Δ multi-linear polynomial and $f = \mathrm{sgn}(p)$. For parameters $0 < \delta \leq 1/2$ and $\lambda \geq 1$, we call p (and f)* (δ,λ)-concentrated *if*

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

where $\mathbf{E}$ *and* $\mathbf{Var}$ *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 4.7 (Concentrated implies close to constant, Lemma 3.26 restated)**.** *For any $0 < \delta \leq 1/2$, if a PTF $f = \mathrm{sgn}(p)$ is δ-concentrated, then f is δ-close to the constant function* $\mathrm{sgn}\left(\mathbf{E}[p]\right)$.

For a multi-linear polynomial $p\colon \{0,1\}^n \to \mathbb{R}$, it easy to see that $\mathbf{E}[p] = p(1/2,\ldots,1/2)$, and so the expectation of p can be *computed efficiently* given access to p (either the evalua-

tion oracle for p, or the full description of p via its coefficients). Thus, the constant function $\mathrm{sgn}(\mathbf{E}[p])$ from Lemma 4.7 is efficiently computable for a given polynomial p.

Recall that 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.

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 4.8 (Random block restriction lemma, Lemma 3.37 restated [3])**.** *For any $0 < \delta < 1$ and any positive 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*

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

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

Lemma 4.9 (Pseudorandom block restriction lemma, Theorem 3.64 restated)**.** *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*

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

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 4.10 (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 $n^{-\alpha}$.*

[3]The original result in Lemma 3.37 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 $\{0,1\}^n$.

- *$\mathcal{R}$ 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

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

Proof. Let M be any monomial in f.

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

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

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

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

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. □

4.3 #SAT algorithm for PTF circuits

In this section, we present our counting (#Circuit-SAT) algorithm in Theorem 4.1 for circuits with sparse PTF gates without any degree restriction on the monomials. We start with some useful tools.

Definition 4.11 (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 $x \in \{0,1\}^n$, $\mathbf{Pr}_{p\sim\mathbf{P}}[f(x) \neq p(x)] \leq \varepsilon$. 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 4.12 (Randomness-efficient probabilistic polynomials for LTFs [Sri13, Tam16])**.** *For any $0 < \varepsilon < 1/2$ and LTF $f\colon \{0,1\}^n \to \{0,1\}$, f has a ε-error probabilistic polynomial $\mathbf{P}$ of degree at most $d = O\left(\sqrt{n \cdot \log(1/\varepsilon)} \cdot \log^5 n\right)$. Moreover, $\mathbf{P}$ is samplable in time $\binom{n}{d} \cdot \mathsf{poly}(n)$ using $O\left(\log^2(n/\varepsilon)\right)$ random bits.*

Theorem 4.13 (Randomness-efficient probabilistic polynomials for AND/OR [Raz87, CW16])**.** *For any $0 < \varepsilon < 1/2$, AND/OR on n variables has a ε-error probabilistic polynomial $\mathbf{P}$ of degree at most $d = O\left(\log(1/\varepsilon)\right)$. Moreover, $\mathbf{P}$ is samplable in time $\binom{n}{d} \cdot \mathsf{poly}(n)$ using $O\left(\log n \cdot \log(1/\varepsilon)\right)$ random bits.*

We need the following useful tools for analyzing polynomials.

Lemma 4.14 (Fast multi-point polynomial evaluation [Yat37, Wil14b])**.** *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 4.15 (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.

Lemma 4.16 (see [CS15b, Section 4.1])**.** *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 $i \in [s]$.*

Lemma 4.17 ([CSS18, Proposition 6.2])**.** *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 $|S| \cdot \mathsf{poly}(n)$*

4.3.1 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 [Wil14b]. 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 4.18. *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*

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

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

Lemma 4.19. *For any $0 < \varepsilon < 1/2$ 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 $d = \sqrt{s}\cdot(\log s)^{O(1)}\cdot\log k\cdot\log(1/\varepsilon)$. Moreover, $\mathbf{P}$ is samplable in time $\binom{n}{d}\cdot\mathsf{poly}(n)$ using $(\log s)^{O(1)}\cdot\log k\cdot\log(1/\varepsilon)$ 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 4.20. *Each f_i, $i \in [k]$, has a ε-error probabilistic polynomial of degree at most $d = \sqrt{s}\cdot(\log s)^{O(1)}\cdot\log(1/\varepsilon)$ that is samplable in time $\binom{n}{d}\cdot\mathsf{poly}(n)$ using $(\log s)^{O(1)}\cdot\log(1/\varepsilon)$ 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 4.13. 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 4.12. 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 $O(\log(1/\varepsilon))$ 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 $\sqrt{s} \cdot (\log s)^{O(1)} \cdot \log(1/\varepsilon)$. □

By Claim 4.20, 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 $\sqrt{s} \cdot (\log s)^{O(1)} \cdot \log k \cdot \log(1/\varepsilon)$. □

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

Proof of Lemma 4.18. 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 4.19. 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 4.15. 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 4.14) 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

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

Setting

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

completes the proof. □

4.3.2 Depth reduction for sparse PTF circuits with few wires

In this section, we show how to use the random restriction lemma for PTFs (Lemma 4.8) to simplify a sparse PTF circuit with few wires.

Lemma 4.21. *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 [CSS18].

Lemma 4.22 (see [CSS18, Section 4.3])**.** *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 $g \in \mathcal{G}$,*

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

  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 4.21. The proof uses Lemma 4.22 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,

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

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 4.10, we have

$$\mathop{\mathbf{Pr}}_{\rho_1 \sim \mathcal{R}_{r_1}} [\deg(f_{\rho_1}) \geq D] \leq s \cdot n^{-c_2/8} \leq r^{\Omega(1)}. \tag{4.2}$$

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 4.8 and Lemma 4.7, 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

$$\sqrt{r_2} \cdot (\log(1/r_2) \cdot \log(1/\delta))^{O(D^2)} \leq n^{\Omega(\beta)} = r^{\Omega(1)}. \tag{4.3}$$

Finally, we have

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\rho \sim \mathcal{R}_{r_1 \cdot r_2}} [f_\rho \text{ is not } \delta\text{-close to an explicit constant}] \\
&\leq \mathop{\mathbf{Pr}}_{\rho_1, \rho_2} [(f_{\rho_1})_{\rho_2} \text{ is not } \delta\text{-close to an explicit constant} \mid \deg(f_{\rho_1}) < D] + \mathop{\mathbf{Pr}}_{\rho_1} [\deg(f_{\rho_1}) \geq D] \\
&\leq r^{\Omega(1)}. && \text{(by Equations (4.2) and (4.3))}
\end{aligned}$$

□

4.3.3 Enumerating minority outputs of sparse PTFs

In our main algorithm, we will need to apply the depth reduction lemma (Lemma 4.21) 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 Lemma 4.16 to reduce to the case of LTF, where we can perform this task efficiently using Lemma 4.17.

Lemma 4.23. *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 Lemma 4.16, 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 Lemma 4.17, 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 $\sum_\sigma S_\sigma \cdot \mathsf{poly}(n) \leq |S| \cdot \mathsf{poly}(n)$. □

4.3.4 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 4.24. *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 $2^{n-n^{\Omega\left(\beta_d^3\right)}} \cdot \mathsf{poly}(n)$.*

Definition 4.25 (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 [CSS18].

Lemma 4.26. *If $s \leq n^{2-5\beta_d}$, then*

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

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 4.21 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 4.21 are not satisfied, we will simply do brute force on all $n^{1-2\beta_d}$ variables. The time to perform this is

$$2^{n-n^{1-2\beta_d}} \cdot \exp(-n^{\varepsilon_d}) \cdot 2^{n^{1-2\beta_d}} \leq 2^{n-n^{\varepsilon_d}}. \tag{4.4}$$

Next, consider a "good" leaf σ that satisfies the conditions in Lemma 4.21. 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 4.23, the running time for this is

$$\left(2^{n^{1-2\beta_d} - \Omega\left(\frac{n^{2\cdot(1-2\beta_d)}}{s}\right)} + 2^{n^{1-2\beta_d}} \cdot \delta\right) \cdot \mathsf{poly}(n), \tag{4.5}$$

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 (4.5) is at most

$$2^{n^{1-2\beta_d} - n^{\Omega\left(\beta_d^3\right)}} \cdot \mathsf{poly}(n). \tag{4.6}$$

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

$$2^t \cdot \mathcal{T}\left(d-1, n^{1-2\beta_d}, t+2n, s\right) \leq 2^{n^{\beta_d}} \cdot \mathcal{T}\left(d-1, n^{1-2\beta_d}, t+2n, s\right). \tag{4.7}$$

Therefore, by Equation (4.6) and Equation (4.7), counting the number of satisfying assignments of C_σ can be done in time

$$2^{n^{1-2\beta_d} - n^{\Omega\left(\beta_d^3\right)}} \cdot \mathsf{poly}(n) + 2^{n^{\beta_d}} \cdot \mathcal{T}\left(d-1, n^{1-2\beta_d}, t+2n, s\right). \tag{4.8}$$

There are at most $L = 2^{n - n^{1-2\beta_d}}$ such leaves. Multiplying L by the running time in Equation (4.8) and combining Equation (4.4) yields the desired running time. □

Given the recursion in Lemma 4.26, we are now ready to prove Theorem 4.24.

Proof of Theorem 4.24. It suffices to show

$$\mathcal{T}\left(d, n, 0, s = n^{2-10\beta_2}\right) \leq 2^{n - n^{\Omega\left(\beta_d^3\right)}} \cdot \mathsf{poly}(n). \tag{4.9}$$

We will iteratively apply Lemma 4.26 until we reach $d = 1$. Then we use the base case algorithm (Lemma 4.18) 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

$$n_i \geq n^{1-2\cdot\sum_{j=i+1}^{d}\beta_j} \geq n^{1-4\beta_{i+1}}. \tag{4.10}$$

By Lemma 4.26, if $s \leq n^{2-5\cdot\beta_d}$, then

$$\mathcal{T}(d, n, t, s) \leq 2^{n-n_{d-1}} \cdot 2^{n^{\beta_d}} \cdot \mathcal{T}\left(d-1, n_{d-1}, t+2n, s\right) + 2^{n - n^{\Omega\left(\beta_d^3\right)}} \cdot \mathsf{poly}(n) \tag{4.11}$$

Now note that using Equation (4.10), 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 (4.11), we unwind $\mathcal{T}(d, n, 0, s)$ $d-1$ times, and we get

$$\mathcal{T}(d,n,0,s) \leq 2^{n-n_1}\cdot 2^{n_2^{\beta_2}}\cdot\mathcal{T}(1,n_1,k,s)+\sum_{i=3}^{d} 2^{n_i^{\beta_i}}\cdot 2^{n-n_{i-1}^{\Omega\left(\beta_{i-1}^3\right)}}\cdot\mathsf{poly}(n_i)+2^{n-n^{\Omega\left(\beta_d^3\right)}}\cdot\mathsf{poly}(n), \tag{4.12}$$

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

We now upper bound Equation (4.12). We first upper bound the first summand in Equation (4.12). By Lemma 4.18 and Equation (4.10), 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,

$$2^{n_i^{\beta_i}}\cdot 2^{n-n_{i-1}^{\Omega\left(\beta_{i-1}^3\right)}} \leq 2^{n_i^{\beta_i}}\cdot 2^{n-n_{i-1}^{E\cdot\beta_i}} \leq 2^{n_i^{\beta_i}}\cdot 2^{n-n^{E\cdot\beta_i/2}} \leq 2^{n-n^{\Omega\left(\beta_d^3\right)}}. \tag{4.13}$$

Therefore, the second summand is also bounded by

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

□

4.3.5 Circuits with gates that are LTFs of few functions

In this section, we describe our #Circuit-SAT algorithm in Theorem 4.2 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 4.27. *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 assign-*

ments of every such circuit C and runs in time at most

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

The proof of the above algorithm is similar to that of Lemma 4.18. We first show that a conjunction of k functions from the class $\mathcal{G}_{s,\Delta}$ has a probabilistic polynomial of degree at most

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

as in the proof of Lemma 4.19. 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 4.13), 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 4.8, 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 4.21, we get that

Lemma 4.28. *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.

Lemma 4.29 (see [CS15b, Section 3.2])**.** *Let $\phi_1,\ldots,\phi_s$ be a sequence of arbitrary Δ-constraints 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 Lemma 4.16, we have the following which is analogous to Lemma 4.23

Lemma 4.30. *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 4.3.4 with a different settings of parameters, which now are $\varepsilon_{d,\Delta}$ and $\beta_{d,\Delta}$.

4.4 Quantified derandomization for PTF circuits

In this section, we prove our quantified derandomization results.

4.4.1 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 4.9). We first show for low-degree PTFs.

Lemma 4.31 (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,*

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

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 4.9 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 4.9 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 4.9, 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,

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

□

To obtain a similar pseudorandom restriction lemma for sparse PTFs, we combine the pseudorandom restriction lemma for low-degree PTFs (Lemma 4.31) and the degree reduction lemma (Lemma 4.10) to get the following pseudorandom restriction lemma for sparse PTFs.

Lemma 4.32 (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,*

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

Proof. The idea is first applying a random restriction from Lemma 4.10 to reduce the degree of the sparse PTFs, and then using a random restriction from Lemma 4.31 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 4.10 (with parameters

$D = 4\Delta$ and $\alpha = \alpha_2/2$), we have

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

Let ρ_2 be a random restriction from Lemma 4.31 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

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

as desired. □

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 4.33 (see the proofs of Lemma 3.35 and Claim 3.70)**.** *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 $(192 \cdot \Delta \cdot \log \delta^{-1})$-wise independent distribution. Then with probability at least $1-\delta$ we have*

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

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.

4.4.2 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 $\mathrm{Appr}_{n,\varepsilon}(\mathcal{G})$ 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 4.34. *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 4.34 implies Theorem 4.3 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 4.32) to reduce the depth of the circuit until the circuit has depth 1. We first show how to do this in one step.

Lemma 4.35. *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 4.35 is similar to that in [Tel18], which is based on the argument in [CSS18], 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 4.32 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 4.32, 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 [CSS18, 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 4.33, the probability that each of these gates 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 4.33, 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 4.34. For $1 \leq i \leq d$, 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 $1 \leq i \leq d$, $\sum_{j=i+1}^{d} \varepsilon_j \leq \varepsilon_i$. Then we have

$$n_i \geq n^{1-3E\cdot\sum_{j=i+1}^{d}\varepsilon_j} \geq n^{1-6E\cdot\varepsilon_{i+1}}. \tag{4.14}$$

Let ρ be a sequence of d random restriction $\rho_1, \cdots, \rho_d$, such that ρ_i is a random restriction from Lemma 4.35 with parameters ε_i and $c = 1$. We say that the restriction ρ_i is successful if the two items in Lemma 4.35 are satisfied.

We claim the following.

Claim 4.36. *Let $K_d = \sum_{i=2}^{d} n_i^{-1} \leq 1/10$. The probability, over ρ, that C_ρ is not (K_d)-close to a $\mathcal{G}$ function is at most $2 \cdot \sum_{i=2}^{d} n_i^{-\varepsilon_i} \leq 1/3$.*

Proof of claim. We prove by induction on the depth d. The base case $d = 2$ follows from Lemma 4.35. Now suppose the claim holds for $d-1$, we show that it holds for d. We have

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\rho}[C_\rho \text{ is not } K_d\text{-close to } \mathcal{G}] \\
\leq &\mathop{\mathbf{Pr}}_{\rho_1,\rho'=\rho_2,\ldots,\rho_d}[(C_{\rho_1})_{\rho'} \text{ is not } K_d\text{-close to } \mathcal{G} \mid \rho_1 \text{ is successful}] + \mathop{\mathbf{Pr}}_{\rho_1}[\rho_1 \text{ is not successful}] \\
\leq &\mathop{\mathbf{Pr}}_{\rho_1,\rho'}[(C_{\rho_1})_{\rho'} \text{ is not } K_d\text{-close to } \mathcal{G} \mid \rho_1 \text{ is successful and } \rho' \text{ is good}] \cdot \mathop{\mathbf{Pr}}_{\rho'}[\rho' \text{ is good}] \\
&+ \mathop{\mathbf{Pr}}_{\rho'}[\rho' \text{ is not good}] + n^{-\varepsilon_d} \\
\leq &\mathop{\mathbf{Pr}}_{\rho_1,\rho'}[(C_{\rho_1})_{\rho'} \text{ is not } K_d\text{-close to } \mathcal{G} \mid \rho_1 \text{ is successful and } \rho' \text{ is good}] \cdot \mathop{\mathbf{Pr}}_{\rho'}[\rho' \text{ is good}] \\
&+ 2 \cdot n^{-\varepsilon_d}.
\end{aligned} \tag{4.15}$$

Now if ρ_1 is successful and ρ' is good, then by Lemma 4.35, $(C_{\rho_1})_{\rho'}$ is n^{-1}-close to some circuit $\tilde{C}_{\rho'}$, where $\tilde{C}$ is a $\left(n', d-1, (n')^{1+7E\cdot\varepsilon_d}, (n')^{2\Delta\cdot\varepsilon_d}, \mathcal{G}\right)$-sparse PTF circuit, with

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

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 (4.15) is at most

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

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 $n_1 \geq n^{1-6/E}$ 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 $\left(\frac{1}{10}\right)$-close to C_ρ. Then it must accept have at most

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

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. □

4.4.3 Quantified derandomization for low-degree PTF circuits

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

Theorem 4.37. *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 $(n, d, n^{1+\varepsilon_d}, \Delta, \mathcal{G})$-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 4.34, using the following one-step pseudorandom restriction for low-degree PTF circuits.

Lemma 4.38. *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 $(n, d, n^{1+\varepsilon}, \Delta, \mathcal{G})$-low-degree PTF circuit C and a random seed of length $(\log n)^{O(\Delta^2)}$, 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 $\Big(n', d-1, (n')^{1+7E\cdot\varepsilon}, \Delta, \mathcal{G}\Big)$-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 4.35 for the sparse PTF gate case, but uses Lemma 4.31 instead of Lemma 4.32. Given Lemma 4.38, we can prove Theorem 4.3 in the same way as proving Theorem 4.3 in the previous section.

4.5 PRG for PTF circuits

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

Theorem 4.39. *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 $s = n^{\frac{1}{\alpha+1}} \cdot \Big(E \cdot 5^{\alpha\cdot\Delta} \cdot \log^2(n) \cdot \log(n/\varepsilon)\Big)^{-1}$ 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 4.40 ([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 $n \cdot \Big(E \cdot 5^{\Delta} \cdot \log^2(n) \cdot \log(1/\varepsilon)\Big)^{-1}$ gates, we have*

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

Next we apply the Nisan-Wigderson construction to the hard function of Theorem 4.40. We will use the following (standard) combinatorial designs.

Claim 4.41 (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 4.39. 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 4.40. By Theorem 4.40 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,

$$\mathop{\mathbf{Pr}}_{z \sim \{0,1\}^\ell}[D(z) = f(z)] \leq \frac{1}{2} + \varepsilon/n. \tag{4.16}$$

Let $S_1, \ldots, S_n$ be the sets from Claim 4.41. 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

$$|\mathop{\mathbf{Pr}}_{x \sim \{0,1\}^n}[C(x) = 1] - \mathop{\mathbf{Pr}}_{y \sim \{0,1\}^r}[C(G_{\alpha,\Delta}(y)) = 1]| > \varepsilon. \tag{4.17}$$

By a standard argument via "reduction from distinguishing to predicting" as in [NW94], Equation (4.17) implies that there exist an $i \in [n]$, and bits $b_{i+1}, \ldots, b_n \in \{0,1\}$, such that

$$\mathop{\mathbf{Pr}}_{z \sim \{0,1\}^\ell}[C'(h_1(z), \ldots, h_i(z), b_{i+1}, \ldots, b_n) = f(z)] > 1/2 + \varepsilon/n, \tag{4.18}$$

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 multilinear 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 (4.18), this PTF circuit C'' computes the function f with probability greater than $1/2 + \varepsilon/n$, contradicting Equation (4.16). □

Next, we show how to obtain the PRG in Corollary 4.5 for PTFs from the result in Theorem 4.39.

Proof of Corollary 4.5. In order to make sure our PRG in Theorem 4.39 fools any degree-Δ PTF, we need to choose a value for α so that the size s in Theorem 4.39 is at least 1, where

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

Let's pick α so that the seed length of our PRG in Theorem 4.39 is

$$r = n^{2/(\alpha+1)} = 2^{L \cdot \sqrt{\Delta \cdot \log n}} \cdot \log^2(1/\varepsilon), \tag{4.19}$$

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

$$n^{\frac{1}{\alpha+1}} = r^{1/2} = 2^{\frac{L}{2} \cdot \sqrt{\Delta \cdot \log n}} \cdot \log(1/\varepsilon). \tag{4.20}$$

On the other hand, Equation (4.19) implies that

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

Plugging the above into the denominator of s, we get

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

which is less than the numerator of s given in Equation (4.20). □

4.6 Open problems

An 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?

Chapter 5

Algorithms and Lower Bounds for Formulas of Low-Communication Leaf Gates

5.1 Background and results

A (De Morgan) Boolean formula over $\{0,1\}$-valued input variables $x_1, \ldots, x_n$ is a binary tree whose internal nodes are labeled by AND or OR gates, and whose leaves are marked with a variable or its negation. The power of Boolean formulas has been intensively investigated since the early years of complexity theory (see, e.g., [Sub61, Nec66, Khr71, And87, PZ93, IN93, Hås98, Tal14, DM18]). The techniques underlying these complexity-theoretic results have also enabled algorithmic developments. These include learning algorithms [Rei11b, ST17], satisfiability algorithms (cf. [Tal15]), compression algorithms [CKK$^+$15], and the construction of pseudorandom generators [IMZ19] for Boolean formulas of different sizes. But despite many decades of research, the current non-trivial algorithms and lower bounds apply only to formulas of less than cubic size, and understanding larger formulas remains a major open problem in circuit complexity.

In many scenarios, however, understanding smaller formulas whose leaves are replaced by certain functions would also be very useful. Motivated by several recent works, we initiate a systematic study of the $\mathsf{FORMULA} \circ \mathcal{G}$ model, i.e., Boolean formulas whose leaves are labelled by an arbitrary function from a fixed class $\mathcal{G}$. This model unifies and generalizes a variety of models that have been previously studied in the literature:

– Oliveira, Pich, and Santhanam [OPS19] show that proving certain lower bounds against formulas of size $n^{1+\varepsilon}$ over parity (XOR) gates would have significant consequences in complexity theory. Note that de Morgan formulas of size $n^{3+\varepsilon}$ can simulate

such devices. Therefore, a better understanding of the $\mathsf{FORMULA} \circ \mathcal{G}$ model even when $\mathcal{G} = \mathsf{XOR}$ is *necessary* before we are able to analyze super-cubic size formulas.[1]

– Tal [Tal17b] obtains almost quadratic lower bounds for the model of bipartite formulas, where there is a fixed partition of the input variables into $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$, and a formula leaf can compute an *arbitrary* function over either $\vec{x}$ or $\vec{y}$. This model was originally investigated by Pudlák, Rödl, and Savický [PRS88], where it was referred to as graph complexity. The model is also equivalent to PSPACE-protocols in communication complexity (cf. [GPW18]).

– Abboud and Bringmann [AB18] consider formulas where the leaves are threshold gates whose input wires can be arbitrary functions applied to either the first or the second half of the input. This extension of bipartite formulas is denoted by $\mathcal{F}_2$ in [AB18]. Their work establishes connections between faster $\mathcal{F}_2$-SAT algorithms, the complexity of problems in P such as Longest Common Subsequence and the Fréchet Distance Problem, and circuit lower bounds.

– Polytopes (i.e. intersection of half-spaces), which corresponds to $\mathcal{G}$ being the family of linear-threshold functions, and the formula contains only AND gates as internal gates. The constructing of PRGs for this model has received significant attention in the literature (see [OST19] and references therein).

We obtain in a unified way several new results for the $\mathsf{FORMULA} \circ \mathcal{G}$ model, for natural classes $\mathcal{G}$ of functions which include parities, linear (and polynomial) threshold functions, and indeed many other functions of interest. In particular, we show that this perspective leads to stronger lower bounds, general satisfiability algorithms, and better pseudorandom generators for a broad class of functions.

5.1.1 Results

We now describe in detail our main results and how they contrast to previous works. Our techniques will be discussed in Section 5.1.2, while a few open problems are mentioned in Section 5.1.3.

We let $\mathsf{FORMULA}[s] \circ \mathcal{G}$ denote the set of Boolean functions computed by formulas containing at most s leaves, where each leaf computes according to some function in $\mathcal{G}$. The set of parity functions and their negations will be denoted by XOR.

We use the following notation for communication complexity. For a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, we let $D(f)$ be the two-party deterministic communication complexity of f, where each party is given an input of $n/2$ bits. Similarly, for a Boolean function

[1] We remark that even a single layer of XOR gates can compute powerful primitives, such as error-correcting codes and hash functions.

$g\colon \{0,1\}^n \to \{0,1\}$, we denote by $R^{(k)}_\delta(g)$ the communication cost of the best k-party *number-on-forehead* (NOF) communication protocol that computes g with probability at least $1-\delta$ on every input, where the probability is taken over the random choices of the protocol. For simplicity, we might omit the superscript (k) from $R^{(k)}_\delta(g)$ when $k=2$. One of our results will also consider k-party *number-in-hand* (NIH) protocols, and this will be clearly indicated in order to avoid confusion. We always assume a canonical partition of the input coordinates in all statements involving k-party communication complexity, unless stated otherwise. We generalize these definitions for a class of functions $\mathcal{G}$ in the natural way. For instance, we let $R^{(k)}_\delta(\mathcal{G}) = \max_{g\in\mathcal{G}} R^{(k)}_\delta(g)$.

Our results refer to standard notions in the literature, but in order to fix notation, Section 5.2 formally defines communication protocols, Boolean formulas, and other notions relevant in this work. We refer to the textbooks [KN97] and [Juk12] for more information about communication complexity and Boolean formulas, respectively. To put our results into context, here we only briefly review a few known upper bounds on the communication complexity of certain classes $\mathcal{G}$.

Parities (XOR) and Bipartite Formulas. Clearly, the deterministic two-party communication complexity of any parity function is at most 2, since to agree on the output it is enough for the players to exchange the parity of their relevant input bits. Moreover, note that the bipartite formula model discussed above precisely corresponds to formulas whose leaves are computed by a two-party protocol of communication cost at most 1.

Halfspaces and Polynomial Threshold Functions (PTFs). Recall that a halfspace, also known as a Linear Threshold Function (LTF), is a Boolean function of the form $\mathsf{sign}(\sum_i^n a_i \cdot x_i - b)$, where each $a_i, b \in \mathbb{R}$ and $x \in \{0,1\}^n$, and that a degree-d PTF is its natural generalization where degree-d monomials are allowed. It is known that if $g(x_1,\ldots,x_n)$ is a halfspace, then its randomized two-party communication complexity, namely $R^{(2)}_\delta(g)$, satisfies $R^{(2)}_\delta(g) = O(\log(n) + \log(1/\delta))$ [Nis94]. On the other hand, if $g(x_1,\ldots,x_n)$ is a degree-d PTF, then $R^{(d+1)}_\delta(g) = O((d\log d)(d\log n + \log(1/\delta)))$ [Nis94, Vio15].

Degree-d Polynomials over $\mathsf{GF}(2)$. It is well known that a degree-d $\mathsf{GF}(2)$-polynomial admits a $(d+1)$-party deterministic protocol of communication cost $d+1$ under *any* variable partition, since in the number-on-forehead model each monomial is entirely seen by some player. In particular, the Inner Product function $\mathsf{IP}_n(x,y) = \sum_i x_i \cdot y_i \pmod 2$ satisfies $R^{(3)}_{1/3}(\mathsf{IP}_n) = O(1)$.

Lower bounds

Prior to this work, the only known lower bound against $\mathsf{FORMULA} \circ \mathsf{XOR}$ or bipartite formulas was the recent result of [Tal17a] showing that IP_n is hard (even on average) against

nearly sub-quadratic formulas. In contrast, we obtain a significantly stronger result and establish lower bounds for different Boolean functions. We define such functions next.

GIP^k_n. The Generalized Inner Product function $\mathsf{GIP}^k_n\colon \{0,1\}^n \to \{0,1\}$ is defined as

$$\mathsf{GIP}^k_n\left(x^{(1)}, x^{(2)}, \ldots, x^{(k)}\right) = \sum_{j=1}^{n/k} \bigwedge_{i=1}^{k} x^{(i)}_j \pmod 2,$$

where $x^{(i)} \in \{0,1\}^{n/k}$ for each $i \in [k]$.

MKtP. In the Minimum Kt Problem, where Kt refers to Levin's time-bounded Kolmogorov complexity[2], we are given a string $x \in \{0,1\}^n$ and a string 1^ℓ. We accept $(x, 1^\ell)$ if and only if $\mathsf{Kt}(x) \leq \ell$.

MCSP. In the Minimum Circuit Size Problem, we are given as input the description of a Boolean function $f\colon \{0,1\}^{\log n} \to \{0,1\}$ (represented as an n-bit string), and a string 1^ℓ. We accept $(f, 1^\ell)$ if and only the circuit complexity of f is at most ℓ.

Theorem 5.1 (Lower bounds)**.** *The following unconditional lower bounds hold:*

1. *If GIP^k_n is $(1/2+\varepsilon)$-close under the uniform distribution to a function in $\mathsf{FORMULA}[s]\circ\mathcal{G}$, then*
$$s \;=\; \Omega\left(\frac{n^2}{k^2\cdot 16^k\cdot\left(R^{(k)}_{\varepsilon/(2n^2)}(\mathcal{G})+\log n\right)^2\cdot\log^2(1/\varepsilon)}\right).$$
2. *If $\mathsf{MKtP}\in\mathsf{FORMULA}[s]\circ\mathcal{G}$, then*
$$s \;=\; \widetilde{\Omega}\left(\frac{n^2}{k^2\cdot 16^k\cdot R^{(k)}_{1/3}(\mathcal{G})}\right).$$
3. *If $\mathsf{MCSP}\in\mathsf{FORMULA}[s]\circ\mathsf{XOR}$, then $s=\widetilde{\Omega}(n^2)$, where $\widetilde{\Omega}$ hides inverse $\mathsf{polylog}(n)$ factors.*

Observe that, while [Tal17a] showed that the Inner Product function IP_n is hard against sub-quadratic bipartite formulas, Theorem 5.1 Item 1 yields lower bounds against formulas whose leaves can compute bounded-degree PTFs and $\mathsf{GF}(2)$-polynomials, including IP_n. Previously, only sub-linear lower bounds were known [Nis94, Vio15] for circuits with PTF gates of similar degree.

[2]For a string $x\in\{0,1\}^*$, $\mathsf{Kt}(x)$ denotes the minimum value $|M|+\log t$ taken over M and t, where M is a machine that prints x when it computes for t steps, and $|M|$ is the description length of M according to a fixed universal machine U.

Let us now comment on the relevance of Items 2 and 3. Both MCSP and MKtP are believed to be computationally much harder than GIP^k_n. However, it is more difficult to analyze these problems compared to GIP^k_n because the latter is mathematically "structured", while the former problems do not seem to be susceptible to typical algebraic, combinatorial, and analytic techniques.

More interestingly, MCSP and MKtP play an important role in the theory of hardness magnification (see [OPS19, CJW19]). In particular, if one could show that MCSP restricted to an input parameter $\ell \leq n^{o(1)}$ is not in $\mathsf{FORMULA}[n^{1+\varepsilon}] \circ \mathsf{XOR}$ for some $\varepsilon > 0$, then it would follow that NP cannot be computed by Boolean formulas of size n^c, where $c \in \mathbb{N}$ is arbitrary. Theorem 5.1 makes partial progress on this direction by establishing the first lower bounds for these problems in the $\mathsf{FORMULA} \circ \mathcal{G}$ model. (We note that the proof of Theorem 5.1 Item 3 requires instances where the parameter ℓ is $n^{\Omega(1)}$.)

Pseudorandom generators

We also get pseudorandom generators (PRGs) against $\mathsf{FORMULA} \circ \mathcal{G}$ for various classes of functions $\mathcal{G}$. Recall that a PRG against a class of functions $\mathfrak{C}$ is a function G mapping short Boolean strings (seeds) to longer Boolean strings, so that every function in $\mathfrak{C}$ accepts G's output on a uniformly random seed with about the same probability as that for an actual uniformly random string. More formally, $G\colon \{0,1\}^\ell \to \{0,1\}^n$ is a PRG that ε-fools $\mathfrak{C}$ if for every Boolean function $h\colon \{0,1\}^n \to \{0,1\}$ in $\mathfrak{C}$, we have

$$\left| \mathop{\mathbf{Pr}}_{z \sim \{0,1\}^\ell}[h(G(z)) = 1] - \mathop{\mathbf{Pr}}_{x \sim \{0,1\}^n}[h(x) = 1] \right| \leq \varepsilon.$$

Furthermore, we require G to run in deterministic time $\mathsf{poly}(n)$ on an input string $z \in \{0,1\}^\ell$. The parameter $\ell = \ell(n)$ is called the seed length of the PRG and is the main quantity to be minimized when constructing PRGs.

There exists a PRG that fools formulas of size s and that has a seed of length $s^{1/3+o(1)}$ [IMZ19]. In particular, there are non-trivial PRGs for n-variate formulas of size nearly n^3. Unfortunately, such PRGs cannot be used to fool even linear size formulas over parity functions, since the naive simulation of these enhanced formulas by standard Boolean formulas requires size n^3. Moreover, it is not hard to see that this simulation is optimal: Andreev's function, which is hard against formulas of nearly cubic size (cf. [Hås98]), can be easily computed in $\mathsf{FORMULA}[O(n)] \circ \mathsf{XOR}$. Given that a crucial idea in the construction of the PRG in [IMZ19] (shrinkage under restrictions) comes from this lower bound proof, new techniques are needed in order to approach the problem in the $\mathsf{FORMULA} \circ \mathsf{XOR}$ model.

More generally, extending a computational model for which strong PRGs are known to allow parities at the bottom layer can cause significant difficulties. A well-known example is AC^0 circuits and their extension to AC^0-XOR. While the former class admits PRGs of

poly-logarithmic seed length (see e.g. [ST19]), the most efficient PRG construction for the latter has seed length $(1-o(1)) \cdot n$ [FSUV13]. Consequently, designing PRGs of seed length $\leq (1-\Omega(1)) \cdot n$ can already be a challenge. We are not aware of previous results on PRGs for $\mathsf{FORMULA} \circ \mathcal{G}$ for any non-trivial class $\mathcal{G}$.

By combining ideas from circuit complexity and communication complexity, we construct PRGs of various seed lengths for $\mathsf{FORMULA} \circ \mathcal{G}$, where $\mathcal{G}$ ranges from the class of parity functions to the much larger class of functions of bounded randomized k-party communication complexity.

Theorem 5.2 (Pseudorandom generators)**.** *Let $\mathcal{G}$ be a class of n-bits functions. Then,*

1. *In the context of parity functions, there is a* PRG *that ε-fools $\mathsf{FORMULA}[s] \circ \mathsf{XOR}$ of seed length*
$$\ell \;=\; O\big(\sqrt{s} \cdot \log(s) \cdot \log(1/\varepsilon) + \log(n)\big)\,.$$

2. *In the context of two-party randomized communication complexity, there is a* PRG *that ε-fools $\mathsf{FORMULA}[s] \circ \mathcal{G}$ of seed length*
$$\ell \;=\; n/2 + O\Big(\sqrt{s} \cdot \Big(R^{(2)}_{\varepsilon/(6s)}(\mathcal{G}) + \log(s)\Big) \cdot \log(1/\varepsilon)\Big)\,.$$
More generally, for every $k(n) \geq 2$, let $\mathcal{G}$ be the class of functions that have k-party number-in-hand (NIH) *$(\varepsilon/6s)$-error randomized communication protocols of cost at most $R^{(k\text{-}\mathsf{NIH})}_{\varepsilon/(6s)}$. There exists a* PRG *that ε-fools $\mathsf{FORMULA}[s] \circ \mathcal{G}$ with seed length*
$$\ell \;=\; n/k + O\Big(\sqrt{s} \cdot \Big(R^{(k\text{-}\mathsf{NIH})}_{\varepsilon/(6s)} + \log(s)\Big) \cdot \log(1/\varepsilon) + \log(k)\Big) \cdot \log(k).$$

3. *In the setting of k-party* NOF *randomized communication complexity, there is a* PRG *that ε-fools $\mathsf{FORMULA}[s] \circ \mathcal{G}$ of seed length*
$$\ell \;=\; n - \frac{n}{O\Big(\sqrt{s} \cdot k \cdot 4^k \cdot \Big(R^{(k)}_{\varepsilon/(2s)}(\mathcal{G}) + \log(n)\Big) \cdot \log(n/\varepsilon)\Big)}.$$

A few comments are in order. Under a standard connection between PRGs and lower bounds (see e.g. [Kab02]), improving the dependence on s in the seed length for $\mathsf{FORMULA}[s] \circ \mathsf{XOR}$ (Theorem 5.2 Item 1) would require the proof of super-quadratic lower bounds against $\mathsf{FORMULA} \circ \mathsf{XOR}$. We discuss this problem in more detail in Section 5.1.3. Note that the additive term $n/2$ is necessary in Theorem 5.2 Item 2, since the model computes in particular every Boolean function on the first $n/2$ input variables (i.e. a protocol of communication cost 1). Similarly, $\ell \geq (1-1/k) \cdot n$ in Theorem 5.2 Item 3. Removing the exponential dependence on k would also require advances in state-of-the-art lower bounds for multiparty communication complexity.

Theorem 5.2 Item 2 has an interesting implication for fooling a well-studied class of functions: *intersections of halfspaces*.[3] Note that an intersection of halfspaces is precisely a polytope, or equivalently, the set of solutions of a 0-1 integer linear program. Such objects have found applications in many fields, including optimization and high-dimensional geometry. After a long sequence of works on the construction of PRGs for bounded-weight halfspaces, (unrestricted) halfspaces, and generalizations of these classes,[4] the following results are known for the intersection of m halfspaces over n input variables. Gopalan, O'Donnell, Wu, and Zuckerman [GOWZ10] gave a PRG for this class for error ε with seed length

$$O\big(m \cdot \log(m/\varepsilon) + \log n\big) \cdot \log(m/\varepsilon)\big).$$

Note that the seed length of their PRG becomes trivial if the number of halfspaces is linear in n. More recently, O'Donnell, Servedio and Tan [OST19] constructed a PRG with seed length

$$\mathsf{poly}(\log(m), 1/\varepsilon) \cdot \log(n).$$

Their PRG has a much better dependence on m, but it cannot be used in the small error regime. For example, the seed length becomes trivial if $\varepsilon = 1/n$. In particular, before this work it was open to construct a non-trivial PRG for the following natural setting of parameters (cf. [OST19, Section 1.2]): intersection of n halfspaces with error $\varepsilon = 1/n$.

We obtain the following consequence of Theorem 5.2 Item 2, which follows from a result of Viola [Vio15] on the k-party *number-in-hand* randomized communication complexity of a halfspace.

Corollary 5.3 (Fooling intersections of halfspaces in the low-error regime)**.** *For every $n, m \in \mathbb{N}$ and $\varepsilon > 0$, there is a pseudorandom generator with seed length*

$$O\Big(n^{1/2} \cdot m^{1/4} \cdot \log(n) \cdot \log(n/\varepsilon)\Big).$$

that ε-fools the class of intersections of m halfspaces over $\{0,1\}^n$.

We note that the PRG from Theorem 5.2 Item 3 can fool, even in the exponentially small error regime, not only intersections of halfspaces, but also small formulas over bounded-degree PTFs.

Finally, Theorem 5.2 Item 2 yields the first non-trivial PRG for formulas over symmetric functions. Let SYM denote the class of symmetric Boolean functions on any number of input variables.

[3] Clearly, the intersection of s functions can be computed by an enhanced formula of size $s + 1$.

[4] We refer to the recent reference [OST19] for an extensive review of the literature in this area.

Corollary 5.4 (Fooling sub-quadratic formulas over symmetric gates)**.** *For every $n, s \in \mathbb{N}$ and $\varepsilon > 0$, there is a pseudorandom generator with seed length*

$$O\Big(n^{1/2} \cdot s^{1/4} \cdot \log(n) \cdot \log(1/\varepsilon)\Big).$$

that ε-fools n-variate Boolean functions in $\mathsf{FORMULA}[s] \circ \mathsf{SYM}$*.*

Prior to this work, Chen and Wang [CW19] proved that the number of satisfying assignments of an n-variate formula of size s over symmetric gates can be approximately counted to an additive error term $\leq \varepsilon \cdot 2^n$ in deterministic time $\exp(n^{1/2} \cdot s^{1/4+o(1)}\sqrt{(\log(n) + \log(s))})$, where $\varepsilon > 0$ is an arbitrary constant. While their upper bound is achieved by a white-box algorithm, Corollary 5.4 provides a (black-box) PRG for the same task.

Satisfiability algorithms

In the #SAT problem for a computational model $\mathcal{C}$, we are given as input the description of a computational device $D(x_1, \ldots, x_n)$ from $\mathcal{C}$, and the goal is to count the number of satisfying assignments for D. This generalizes the SAT problem for $\mathcal{C}$, where it is sufficient to decide whether D is satisfiable by some assignment.

In this section, we show that #SAT algorithms can be designed for a broad class of functions. We consider the $\mathsf{FORMULA} \circ \mathcal{G}$ model for classes $\mathcal{G}$ that admit two-party communication protocols of bounded cost. We establish a general result in this context which can be used to obtain algorithms for previously studied classes of Boolean circuits.

To put our #SAT algorithms for $\mathsf{FORMULA} \circ \mathcal{G}$ into context, we first mention relevant related work on the satisfiability of Boolean formulas. Recall that in the very restricted setting of CNF formulas, known algorithms run (in the worst-case) in time $2^{n-o(n)}$ when the input formulas can have a super-linear number of clauses (cf. [DH09]). On the other hand, for the class of general formulas, there is a better-than-brute-force algorithm for formulas of size almost n^3. In more detail, for any $\varepsilon > 0$, there is a deterministic #SAT algorithm for $\mathsf{FORMULA}[n^{3-\varepsilon}]$ that runs in time $2^{n-n^{\Omega(\varepsilon)}}$ [Tal15]. No results are known for formulas of cubic size and beyond, and for the reasons explained in Section 5.1.1, the algorithm from [Tal15] cannot even be applied to $\mathsf{FORMULA} \circ \mathsf{XOR}$.

Before stating our results, we discuss the input encoding in the #SAT problem for $\mathsf{FORMULA} \circ \mathcal{G}$. The top formula F is represented in some canonical way, while for each leaf ℓ of F, the input string contains the description of a protocol Π_ℓ computing a function in $\mathcal{G}$. Our results are robust to the encoding employed for Π_ℓ. Recall that a protocol for a two-party function is specified by a protocol tree and a sequence of functions, where each function is associated with some internal node of the tree and depends on $n/2$ input bits. Since a protocol of communication cost $o(n)$ has a protocol tree containing at most $2^{o(n)}$ nodes, it can be specified by a string of length $2^{n/2+o(n)}$. Our algorithms will run in time closer to 2^n, and using a fully explicit input representation for the protocols is not an issue.

Another possibility for the input representation is to use "computational efficient" protocols. Informally, the next bit messages of such protocols can be computed in polynomial time from the current transcript of the protocol and a player input. An advantage of this representation is that an input to our #SAT problem can be succinctly represented. We observe that these input representations can be generalized to randomized two-party protocols in natural ways. We refer to Section 5.2 for a formal presentation.

We obtain non-trivial satisfiability algorithms assuming upper bounds on the two-party deterministic and randomized communication complexities of functions in $\mathcal{G}$.

Theorem 5.5 (Satisfiability algorithms)**.** *The following results hold.*

1. *There is a deterministic* #SAT *algorithm for* $\mathsf{FORMULA}[s] \circ \mathcal{G}$ *that runs in time*

$$2^{n-t} \cdot \mathsf{poly}(n,s), \text{ where } t = \Omega\left(\frac{n}{\sqrt{s} \cdot \log^2(s) \cdot D(\mathcal{G})}\right).$$

2. *There is a randomized* #SAT *algorithm for* $\mathsf{FORMULA}[s] \circ \mathcal{G}$ *that runs in time*

$$2^{n-t} \cdot \mathsf{poly}(n,s), \text{ where } t = \Omega\left(\frac{n}{\sqrt{s} \cdot \log^2(s) \cdot R_{1/3}(\mathcal{G})}\right)^{1/2}.$$

Theorem 5.5 readily provides algorithms for many circuit classes. For instance, since one can effectively describe a randomized communication protocol for linear threshold functions [Nis94, Vio15], the algorithm from Theorem 5.5 Item 2 can be used to count the number of satisfying assignments of Boolean devices from $\mathsf{FORMULA}[n^{1.99}] \circ \mathsf{LTF}$.

Corollary 5.6 (#SAT algorithm for formulas of linear threshold functions)**.** *There is a randomized* #SAT *algorithm for* $\mathsf{FORMULA}[s] \circ \mathsf{LTF}$ *that runs in time*

$$2^{n-t} \cdot \mathsf{poly}(n,s), \text{ where } t = \Omega\left(\frac{n}{\sqrt{s} \cdot \log^2(s) \cdot \log(n)}\right)^{1/2}.$$

In connection with Corollary 5.6, prior to this work essentially two lines of research have been pursued. #SAT and/or SAT algorithms were known for bounded-depth circuits of almost-linear size whose gates can compute LTFs or sparse PTFs (see [KL18] and references therein), and for sub-exponential size bounded-depth ACC^0 circuits with two layers of LTFs at the bottom, assuming a sub-quadratic number of them in the layer next to the input variables (see [ACW16] for this result and further related work). Corollary 5.6 seems to provide the first non-trivial SAT algorithm that operates with unbounded-depth Boolean devices containing a layer with a sub-quadratic number of LTFs.

Theorem 5.5 can be seen as a generalization of several approaches to designing SAT algorithms appearing in the literature, which often employ ad-hoc constructions to convert

bottlenecks in the computation of devices from a class $\mathcal{C}$ into non-trivial SAT algorithms for $\mathcal{C}$. We observe that, before this work, [PW10] had made a connection between faster SAT algorithms for CNFs and the 3-party communication complexity of a specific function. Their setting is different though: it seems to work only for CNFs, and they rely on conjectured upper bounds on the communication complexity of a particular problem. More recently, [CW19] employed quantum communication protocols to design *approximate counting* algorithms for several problems.[5] In comparison to previous works, to our knowledge Theorem 5.5 is the first unconditional result that yields faster #SAT algorithms via communication complexity in a generic way. [6]

Learning algorithms

We describe a learning algorithm for the $\mathsf{FORMULA} \circ \mathsf{XOR}$ class in Leslie Valiant's challenging PAC-learning model [Val84]. Recall that a (PAC) learning algorithm for a class of functions $\mathcal{C}$ has access to labelled examples $(x, f(x))$ from an unknown function $f \in \mathcal{C}$, where x is sampled according to some (also unknown) distribution $\mathcal{D}$. The goal of the learner is to output, with high probability over its internal randomness and over the choice of random examples (measured by a confidence parameter δ), a hypothesis h that is close to f under $\mathcal{D}$ (measured by an error parameter ε). We refer to [KV94] for more information about this learning model, and to Section 5.2 for its standard formalization.

It is known that formulas of size s can be PAC-learned in time $2^{\widetilde{O}(\sqrt{s})}$ [Rei11b]. Therefore, formulas of almost quadratic size can be non-trivially learned from random samples of an arbitrary distribution. A bit more formally, we say that a learning algorithm is *non-trivial* if it runs in time $2^n/n^{\omega(1)}$, i.e., noticeably faster than the trivial brute-force algorithm that takes time $2^n \cdot \mathsf{poly}(n)$. Obtaining non-trivial learning algorithms for various circuit classes is closely connected to the problem of proving explicit lower bounds against the class [OS17] (see also [ST17] for a systematic investigation of such algorithms). We are not aware of the existence of non-trivial learning algorithms for super-quadratic size formulas. However, it seems likely that such algorithms exist at least for formulas of near cubic size. As explained in Section 5.1.1, this would still be insufficient for the learnability of classes such as (linear size) $\mathsf{FORMULA} \circ \mathsf{XOR}$.

We explore structural properties of $\mathsf{FORMULA} \circ \mathsf{XOR}$ employed in previous results and boosting techniques from learning theory to show that sub-quadratic size devices from this class can be PAC-learned in time $2^{O(n/\log n)}$.

[5] Recall that approximately counting satisfying assignments is substantially easier than solving #SAT, for which the fastest known algorithms run in time $2^{(1-o(1))n}$.

[6] It has been brought to our attention that Avishay Tal has independently discovered a SAT algorithm for bipartite formulas of sub-quadratic size (see the discussion in [AB18, Page 7]), which corresponds to a particular case of Theorem 5.5.

Theorem 5.7 (PAC-learning $\mathsf{FORMULA} \circ \mathsf{XOR}$ in sub-exponential time)**.** *For every constant $\gamma > 0$, there is an algorithm that* PAC *learns the class of n-variate Boolean functions $\mathsf{FORMULA}[n^{2-\gamma}] \circ \mathsf{XOR}$ to accuracy ε and with confidence δ in time $\mathsf{poly}(2^{n/\log n}, 1/\varepsilon, \log(1/\delta))$.*

We make a few comments on potential extensions of this result to larger classes of formulas and on the running time of this learning algorithm.

Note that a sub-exponential running time cannot be achieved for $\mathsf{FORMULA} \circ \mathcal{G}$ when we consider the communication complexity of $\mathcal{G}$. Again, the class is too large, for the same reason discussed in Section 5.1.1. It might still be possible to design a non-trivial learning algorithm in this case, but this would possibly require the introduction of new lower bound techniques for $\mathsf{FORMULA} \circ \mathsf{XOR}$.

In contrast to the algorithm mentioned above that learns (standard) formulas of size $s \leq n^{2-o(1)}$ in time $2^{\widetilde{O}(\sqrt{s})}$, the algorithm from Theorem 5.7 does not learn smaller formulas over parities in time faster than $2^{O(n/\log n)}$. We discuss this in more detail in Section 5.1.2 and Section 5.1.3.

Finally, we mention a connection to cryptography that provides a conditional upper bound on the size of $\mathsf{FORMULA} \circ \mathsf{XOR}$ circuits that can be learned in time $2^{o(n)}$. It is well known that if a circuit class $\mathcal{C}$ can compute pseudorandom functions (or some variants of this notion), then it cannot be learned in various learning models (see e.g. [KV94]). It has been recently conjectured that depth-two $\mathsf{MOD}_3 \circ \mathsf{XOR}$ circuits of linear size can compute weak pseudorandom functions of exponential security [BIP+18, Conjecture 3.7]. If this conjecture holds, then such circuits cannot be learned in time $2^{o(n)}$. Since MOD_3 gates over a linear number of input wires can be simulated by formulas of size at most $O(n^{2.8})$ [Ser17], under this cryptographic assumption it is not possible to learn $\mathsf{FORMULA}[n^{2.8}] \circ \mathsf{XOR}$ in time $2^{o(n)}$, even if the learner only needs to succeed under the uniform distribution.

5.1.2 Techniques

In order to explain our techniques, we focus for the most part on the design of PRGs for $\mathsf{FORMULA} \circ \mathcal{G}$ when $\mathcal{G}$ is of bounded two-party randomized communication complexity (a particular case of Theorem 5.2 Item 2). This proof makes use of various ingredients employed in other results. After sketching this argument, we say a few words about our strongest lower bound (Theorem 5.1 Item 1) and the satisfiability and learning algorithms (Theorem 5.5 and Theorem 5.7, respectively).

We build on a powerful result showing that any small De Morgan formula can be approximated pointwise by a low-degree polynomial:

(A) For every formula $F(y_1, \ldots, y_m)$ of size s, there is a polynomial $p(y_1, \ldots, y_m) \in \mathbb{R}[y_1, \ldots, y_m]$ of degree $O(\sqrt{s})$ such that $|F(a) - p(a)| \leq 1/10$ on every $a \in \{0,1\}^m$.

The only known proof of this result [Rei11b] relies on a sequence of works [BBC+01, LLS06, HLS07, FGG08, Rei09, ACR+10, RS12] on quantum query complexity, generalizing Grover's

search algorithm for the OR predicate [Gro96] to arbitrary formulas. The starting point of many of our results is a consequence of **(A)** which is implicit in the work of Tal [Tal17a].

(B) Let $\mathcal{D}$ be a distribution over $\{0,1\}^m$, and $F \in \mathsf{FORMULA}[s] \circ \mathcal{G}$. Then, for every function f,

$$\text{if } \Pr_{x\sim\mathcal{D}}[F(x) = f(x)] \geq 1/2 + \varepsilon \text{ then } \Pr_{x\sim\mathcal{D}}[h(x) = f(x)] \geq 1/2 + \exp(-t)$$

for some function h which is the XOR of at most t functions in $\mathcal{G}$, where $t = \widetilde{\Theta}(\sqrt{s}\cdot\log(1/\varepsilon))$.

Intuitively, if we could understand well enough the XOR of any small collection of functions in $\mathcal{G}$, then we can translate this into results for $\mathsf{FORMULA}[s] \circ \mathcal{G}$, as long as $s \ll n^2$. We adapt the techniques behind **(B)** to provide a general approach to constructing PRGs against $\mathsf{FORMULA} \circ \mathcal{G}$:

Main PRG Lemma. In order for a distribution $\mathcal{D}$ to ε-fool the class $\mathsf{FORMULA}[s] \circ \mathcal{G}$, it is enough for it to $\exp(-t)$-fool the class $\mathsf{XOR}_t \cdot \mathcal{G}$, where $t = \widetilde{\Theta}(\sqrt{s} \cdot \log(1/\varepsilon))$.

Recall that, in Theorem 5.2 Item 2, we consider a class $\mathcal{G}$ of functions that admit two-party randomized protocols of cost $R = R^{(2)}_{\varepsilon/6s}(\mathcal{G})$. It is easy to see that the XOR of any t functions from $\mathcal{G}$ is a function that can be computed by a protocol of cost at most $t\cdot R$. Thus the lemma above shows that it is sufficient to fool, to exponentially small error, a class of functions of bounded two-party randomized communication complexity. Moreover, since a randomized protocol can be written as a convex combination of deterministic protocols, it is possible to prove that fooling functions of bounded deterministic communication complexity is enough.

Pseudorandom generators in the two-party communication model have been known since [INW94]. Their construction exploits that the Boolean matrix associated with a function of small communication cost can be partitioned into a not too large number of monochromatic rectangles. We provide in Appendix A.2 a slightly modified and self-contained construction based on explicit extractors. It achieves the following parameters: There is an explicit PRG that δ-fools any n-bit function of two-party communication cost D and that has seed length $n/2 + O(D + \log(1/\delta))$. This PRG has non-trivial seed length even when the error is exponentially small, as required by our techniques. One issue here is that the INW PRG was only shown to fool functions with low *deterministic* communication complexity. To obtain our PRGs for $\mathsf{FORMULA} \circ \mathcal{G}$ when $\mathcal{G}$ admits low-cost *randomized* protocols, we first extend the analysis of the INW PRG to show that it also fools functions with low *randomized* communication complexity. Combining this construction with the aforementioned discussion completes the proof of Theorem 5.2 Item 2.

The argument just sketched reduces the construction of PRGs for $\mathsf{FORMULA} \circ \mathcal{G}$ when functions in $\mathcal{G}$ admit low-cost *randomized* protocols to the analysis of PRGs for functions that admit relatively low-cost *deterministic* protocols. Our lower bound proof for GIP^k_n in Theorem 5.1 Item 1 proceeds in a similar fashion. We combine statement **(B)** described above with other ideas to show:

Transfer Lemma (Informal). If a function correlates with some small formula whose leaf gates have low-cost *randomized* k-party protocols, then it also non-trivially correlates with some function that has relatively low-cost *deterministic* k-party protocols.

Given this result, we are able to rely on a strong average-case lower bound for IP_n^k against k-party deterministic protocols from [BNS92] to conclude that GIP_n^k is hard for $\mathsf{FORMULA} \circ \mathcal{G}$.

Our #SAT algorithms combine the polynomial representation of the top formula provided by **(A)**, for which we show that such a polynomial can be obtained *explicitly*, with a decomposition of the Boolean matrix at each leaf that is induced by a corresponding low-cost randomized or deterministic two-party protocol. A careful combination of these two representations allows us to adapt a standard technique employed in the design of non-trivial SAT algorithms (fast rectangular matrix multiplication) to obtain non-trivial savings in the running time.

Finally, our learning algorithm for $\mathsf{FORMULA} \circ \mathsf{XOR}$ is a consequence of statement **(B)** above coupled with standard tools from learning theory. In a bit more detail, since a parity of parities is just another parity function, **(B)** implies that, under any distribution, every function in $\mathsf{FORMULA}[n^{1.99}] \circ \mathsf{XOR}$ is weakly correlated with some parity function. Using the agnostic learning algorithm for parity functions of [KMV08], it is possible to weakly learn $\mathsf{FORMULA}[n^{1.99}] \circ \mathsf{XOR}$ in time $2^{O(n/\log n)}$. This weak learner can then be transformed into a (strong) PAC learner using standard boosting techniques [Fre90], with only a polynomial blow-up over its running time.

5.1.3 Concluding remarks

The main message of our results is that the *computational power* of a subquadratic-size top formula is *not* significantly enhanced by leaf gates of *low communication complexity*. We believe that the idea of decomposing a Boolean device into a computational part and a layer of communication protocols will find further applications in lower bound proofs and algorithm design.

One of our main open problems is to discover a method that can analyze $\mathsf{FORMULA}[s] \circ \mathcal{G}$ when $s \gg n^2$. For instance, is it possible to adapt existing techniques to show an explicit lower bound against $\mathsf{FORMULA}[n^{2.01}] \circ \mathcal{G}$, or achieving this is just as hard as breaking the cubic barrier for formula lower bounds? Results in this direction would be interesting even for $\mathcal{G} = \mathsf{XOR}$.

Finally, we would like to mention a few questions connected to our results and their applications. Is it possible to combine the techniques behind Corollary 5.3 and [OST19] to design a PRG of seed length $n^{o(1)}$ and error $\varepsilon = 1/n$ for the intersection of n halfspaces? Can we design a satisfiability algorithm for formulas over k-party number-on-forehead communication protocols? Is it possible to learn $\mathsf{FORMULA}[s] \circ \mathsf{XOR}$ in time $2^{\widetilde{O}(\sqrt{s})}$? (The learning

algorithm for formulas from [Rei11b] relies on techniques from [KKMS08], and it is unclear how to extend them to the case of $\mathsf{FORMULA} \circ \mathsf{XOR}$.)

Organization of this chapter. Theorem 5.1 Item 1 is proved in Section 5.3, while Items 2 and 3 rely on our PRG constructions and are deferred to Section 5.4. The latter describes a general approach to constructing PRGs for $\mathsf{FORMULA} \circ \mathcal{G}$. It includes the proof of Theorem 5.2 and other applications. Our satisfiability algorithms (Theorem 5.5) appear in Section 5.5. Finally, Section 5.6 discusses learning results for $\mathsf{FORMULA} \circ \mathsf{XOR}$ and contains a proof of Theorem 5.7.

5.2 Preliminaries

Notation

In this chapter, we will mainly use $\{-1, 1\}$ as the Boolean basis. In some parts of this chapter, we will use the $\{0, 1\}$ basis for the simplicity of the presentation. This will be specified in corresponding sections.

De Morgan formulas and extensions

In this work, we denote by $\mathsf{FORMULA}[s]$ the class of Boolean functions computable by size-s De Morgan formulas. Let $\mathcal{G}$ denote some class of Boolean functions; then, we denote by $\mathsf{FORMULA}[s] \circ \mathcal{G}$ the class of functions computable by some size-s De Morgan formula where its leaves are labelled by functions in $\mathcal{G}$.

Approximating polynomials

Definition 5.8 (Point-wise approximation)**.** *For a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$, we say that the function $\tilde{f}\colon \{-1,1\}^n \to \mathbb{R}$ ε-approximates f if for every $z \in \{-1,1\}^n$,*

$$\left| f(z) - \tilde{f}(z) \right| \leq \varepsilon.$$

We will need the following powerful result for the approximating degree of De Morgan formulas.

Theorem 5.9 ([Rei11b], see also [BNRdW07])**.** *Let $s > 0$ be an integer and $0 < \varepsilon < 1$. Any De Morgan formula $F\colon \{-1,1\}^n \to \{-1,1\}$ of size s has a ε-approximating polynomial of degree $d = O(\sqrt{s} \cdot \log(1/\varepsilon))$. That is, there exists a degree-d polynomial $p\colon \{-1,1\}^n \to \mathbb{R}$ over the reals such that for every $z \in \{-1,1\}^n$,*

$$|p(z) - F(z)| \leq \varepsilon.$$

Note that Theorem 5.9 still holds if we use $\{0, 1\}$ as the Boolean basis.

Communication complexity

We use standard definitions from communication complexity. We consider the standard two party model of Yao and its generalizations to multiparty setting. We denote deterministic communication complexity of a Boolean function by $D(f)$ in the two party setting. We refer to [KN97] for standard definitions from communication complexity.

Definition 5.10. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. The* communication matrix of f*, namely M_f, is a $2^{n/2} \times 2^{n/2}$ matrix defined by $(M_f)_{x,y} := f(x,y)$.*

Definition 5.11. *A* rectangle *is a set of the form $A \times B$, for $A, B \subseteq \{0,1\}^n$. A* monochromatic rectangle *is a rectangle S such that for all pairs $(x,y) \in S$ the value $f(x,y)$ is the same.*

Lemma 5.12. *Let Π be a protocol that computes $f\colon \{0,1\}^n \to \{0,1\}$ with at most D bits of communication. Then, Π induces a partition of M_f into at most 2^D monochromatic rectangles.*

Given a protocol, its *transcript* is the sequence of bits communicated.

Lemma 5.13. *For every transcript z of some communication protocol, the set of inputs (x,y) that generate z is a rectangle.*

Below, we recount the definitions of two multiparty communication models used in this work, namely the number-on-forehead and the number-in-hand models.

Definition 5.14 ("Number-on-forehead" communication model; informal)**.** *In the k-party "number-on-forehead" communication model, there are k players and k strings $x_1, \ldots, x_k \in \{0,1\}^{n/k}$ and player i gets all the strings except for x_i. The players are interested in computing a value $f(x_1, \ldots, x_k)$, where $f\colon \{0,1\}^n \to \{0,1\}$ is some fixed function. We denote by $D^{(k)}(f)$ the number of bits that must be exchanged by the best possible number on forehead protocol solving f.*

We also use the following weaker communication model.

Definition 5.15 ("Number-in-hand" communication model; informal)**.** *In the k-party "number-in-hand" communication model, there are k players and k strings $x_1, \ldots, x_k \in \{0,1\}^{n/k}$ and player i gets only x_i. The players are interested in computing a value $f(x_1, \ldots, x_k)$, where $f\colon \{0,1\}^n \to \{0,1\}$ is some fixed function. We denote by $D^{(k\text{-NIH})}(f)$ the number of bits that must be exchanged by the best possible communication protocol.*

Note that $D^{(k\text{-NIH})}(f) \leq (1 - 1/k) \cdot n + 1$, for any n-variate Boolean function f, as if $k-1$ players write on the blackboard their string, then the player that did not reveal her input may compute $f(x_1, \ldots, x_k)$ on her own and then publish it.

For the communication models mentioned above, there are also bounded-error randomized versions, denoted by R_δ, $R_\delta^{(k)}$, and $R_\delta^{(k\text{-NIH})}$, respectively, where $0 < \delta < 1$ is an upper bound on the error probability of the protocol. In this setting, the players have access to some shared random string, say r, and the aforementioned error probability of the protocol is considered over the possible choices of r. Moreover, we require the error to be at most δ on each fixed choice of inputs.

We can extend the definitions of the communication complexity measures, defined above, to classes of Boolean functions, in a natural way. That is, for any communication complexity measure $M \in \left\{D, D^{(k)}, D^{(k\text{-NIH})}, R_\delta, R_\delta^{(k)}, R_\delta^{(k\text{-NIH})}\right\}$ and for any class of Boolean functions $\mathcal{G}$, we may define

$$M(\mathcal{G}) := \max_{g \in \mathcal{G}} M(g)\,.$$

We note that throughout this chapter, we denote by n the number of input bits for the function regardless the communication models. In the k-party communication setting (either NOF or NIH), we assume without loss of generality that n is divisible by k.

Learning

For a function $f : \{0,1\}^n \to \{0,1\}$ and a distribution $\mathcal{D}$ supported over $\{0,1\}^n$, we denote by $\mathrm{EX}(f, \mathcal{D})$ a randomized oracle that outputs independent identically distributed labelled examples of the form $(x, f(x))$, where $x \sim \mathcal{D}$.

Definition 5.16 (PAC learning model [Val84])**.** *Let $\mathcal{C}$ be a class of Boolean functions. We say that a randomized algorithm A* learns *$\mathcal{C}$ if, when A is given oracle access to $\mathrm{EX}(f, \mathcal{D})$ and inputs 1^n, ε, and δ, the following holds. For every n-variate function $f \in \mathcal{C}$, distribution $\mathcal{D}$ supported over $\{0,1\}^n$, and real-valued parameters $\varepsilon > 0$ and $\delta > 0$, $A^{\mathrm{EX}(f,\mathcal{D})}(1^n, \varepsilon, \delta)$ outputs with probability at least $1 - \delta$ over its internal randomness and the randomness of the example oracle $\mathrm{EX}(f, \mathcal{D})$ a description of a hypothesis $h : \{0,1\}^n \to \{0,1\}$ such that*

$$\mathop{\mathbf{Pr}}_{x \sim \mathcal{D}}[f(x) = h(x)] \;\geq\; 1 - \varepsilon.$$

The sample complexity *of a learning algorithm is the maximum number of random examples from $\mathrm{EX}(f, \mathcal{D})$ requested during its execution.*

5.3 Lower bounds

In this section, we prove an average-case lower bound for the generalized inner product function, against $\mathsf{FORMULA} \circ \mathcal{G}$, where $\mathcal{G}$ is the set of functions that have low-cost randomized communication protocols in the number-on-forehead setting. This corresponds to Item 1 of Theorem 5.1. Items 2 and 3 rely on our PRG constructions, and the proofs are deferred to Section 5.4.

Theorem 5.17. *For any integer $k \geq 2$, $s > 0$ and any class of functions $\mathcal{G}$, let $C\colon \{-1,1\}^n \to \{-1,1\}$ be a function in $\mathsf{FORMULA}[s] \circ \mathcal{G}$ such that*

$$\mathop{\mathbf{Pr}}_{x\sim\{-1,1\}^n}\left[C(x) = \mathsf{GIP}^k_n(x)\right] \geq 1/2 + \varepsilon.$$

Then

$$s = \Omega\left(\frac{n^2}{k^2 \cdot 16^k \cdot \left(R^{(k)}_{\varepsilon/(2n^2)}(\mathcal{G}) + \log n\right)^2 \cdot \log^2(1/\varepsilon)}\right).$$

We need a couple useful lemmas from [Tal16], whose proofs are presented in Section A.1 (Lemma A.1 and Lemma A.2) for completeness.

Lemma 5.18 ([Tal16])**.** *Let $\mathcal{D}$ be a distribution over $\{-1,1\}^n$, and let $f, C\colon \{-1,1\}^n \to \{-1,1\}$ be such that*

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[C(x) = f(x)] \geq 1/2 + \varepsilon.$$

Let $\tilde{C}\colon \{-1,1\}^n \to \mathbb{R}$ be a ε-approximating function of C, i.e., for every $x \in \{-1,1\}^n$, $|C(x) - \tilde{C}(x)| \leq \varepsilon$. Then,

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\tilde{C}(x) \cdot f(x)] \geq \varepsilon.$$

Lemma 5.19 ([Tal16])**.** *Let $\mathcal{D}$ be a distribution over $\{-1,1\}^n$ and let $\mathcal{G}$ be a class of functions. For $f\colon \{-1,1\}^n \to \{-1,1\}$, suppose that $D\colon \{-1,1\}^n \to \{-1,1\} \in \mathsf{FORMULA}[s] \circ \mathcal{G}$ is such that*

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[D(x) = f(x)] \geq 1/2 + \varepsilon_0.$$

Then there exists some $h\colon \{-1,1\}^n \to \{-1,1\} \in \mathsf{XOR}_{O\left(\sqrt{s}\cdot\log(1/\varepsilon_0)\right)} \circ \mathcal{G}$ such that

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[h(x) \cdot f(x)] \geq \frac{1}{s^{O\left(\sqrt{s}\cdot\log(1/\varepsilon_0)\right)}}.$$

We also need the following communication-complexity lower bound for GIP.

Theorem 5.20 ([BNS92, Theorem 2])**.** *For any $k \geq 2$, any function that computes GIP^k_n on more than $1/2 + \delta$ fraction of the inputs (over uniformly random inputs) must have k-party deterministic communication complexity at least $\Omega\Big(n/(k \cdot 4^k) - \log(1/\delta)\Big)$.*

We first show that if a function correlates with some small formula, whose leaves are functions with low *randomized* communication complexity, then it also correlates non-trivially with some function of relatively low *deterministic* communication complexity.

Lemma 5.21. *For any distribution $\mathcal{D}$ over $\{-1,1\}^n$, and any class of functions $\mathcal{G}$, let $f\colon \{-1,1\}^n \to \{-1,1\}$ and $C\colon \{-1,1\}^n \to \{-1,1\} \in \mathsf{FORMULA}[s] \circ \mathcal{G}$ be such that*

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[C(x) = f(x)] \geq 1/2 + \varepsilon.$$

Then there exists a function h, with k-party deterministic communication complexity at most

$$O\left(R^{(k)}_{\varepsilon/(2s)}(\mathcal{G}) \cdot \sqrt{s} \cdot \log(1/\varepsilon)\right),$$

such that

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[h(x) = f(x)] \geq 1/2 + 1/s^{O(\sqrt{s}\cdot\log(1/\varepsilon))}.$$

Proof. Let $C = F(g_1, g_2 \ldots, g_s)$ be the function in $\mathsf{FORMULA}[s]\circ\mathcal{G}$, where F is a formula and $g_1, g_2, \ldots, g_s$ are leaf functions from the class $\mathcal{G}$. For each g_i, consider a k-party randomized protocol Π_i of cost at most $R = R^{(k)}_{\varepsilon/(2s)}(\mathcal{G})$ that has an error $\varepsilon/(2s)$. Now consider the following function

$$\tilde{C}(x) := \mathop{\mathbf{E}}_{\Pi_1,\Pi_2,\ldots,\Pi_s}[D(x)],$$

where

$$D(x) := F(\Pi_1(x), \Pi_2(x), \ldots, \Pi_s(x)).$$

Note that for any fixed choice of $(\Pi_1, \Pi_2, \ldots, \Pi_s)$, D is a formula whose leaves are functions with *deterministic* communication complexity at most R. Next, we show the following.

Claim 5.22. *The function $\tilde{C}$ ε-approximates C.*

Proof of Claim 5.22. First note that since each Π_i is a $(\varepsilon/(2s))$-error randomized protocol, by taking the union bound over the s leaf functions, we have that for every input $x \in \{-1,1\}^n$,

$$\mathop{\mathbf{Pr}}_{\Pi_1,\Pi_2,\ldots,\Pi_s}[\Pi_1(x) = g_1(x) \wedge \Pi_2(x) = g_2(x) \wedge \cdots \wedge \Pi_s(x) = g_s(x)] \geq 1 - \varepsilon/2.$$

Denote by $\mathcal{E}$ the event $\Pi_1(x) = g_1(x) \wedge \Pi_2(x) = g_2(x) \wedge \cdots \wedge \Pi_s(x) = g_s(x)$. We have for every $x \in \{-1,1\}^n$,

$$\begin{aligned}
\tilde{C}(x) &= \mathop{\mathbf{E}}_{\Pi_1,\Pi_2,\ldots,\Pi_s}[D(x)] \\
&= \mathbf{E}[D(x) \mid \mathcal{E}] \cdot \mathbf{Pr}[\mathcal{E}] + \mathbf{E}[D(x) \mid \neg\mathcal{E}] \cdot \mathbf{Pr}[\neg\mathcal{E}] \\
&= C(x) \cdot \mathbf{Pr}[\mathcal{E}] + \mathbf{E}[D(x) \mid \neg\mathcal{E}] \cdot \mathbf{Pr}[\neg\mathcal{E}].
\end{aligned}$$

On the one hand, we have

$$\tilde{C}(x) = C(x) \cdot \mathbf{Pr}[\mathcal{E}] + \mathbf{E}[D(x) \mid \neg\mathcal{E}] \cdot \mathbf{Pr}[\neg\mathcal{E}] \leq C(x) + \varepsilon/2.$$

On the other hand, we get

$$\tilde{C}(x) = C(x) \cdot \mathbf{Pr}[\mathcal{E}] + \mathbf{E}[D(x) \mid \neg\mathcal{E}] \cdot \mathbf{Pr}[\neg\mathcal{E}] \geq C(x) \cdot (1 - \varepsilon/2) + (-1) \cdot (\varepsilon/2) \geq C(x) - \varepsilon.$$

This completes the proof of the claim. □

Now by Claim 5.22 and Lemma 5.18, we have

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\tilde{C}(x)\cdot f(x)] \geq \varepsilon. \tag{5.1}$$

By the definition of $\tilde{C}$, Equation (5.1) implies that there exists some D, which is a formula whose leaves are functions with *deterministic* communication complexity at most R, such that

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[D(x)\cdot f(x)] \geq \varepsilon,$$

which implies

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[D(x) = f(x)] \geq 1/2 + \varepsilon/2.$$

Then by Lemma 5.19, there exists a function h, which can be expressed as the XOR of at most $O(\sqrt{s}\cdot\log(1/\varepsilon))$ leaf functions in D, such that

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[h(x)\cdot f(x)] \geq \frac{1}{s^{O(\sqrt{s}\cdot\log(1/\varepsilon))}},$$

which again implies

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[h(x) = f(x)] \geq \frac{1}{2} + \frac{1}{s^{O\left(\sqrt{s}\cdot\log(1/\varepsilon)\right)}}.$$

Finally, note that the k-party deterministic communication complexity of h is at most

$$O(R\cdot\sqrt{s}\cdot\log(1/\varepsilon)),$$

where $R = R^{(k)}_{\varepsilon/(2s)}(\mathcal{G})$. □

We are now ready to show Theorem 5.17.

Proof of Theorem 5.17. Consider Lemma 5.21 with f being GIP^k_n and $\mathcal{D}$ being the uniform distribution. Consider Theorem 5.20 with $\delta = 1/s^{O(\sqrt{s}\cdot\log(1/\varepsilon))}$. We have

$$O\left(R^{(k)}_{\varepsilon/(2s)}(\mathcal{G})\cdot\sqrt{s}\cdot\log(1/\varepsilon)\right) \geq n/(k4^k) - O\left(\sqrt{s}\cdot\log(s)\cdot\log(1/\varepsilon)\right),$$

which implies

$$s \geq \Omega\left(\frac{n^2}{k^2\cdot 16^k\cdot\left(R^{(k)}_{\varepsilon/(2n^2)}(\mathcal{G}) + \log n\right)^2\cdot\log^2(1/\varepsilon)}\right).$$ □

5.4 Pseudorandom generators

Some of our PRGs are obtained from a general framework that allows us to reduce the task of fooling $\mathsf{FORMULA}\circ\mathcal{G}$ to the task of fooling the class of functions which are the parity or conjunction of few functions from $\mathcal{G}$.

5.4.1 The general framework

We show that in order to get a PRG for the class of subquadratic-size formulas with leaf gates in $\mathcal{G}$, it suffices to get a PRG for very simple sublinear-size formulas: either $\mathsf{XOR} \circ \mathcal{G}$ or $\mathsf{AND} \circ \mathcal{G}$.

Theorem 5.23 (PRG for $\mathsf{FORMULA} \circ \mathcal{G}$ from PRG for $\mathsf{XOR} \circ \mathcal{G}$ or $\mathsf{AND} \circ \mathcal{G}$)**.** *Let $\mathcal{G}$ be a class of gates on n bits. For any integer $s > 0$ and any $0 < \varepsilon < 1$, there exists a constant $c > 0$ such that the following holds. If a distribution $\mathcal{D}$ over $\{-1,1\}^n$ $\left(2^{-c \cdot \sqrt{s} \cdot \log(s) \cdot \log(1/\varepsilon)}\right)$-fools the XOR (parity) or the AND (conjunction) of $c \cdot \sqrt{s} \cdot \log(1/\varepsilon)$ arbitrary functions from $\mathcal{G}$, then $\mathcal{D}$ also ε-fools $\mathsf{FORMULA}[s] \circ \mathcal{G}$.*

Proof. We first show the case where $\mathcal{D}$ fools the parity of a few functions from $\mathcal{G}$. The proof can be easily adapted to the case of conjunction.

Let $C = F(g_1, g_2 \ldots, g_s)$ be a function in $\mathsf{FORMULA}[s] \circ \mathcal{G}$, where F is a formula, and $g_1, g_2, \ldots, g_s$ are functions from the class $\mathcal{G}$. Let U be the uniform distribution over $\{-1,1\}^n$. We need to show

$$\mathbf{E}[C(\mathcal{D})] \overset{\varepsilon}{\approx} \mathbf{E}[C(U)]. \tag{5.2}$$

Let p be a $(\varepsilon/3)$-approximating polynomial for F given by Theorem 5.9. Note that the degree of p is

$$d = O(\sqrt{s} \cdot \log(1/\varepsilon)).$$

Let us replace F, the formula part of C, with p and let

$$\tilde{C} := p(g_1, g_2 \ldots, g_s).$$

Since $\tilde{C}$ point-wisely approximates C, we have

$$\mathbf{E}[\tilde{C}(U)] \overset{\varepsilon/3}{\approx} \mathbf{E}[C(U)],$$

and

$$\mathbf{E}[\tilde{C}(\mathcal{D})] \overset{\varepsilon/3}{\approx} \mathbf{E}[C(\mathcal{D})].$$

Then to show Equation (5.2), it suffices to show

$$\mathbf{E}[\tilde{C}(\mathcal{D})] \overset{\varepsilon/3}{\approx} \mathbf{E}[\tilde{C}(U)].$$

We have

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\tilde{C}(x)] &= \mathop{\mathbf{E}}_{x\sim D}\left[\sum_{\substack{S\subseteq[s]:\\|S|\le d}} \hat{p}(S)\cdot\prod_{i\in S} g_i(x)\right] \\ &= \sum_{\substack{S\subseteq[s]:\\|S|\le d}} \hat{p}(S)\cdot\mathop{\mathbf{E}}_{x\sim\mathcal{D}}\left[\prod_{i\in S} g_i(x)\right]. \end{aligned} \tag{5.3}$$

Now note that for each $S\subseteq[s]$, $\prod_{i\in S} g_i(x)$ computes the XOR of at most d functions from $\mathcal{G}$. Using the fact the distribution $\mathcal{D}$ $\left(\delta = 1/2^{c\cdot\sqrt{s}\cdot\log(s)\cdot\log(1/\varepsilon)}\right)$-fools the XOR of any d functions from $\mathcal{G}$, we get

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\tilde{C}(x)] &= \sum_{\substack{S\subseteq[s]:\\|S|\le d}} \hat{p}(S)\cdot\mathop{\mathbf{E}}_{x\sim D}\left[\prod_{i\in S} g_i(x)\right] \\ &= \sum_{\substack{S\subseteq[s]:\\|S|\le d}} \hat{p}(S)\cdot\left(\mathop{\mathbf{E}}_{x\sim U}\left[\prod_{i\in S} g_i(x)\right] + \delta_S\right) && \text{(where } |\delta_S|\le\delta\text{)} \\ &= \sum_{\substack{S\subseteq[s]:\\|S|\le d}} \left(\hat{p}(S)\cdot\mathop{\mathbf{E}}_{x\sim U}\left[\prod_{i\in S} g_i(x)\right] + \hat{p}(S)\cdot\delta_S\right) \\ &= \sum_{\substack{S\subseteq[s]:\\|S|\le d}} \hat{p}(S)\cdot\mathop{\mathbf{E}}_{x\sim U}\left[\prod_{i\in S} g_i(x)\right] + \sum_{\substack{S\subseteq[s]:\\|S|\le d}} \hat{p}(S)\cdot\delta_S \\ &= \mathop{\mathbf{E}}_{x\sim U}[\tilde{C}(x)] + \sum_{\substack{S\subseteq[s]:\\|S|\le d}} \hat{p}(S)\cdot\delta_S. \end{aligned}$$

It remains to show

$$\left|\sum_{\substack{S\subseteq[s]:\\|S|\le d}} \hat{p}(S)\cdot\delta_S\right| \le \varepsilon/3.$$

Note that because $p(z)\in[1-\varepsilon/3, 1+\varepsilon/3]$ for every $z\in\{-1,1\}^s$, we have

$$|\hat{p}(S)| = \left|\mathop{\mathbf{E}}_{z\sim\{-1,1\}^s}\left[p(z)\cdot\prod_{i\in S} z_i\right]\right| \le 1+\varepsilon/3 < 2.$$

Then,

$$\left|\sum_{\substack{S\subseteq[s]:\\ |S|\le d}} \hat{p}(S)\cdot\delta_S\right| \le \sum_{\substack{S\subseteq[s]:\\ |S|\le d}} |\hat{p}(S)|\cdot|\delta_S| \le \delta\cdot\sum_{\substack{S\subseteq[s]:\\ |S|\le d}} |\hat{p}(S)| \le \delta\cdot s^{O(\sqrt{s}\cdot\log(1/\varepsilon))} \le \varepsilon/3,$$

where the last inequality holds for some sufficiently large constant c.

To show the case of conjunction, we can write the approximating polynomial as the sum of all degree-d monomials, each of which is the AND of at most d variables. One way to do this is to use the domain $\{0,1\}$ instead of $\{-1,1\}$ in the above argument. We need to show that the coefficients in this case still have small magnitude.

Claim 5.24. *Let $p\colon \{-1,1\}^n \to \mathbb{R}$ be a degree-d polynomial of the form*

$$p(x) = \sum_{\substack{S\subseteq[n]:\\ |S|\le d}} \hat{p}(S)\cdot\prod_{i\in S} x_i,$$

and let $q\colon \{0,1\}^n \to \mathbb{R}$ be the corresponding polynomial of p over the domain $\{0,1\}^n$, of the form

$$q(y) = \sum_{\substack{T\subseteq[n]:\\ |T|\le d}} \hat{q}(T)\cdot\prod_{i\in T} y_i.$$

Then,

$$|q|_1 = \sum_{\substack{T\subseteq[n]:\\ |T|\le d}} |\hat{q}(T)| \le n^{O(d)}\cdot\max_{\substack{S\subseteq[n]:\\ |S|\le d}} |\hat{p}(S)|.$$

Proof. We have

$$\begin{aligned}
q(y_1,y_2,\ldots,y_n) &= p(1-2y_1,1-2y_2,\ldots,1-2y_n)\\
&= \sum_{\substack{S\subseteq[n]:\\ |S|\le d}} \hat{p}(S)\cdot\prod_{i\in S}(1-2y_i)\\
&= \sum_{\substack{S\subseteq[n]:\\ |S|\le d}} \hat{p}(S)\cdot\left(\sum_{\ell\in\{0,1\}^{|S|}}\prod_{\substack{j\in S:\\ \ell_j=1}}(-2)^{|\ell|}\cdot y_j\right) && (\text{where } |\ell| = \textstyle\sum_{i=1}^{|S|}\ell_i)\\
&= \sum_{\substack{S\subseteq[n]:\\ |S|\le d}}\sum_{\ell\in\{0,1\}^{|S|}} \hat{p}(S)\cdot(-2)^{|\ell|}\cdot\prod_{\substack{j\in S:\\ \ell_j=1}} y_j. && (\text{where } |\ell| = \textstyle\sum_{i=1}^{|S|}\ell_i)
\end{aligned}$$

For a pair (S, ℓ) where $S \subseteq [n]$, $|S| \leq d$ and $\ell \in \{0,1\}^{|S|}$, let us define the polynomial $q_{(S,\ell)}$ as

$$q_{(S,\ell)}(y) = \hat{p}(S) \cdot (-2)^{|\ell|} \cdot \prod_{\substack{j \in S: \\ \ell_j = 1}} y_j.$$

Note that there are at most $n^d \cdot 2^d$ many pairs of such (S, ℓ)'s and for each (S, ℓ), we have

$$|q_{(S,\ell)}|_1 = \left|\hat{p}(S) \cdot (-2)^{|\ell|}\right| \leq 2^d \cdot |\hat{p}(S)|.$$

Finally we have

$$|q|_1 = \left|\sum_{(S,\ell)} q_{(S,\ell)}\right|_1 \leq \sum_{(S,\ell)} |q_{(S,\ell)}|_1 \leq n^d \cdot 2^d \cdot 2^d \cdot \max_{\substack{S \subseteq [n]: \\ |S| \leq d}} |\hat{p}(S)|,$$

as desired. □ (Claim 5.24)

This completes the proof of Theorem 5.23. □ (Theorem 5.23)

5.4.2 Formulas of low-communication functions in the number-in-hand setting

In this subsection, we will use $\{0,1\}$ as the Boolean basis.

Theorem 5.25. *For any integers $k \geq 2$, $s > 0$ and any $0 < \varepsilon < 1$, let $\mathcal{G}$ be the class of functions that have k-party number-in-hand $(\varepsilon/6s)$-error randomized communication protocols of cost at most R. There exists a PRG that ε-fools $\mathsf{FORMULA}[s] \circ \mathcal{G}$ with seed length*

$$n/k + O\left(\sqrt{s} \cdot (R + \log(s)) \cdot \log(1/\varepsilon) + \log(k)\right) \cdot \log(k).$$

We need the following PRG that fools single functions with low communication complexity in the number-in-hand model. The proof is presented in Section A.2 (Theorem A.3) for completeness.

Theorem 5.26 ([ASWZ96, INW94])**.** *For any $k \geq 2$, there exists a PRG that δ-fools any n-bits functions with k-party number-in-hand deterministic communication complexity of at most D', with seed length*

$$n/k + O\left(D' + \log(1/\delta) + \log(k)\right) \cdot \log(k).$$

Next, we show a PRG for $\mathsf{FORMULA} \circ \mathcal{G}$, where $\mathcal{G}$ is the class of functions with low-cost communication protocols in the number-in-hand setting. We first show for the case of deterministic protocols.

Theorem 5.27. *For any integers $k \geq 2$ and $s > 0$, let $\mathcal{G}$ be the class of functions whose k-party number-in-hand deterministic communication complexity are at most D. There is a PRG that ε-fools $\mathsf{FORMULA}[s] \circ \mathcal{G}$ of size s with seed length*

$$n/k + O\left(\sqrt{s} \cdot \log(1/\varepsilon) \cdot (D + \log(s)) + \log(k)\right) \cdot \log(k).$$

Proof. By Theorem 5.23, it suffices to show a PRG that $\left(\delta = 1/2^{c \cdot \sqrt{s} \cdot \log(s) \cdot \log(1/\varepsilon)}\right)$-fools every function that is the XOR of $t = c \cdot \sqrt{s} \cdot \log(1/\varepsilon)$ arbitrary functions from $\mathcal{G}$. Note that such a function has deterministic communication complexity at most $D' = t \cdot D$. Then Theorem 5.27 follows from Theorem 5.26. □

We now establish the randomized case.

Proof of Theorem 5.25. Let C be a function in $\mathsf{FORMULA}[s] \circ \mathcal{G}$. For each of the leaf functions in C, consider a k-party number-in-hand randomized protocol of cost at most R that has an error at most $\varepsilon/(6s)$. By taking a union bound over the s leaf functions and by viewing a randomized protocol as a distribution of deterministic protocols (as shown in the proof of Claim 5.22), we get the following which is a (point-wisely) $(\varepsilon/3)$-approximating function for C:

$$\tilde{C}(x) := \sum_i p_i \cdot D_i(x),$$

where each $p_i \in [0, 1]$ is some probability density value (so $\sum_i p_i = 1$), and each D_i is a formula whose leaves are functions with *deterministic* communication complexity at most R. Then to ε-fool C, it suffices to $(\varepsilon/3)$-fool its $(\varepsilon/3)$-approximating function $\tilde{C}$. Also, since $\tilde{C}$ is a convex combination of the D_i's, it suffices to $(\varepsilon/3)$-fools all the D_i's. We will do this using the PRG form Theorem 5.27. We get that there exists a PRG that $(\varepsilon/3)$-fools each D_i with seed length

$$n/k + O\left(\sqrt{s} \cdot (R + \log(s)) \cdot \log(1/\varepsilon) + \log(k)\right) \cdot \log(k),$$

as desired. □

5.4.3 Applications: Fooling formulas of SYMs, LTFs, XORs, and AC^0 circuits

$\mathsf{FORMULA} \circ \mathsf{SYM}$ and $\mathsf{FORMULA} \circ \mathsf{LTF}$

Here, we show how the PRG in Theorem 5.25 implies PRGs for $\mathsf{FORMULA} \circ \mathsf{LTF}$ and $\mathsf{FORMULA} \circ \mathsf{SYM}$.

Theorem 5.28. *For any size $s > 0$ and $0 < \varepsilon < 1$, there exists a PRG that ε-fools $\mathsf{FORMULA}[s] \circ \mathsf{LTF}$ with seed length*

$$O\left(n^{1/2} \cdot s^{1/4} \cdot \log(n) \cdot \log(n/\varepsilon)\right).$$

For $\mathsf{FORMULA}[s] \circ \mathsf{SYM}$*, the seed length is*

$$O\left(n^{1/2} \cdot s^{1/4} \cdot \log(n) \cdot \log(1/\varepsilon)\right).$$

We need the fact that the class of LTF has low communication complexity in the number-in-hand model. Consider the following k-party SUM-GREATER$_m$ problem where the i-th party holds a m-bit number z_i in hand and they want to determine whether $\sum_{i=1}^{k} z_i > \theta$, where θ is a fixed number known to all the parties. Nisan [Nis94] gave an efficient randomized protocol (with public randomness) for this problem.

Theorem 5.29 ([Nis94][7])**.** *Let $m > 0$ be an integer. For any integer $2 \leq k \leq m^{O(1)}$, and any $0 < \delta < 1$, there exists a δ-error randomized protocol of cost $O(k \cdot \log(m) \cdot \log(m/\delta))$ for the k-party* SUM-GREATER$_m$ *problem.*

By Theorem 5.29 and the fact that every linear threshold function on n bits has a representation such that the weights are $O(n \log(n))$ integers [MTT61], we get the following.

Corollary 5.30. *For every $k \geq 2$ and $0 < \delta < 1$, the k-party number-in-hand δ-error randomized communication complexity of* LTF *is $O(k \cdot \log(n) \cdot \log(n/\delta))$.*

Proof of Theorem 5.28. By Corollary 5.30 and Theorem 5.25, for *every* $k \geq 2$ we get a PRG for $\mathsf{FORMULA} \circ \mathsf{LTF}$ of seed length

$$n/k + O\left(\sqrt{s} \cdot k \cdot \log(n) \cdot \log(ns/\varepsilon) \cdot \log(1/\varepsilon) + \log(k)\right) \cdot \log(k).$$

By choosing

$$k = \frac{n^{1/2}}{s^{1/4} \cdot \log(n) \cdot \log(n/\varepsilon)},$$

the claimed seed length follows from a simple calculation.

For $\mathsf{FORMULA} \circ \mathsf{SYM}$, note that every n-bit symmetric function has a deterministic k-party number-in-hand communication protocol of cost at most $k \cdot \log(n)$. Then the rest can be shown using a similar argument as above (by choosing $k = n^{1/2} / \left(s^{1/4} \cdot \log(n)\right)$). □

$\mathsf{FORMULA} \circ \mathsf{XOR}$

For the case of $\mathsf{FORMULA} \circ \mathsf{XOR}$, we get a PRG with better seed length.

Theorem 5.31. *For any size $s > 0$ and $0 < \varepsilon < 1$, there exists a PRG that ε-fools* $\mathsf{FORMULA}[s] \circ \mathsf{XOR}$ *with seed length*

$$O\left(\sqrt{s} \cdot \log(s) \cdot \log(1/\varepsilon) + \log(n)\right).$$

[7] Viola [Vio15] gave a δ-error randomized protocol for the k-party SUM-GREATER$_m$ problem of cost $O(k \cdot \log(k) \cdot \log(m/\delta))$, which is better than Nisan's protocol when $k = m^{o(1)}$.

Proof. By Theorem 5.23, to fool $\mathsf{FORMULA}[s] \circ \mathcal{G}$, it suffices to $\left(\delta = 1/2^{O\left(\sqrt{s}\cdot\log(s)\cdot\log(1/\varepsilon)\right)}\right)$-fool the XOR of a few functions from $\mathcal{G}$, where $\mathcal{G}$ in this case is the set of all XOR functions. Note that the XOR of any set of XOR functions simply computes some XOR function. Therefore, we can use small-bias distribution, which fools every XOR function, to fool $\mathsf{FORMULA}[s] \circ \mathsf{XOR}$. Finally, note that there are known constructions for δ-bias distributions that use $O(\log(n/\delta))$ random bits (see e.g. [AGHP92]). □

Using the "locality" of this PRG for $\mathsf{FORMULA} \circ \mathsf{XOR}$, we get a lower bound for MCSP against subquadratic-size formulas of XORs.

Theorem 5.32. *For every integer $s > 0$, if* MCSP *on N-bit can be computed by some function in $\mathsf{FORMULA}[s] \circ \mathsf{XOR}$, then $s = \tilde{\Omega}(N^2)$.*

Proof sketch. There is a standard construction of δ-bias distribution that is local (see e.g. [AGHP92, Construction 3] and [CKLM20, Fact 18]) in the following sense: there exists a circuit of size at most $\tilde{O}(\log(n/\delta) \cdot \log(n))$ such that given a seed of length $O(\log(n/\delta))$ and a index $j \in [n]$, outputs the j-th bit of the distribution. Local PRGs imply MCSP lower bounds (see [CKLM20, Section 3]). □

$\mathsf{FORMULA} \circ \mathsf{AC}^0$

Another application of Theorem 5.23 is to take $\mathcal{G}$ to be the set all functions that can be computed by small constant-depth circuits (AC^0). Note the state-of-the-art PRG against size-M depth-d AC^0 has a seed length of $\log^{d+O(1)}(Mn) \cdot \log(1/\varepsilon)$ [ST19]. Below, let $\mathsf{AC}^0_{d,M}$ denote the class of depth-d circuits of size at most M.

Theorem 5.33. *For any size $s, m > 0$ and $0 < \varepsilon < 1$, there exists a PRG that ε-fools $\mathsf{FORMULA} \circ \mathsf{AC}^0_{d,M}$ of size s with seed length*

$$\log^{d+O(1)}(Mn) \cdot \sqrt{s} \cdot \log(s) \cdot \log(1/\varepsilon).$$

Moreover, by inspecting the construction of PRG in [ST19], it is not difficult to see that the PRG is also local; there exists a circuit of size at most $\lambda = \log^{d+O(1)}(Mn) \cdot \log(1/\varepsilon)$ such that given a seed of length $O\log^{d+O(1)}(Mn) \cdot \log(1/\varepsilon)$ and a index $j \in [n]$, outputs the j-th bit of the PRG. As a result, we get MCSP lower bounds from the this PRG.

Theorem 5.34. *For every $s, d, M \in \mathbb{N}$, if* MCSP *on N-bit can be computed by some function in $\mathsf{FORMULA}[s] \circ \mathsf{AC}^0_{d,M}$, then*

$$s \geq N^2 / \log^{2d+O(1)}(Mn).$$

5.4.4 Formulas of low number-on-forehead communication leaf gates

In this section, we show a PRG with mild seed length for formulas of functions with low *multi-party number-on-forehead* communication complexity.

Theorem 5.35. *Let $\mathcal{G}$ be a class of n-bits functions. For any size $s > 0$, there exists a PRG that ε-fools $\mathsf{FORMULA}[s] \circ \mathcal{G}$, with seed length*

$$n - \frac{n}{O\left(\sqrt{s} \cdot k \cdot 4^k \cdot \left(R^{(k)}_{\varepsilon/(2s)}(\mathcal{G}) + \log(n)\right) \cdot \log(n/\varepsilon)\right)}.$$

The PRG is constructed using the hardness vs. randomness paradigm.

Hardness based PRGs

We show how to construct the PRG using the average-case hardness result for formulas of functions with low multi-party communication complexity (Theorem 5.17). We start with some notations. For $x \in \{-1,1\}^m$ and an integer k such that k divides m, we consider a partition of x into k equal-sized consecutive blocks and write $x = x^{(1)}, x^{(2)}, \ldots, x^{(k)}$, where $x^{(i)} \in \{-1,1\}^{m/k}$ for each $i \in [k]$.

Lemma 5.36. *For any integers $m, t, k > 0$ such that k divides m, t, let $\mathcal{G}$ be a class of functions on $mt + t$ bits, and let $G\colon \{-1,1\}^{m \times t} \to \{-1,1\}^{mt+t}$ be*

$$\begin{aligned} &G(x_1, x_2, \ldots, x_t) \\ &= \left(x_1^{(i)}, x_2^{(i)}, \ldots, x_t^{(i)}, \mathsf{GIP}^k_m\left(x_{(i-1)\cdot(t/k)+1}\right), \mathsf{GIP}^k_m\left(x_{(i-1)\cdot(t/k)+2}\right), \ldots, \mathsf{GIP}^k_m\left(x_{i\cdot(t/k)+1}\right)\right)_{i \in [k]}, \end{aligned}$$

where $x_1, x_2, \ldots, x_t \in \{-1,1\}^m$. Then G is a PRG that $(t \cdot \varepsilon)$-fools $\mathsf{FORMULA} \circ \mathcal{G}$ of size

$$s = \Omega\left(\frac{m^2}{k^2 \cdot 16^k \cdot \left(R^{(k)}_{\varepsilon/(2m^2)}(\mathcal{G}) + \log m\right)^2 \cdot \log^2(1/\varepsilon)}\right).$$

Proof. The high level idea is as follows. We argue that if there is a $\mathsf{FORMULA} \circ \mathcal{G}$ of the claimed size that breaks the PRG, then there is a $\mathsf{FORMULA} \circ \mathcal{G}'$ of the same size that computes GIP on m bits, where $\mathcal{G}'$ has a k-party communication complexity that is at most that of $\mathcal{G}$ *with respect to the m-bit input*, and hence contradicts the $\mathsf{FORMULA} \circ \mathcal{G}'$ complexity of the generalized inner product function. The resulting formula is obtained by fixing some input bits of the original $\mathsf{FORMULA} \circ \mathcal{G}$ which breaks the PRG.

We use a hybrid argument. First consider the distribution given by G, where we replace each $\mathsf{GIP}(x_j)$ ($j \in [t]$) with a uniformly random bit; let us denote those random bits as U_j for $j \in [t]$ (note that this is just the uniform distribution). Then for each $j \in [t]$, define H_j to be the distribution that we substitute back $\mathsf{GIP}(x_1), \mathsf{GIP}(x_2), \ldots, \mathsf{GIP}(x_j)$ for the corresponding uniform bits in the previous distribution.

For the sake of contradiction, suppose there exists a $\mathsf{FORMULA} \circ \mathcal{G}$ C of size s such that

$$|\mathbf{Pr}[C(H_t) = 1] - \mathbf{Pr}[F(H_0) = 1]| > t \cdot \varepsilon.$$

By the triangle inequality, there exists a $1 \leq j \leq k$ such that

$$|\mathbf{Pr}[C(H_j) = 1] - \mathbf{Pr}[C(H_{j-1}) = 1]| > \varepsilon.$$

Then by averaging, there exist some fixings of $x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_t$ and $U_{j+1}, \ldots, U_t$ to C such that the above inequality still holds. Let us denote by C' the circuit obtained by C after such fixings and assume without loss of generality $(k-1)t/k \leq j \leq t$. Then we have

$$\left|\mathbf{Pr}\left[C'\left(x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(k)}, \mathsf{GIP}(x_j)\right) = 1\right] - \mathbf{Pr}\left[C'\left(x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(k)}, U_j\right) = 1\right]\right| > \varepsilon. \tag{5.4}$$

By a standard "unpredictability implies pseudorandomness" argument [Yao82], we can show that there is some circuit C'', obtained from C' by fixing some value for the last bit, that computes the generalized inner product function on m bits with probability greater than $1/2 + \varepsilon$ over uniformly random inputs. Note that the size of C'' is the same as C' (hence also C) , and also C'' can be computed by some $\mathsf{FORMULA} \circ \mathcal{G}'$, where $R^{(k)}_{\delta}(\mathcal{G}') \leq R^{(k)}_{\delta}(\mathcal{G})$ for every δ. This contradicts hardness of GIP for such circuits (Theorem 5.17). □

We are now ready to prove Theorem 5.35.

Proof of Theorem 5.35. Consider Lemma 5.36. Let $n = mt + t$, and we have $m = (\frac{n}{t} - 1)$. Then Lemma 5.36 gives a PRG that ε-fools $\mathsf{FORMULA} \circ \mathcal{G}$ of size

$$\begin{aligned} s &= \Omega\left(\frac{m^2}{k^2 \cdot 16^k \cdot \left(R^{(k)}_{\varepsilon/(2m^2)}(\mathcal{G}) + \log m\right)^2 \cdot \log^2(t/\varepsilon)}\right) \\ &\geq \Omega\left(\left(\frac{n}{t}\right)^2 / \left(k^2 \cdot 16^k \cdot \left(R^{(k)}_{\varepsilon/(2n^2)}(\mathcal{G}) + \log n\right)^2 \cdot \log^2(n/\varepsilon)\right)\right), \end{aligned}$$

which yields

$$t \geq \Omega\left(\frac{n}{\sqrt{s} \cdot k \cdot 4^k \cdot \left(R^{(k)}_{\varepsilon/(2n^2)}(\mathcal{G}) + \log n\right) \cdot \log(n/\varepsilon)}\right).$$

Note that the seed length in this case is $n - t$. □

MKtP lower bounds

The PRG in Theorem 5.35 is sufficient to give an MKtP lower bound for formulas of functions with low multi-party communication complexity.

Theorem 5.37. *For any integer $s > 0$ and any class of N-bit function $\mathcal{G}$, if* MKtP *on N-bit can be computed by some function $\mathsf{FORMULA}[s] \circ \mathcal{G}$, then*

$$s = \frac{N^2}{k^2 \cdot 16^k \cdot R^{(k)}_{1/3}(\mathcal{G}) \cdot \mathsf{polylog}(N)}.$$

Proof. Let C be a function in $\mathsf{FORMULA} \circ \mathcal{G}$ of size less than

$$\frac{N^2}{k^2 \cdot 16^k \cdot R_{1/3}^{(k)}(\mathcal{G}) \cdot \log^c(N)}$$

where $c > 0$ is some sufficiently large constant. By Theorem 5.35, we have that there is a PRG that $(1/3)$-fools C and its seed length is

$$N - \mathsf{polylog}(N).$$

Also, since the PRG is polynomial-time computable, we get that for every seed, the output of the PRG has Kt complexity at most $\theta = N - \mathsf{polylog}(N)$. However, consider the MKtP function with a threshold parameter θ; this function is not fooled by such a PRG, since it accepts every output of the PRG and rejects a uniformly random string with high probability. □

5.5 Satisfiability algorithms

In this section, we will use $\{0,1\}$ as the Boolean basis.

5.5.1 Computational efficient communication protocols

Definition 5.38 (Computational efficient communication protocols)**.** *Let $t\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. We say that a two-party communication protocol is* t-efficient *if for each of the parties, given an input x and some previously sent messages $\pi \in \{0,1\}^*$, the next message to send can be computed in time $t(|x|, |\pi|)$ ($\perp$ is being output if there is no next message). We say that such a protocol is* explicit *if $t(|x|, |\pi|) = 2^{o(|x|+|\pi|)}$.*

Lemma 5.39. *Let $f\colon \{0,1\}^n \to 1$ and let Π be a t-efficient communication protocol for f with communication cost at most D. Then the protocol tree of Π can be output in time $O\left(D \cdot t(n/2, D) \cdot 2^{n/2} \cdot 2^D\right)$. That is, there exists an algorithm that outputs a list of all (partial and full) transcripts of length at most D and the rectangles associated with each of the transcripts.*

Proof. It suffices to show that, given an input $x \in \{0,1\}^{n/2}$ and a transcript $\ell \in \{0,1\}^{\leq D}$, we can decide whether x belongs to the rectangle indexed by ℓ in time $D \cdot t(n/2, D)$. Suppose x is the input for Alice (resp. Bob), and we want to decide whether x belongs to the rectangle indexed by π. We can carry out the communication task by simulating the behavior of Alice (resp. Bob) using the protocol Π and simulating Bob's (resp. Alice's) behavior using the transcript π, and check whether the messages sent by Alice (resp. Bob) is consistent with the transcript π. This takes time at most $D \cdot t(n/2, D)$. To construct the tree, we do the above for every (partial and full) transcript $\pi \in \{0,1\}^{\leq D}$ and every input $x \in \{0,1\}^{n/2}$ for Alice (resp. Bob). The total running time is $O\left(D \cdot t(n/2, D) \cdot 2^{n/2} \cdot 2^D\right)$. □

For a protocol Π, we denote by Leaves(Π) the set of full transcripts of Π.

Remark. We note that, in the *white-box* context of the satisfiability problem, there is no need to assume a canonical partition of the input variables among the players. For instance, a helpful partition can either be given as part of the input, or computed by the algorithm. As a consequence, in instantiations of Theorem 5.5 for a particular circuit class $\mathcal{C}$, it is sufficient to be able to convert the input circuit from $\mathcal{C}$ into some device from $\mathsf{FORMULA} \circ \mathcal{G}$ for which protocols of bounded communication cost can be described.

5.5.2 Explicit approximating polynomials for formulas

From Theorem 5.9, we know that every size-s formula has a degree-$O(\sqrt{s})$ polynomial that point-wisely approximates it. In our SAT algorithms, we will need to *explicitly construct* such an approximating polynomial given a formula. One way to do this is to use an *efficient* quantum query algorithm for formulas. It is known that a quantum query algorithm for a function f using at most T queries implies an approximating polynomial for f of degree at most $2T$ [BBC$^+$01], and by classically simulating such an quantum algorithm, one can show that the approximating polynomial can be obtained in time that is polynomial in the number of its monomials, in addition to the time for the classical simulation. For our task, we can use the result of Reichardt [Rei11a] which showed an *efficient* quantum algorithm for evaluating size-s formulas with $O(\sqrt{s} \cdot \log s)$ queries[8]. Here, we present an alternate way to construct approximating polynomials for De Morgan formulas which rely only on the *existence* of such polynomials, without requiring an efficient quantum query algorithm. This "black-box" approach was suggested to us by an anonymous reviewer.

We first need the following structural lemma for formulas.

Lemma 5.40 ([IMZ12a, Tal14])**.** *For every integer $s > 0$, there exists an algorithm such that given a size-s De Morgan formula F, runs in $\mathsf{poly}(s)$ time and outputs a top formula F' with $O(\sqrt{s})$ leaves and each leaf of F' is a sub-formula with $O(\sqrt{s})$ input leaves.*

Lemma 5.41. *For any integer $s > 0$ and any $0 < \varepsilon < 1$, there exists an algorithm of running time $s^{O\left(\sqrt{s}\cdot\log(s)\cdot\log(1/\varepsilon)\right)}$ such that given a De Morgan formula F of size s, outputs an ε-approximating polynomial of degree $O(\sqrt{s}\cdot\log(s)\cdot\log(1/\varepsilon))$ for F. That is, the algorithm outputs a multi-linear polynomial (as sum of monomials) over the reals such that for every $x \in \{0,1\}^n$,*

$$|p(x) - F(x)| \leq \varepsilon.$$

[8]It is also known that there exists a quantum query algorithm for evaluating size-s formulas with $O\left(\sqrt{s}\right)$ queries [Rei11b], which implies the existence of an approximating polynomial for size-s formulas of degree $O\left(\sqrt{s}\right)$ (see Theorem 5.9). However, because this algorithm is not known to be efficient, it is unclear whether such an approximating polynomial can be constructed efficiently with respect to the number of monomials.

Proof. We first note that it suffices to construct a (1/3)-approximating polynomial for F with degree $D = O(\sqrt{s} \cdot \log(s))$. This is because given a (1/3)-approximating polynomial one can obtain explicitly an ε-approximating polynomial of degree $D \cdot O(\log(1/\varepsilon))$, by feeding $O(1/\varepsilon)$ copies of the (1/3)-approximating polynomial to the polynomial computing MAJORITY on $O(1/\varepsilon)$ bits [BNRdW07] (see also [Tal14, Appendix B]).

We first invoke Lemma 5.40 on F to obtain a top formula F' with $t = O(\sqrt{s})$ leaves, each of which is a sub-formula of size $O(\sqrt{s})$. We construct a (1/20)-approximating (multilinear) polynomial P for the top formula F', which has degree $d_1 = O(s^{1/4})$ by Theorem 5.9. Note that P can be constructed in time $2^{O(\sqrt{s})}$ because F' has at most $O(\sqrt{s})$ leaves. Next, for each of the t sub-formulas, denoted as $F_1, F_2, \ldots, F_t$, we construct a $(1/(20t))$-approximating polynomial. Note that these polynomials have degree $d_1 = O(s^{1/4} \cdot \log(s))$ and can be constructed in time $2^{O(\sqrt{s})}$. Let's denote these t polynomials as $Q_1, Q_2, \ldots, Q_t$. Now for each Q_i ($i \in [t]$), we define

$$q_i(x) = \frac{Q_i(x) + 1/(20t)}{1 + 1/(10t)}.$$

The final approximating polynomial for F is given as

$$p(x) = P\left(q_1(x), q_2(x), \ldots, q_t(x)\right).$$

Note that p has degree $d_1 \cdot d_2 = O(\sqrt{s} \cdot \log(s))$ and can be constructed (as sum of monomials) in time $s^{O(\sqrt{s} \cdot \log(s))}$. It remains to show that p (1/3)-approximates F.

For $0 \leq q \leq 1$, let N_q be the distribution over $\{0,1\}$ such that $\mathbf{Pr}_{y \sim N_q}[y = 1] = q$. Then for an fixed input $x \in \{0,1\}^s$, we have

$$p(x) = \mathop{\mathbf{E}}_{y_i \sim N_{q_i(x)}} [P(y_1, y_2, \ldots, y_t)]. \tag{5.5}$$

Let $\mathcal{E}$ be the event that $y_i = F_i(x)$ for all $i \in [t]$. Note that

$$\delta := \mathop{\mathbf{Pr}}_{y_i \sim N_{q_i(x)}} [\neg \mathcal{E}] \leq 1/10. \tag{5.6}$$

To see Equation (5.6), note that for every $i \in [t]$, if $F_i(x) = 0$, then $0 \leq q_i(x) \leq 1/(10t)$, which implies

$$\mathop{\mathbf{Pr}}_{y_i \sim N_{q_i(x)}} [y_i \neq F_i(x)] \leq 1/(10t).$$

Similar for the case when $F_i(x) = 1$ (which implies $1 - 1/(10t) < q_i(x) \leq 1$). Then Equation (5.6) follows from a union bound. Now we can re-write Equation (5.5) as

$$\begin{aligned} p(x) &= \mathbf{E}[P(y_1, y_2, \ldots, y_t) \mid \mathcal{E}] \cdot \mathbf{Pr}[\mathcal{E}] + \mathbf{E}[P(y_1, y_2, \ldots, y_t) \mid \neg \mathcal{E}] \cdot \mathbf{Pr}[\neg \mathcal{E}] \\ &= \left(F'(F_1(x), F_2(x), \cdot F_t(x)) \pm 1/20\right) \cdot (1 - \delta) + \mathbf{E}[P(y_1, y_2, \ldots, y_t) \mid \neg \mathcal{E}] \cdot \delta. \end{aligned}$$

Note that $P(y) \in [-1/(20t), 1 + 1/(20t)]$ for every $y \in \{0,1\}^t$, and that $\delta \leq 1/10$. A simple calculation shows that

$$p(x) = F'(F_1(x), F_2(x), \ldots, F_t(x)) \pm \frac{1}{3},$$

as desired. □

5.5.3 The #SAT algorithm

In this subsection, we present our #SAT algorithm.

Theorem 5.42. *For any integer $s > 0$, there exists a deterministic #SAT algorithm for $\mathsf{FORMULA}[s] \circ \mathcal{G}$, where $\mathcal{G}$ is the class of functions with explicit two-party deterministic protocols of communication cost at most D, that runs in time*

$$2^{n - \Omega\left(\frac{n}{\sqrt{s} \cdot \log^2(s) \cdot D}\right)} \cdot \mathsf{poly}(n, s).$$

In the case $\mathcal{G}$ is the class of functions with explicit randomized protocols of communication cost at most R, there exists an analogous randomized algorithm with a running time

$$2^{n - \Omega\left(\frac{n}{\sqrt{s} \cdot \log^2(s) \cdot R}\right)^{1/2}} \cdot \mathsf{poly}(n, s).$$

The algorithm is based on the framework for designing satisfiability algorithms developed by Williams [Wil14b]. The idea is to transform a given circuit into a "sparse polynomial" and solve satisfiability by evaluating the polynomial on all points in a faster-than-brute-force manner.

We first need the following fast matrix multiplication algorithm for "narrow" matrices.

Theorem 5.43 ([Cop82])**.** *Multiplication of an $N \times N^{.172}$ matrix with an $N^{.172} \times N$ matrix can be done in $O(N^2 \log^2 N)$ arithmetic operations over any field.*

For an even number $n > 0$, and $x \in \{0,1\}^n$, we denote by x^{L} (resp. x^{R}) the first half of x and $x^{\mathrm{R}} \in \{0,1\}^{n/2}$ the second half. We now prove Theorem 5.42.

Proof of Theorem 5.42. We first prove the deterministic case.

Let $C = F(g^1, g^2 \ldots, g^s)$ be a device in $\mathsf{FORMULA} \circ \mathcal{G}$ where F is a formula and $g^1, g^2, \ldots, g^s$ are functions that have a explicit communication protocol of cost at most D. The first step is to output the protocol tree for each g^i ($i \in [s]$). Since each g^i has explicit protocol of cost at most D, by Lemma 5.39, these protocol trees can be output in time $s \cdot 2^{n/2 + D + o(n)} \leq 2^{n/1.9}$ (here we assume $D = o(n)$ and $s \leq n^2$ otherwise the theorem holds trivially).

Let n' be an integer whose value is determined later. Let T be a set of n' variables such that T contains $n'/2$ variables from the first half of the n variables and the rest are from the

second half. For a partial assignment $z \in \{0,1\}^{n'}$ to T, denote by C_z the restricted function of C where the variables in T are fixed according to z. To count the number of satisfying assignments of C, we need to compute the following quantity:

$$\sum_{x \in \{0,1\}^{n-n'}} \sum_{z \in \{0,1\}^{n'}} C_z(x). \tag{5.7}$$

Now consider

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

We will try to obtain the value of $Q(x)$ for every $x \in \{0,1\}^{n-n'}$, in time about $2^{n-n'}$, which will allow us to compute the quantity in Equation (5.7) in time $\tilde{O}(2^{n-n'})$ by summing $Q(x)$ over all the x's. We do this by first transforming Q into an *approximating* polynomial with not-too-many monomials, and each monomial is a product of *functions that only rely on either the first or the second half of* x. With such a polynomial, we can perform fast multipoint evaluation using the fast matrix multiplication algorithm in Theorem 5.43.

For each $z \in \{0,1\}^{n'}$, we view the formula C_z as $F(g_z^1, g_z^2, \ldots, g_z^s)$, where F is the De Morgan formula part of C_z and $g_z^1, g_z^2, \ldots, g_z^s$ are the leaf gates. Let us now replace F by a ε-approximating polynomial p, where $\varepsilon = 1/\left(3 \cdot 2^{n'}\right)$, using Lemma 5.41. Note that the degree of p is at most

$$d \leq O(\sqrt{s} \cdot \log(s) \cdot \log(1/\varepsilon)) \leq O(\sqrt{s} \cdot \log(s) \cdot n').$$

Now consider the following

$$Q'(x) = \sum_{z \in \{0,1\}^{n'}} p(g_z^1(x), g_z^2(x), \ldots, g_z^s(x)).$$

First, note that by the value that we've chosen for the approximating error ε, we have that, for every x,

$$\left|Q'(x) - Q(x)\right| \leq 2^{n'} \cdot \varepsilon = 1/3.$$

In other words, given $Q'(x)$, we can recover the value of $Q(x)$, which is supposed to be an integer.

Next, we perform fast multipoint evaluation on Q'. First of all, we re-write Q' as follows:

$$Q'(x) = \sum_{z \in \{0,1\}^{n'}} \sum_{\substack{S \subseteq [s]: \\ |S| \leq d}} \hat{p}(S) \cdot \prod_{i \in S} g_z^i(x). \tag{5.8}$$

Now let Π_i be the protocol of g^i, we can re-write g^i_z as follows:

$$g^i_z(x) = \sum_{\pi_i \in \text{Leaves}(\Pi_i)} \alpha^i\left(z^{\text{L}}x^{\text{L}}, \pi_i\right) \cdot \beta^i\left(z^{\text{R}}x^{\text{R}}, \pi_i\right), \tag{5.9}$$

where $\alpha^i\left(z^{\text{L}}x^{\text{L}}, \pi_i\right)$ (resp. $\beta^i\left(z^{\text{R}}x^{\text{R}}, \pi_i\right)$) is 1 if and only if $\left(z^{\text{L}}x^{\text{L}}\right)$ (resp. $\left(z^{\text{R}}x^{\text{R}}\right)$) belongs to the rectangle indexed by π_i and the function value of that rectangle is 1. Note that for each $i \in [s]$, given the pre-computed protocol tree of the Π_i, α^i and β^i can be computed in polynomial time (for example, using binary search). After plugging Equation (5.9) into Equation (5.8) for every $i \in [s]$ and rearranging, we get

$$Q'(x) = \sum_{z \in \{0,1\}^{n'}} \sum_{\substack{S \subseteq [s]: \\ |S| \leq d}} \sum_{\substack{\vec{\pi} = (\pi_i)_{i \in S}: \\ \pi_i \in \text{Leaves}(\Pi_i)}} \hat{p}(S) \cdot \prod_{i \in S} \alpha^i\left(z^{\text{L}}x^{\text{L}}, \pi_i\right) \cdot \prod_{i \in S} \beta^i\left(z^{\text{R}}x^{\text{R}}, \pi_i\right). \tag{5.10}$$

Note that Q' can be expressed as the sum of at most m terms, where

$$m \leq 2^{n'} \cdot s^{O(\sqrt{s} \cdot \log(s) \cdot n')} \cdot 2^{O(\sqrt{s} \cdot \log(s) \cdot n' \cdot D)} \leq 2^{O(\sqrt{s} \cdot \log^2(s) \cdot D \cdot n')}.$$

Note that given Lemma 5.41, we can obtain Q' in time

$$2^{O(\sqrt{s} \cdot \log^2(s) \cdot D \cdot n')}. \tag{5.11}$$

Next, we construct a $2^{(n-n')/2} \times m$ matrix A and a $m \times 2^{(n-n')/2}$ matrix B as follows:

$$A_{x^{\text{L}}, (z, S, \vec{\pi})} = \hat{p}(S) \cdot \prod_{i \in S} \alpha^i\left(z^{\text{L}}x^{\text{L}}, \pi_i\right),$$

and

$$B_{(z, S, \vec{\pi}), x^{\text{R}}} = \prod_{i \in S} \beta^i\left(z^{\text{R}}x^{\text{R}}, \pi_i\right).$$

It is easy to see that for each $x \in \{0,1\}^{n-n'}$,

$$Q'(x) = (A \cdot B)_{x^{\text{L}}, x^{\text{R}}}.$$

We now want to compute $A \cdot B$. Therefore, we want $m \leq 2^{\cdot 172(n-n')/2}$ so that computing $A \cdot B$ can be done in time $\tilde{O}(2^{n-n'})$ using Theorem 5.43. For this we can set n' to be

$$n' = \frac{n}{c \cdot \sqrt{s} \cdot \log^2(s) \cdot D},$$

where $c > 0$ is some sufficiently large constant. Together with the running time in Equation (5.11), The total running time of the algorithm is therefore

$$2^{n-\Omega\left(\frac{n}{\sqrt{s}\cdot\log^2(s)\cdot D}\right)}\cdot \mathsf{poly}(n).$$

For the randomized case, for each g^i ($i \in [s]$), we consider a randomized protocol Π_i that has error $\varepsilon' \leq 1/(3 \cdot s \cdot 2^{n'})$, and replace g^i with a randomly picked protocol from Π_i, so we can say that for every $x \in n - n'$, the algorithm computes $Q(x)$ (or $Q'(x)$) with probability at least $2/3$ (via a union bound over all the g^i's and a union bound over all the z's in $\{0,1\}^{n'}$). Then we can repeat the above algorithm $\mathsf{poly}(n)$ times and obtain $Q(x)$ for all $x \in \{0,1\}^{n-n'}$ correctly with high probability. Note that the error of any randomized protocol with communication complexity R can be reduced to ε' by blowing up the communication complexity by a factor of $O(\log(1/\varepsilon'))$. In this case the, (as we are considering longer transcripts) the number of terms in Q' (as in Equation (5.10)) will be

$$2^{O(\sqrt{s}\cdot\log^2(s)\cdot R\cdot(n')^2)},$$

and we need to set accordingly

$$n' = \Omega\left(\frac{n}{\sqrt{s}\cdot\log^2(s)\cdot R}\right)^{1/2},$$

which gives the claimed running time for the randomized case. □

In fact, using the ideas above we can also get a randomized #SAT algorithm for the more expressive class $\mathsf{FORMULA}\circ\mathsf{AC}^0_{d,M}\circ\mathcal{G}$, where $\mathsf{AC}^0_{d,M}$ is the class of depth-d size-M circuits and $\mathcal{G}$ is the class of functions that have low-communication complexity[9], by combining with the fact that AC^0 circuits have low-degree *probabilistic polynomials over the reals* (a probabilistic polynomial of a function f is a distribution on polynomials such that for every input x, a randomly picked polynomial from the distribution agrees with f on the input x). More specifically, we have the following.

Theorem 5.44. *For any integers $s, d, M > 0$, there exists a randomized #SAT algorithm for $\mathsf{FORMULA}[s]\circ\mathsf{AC}^0_{d,M}\circ\mathcal{G}$, where $\mathcal{G}$ is the class of functions with explicit two-party deterministic protocols of communication cost at most D, the algorithm outputs the number of satisfying assignments in time*

$$2^{n-\left(\frac{n}{\sqrt{s}\cdot\log^2(s)\cdot(\log M)^{O(d)}\cdot D}\right)^{1/2}}\cdot\mathsf{poly}(n,s).$$

[9]Here we define the size of a $\mathsf{AC}^0_{d,M}$ circuit to be the number of wires. Note that a circuit in $\mathsf{FORMULA}\circ\mathsf{AC}^0_{d,M}\circ\mathcal{G}$ can have M functions from $\mathcal{G}$ at the bottom.

In the case $\mathcal{G}$ is the class of functions with explicit randomized protocols of communication cost at most R, there exists an analogous randomized algorithm with a running time

$$2^{n-\left(\frac{n}{\sqrt{s}\cdot\log^2(s)\cdot(\log M)^{O(d)}\cdot R}\right)^{1/3}}\cdot\mathsf{poly}(n,s).$$

Proof sketch. We show the case where $\mathcal{G}$ has low randomized communication complexity. Let

- $\varepsilon_1 = 1/\left(3\cdot 2^{n'}\right)$,
- $\varepsilon_2 = 1/\left(6\cdot s\cdot 2^{n'}\right)$ and
- $\varepsilon_3 = 1/\left(6\cdot M\cdot 2^{n'}\right)$.

As in the proof of Theorem 5.42, we can replace the formula part of $\mathsf{FORMULA}[s]\circ\mathsf{AC}^0_{d,M}\circ\mathcal{G}$ with a ε_1-approximating polynomial of degree

$$O(\sqrt{s}\cdot\log(s)\cdot\log(1/\varepsilon_1)) = O(\sqrt{s}\cdot\log(s)\cdot n').$$

Then we replace the $\mathsf{AC}^0_{d,M}$ circuit with a randomly picked polynomial from a ε_2-error probabilistic polynomial. By [HS19], such a probabilistic polynomial is constructive and has degree at most

$$(\log M)^{O(d)}\cdot\log(1/\varepsilon_2) = (\log M)^{O(d)}\cdot(n'+\log(s)).$$

Finally, we replace each of the bottom functions, which is from $\mathcal{G}$, with a randomly picked protocol from a randomized protocol with error ε_3, and hence has cost at most

$$R\cdot O(\log(1/\varepsilon_3)) = O(R\cdot(n'+\log(M))).$$

As a result, we can express Q' as a polynomial with at most

$$2^{O\left(\sqrt{s}\cdot\log^2(s)\cdot(\log M)^{O(d)}\cdot R\cdot(n')^3\right)}$$

monomials, whose variables are functions that depend on either the first half or the second half of x. Note that with our choices of ε_2 and ε_3, for every $x\in\{0,1\}^{n-n'}$, the algorithm computes $Q(x)$ correctly that with probability at least $2/3$ (by union bounds). By the same reasoning as in the proof of Theorem 5.42, we get a randomized #SAT algorithm with running time

$$2^{n-\left(\frac{n}{\sqrt{s}\cdot\log^2(s)\cdot(\log M)^{O(d)}\cdot R}\right)^{1/3}}\cdot\mathsf{poly}(n),$$

as desired. □

It is worth noting that unlike Theorem 5.42, the algorithm in Theorem 5.44 is *randomized* even if $\mathcal{G}$ is the class of functions with low *deterministic* communication complexity, because of the use of probabilistic polynomials for the AC^0 circuits.

5.6 Learning algorithms

In this section, we prove the following learning result for the $\mathsf{FORMULA} \circ \mathsf{XOR}$ model.

Theorem 5.45. *For every constant $\gamma > 0$, there is an algorithm that* PAC *learns the class of n-variate Boolean functions $\mathsf{FORMULA}[n^{2-\gamma}] \circ \mathsf{XOR}$ to accuracy ε and with confidence δ in time $\mathsf{poly}(2^{n/\log n}, 1/\varepsilon, \log(1/\delta))$.*

We first review some useful results that pertain to agnostically learning parities as well as boosting of learning algorithms.

5.6.1 Agnostically learning parities and boosting

For a parameter $n \geq 1$, let Δ be a distribution on labelled examples (x, y) supported over $\{0,1\}^n \times \{0,1\}$, and assume that for each x there is at most one y such that $(x, y) \in \mathsf{Support}(\Delta)$. For a function $h\colon \{0,1\}^n \to \{0,1\}$, we denote by $\operatorname{err}_\Delta(h)$ the error of h under this distribution:

$$\operatorname*{err}_{\Delta}(h) \;=\; \mathop{\mathbf{Pr}}_{(x,y)\sim\Delta}[h(x) \neq y]\,.$$

Similarly, for a class of functions $\mathcal{C}$, we let $\operatorname{opt}_\Delta(\mathcal{C})$ be the error of the best function in the class:

$$\operatorname*{opt}_{\Delta}(\mathcal{C}) \;=\; \min_{h\in\mathcal{C}} \operatorname*{err}_{\Delta}(h)\,.$$

We will need a result established by Kalai, Mansour, and Verbin [KMV08], which gives a non-trivial time agnostic learning algorithm for the class of parities.

Lemma 5.46 ([KMV08])**.** *Let XOR be the class of parity functions on n variables. Then, for any constant $\zeta > 0$, there is a randomized learning algorithm W such that, for every parameter $n \geq 1$ and distribution Δ over labelled examples, when W is given access to independent samples from Δ it outputs with high probability a circuit computing a hypothesis $h : \{0,1\}^n \to \{0,1\}$ such that*

$$\operatorname*{err}_{\Delta}(h) \;\leq\; \operatorname*{opt}_{\Delta}(\mathsf{XOR}) + 2^{-n^{1-\zeta}}.$$

The sample complexity and running time of W is $2^{O(n/\log n)}$.

Recall that a boosting procedure for learning algorithms transforms a weak learner that outputs a hypothesis that is just weakly correlated with the unknown function into a (strong) PAC learning algorithm for the same class (i.e., a learner in the sense of Definition 5.16).

We refer for instance to [KV94] for more information about boosting in learning theory. We shall make use of the following boosting result by Freund [Fre90].

Lemma 5.47 ([Fre90])**.** *Let W be a (weak) learner for a class $\mathcal{C}$ that runs in time $t(n)$ and outputs (under any distribution) a hypothesis of error up to $1/2 - \beta$, for some constructive function $\beta(n) > 0$. Then, there exists a* PAC *learning algorithm for $\mathcal{C}$ that runs in time* $\mathsf{poly}(n, t, 1/\varepsilon, 1/\beta, \log(1/\delta))$.

5.6.2 PAC-learning small formulas of parities

We are ready to show that sub-quadratic size formulas over parity functions can be learned in time $2^{O(n/\log n)}$. First, we argue that Lemma 5.46 provides a weak learner that works under any distribution $\mathcal{D}$ supported over $\{0,1\}^n$. This will follow from Lemma 5.19, which shows that any function in $\mathsf{FORMULA}[s] \circ \mathsf{XOR}$ is correlated with some parity function with respect to $\mathcal{D}$. We then obtain a standard PAC learner via the boosting procedure from Lemma 5.47.

Proof of Theorem 5.45. Let $\mathcal{C} = \mathsf{FORMULA} \circ \mathsf{XOR}$, where $s = n^{2-\gamma}$ for some constant $\gamma > 0$. For any function $f \in \mathsf{FORMULA}[s] \circ \mathsf{XOR}$ and distribution $\mathcal{D}$ supported over $\{0,1\}^n$, Lemma 5.19 shows that there exists a parity function $\chi = \chi(f, \mathcal{D})$ such that

$$\Pr_{x \sim \mathcal{D}}[f(x) = \chi(x)] \;\geq\; \frac{1}{2} + \frac{1}{2^{n^{1-\lambda}}},$$

for some $\lambda = \lambda(\gamma) > 0$ independent of n, under the assumption that n is sufficiently large. Let $\Delta = \Delta(\mathcal{D}, f)$ be the distribution over labelled examples induced by $\mathcal{D}$ and f. Note that $\mathrm{opt}_\Delta(\mathsf{XOR}) \leq 1/2 - \exp(n^{1-\lambda})$. Consequently, by invoking Lemma 5.46 with parameter $\zeta = \lambda$, it follows that $\mathsf{FORMULA}[n^{2-\gamma}] \circ \mathsf{XOR}$ can be learned under an arbitrary distribution to error $\beta(n) \leq 1/2 - \exp(n^{1-\Omega(1)})$ in time $t(n) = 2^{O(n/\log n)}$. Consequently, we can obtain a PAC learner algorithm for $\mathsf{FORMULA}[n^{2-\gamma}] \circ \mathsf{XOR}$ via Lemma 5.47 that runs in time $\mathsf{poly}(n, t(n), 1/\varepsilon, 1/\beta, \log(1/\delta)) = \mathsf{poly}(2^{n/\log n}, 1/\varepsilon, \log(1/\delta))$. □

Chapter 6

Circuit Lower Bounds for MCSP

6.1 Background and results

Given the truth table of some Boolean function f and a size parameter θ, the minimum circuit size problem (MCSP) asks whether f can be computed by a circuit of size at most θ. Understanding the exact complexity of MCSP is an important open problem in computational complexity theory, dating back to the 1950s [Tra84].

It is easy to see that MCSP is in NP. A popular conjecture is that MCSP is also NP-hard. However, despite serious efforts over the years, such a proof is still unknown. Given that it is difficult to show that MCSP is hard, perhaps the problem is easy? It turns out that this cannot be the case under some plausible cryptographic assumptions. More specifically, it is known that if one-way functions exist, then MCSP is not in P [KC00]. As proving an *unconditional* lower bound for MCSP seems far beyond the reach of currently known techniques, can we at least prove unconditional lower bounds for MCSP against some restricted computational models?[1]

Two of the most studied restricted computational models in complexity theory are constant-depth circuits (AC^0) and De Morgan formulas. For AC^0 circuits, the best-known lower bound is about PARITY: PARITY on N variables requires depth-d AC^0 circuits of size $2^{\Omega\left(N^{1/(d-1)}\right)}$ [Hås89]. For De Morgan formulas, the state-of-the-art lower bound is almost cubic, namely $N^{3-o(1)}$, for some polynomial-time computable function [Hås98, Tal14, Tal17a, DM18].

Notably, there are also lower bounds against these models for MCSP. Allender et al. [ABK+06] showed that MCSP, on functions represented as a truth table of length N, cannot be computed by polynomial-size constant-depth AC^0 circuits. In fact, by a more careful analysis of their argument, one can get a lower bound of $2^{N^{1/(c\cdot d+O(1))}}$, for a constant $c \geq 2$. However, such a lower bound still has a worse dependence on the depth compared

[1] A recent line of research on *hardness magnification* [OS18, OPS19] provides another motivation for proving relatively weak lower bounds for restricted circuit models against certain "gap variants" of MCSP. Such lower bounds are shown to imply much stronger (superpolynomial) lower bounds.

to the PARITY lower bound. For De Morgan formulas, Hirahara and Santhanam [HS17] showed that computing MCSP requires De Morgan formulas of size $N^{2-o(1)}$.

Given these two MCSP lower bounds and the best-known lower bounds against these two models, it is natural to ask whether we can get MCSP lower bounds against small-depth circuits and De Morgan formulas that match the state-of-the-art lower bounds against these models. More specifically, can we show that computing MCSP requires depth-d AC^0 circuits of size $2^{N^{1/(d+O(1))}}$ and De Morgan formulas of size $N^{3-o(1)}$? Furthermore, can we show lower bounds for MCSP against some other restricted models that match their state-of-the-art lower bounds? In this work, we answer these questions in the affirmative.

6.1.1 Results

Our first result is an almost-cubic De Morgan formula lower bound for MCSP.

Theorem 6.1. *Any De Morgan formula computing* MCSP *on truth tables of length N must have size at least $N^3/2^{O(\log^{2/3} N)}$.*

We also get almost-quadratic lower bounds against formulas over an arbitrary basis as well as general branching programs; these almost match the best-known lower bounds against these models [Nec66].

Theorem 6.2. *Let C be either a formula over any basis or a branching program that computes* MCSP *on truth tables of length N. Then C must have size at least $N^2/2^{O(\sqrt{\log N})}$.*

For small-depth circuits, we have the following improved lower bound for MCSP, which its dependence on the depth matches the one in the PARITY lower bound, up to a small additive constant.

Theorem 6.3. *For every $d > 2$ and every constant $\gamma > 0$, any depth-d AC^0 circuit computing* MCSP *on truth tables of length N must have size $2^{\Omega(N^{1/(d+2+\gamma)})}$.*

For the special case of depth-2 circuits, we can have an almost optimal lower bound.

Theorem 6.4. *Any* CNF *or* DNF *computing* MCSP *on truth tables of length N must have size $2^{\Omega(N)}$.*

Also, in this work, we give a fine-grained analysis of the approach of obtaining MCSP lower bounds from average-case hardness via the Nisan-Wigderson framework (see Section 6.7).

6.1.2 Techniques

For a class $\mathfrak{C}$ of N-variate Boolean functions, a pseudorandom generator (PRG) against $\mathfrak{C}$ is a deterministic efficiently-computable function G mapping short binary strings (seeds) to

longer binary strings so that every function in $\mathfrak{C}$ accepts G's output on a uniformly random seed with about the same probability as that for an actual uniformly random string.

A key notion in this work is that of a local PRG. We say that a PRG is *local* if its N-bit output (viewed as the truth table of some function) has small circuit complexity. More precisely, for any fixed seed to the PRG, there exists a small circuit such that, given $j \in [N]$ as an input, the circuit computes the j-th bit of the PRG output, where the complexity of the circuit is measured relative to its input length, namely $\log N$. Note that our notion of local PRGs does not require that the PRG in question is explicit; that is, we do not require that a local PRG can be computed by some uniform algorithm.

Local PRGs in the context of MCSP (and related problems) have been studied in previous works (see, e.g., [ABK+06, OS17, HS17, Hir18]). In this work, we refine the previous approaches, and obtain stronger circuit lower bounds by establishing strong locality properties of certain PRG constructions.[2]

MCSP lower bounds from local PRGs

Suppose we have a local PRG against some class of circuits $\mathfrak{C}$ of size s, and we want to show that MCSP cannot be computed by any size-s circuit in $\mathfrak{C}$. Suppose some size-s circuit C in $\mathfrak{C}$ computes MCSP. Using the fact that a random function has almost maximum circuit complexity, we have that C will output FALSE on most of its inputs (by setting the size parameter θ to be a non-trivial quantity that is asymptotically smaller than $2^n/n$, where n is the input length of the function). If we replace the uniformly random inputs with the outputs of the local PRG, then, by the definition of PRG, C will still output FALSE with large probability. However, since the PRG is local, all of its outputs have circuit complexity smaller than the size parameter θ, and hence must be accepted by C. A contradiction.

To get a strong lower bound, we would like to make the above argument to work for large s. Note that the local complexity of the PRG, $\lambda(s)$, is a function on the size of the circuit C, and we need this local complexity to be "non-trivial" in order to reach a contradiction. Therefore, we want to choose s so that this local complexity remains asymptotically smaller than $2^n/n$. As a result, the final lower bound (i.e., the largest s that we can choose) is determined by the local complexity λ. So the main question we study in our work is: What is the smallest local complexity of a PRG against a given circuit class?

[2]Note that, as one of our ICALP'19 reviewers pointed out, the notion of a local PRG can be also found in the context of cryptography [CM01], where a PRG $G : \{0,1\}^n \to \{0,1\}^m$ is called *k-local*, for some constant $k > 0$, if every output PRG bit $G(x)_j$, for any $x \in \{0,1\}^n$ and $j \in [m]$, depends only on k input bits $x_{i_1}, \ldots, x_{i_k}$, for $i_1, \ldots, i_k \in [n]$. In our work, however, locality refers to the circuit complexity of the PRG at hand and the output bits of our PRGs may depend on a super-constant number of input bits.

MCSP lower bound against De Morgan formulas

Our formula lower bound for MCSP is obtained by applying the framework described above to a local PRG against formulas. The state-of-the-art PRG against formulas is given by Impagliazzo, Meka, and Zuckerman [IMZ19], which we refer to as the IMZ PRG. Their PRG has a seed length of $s^{1/3+o(1)}$ for size s formulas (note that such a PRG is useful against sub-cubic formulas only). If we want to utilize the IMZ PRG to get an MCSP lower bound against formulas, we will need to argue that the IMZ PRG is local.

In fact, in order to get an almost-cubic lower bound, we will need such a PRG to be strongly local in the sense that any single output bit of the PRG (on any given fixed seed) can be computed by a circuit of size comparable to its seed length, which is $s^{1/3+o(1)}$. However, by inspecting the construction, the IMZ PRG does not seem to have such a property, and a straightforward implementation seems to require a circuit of size at least $s^{2/3}$ (see Section B.2 for more details), which yields a weaker lower bound for MCSP.

To overcome this issue, we present an alternative PRG useful against sub-cubic formulas which is strongly local. The construction of this PRG can be viewed as a modification of the IMZ PRG. At a high level, it is based on the Ajtai-Wigderson construction [AW89], which is a framework for constructing PRGs against computations that can be simplified under (pseudo)random restrictions. This framework is then combined with the ideas for reducing (recycling) random bits using an extractor, by exploiting communication bottlenecks in computations [NZ96]. Our modification, particularly the utilization of the Ajtai-Wigderson construction, allows us to compute any output bit of the PRG efficiently by reducing the number of calls to the extractor. Using some crucial observations on the circuit complexity of certain pseudorandom objects, we get a PRG that is locally computable by a $s^{1/3+o(1)}$-size circuit.[3]

MCSP lower bounds against formulas over an arbitrary basis or branching programs

The MCSP lower bounds against formulas over an arbitrary basis or branching programs are obtained similarly to those for De Morgan formulas. The idea is to construct strongly local PRGs against these models by modifying the PRGs in [IMZ19]. Then, by applying our "MCSP circuit lower bounds from local PRGs" framework, we get the desired lower bounds.

[3]It is also possible to use the original IMZ PRG to obtain an almost-cubic formula lower bound for MCSP. We can show that the IMZ PRG, although not fully strongly local, is "almost strongly local" in the sense that *most* of its outputs have very small circuit complexity; see Section B.2.

MCSP **lower bounds against** AC^0

We use a local PRG against AC^0 to get MCSP lower bounds. To get a lower bound matching the one in Theorem 6.3, we can use the state-of-the-art PRG against AC^0 by Trevisan and Xue [TX13], which has a seed length of $(\log s)^{d+O(1)}$ for size-s depth-d AC^0 circuits. By a careful analysis of the construction of this PRG, we can show that the Trevisan-Xue PRG is strongly local and can be used to get an MCSP lower bound that is close to the one stated in Theorem 6.3. However, in this work, we will present a more direct proof of such a lower bound by using the pseudorandom switching lemma for constant-depth circuits, which is due to Trevisan and Xue [TX13] as well, and is a key ingredient in their PRG.

The idea is to show that for any small-depth circuit of size less than the claimed lower bound, there is some locally computable restriction that turns the circuit into a constant function, but leaves many variables unrestricted. However, MCSP cannot be constant under such a restriction, because depending on the partial assignment to the unrestricted variables, the resulting input function (which is composed of the restriction and the partial assignment) can be either easy or hard. Such an approach based on pseudorandom restrictions can also be applied to the special case of depth-2 circuits to get almost optimal CNF (and DNF) lower bounds for MCSP.

Organization of this chapter. We give the necessary background in Section 6.2. In Section 6.3, we describe our framework of using local PRGs to obtain lower bounds for MCSP. We prove the almost-cubic De Morgan formula lower bound for MCSP (Theorem 6.1) in Section 6.4, and the almost-quadratic lower bounds against formulas over an arbitrary basis and branching programs (Theorem 6.2) in Section 6.5. The improved AC^0 lower bounds for MCSP (Theorem 6.3 and Theorem 6.4) are proved in Section 6.6. In Section 6.7, we discuss the framework of proving MCSP lower bounds from average-case hardness. Finally, we give some open problems in Section 6.8.

6.2 Preliminaries

Notation

Throughout this chapter, we will use $\{0,1\}$ as the Boolean domain.

For any computational model, we use the term *size* to refer to its complexity measure. For example, if the model is circuits of some fixed depth, then the size is the number of gates in the circuit.

For a positive integer n that is a power of two,[4] we use the following notation:

[4]We may sometimes implicitly assume that some quantity, such as the number of variables, or circuit size, is a "nice" number (e.g., a power of two). This can always be made true by adding dummy variables or

- $[n]$ denotes the set $\{1, \ldots, n\}$. We will sometimes identify $[n]$ with $\{0,1\}^{\log n}$, in a natural way.
- $\mathbb{F}_n$ denotes the field with n elements. Again, we will sometimes identify $\mathbb{F}_n$ with $\{0,1\}^{\log n}$ where the elements in $\mathbb{F}_n$ are represented by $(\log n)$-bit strings.
- U_n denotes the uniform distribution over $\{0,1\}^n$.
- For a function $f\colon \{0,1\}^n \to \{0,1\}$, $\mathtt{tt}(f) \in \{0,1\}^{N=2^n}$ denotes the truth table of f, and $\mathtt{CC}(f)$ denotes its circuit complexity, that is, the size of the smallest Boolean circuit that computes f.

Useful facts about Boolean circuits and pseudorandomness

We refer to a textbook as [Juk12] for a general introduction to Boolean circuits.

Proposition 6.5. *A Boolean circuit of size s can be specified using $O(s \log s)$ bits. Hence there are at most $2^{O(s \log s)} = s^{O(s)}$ distinct circuits of size at most s.*

Theorem 6.6 ([Sha49])**.** *The fraction of functions on n variables that have a circuit of size less than $2^n/(3n)$ is $o(1)$.*

Lemma 6.7. *For any integer $t > 0$, there exists a circuit C of size $\widetilde{O}(t)$ such that, given any string $x \in \{0,1\}^t$, the circuit does the following:*

- *If $x = 0^t$, then C outputs $(0, 0^{\log t})$.*
- *If $x \neq 0^t$, then C outputs $(1, q)$, where $q \in \{0,1\}^{\log t}$ is the index of the first bit in x that is not 0.*

Proof. Define $z^{(0)} = \left(0, 0^{\log t}\right)$ and $z^{(i)}$, for any $i = 1, \ldots, t$, recursively as follows:

$$z^{(i)} = \begin{cases} z^{(i-1)}, & \text{if } (z^{(i-1)})_1 = 1, \\ z^{(i-1)}, & \text{if } (z^{(i-1)})_1 = 0 \text{ and } x_i = 0, \text{ and} \\ (1, i), & \text{if } (z^{(i-1)})_1 = 0 \text{ and } x_i = 1. \end{cases}$$

Note that each $z^{(i)}$ can be computed in $\mathsf{polylog}(t)$ size given $z^{(i-1)}$. Using a circuit of size $\widetilde{O}(t)$ we can compute $z^{(t)}$, which is our output. □

The following circuit upper bound for the addressing (storage access) function is well-known (see, e.g., [Weg87]); we include a proof for completeness.

dummy gates, which may change the respective quantity by a small amount, and all of our results will still hold asymptotically.

Lemma 6.8. *For any integers $t, m > 0$, there exists a circuit of size $O(t \cdot m)$ such that, given any string $y = (y_1, \ldots, y_t)$, where $y_i \in \{0,1\}^m$, for each i, and an index $i \in \{0,1\}^{\log t}$, the circuit outputs y_i.*

Proof. We first look at the first bit (i.e., the least significant bit in binary) of i and output either the first half of y (i.e., $y_1, \ldots, y_{t/2}$), if the first bit is 0, or the second half (i.e., $y_{(t/2)+1}, \ldots, y_t$), if the first bit is 1; denote this output as $y^{(1)}$. This can be done by a circuit of size $c \cdot t \cdot m$, for some constant $c > 0$. Then, we look at the second bit of i and output either the first half or the second half of $y^{(1)}$, denoted as $y^{(2)}$. This can be done by a circuit of size $c \cdot t \cdot m/2$. We repeat the above process $\log t$ times, in total, until we get $y^{(\log t)}$, which is y_i. The circuit complexity of this procedure is

$$\sum_{k=1}^{\log t} (c \cdot t \cdot m)/2^{k-1} = O(t \cdot m). \qquad \square$$

We will need the following concentration bound for k-wise independent distributions, which is an application of Cantelli's inequality.

Proposition 6.9. *For any $0 < p < 1$, let $X_1, \ldots, X_n$ be pair-wise independent variables over $\{0,1\}$ such that $\mathbf{Pr}[X_i = 1] = p$ for each $i \in [n]$. Then, it is the case that*

$$\mathbf{Pr}[X \leq pn/2] \leq \frac{4}{pn}.$$

The following simple fact will be convenient for us.

Lemma 6.10. *Let X and Y be two random variables that take values in $\{0,1\}$ and $\mathcal{E}$ be some event. If*

- $|\mathbf{E}[X \mid \mathcal{E}] - \mathbf{E}[Y \mid \mathcal{E}]| \leq \varepsilon_1$ *and*
- $\mathbf{Pr}[\neg\mathcal{E}] \leq \varepsilon_2$,

then $|\mathbf{E}[X] - \mathbf{E}[Y]| \leq \varepsilon_1 + \varepsilon_2$.

Proof. We have

$$\mathbf{E}[X] = \mathbf{E}[X \mid \mathcal{E}] \cdot \mathbf{Pr}[\mathcal{E}] + \mathbf{E}[X \mid \neg\mathcal{E}] \cdot \mathbf{Pr}[\neg\mathcal{E}],$$

and

$$\mathbf{E}[Y] = \mathbf{E}[Y \mid \mathcal{E}] \cdot \mathbf{Pr}[\mathcal{E}] + \mathbf{E}[Y \mid \neg\mathcal{E}] \cdot \mathbf{Pr}[\neg\mathcal{E}].$$

Then,

$$\begin{aligned}\mathbf{E}[X] - \mathbf{E}[Y] &= (\mathbf{E}[X \mid \mathcal{E}] - \mathbf{E}[Y \mid \mathcal{E}]) \cdot \mathbf{Pr}[\mathcal{E}] + (\mathbf{E}[X \mid \neg\mathcal{E}] - \mathbf{E}[Y \mid \neg\mathcal{E}]) \cdot \mathbf{Pr}[\neg\mathcal{E}] \\ &\leq |\mathbf{E}[X \mid \mathcal{E}] - \mathbf{E}[Y \mid \mathcal{E}]| + \mathbf{Pr}[\neg\mathcal{E}] \\ &\leq \varepsilon_1 + \varepsilon_2.\end{aligned}$$

The fact $\mathbf{E}[Y] - \mathbf{E}[X] \leq \varepsilon_1 + \varepsilon_2$ can be similarly shown. □

6.3 The "MCSP circuit lower bounds from local PRGs" framework

We first describe how to use local PRGs to obtain circuit lower bounds for MCSP.

Definition 6.11 (Local PRGs)**.** *Let $\lambda\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a size function. For any Boolean computational model and size $s > 0$, we say that a function $G\colon \{0,1\}^{r=r(N,s)} \to \{0,1\}^N$ is a $(N, s, \lambda(N,s))$-*local PRG against the model *if*

- *G 1/3-fools every device f on N variables of size s in the model; that is,*

$$\left| \mathop{\mathbf{E}}_{z \sim \{0,1\}^r}[f(G(z))] - \mathop{\mathbf{E}}_{x \sim \{0,1\}^N}[f(x)] \right| \leq 1/3,$$

  and

- *for any seed $z \in \{0,1\}^r$, the function $g_z\colon \{0,1\}^{\log N} \to \{0,1\}$ defined as $g_z(j) = G(z)_j$ can be computed by a general circuit of size at most $\lambda(N,s)$.*

Remark 6.12. *Definition 6.11 is a notable departure from earlier work on PRGs, in that there is no requirement that a local PRG is easy to compute. Instead, the utility of the PRG is derived from the requirement that each of the functions g_z is easy to compute.*

Theorem 6.13. *There exists a constant $c > 0$ such that the following holds. For any computational model, let s be such that* MCSP *on truth tables of length N can be computed by a device of size s in the model. If there exists some $(N, s, \lambda(N,s))$-local PRG, of any seed-length, against the model, then $\lambda(N,s) \geq \frac{N}{c \log N}$.*

Proof. Let C be a device in the computational model such that C computes MCSP on truth tables of length N. Suppose C has size s, and let G be a $(N, s, \lambda(N,s))$-local PRG against C with some seed length r.

For the sake of contradiction, suppose that

$$\lambda(N,s) < \frac{N}{c \log N}.$$

On the one hand, since most functions require circuits of size greater than $\frac{N}{c \log N}$ (Theorem 6.6) and C computes MCSP, we have

$$\mu = \mathop{\mathbf{Pr}}_{\mathtt{tt}(f) \sim \{0,1\}^N} [C(\mathtt{tt}(f), \lambda(N,s)) = 0] \geq 1/2.$$

Also, since G fools C, we have

$$\mathop{\mathbf{Pr}}_{z \sim \{0,1\}^r} [C(G(z), \lambda(N,s)) = 0] \geq \mu - 1/3 \geq 1/6.$$

On the other hand, because G is $(N, s, \lambda(N,s))$-local, we must have $C(G(z), \lambda(N,s)) = 1$, for every z. A contradiction. □

It is easy to see that a local hitting set generator (HSG) is sufficient for the above argument to work. HSGs are a weak version of PRGs with the following property: For every function f in the class, if f accepts many of its inputs, then a HSG outputs such an input for at least one of its seeds.

6.4 Almost-cubic De Morgan formula lower bounds for MCSP

In this section, we present our almost-cubic De Morgan formula lower bound for MCSP. By saying "formula" within this section, we refer to formulas over the De Morgan basis (AND, OR, and NOT). By the size of a formula, we mean its usual leaf complexity, i.e., the number of leaves in the tree representation of the formula.

Theorem 6.14 (Theorem 6.1, restated)**.** *Any De Morgan formula computing* MCSP *on truth tables of length N must have size at least $N^3/2^{O\left(\log^{2/3} N\right)}$.*

We will construct a strongly local PRG useful against sub-cubic formulas. That is, given as input an index j, the j-th bit of the PRG can be computed by a circuit of size that is comparable to its seed length, which in our case is around $s^{1/3}$ for size s formulas.

Lemma 6.15. *For any $s \geq N$, there exists a $\left(N, s, s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}\right)$-local PRG against De Morgan formulas.*

Given the local PRG in Lemma 6.15, we can combine it with our Theorem 6.13 to obtain a formula lower bound for MCSP.

Proof of Theorem 6.14. Let $s \leq N^3$ be such that MCSP on truth tables of length N can be computed by some formula of size s. By Theorem 6.13 and Lemma 6.15, we have

$$s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)} \geq N/(c \log N);$$

then, $s \geq N^3 / \left(2^{O\left(\log^{2/3} N\right)} c^3 \log^3 N\right)$. □

The rest of this section is devoted to proving Lemma 6.15.

6.4.1 Almost-linear-size k-independent generators

The PRG in Lemma 6.15 will use k-wise independent distributions. Recall that a multidimensional distribution is called *k-wise independent* if any k coordinates of the distribution are uniformly distributed (see Definition 2.6).

A *k-independent generator* is a function from binary strings to binary strings that takes as input a random seed and stretches that seed to a string that follows a k-wise independent distribution. We will need efficient and local constructions for k-independent generators as well as some other pseudorandom objects. These objects can be constructed using finite fields; we need the following result, which says that finite field arithmetic can be performed by almost-linear-size circuits.

Fact 6.16 (See, e.g., [vzGG13, GS13])**.** *For any integer $\ell > 0$, let the elements in $\mathbb{F}_{2^\ell}$ be represented by ℓ-bit strings. Then, addition over $\mathbb{F}_{2^\ell}$ can be performed by a circuit of size $O(\ell)$ and multiplication over $\mathbb{F}_{2^\ell}$ can be performed by a circuit of size $\widetilde{O}(\ell)$.*

We now describe an efficient construction for k-independent generator, using the fact that finite field arithmetic can be done using *almost linear-size* circuits.

Lemma 6.17. *For any integer $k > 0$, there exists a k-independent generator $G\colon \{0,1\}^r \to [m]^n$, with $r = k \cdot \max\{\log n, \log m\}$, such that the following holds. There exists a circuit of size*

$$k \cdot \max\{\widetilde{O}(\log n), \widetilde{O}(\log m)\}$$

such that, given $j \in \{0,1\}^{\log n}$ and a seed $z \in \{0,1\}^r$, the circuit computes the j-th coordinate of $G(z)$.

Proof. Let $n' = \max\{n, m\}$ and suppose $n' = 2^\ell$. We view the elements in $\mathbb{F}_{n'}$ as ℓ-bit strings. Consider the following function $g\colon \mathbb{F}_{n'} \times \mathbb{F}_{n'}^k \to \mathbb{F}_{n'}$:

$$g(i, z_0, \ldots, z_{k-1}) = z_0 + z_1 \cdot i + \cdots + z_{k-1} \cdot i^{k-1}.$$

It is known (see [Vad12, Proposition 3.33]) that the function $G\colon \mathbb{F}_{n'}^k \to \mathbb{F}_{n'}^{n'}$ given as

$$G(z_0, \ldots, z_{k-1}) = \left(g(1, z_0, \ldots, z_{k-1}), \ldots, g(n', z_0, \ldots, z_{k-1})\right),$$

is a k-independent generator.

Using Fact 6.16 it is easy to implement a circuit of size $k \cdot \widetilde{O}(\ell)$ that computes $g(j, z)$. Note that to get an output in $[m]$ we can simply output the first $\log m$ bits of $G(z)_j$, since the field has characteristic 2. □

6.4.2 Almost-linear-size extractors

Our PRG will make use of randomness extractors. Here, we describe an extractor that is computable by a circuit of size that is almost linear in the length of its input. We start by reviewing some basic definitions regarding extractors.

Definition 6.18 (ε-closeness and statistical distance)**.** *Let $0 \leq \varepsilon \leq 1$. We say two distributions X and Y (over some universe D) are* ε-close *if their* statistical distance*, defined as*

$$\max_{T:D\to\{0,1\}} \left|\mathbf{Pr}[T(X)=1] - \mathbf{Pr}[T(Y)=1]\right|,$$

is at most ε.

Definition 6.19 (Min-entropy)**.** *Let X be a random variable. The* min-entropy *of X, denoted by $H_\infty(X)$, is the largest real number k such that $\mathbf{Pr}[X=x] \leq 2^{-k}$ for every x in the range of X. If X is a distribution over $\{0,1\}^\aleph$ with $H_\infty(X) \geq k$, then X is called a* $(\aleph, k)$-source*.*

Definition 6.20 (Extractors)**.** *A function $E\colon \{0,1\}^\aleph \times \{0,1\}^d \to \{0,1\}^m$ is an* (k,ε)-extractor *if, for any $(\aleph,k)$-source X, the distribution $E(X,U_d)$ is ε-close to U_m.*

We now state the extractor, which for a high min-entropy source extracts a constant fraction of the min-entropy, using seeds of polylogarithmic length. The construction and circuit complexity of this extractor are presented in Section B.1.

Lemma 6.21 (Almost-linear-size extractors, following [NZ96])**.** *There exists some randomness extractor $E\colon \{0,1\}^\aleph \times \{0,1\}^d \to \{0,1\}^m$ that is an $(\aleph/2,\varepsilon)$-extractor with $m = \Omega(\aleph)$ and $d = \mathsf{polylog}(\aleph/\varepsilon)$. Moreover, E can be computed by a circuit of size $\aleph \cdot \mathsf{polylog}(\aleph/\varepsilon)$.*

6.4.3 Strongly local PRG useful against sub-cubic De Morgan formulas

For a formula F, let $L(F)$ denote the size (which is measured by the number of leaves) of F. We need the following pseudorandom shrinkage lemma for De Morgan formulas, which says that there exists a p-regular restriction, where the unrestricted variables are selected pseudorandomly and the restricted variables are fixed truly-randomly, such that *with high probability* the size of the restricted formula will "shrink" by a factor of p^2.

Lemma 6.22 (Pseudorandom shrinkage lemma, [IMZ19, Lemma 4.8][5])**.** *There exists a constant $c_0 > 0$ such that the following holds. For any constant $c > c_0$, any $s \geq N$,*

[5]The pseudorandom shrinkage lemma in [IMZ19] is not stated in this form, but rather selects the unrestricted variables and fixes the restricted variables both pseudorandomly (based on limited independence). This immediately implies the above lemma, where the restricted variables are set independently (and hence also k-wise independently, for any k). Further, the last statement follows from the fact that the restricted variables are chosen by a k-wise independent distribution, which can be computed locally; see Lemma 6.17.

$p \geq s^{-1/2}$, and any De Morgan formula F on N variables of size s, there exists a p-regular pseudorandom selection $\mathcal{D}$ over N variables that is samplable using $r = 2^{O(\log^{2/3} s)}$ random bits such that

$$\mathop{\mathbf{Pr}}_{\sigma\sim\mathcal{D},x\sim\{0,1\}^N}\left[L\Big(F_{(\sigma,x)}\Big) \geq 2^{3\cdot c\cdot\log^{2/3} s}\cdot p^2\cdot s\right] \leq s^{-c}.$$

Moreover, there exists a circuit of size $2^{O(\log^{2/3} s)}$ such that, given $j \in \{0,1\}^{\log N}$ and a seed $z \in \{0,1\}^r$, the circuit computes the j-th coordinate of $\mathcal{D}(z)$.

We are now ready to show our PRG in Lemma 6.15.

Proof of Lemma 6.15. The construction is as follows: We first sample a p-regular pseudorandom selection from Lemma 6.22. Then, we fill the star coordinates, specified by the pseudorandom selection, in the output string with the output of some extractor which takes a min-entropy source sample and a short seed. (More precisely, the star coordinates are filled with the output of some limited-independence generator that takes the output of an extractor as a seed.) We then sample another pseudorandom selection, and fill the star coordinates specified by this pseudorandom selection but this time only for those that have not been filled in previous steps, again with the output of the same extractor using the *same min-entropy source sample* but a *different short seed*. We continue this way until all the coordinates are filled.

More formally, our PRG uses the following parameters:[6]

- $p = 1/s^{1/3}$, the expected fraction of unrestricted variables in each of the pseudorandom selections;
- $\varepsilon = 1/\mathsf{poly}(N)$ and $\varepsilon_0 = \varepsilon/(10t)$, which specify the error of the PRG;
- $t = \ln(4N/\varepsilon)/p = s^{1/3}\cdot O(\log N)$, the number of steps needed so that all the coordinates will be filled with probability $1-\varepsilon/4$;
- $s_0 = p^2\cdot s\cdot 2^{O\left(\log^{2/3} s\right)} = s^{1/3}\cdot 2^{O\left(\log^{2/3} s\right)}$, the size of the formula after being simplified by a pseudorandom restriction;
- $k \geq s_0 = s^{1/3}\cdot 2^{O\left(\log^{2/3} s\right)}$, the amount of independence needed to fool the simplified formula, and $r_k = k\cdot\log N$ the seed length for the k-independent generator;
- $\aleph$, the length of the min-entropy source for the extractor, which is such that $\aleph \geq 2\cdot\log(1/\varepsilon_0) + c\cdot s_0\cdot\log s_0$, where $c > 0$ is some constant, and that $\Omega(\aleph) \geq r_k$. We can take $\aleph = s^{1/3}\cdot 2^{O\left(\log^{2/3} s\right)}$;
- $d = \mathsf{polylog}(\aleph/\varepsilon_0) = \mathsf{polylog}(N)$, the seed length of the extractor;

[6]In fact, there are mainly two types of parameters here. Those that are close to $s^{1/3}$, which are $1/p, t, s_0, k, N$, and those that are close to $N^{o(1)}$, which are d and ℓ.

- $\ell = 2^{O(\log^{2/3} s)}$, the number of random bits for sampling a pseudorandom selection.

Construction. The PRG takes a seed $(X, Y_1, \ldots, Y_t, \gamma_1, \ldots, \gamma_t) \in \{0,1\}^r$, where

- $X \in \{0,1\}^{\aleph}$ is the min-entropy source sample of an extractor,
- $Y_i \in \{0,1\}^{\mathsf{polylog}(N)}$, for each $i \in [t]$, is the seed of an extractor, and
- $\gamma_i \in \{0,1\}^{\ell}$, for each $i \in [t]$, is the seed for sampling a pseudorandom selection.

The construction of the PRG proceeds in the following two stages.

Stage 1. Compute a sequence of t p-regular pseudorandom selections

$$\sigma_1, \ldots, \sigma_t,$$

using Lemma 6.22, with the seeds $\gamma_1, \ldots, \gamma_t$. Below, we denote the star coordinates in σ_i by $\sigma_i^{-1}(*)$. Let $S_1, \ldots, S_t \subseteq [N]$ be t disjoint sets defined by

$$S_i = \sigma_i^{-1}(*) \setminus (S_1 \cup \cdots \cup S_{i-1}).$$

Stage 2. Define $Z_1, \ldots, Z_t \in \{0,1\}^N$ by

$$Z_i = G_k(E(X, Y_i)),$$

where $E\colon \{0,1\}^{\aleph} \times \{0,1\}^d \to \{0,1\}^{\Omega(\aleph)}$ is an $(\aleph/2, \varepsilon_0)$-extractor and $G_k\colon \{0,1\}^{r_k} \to \{0,1\}^N$ is a k-independent generator. The final output of our PRG is the binary string that has the values $Z_i|_{S_i}$ in the positions indexed by S_i, for all $i \in [t]$, where $Z_i|_{S_i}$ denotes the bit values of Z_i projected to the set S_i. (We fix those positions that are not in any of the S_i's to be 0.) Stage 2 of the PRG construction is depicted in Figure 6.1.

Correctness. Next, we show that the above PRG ε-fools N-variate formulas of size s. First, note that, by our choice of t, with probability $1 - \varepsilon/4$, $S = S_1 \cup \cdots \cup S_t$ covers all N coordinates. To proceed, we will use a *hybrid argument*. Let G denote the distribution given by the PRG described above. Let U be the uniform distribution. Note that if in the above construction we replace Z_i, for all $i \in [t]$, with U, then we would get a uniform distribution. Now we can start from there and we will gradually replace U with the Z_i's step-by-step for a total of t steps. We will argue that after each replacement step, the expected value of the function does not change by much. Let B_i be the distribution where we have replaced U with Z_i in the first i steps and $S = [N]$. That is,

$$B_i = \left(Z_1|_{S_1}, \ldots, Z_i|_{S_i}, U|_{S_{i+1}}, \ldots, U|_{S_t} \right) = \left(Z_1|_{S_1}, \ldots, Z_i|_{S_i}, U|_{S_{i+1} \cup \cdots \cup S_t} \right)$$

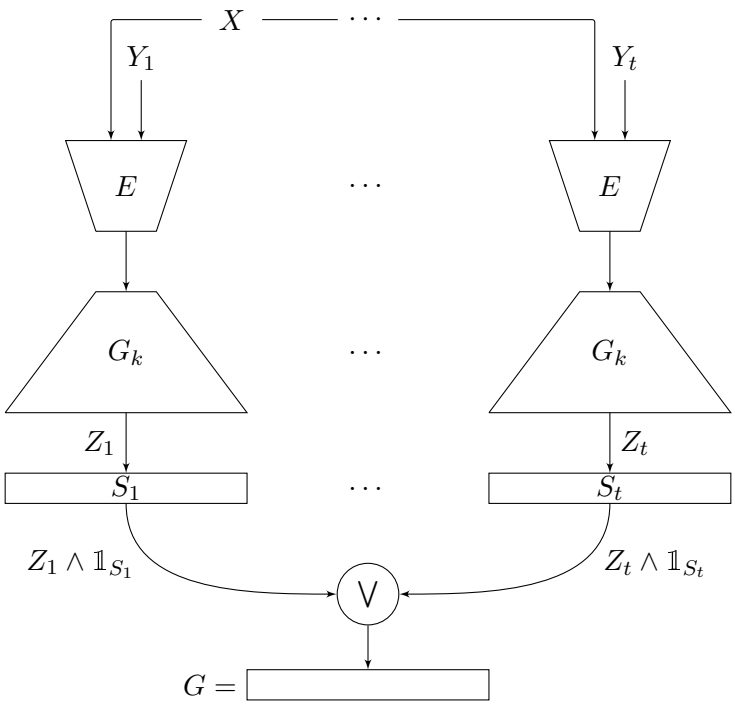

Figure 6.1: Construction of the PRG in Stage 2 of Lemma 6.15, Stage 2. For each $i \in [t]$, $\mathbb{1}_{S_i} \in \{0,1\}^N$ denotes the characteristic Boolean vector of the set S_i, where $S_i \subseteq [N]$ is the set of star coordinates, in the i-th pseudorandom selection, that did not appear in the preceding sets $S_1, \ldots, S_{i-1}$. Also, $\wedge$ denotes a coordinate-wise AND operation (i.e., coordinate-wise *multiplication* of Boolean vectors) and $\bigvee$ is a coordinate-wise OR operation.

and we want to show that $|\mathbf{E}[f(U)] - \mathbf{E}[f(G)]| = |\mathbf{E}[f(B_0)] - \mathbf{E}[f(B_t)]| \leq \varepsilon$. Let

$$A_i = \left(Z_1|_{S_1}, \ldots, Z_i|_{S_i}, U|_{S_{i+1}}, \ldots, U|_{S_t}, U|_{[N]\setminus S} \right) = \left(Z_1|_{S_1}, \ldots, Z_i|_{S_i}, U|_{\,S_{i+1} \cup \cdots \cup S_t \cup ([N]\setminus S)} \right)$$

be the version of the distribution B_i in the case where $S \subsetneq [N]$. Let $\mathcal{C}$ denote the event $S = [N]$; by Lemma 6.10, for all i, we get that

$$|\mathbf{E}[f(A_i)] - \mathbf{E}[f(B_i)]| \leq |\mathbf{E}[f(A_i) \mid \mathcal{C}] - \mathbf{E}[f(B_i) \mid \mathcal{C}]| + \Pr[\neg\mathcal{C}] \leq 0 + \varepsilon/4 = \varepsilon/4.$$

Therefore, it would suffice to establish $|\mathbf{E}[f(A_0)] - \mathbf{E}[f(A_t)]| \leq \varepsilon/2$ since this inequality would imply the desired $|\mathbf{E}[f(B_0)] - \mathbf{E}[f(B_t)]| \leq \varepsilon$.

Note that using the distributions would B_i require that $S = [N]$ and this could result in dependencies among the sets S_i. This is the reason for introducing the distributions A_i; we shall later make use of the fact that the selections σ_i, that come up in the definitions of the sets S_i, are independent.

Now, for the sake of contradiction, suppose there exists a size-s formula f on N variables such that

$$|\mathbf{E}[f(A_0)] - \mathbf{E}[f(A_t)]| > \varepsilon/2.$$

By the triangle inequality, there exists an $0 \leq i < t$ such that

$$|\mathbf{E}[f(A_i)] - \mathbf{E}[f(A_{i+1})]| > \varepsilon/(2t). \tag{6.1}$$

Let us say that both the expectations in Equation (6.1) are over

$$\sigma_1, \ldots, \sigma_{i+1}, Y_1, \ldots, Y_{i+1}, X, U,$$

and we remove the absolute value without loss of generality. Then, we have

$$\mathop{\mathbf{E}}_{\substack{\sigma_1,\ldots,\sigma_i,\\ Y_1,\ldots,Y_i,\\ X}} \left[\mathop{\mathbf{E}}_{\sigma_{i+1},Y_{i+1},U} [f(A_i)] - \mathop{\mathbf{E}}_{\sigma_{i+1},Y_{i+1},U} [f(A_{i+1})] \right] > \varepsilon/(2t). \tag{6.2}$$

Denote $W_i = (\sigma_1, \ldots, \sigma_i, Y_1, \ldots, Y_i, X)$, and let f' be the random function (where the randomness is over W_i) defined as

$$f' = f(Z_1|_{S_1}, \ldots, Z_i|_{S_i}, \quad \cdot \quad).$$

That is, f' is the restricted function after the first i steps. Then, the left hand side of Equation (6.2) becomes

$$\begin{aligned} \mathop{\mathbf{E}}_{W_i} \big[& \mathop{\mathbf{E}}_{\sigma_{i+1},U} \left[f'(U|_{S_{i+1}}, U|_{S_{i+2} \cup \cdots \cup S_t, ([N] \setminus S)}) \right] \\ & - \mathop{\mathbf{E}}_{\sigma_{i+1},Y_{i+1},U} \left[f'(Z_{i+1}|_{S_{i+1}}, U|_{S_{i+2} \cup \cdots \cup S_t \cup ([N] \setminus S)}) \right] \big]. \end{aligned} \tag{6.3}$$

Note that, at this point, we can view $\rho_{i+1} = (\sigma_{i+1}, U)$ as a pseudorandom restriction (in the sense of Lemma 6.22) applied to f'. Next, let f'' be the random function defined as the restricted function of f' under ρ_{i+1} (note that the randomness is over W_i, and also the pseudorandom restriction ρ_{i+1}). Now Equation (6.3) becomes

$$\mathop{\mathbf{E}}_{W_i,\rho_{i+1}} \left[\mathop{\mathbf{E}}_{U} [f''(U)] - \mathop{\mathbf{E}}_{Y_{i+1}} [f''(Z_{i+1})] \right]. \tag{6.4}$$

Note that in the above, we abuse the notation and use U and Z_{i+1} to denote $U|_{S_{i+1}}$ and $Z_{i+1}|_{S_{i+1}}$, respectively.

Next we want to show that the difference between the two expectations in Equation (6.4) is at most $3\varepsilon_0 = 3\varepsilon/(10t) \leq \varepsilon/(2t)$, which would give a contradiction, by Equation (6.2). The intuition is the following. On the one hand, f'' is obtained by a pseudorandom restriction ρ_{i+1}, and so, with high probability, it has size at most s_0. On the other hand, Z_{i+1} is obtained

using an extractor that is supposed to extract enough random bits for an s_0-independent generator.

The issue, however, is that f'' depends on X, the source sample of the extractor. Therefore, f'' may contain information about X, so that X is not truly random anymore. Nonetheless, being a formula of size at most s_0, f'' cannot contain too much information, and so cannot take too much entropy away from X. We make this argument more formal next.

Let us define the set of good functions for f'', namely

$$\mathcal{F} = \left\{ g \mid L(g) \leq s_0 \quad \text{and} \quad \Pr_{W_i, \rho_{i+1}}[f'' = g] \geq \varepsilon_0 / s_0^{cs_0} \right\},$$

where c is some constant. Let $\mathcal{E}$ denote the event $f'' \in \mathcal{F}$. We first show the following.

Claim 6.23. *It is the case that* $\mathbf{Pr}[\neg\mathcal{E}] \leq 2\varepsilon_0$.

Proof of Claim 6.23. We have

$$\begin{aligned}\mathbf{Pr}[\neg\mathcal{E}] &= \mathbf{Pr}[(f'' \notin \mathcal{F}) \wedge (L(f'') > s_0)] + \mathbf{Pr}[(f'' \notin \mathcal{F}) \wedge (L(f'') \leq s_0)] \\ &\leq \mathbf{Pr}[(L(f'') > s_0)] + \mathbf{Pr}[(f'' \notin \mathcal{F}) \wedge (L(f'') \leq s_0)].\end{aligned}$$

Note that, by the pseudorandom shrinkage lemma (Lemma 6.22), we have

$$\mathbf{Pr}[L(f'') > s_0] \leq \varepsilon_0;$$

in fact, our choices of s_0 and ε_0 were informed by our intention to make the above inequality hold. Also note that under the condition that $L(f'') \leq s_0$, there can be at most $s_0^{O(s_0)}$ choices for f'', since a formula of size s_0 can be specified using $O(s_0 \log s_0)$ bits (Proposition 6.5). Therefore,

$$\mathbf{Pr}[(f'' \notin \mathcal{F}) \wedge (L(f'') \leq s_0)] \leq s_0^{O(s_0)} \cdot \varepsilon_0 / s_0^{cs_0} \leq \varepsilon_0. \qquad \square$$

Let us now analyze Equation (6.4) while conditioning on the event $\mathcal{E}$. We show the following.

Claim 6.24. *It is the case that* $\mathbf{E}[f''(U) \mid \mathcal{E}] - \mathbf{E}[f''(Z_{i+1}) \mid \mathcal{E}] \leq \varepsilon_0$.

Proof of Claim 6.24. First note that conditioning on $\mathcal{E}$, X still has a large min-entropy. More precisely, for every $g \in \mathcal{F}$ it is the case that

$$H_\infty(X \mid f'' = g) \geq \aleph/2.$$

This is because, for every x, we have

$$\mathbf{Pr}[X = x \mid f'' = g] \leq \frac{\mathbf{Pr}[X = x]}{\mathbf{Pr}[f'' = g]} \leq \frac{2^{-\aleph}}{\varepsilon_0 / s_0^{c \cdot s_0}} = 2^{-(\aleph - \log(1/\varepsilon_0) - c \cdot s_0 \cdot \log s_0)} \leq 2^{-\aleph/2}.$$

Then, by the definition of the extractor, we have

$$\mathbf{E}\left[f''(G_k(U)) \mid \mathcal{E}\right] - \mathbf{E}\left[f''(Z_{i+1}) \mid \mathcal{E}\right] \leq \varepsilon_0.$$

Finally, note that

$$\mathbf{E}\left[f''(G_k(U)) \mid \mathcal{E}\right] = \mathbf{E}\left[f''(U) \mid \mathcal{E}\right],$$

since s_0-wise independent distributions fool size-s_0 formulas. □

Combining Claim 6.23, Claim 6.24, and Lemma 6.10, we get that the quantity in Equation (6.4) is at most $3\varepsilon_0$, which leads to a contradiction. This completes the proof of the correctness.

Locality. To see that the j-th bit of the PRG can be computed using a circuit of size $s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$, we observe the following equivalent construction:

1. Compute the j-th bits of the t pseudorandom selections $(\sigma_1)_j, \ldots, (\sigma_t)_j$.
2. Retrieve Y_q, where q is the smallest integer such that $(\sigma_q)_j$ is a star.
3. Compute $(Z_q)_j = G_k(E(X, Y_q))_j$ as the j-th bit of the PRG.

Note that Step 1 can be done using a circuit of size $t \cdot 2^{O\left(\log^{2/3} s\right)} = s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$, by the pseudorandom shrinkage lemma (Lemma 6.22). Also, Step 2 can be done by first computing q from the sequence $((\sigma_i)_j)_{i\in[t]}$ using a circuit of size $\widetilde{O}(t)$ (Lemma 6.7), and then outputting Y_q from $(Y_i)_{i\in[t]}$ using a circuit of size $t \cdot \mathsf{polylog}(N)$ (Lemma 6.8). Finally, Step 3 can be done by a circuit of size $\widetilde{O}(\aleph)$ using the efficient extractor (Lemma 6.21) and the limited-independence generator (Lemma 6.17). □

6.5 Almost-quadratic lower bounds against arbitrary basis formulas and branching programs

Here, we prove MCSP lower bounds against formulas, over an arbitrary basis, and branching programs. These lower bounds are obtained similarly to those for De Morgan formulas in the previous section. The idea is to construct strongly local PRGs against these models by modifying the PRGs in [IMZ19].

The following pseudorandom shrinkage lemma for formulas over an arbitrary basis as well as branching programs is an analogue of Lemma 6.22.

Lemma 6.25 ([IMZ19, Lemma 4.2 and Lemma 5.3])**.** *There exists a constant $c_0 > 0$ such that the following holds. For any constant $c > c_0$ and any $s \geq N$, let $p = s^{-1/2}$ and F be a formula over any basis (or a branching program) on N variables of size s; then, there exists a*

p-regular pseudorandom selection $\mathcal{D}$ over N variables that is samplable using $r = \mathsf{polylog}(N)$ random bits such that

$$\mathop{\mathbf{Pr}}_{\sigma\sim\mathcal{D},x\sim\{0,1\}^N}\left[L\Big(F_{(\sigma,x)}\Big) \geq 2^{3\cdot\sqrt{c\cdot\log s}}\cdot p\cdot s\right] \leq 2\cdot s^{-c}.$$

Moreover, there exists a circuit of size $2^{O(\log^{2/3} s)}$ such that, given $j \in \{0,1\}^{\log N}$ and a seed $z \in \{0,1\}^r$, the circuit computes the j-th bit of $\mathcal{D}(z)$.

Using the above pseudorandom shrinkage lemma and an argument as in the proof of the strongly local PRG against De Morgan formulas (Lemma 6.15), we get the following local PRGs.

Lemma 6.26. *For any $s \geq n$, there exists a $\Big(N, s, s^{1/2}\cdot 2^{O\left(\sqrt{\log s}\right)}\Big)$-local PRG against size-$s$ formulas over an arbitrary basis (or branching programs).*

The MCSP lower bound in Theorem 6.2 follows from Lemma 6.26 and Theorem 6.13.

6.6 Improved AC^0 lower bounds for MCSP

In this section, we show improved lower bounds for MCSP against constant-depth circuits.

6.6.1 The case of depth $d > 2$

We first show an improved lower bound against depth-d circuits that almost matches the lower bound for PARITY.

Theorem 6.27 (Theorem 6.3, restated)**.** *For every $d > 2$ and every constant $\gamma > 0$, any depth-d AC^0 circuit computing* MCSP *on truth tables of length N must have size $2^{\Omega\left(N^{1/(d+2+\gamma)}\right)}$.*

The above result is proved using the following structural property of small-depth circuits, which says that, for any such circuit, there exists some locally computable restriction that simplifies the circuits to a constant while leaving many variables unrestricted.

Lemma 6.28. *For any size-s depth-d circuit C, there exists a restriction $\rho \in \{0,1,*\}^N$ such that*

- *C_ρ is a constant function,*
- *$\left|\rho^{-1}(*)\right| \geq \frac{N}{O(\log s)^{d-2}} - \log s$, and*
- *there exists a circuit of size $d\cdot\log(N)\cdot\widetilde{O}\left(\log^3 s\right)$ such that, given $j \in \{0,1\}^{\log N}$, the circuit computes the j-th coordinate of ρ.*

We now prove Theorem 6.27 using Lemma 6.28.

Proof of Theorem 6.27. Let C be a depth-d AC^0 circuit on $\{0,1\}^N \times \{0,1\}^{\log N}$ such that C computes MCSP on truth tables of length N, and let s be the size of C.

For a size parameter $\lambda = d \cdot \log(N) \cdot \widetilde{O}\left(\log^3 s\right)$, let $C' = C(\cdot, \lambda)$. Let ρ be a restriction from Lemma 6.28 for C'. By Lemma 6.28, we have that C'_ρ is a constant function. First, note that

$$C'_\rho\left(0^{|\rho^{-1}(*)|}\right) = 1.$$

To see this, note that

$$C'_\rho\left(0^{|\rho^{-1}(*)|}\right) = C\left(\mathtt{tt}(f), \lambda\right),$$

where C computes MCSP and $f\colon \{0,1\}^{\log N} \to \{0,1\}$ is the following:

$$f(j) = \begin{cases} 0, & \text{if } \rho_j = 0 \text{ or } \rho_j = *, \\ 1, & \text{if } \rho_j = 1. \end{cases}$$

By Item 3 of Lemma 6.28, such a function f can be computed by a λ-size circuit. On the other hand, there can be $2^{|\rho^{-1}(*)|}$ different functions corresponding to the different partial assignments to the unrestricted variables. Since there are at most $2^{O(\lambda \log \lambda)}$ different circuits of size at most λ, in order for C'_ρ to be constant and equal to 1, we must have

$$2^{O(\lambda \log \lambda)} \geq 2^{|\rho^{-1}(*)|} = 2^{\frac{N}{O(\log s)^{d-2}} - \log s},$$

which, by a simple calculation, implies $s = 2^{\Omega\left(N^{1/(d+2+\gamma)}\right)}$, for any constant $\gamma > 0$. □

The proof of Lemma 6.28 uses the pseudorandom switching lemma due to Trevisan and Xue [TX13], which we revisit below. The (pseudorandom) switching lemma says that a depth-2 circuit is likely to be simplified after being hit by a (pseudo)random restriction.

Below, when we refer to the size of a DNF or CNF we mean the number of its terms or clauses, respectively.

Lemma 6.29 (Pseudorandom switching lemma, [TX13, Lemma 7])**.** *For any integers $d, t > 0$, $s \geq N$, and any $(480/N)^{1/(d-2)} < p < 1$ and $0 < \varepsilon_0 < 1$, there exists a distribution $\mathcal{D}$ over $\{0,1\}^{N \cdot \log(1/p)} \times \{0,1\}^N$ that ε_0-fools $\left(s_0 = s \cdot 2^{w \cdot (\log(1/p)+1)}\right)$-clause CNFs, then*

$$\Pr_{\rho \sim \mathcal{D}}[F_\rho \textit{ does not have a depth-}t\textit{ decision tree}] \leq 2^{t+w+1} \cdot (5pw)^w + \varepsilon_0 \cdot 2^{(t+1)(2w+\log s)}.$$

Lemma 6.30 (Following [TX13, Theorem 11])**.** *For any integers $d, t > 0$, $s \geq N$, and any $1/n < p < 1$ and $0 < \varepsilon_0 < 1$, there exists a distribution $\mathcal{D}$ over $\{0,1\}^{N \cdot \log(1/p)} \times \{0,1\}^N$ for sampling a pseudorandom restriction such that*

- *for any size-s depth-d circuit C on N variables, we have that*

$$\Pr_{\rho\sim\mathcal{D}}\left[C_\rho \text{ is not a } t\text{-DNF or } t\text{-CNF}\right] \leq s\cdot\left(2^{2t+1}\cdot(10p\log s)^t+\varepsilon_0\cdot 2^{(t+1)(2t+\log s)}\right),$$

- *with probability at least $2/3$ the number of unrestricted variables is $\frac{p^{d-2}}{80}\cdot N$, and*
- *there exists a circuit of size $d\cdot k\cdot\widetilde{O}(\log N)$ such that, given $j\in\{0,1\}^{\log N}$ and a seed $z\in\{0,1\}^{d\cdot k\cdot O(\log N)}$, the circuit computes the j-th coordinate of ρ (as an element in $\{0,1,*\}$), where*

$$k=O\left((\log(s)+t\cdot\log(1/p))^2+(\log(s)+t\cdot\log(1/p))\cdot\log(1/\varepsilon_0)\right).$$

Proof (sketch). The proof is similar to that of Theorem 11 in [TX13]. The idea is to apply the pseudorandom switching lemma (Lemma 6.29) repeatedly. Each time, we sample a pseudorandom restriction using some distribution that ε_0-fools CNFs of size $s_0=s\cdot 2^{w\cdot(\log(1/p)+1)}$, for $w=t$. By Lemma 6.29, each time, with high probability, the two bottom layers can be computed by depth-t decision trees, so we can switch them to t-DNFs or t-CNFs, and hence reduce the depth of the circuit by one as we merge them with the layer above.

One difference here from the argument in [TX13] is that we only apply the pseudorandom switching lemma $d-1$ times, instead of d times, since we only need the final restricted circuit to be a t-DNF or t-CNF (rather than a depth-t decision tree as in the original statement of [TX13], which requires an additional application of the pseudorandom switching lemma). Note that we use parameter $p=1/40$ for the first iteration. Another difference is that, to sample a pseudorandom restriction, we use a k-wise independent distribution (say over $[1/p]^{2N}$), instead of using the PRG against depth-2 circuits in [DETT10], where

$$k=O\left(\log(s_0/\varepsilon_0)\cdot\log s_0\right)=O\left((\log(s)+t\cdot\log(1/p))^2+(\log(s)+t\cdot\log(1/p))\cdot\log(1/\epsilon_0)\right),$$

and we use the fact that such a k-wise independent distribution ε_0-fools s_0-clause CNFs [Tal17b, Theorem 22].[7]

Note that the expected number of unrestricted variables is $\frac{p^{d-2}}{40}\cdot N$. Then Item 2 follows from the fact that the random restriction is pair-wise independent and Proposition 6.9.

Finally, it is easy to get Item 3 using Lemma 6.17. □

We are now ready to show Lemma 6.28.

[7]The PRG in [DETT10] is based on a *small-biased distribution*. While it has smaller seed length, compared to a k-wise independent distribution, it does not seem to offer any advantage in terms of the local circuit complexity of computing the PRG.

Proof of Lemma 6.28. By Lemma 6.30, using the parameters $t = O(\log s)$, $p = 1/O(\log s)$, and $\varepsilon_0 = 1/2^{O(\log^2 s)}$, we get a restriction ρ_0 such that the circuit restricted by ρ_0 is a width-$O(\log s)$ DNF or CNF, with probability at least $1 - 1/\mathsf{poly}(N)$. Note that, by Item 2 of Lemma 6.30, ρ_0 leaves at least $\frac{N}{O(\log s)^{d-2}}$ variables unrestricted, with constant probability. Therefore, by a union bound, with some constant probability, we get a restriction ρ_0 that both simplifies the circuit to be a width-$(\log s)$ DNF or CNF and that leaves $\frac{N}{O(\log s)^{d-2}}$ variables unrestricted. Note that once we have such a restriction, we can make the restricted circuit constant by further fixing at most $\log s$ variables; denote this restriction by ρ_1. The final restriction is $\rho = \rho_0 \circ \rho_1$.

We now show the last item. Note that our final restriction consists of two parts, ρ_0 and ρ_1, where ρ_0 is a restriction from Lemma 6.30 and ρ_1 is a restriction that fixes $\log s$ variables. To compute the final restriction, given an index $j \in \{0,1\}^{\log N}$, we can first check if the j-th variable is fixed by ρ_1 and output the fixing value if it is the case. This can be done by hard-wiring the $\log s$ variables that are fixed by ρ_1 and their corresponding fixing values. It is easy to see that the above can be done using a circuit of size at most $O(\log s \cdot \log N)$. Otherwise, we can output the j-th coordinate of ρ_0, which can be done with a circuit of size $d \cdot \log(N) \cdot \widetilde{O}\left(\log^3 s\right)$, by Item 3 of Lemma 6.30. □

6.6.2 The case of depth 2

Here, we show that computing MCSP requires depth-2 circuits of almost maximum size.[8]

Theorem 6.31 (Theorem 6.4, restated)**.** *Any* CNF *or* DNF *computing* MCSP *on truth tables of length N must have size $2^{\Omega(N)}$.*

To prove Theorem 6.31, we will utilize the following lemma and corollary.

Lemma 6.32. *Let $0 < \delta < 1$. Any N-variate* CNF *or* DNF *of width $s \leq 2^{\delta N}$ can be fixed to a constant by applying a restriction that sets $O\left(\sqrt{\delta}N\right)$ variables.*

Proof. We will show the lemma for the case of DNFs. The proof can be easily adapted to the case of CNFs. Fix a constant $A := 1/\sqrt{\delta}$. We choose the restriction in question in two phases. In Phase 1, we show that we can set at most $O\left(\sqrt{\delta}N\right)$ variables to get the width of the DNF down to $A \log s = \sqrt{\delta}N$. In Phase 2, we can easily set any term of the remaining DNF to fix the function. Since Phase 2 is trivial, let us henceforth focus on Phase 1.

To this end, imagine that we choose a uniformly random input variable x_i (for some i) and set it to a random value. If T is any term in the DNF of size greater than $A \log s$, then T is set to 0 with probability at least

$$\frac{A \log s}{2N}.$$

[8]The results of this subsection exactly follow the guidelines of one of our ToCT reviewers.

Repeating this process $t := 2\sqrt{\delta}N$ times, we see that the probability that T survives is at most

$$\left(1 - \frac{A \log s}{2N}\right)^t = \left(1 - \frac{\log s}{2\sqrt{\delta}N}\right)^{2\sqrt{\delta}N} \leq \exp(-\log s) < \frac{1}{s}.$$

By a union bound over the (at most) s terms of the DNF in question, there is a restriction ρ that restricts $O\left(\sqrt{\delta}N\right)$ variables such that ρ sets all the terms of width greater than $A \log s$ to 0. This completes Phase 1. □

Corollary 6.33. *For any size-s depth-2 circuit C, there exists a restriction $\rho \in \{0,1,*\}^N$ such that*

- *C_ρ is a constant function,*
- *$|\rho^{-1}(*)| \geq \Omega(N)$, and*
- *there exists a circuit of size $\log s / \Omega(\log \log s)$ such that, given $j \in \{0,1\}^{\log N}$, the circuit computes the j-th coordinate of ρ.*

Proof. By Lemma 6.32. We shall verify that all of three properties of Corollary 6.33 hold for the restriction ρ that is proved to exist by Lemma 6.32. Items 1 and 2 trivially hold, as C_ρ is a constant by the construction of ρ and $|\rho^{-1}(\{0,1\})| = O\left(\sqrt{\delta}N\right)$, respectively. Item 3 follows by a result of Lupanov [GII+19, Theorem 2.2] for the (biased) Boolean function that sets all the unrestricted variables to 0. □

We are now able to prove the main result of this subsection (Theorem 6.31) by using Corollary 6.33.

Proof of Theorem 6.31 (sketch). One may prove Theorem 6.31 by using Corollary 6.33 exactly as we did in the proof of Theorem 6.27 with Lemma 6.28. □

6.7 MCSP circuit lower bounds from average-case hard functions

6.7.1 The Nisan-Wigderson generator

It is well known in the field of derandomization that, if we have a function that is average-case hard against some circuit class $\mathfrak{C}$, we can get a PRG for $\mathfrak{C}$ by plugging the hard function into the Nisan-Wigderson framework [NW94] (provided that the hard function is not too hard to compute and that $\mathfrak{C}$ satisfies some mild conditions). The construction involves computing some combinatorial design with some suitably chosen parameters; a design is a list of subsets (over some universe) that have some combinatorial properties (see Definition 6.34). Also, to compute a single bit of such a PRG, we need to compute the corresponding subset of the design. There are known design constructions such that any

single subset of the design can be computed efficiently and locally (without computing the whole design). Therefore, using such a local design, we can get a locally computable PRG which can be used to obtain an MCSP lower bound against $\mathfrak{C}$.

The idea of using Nisan-Wigderson PRGs to study MCSP and related problems has been explored before (e.g. [ABK$^+$06, OS17, Hir18]). However, the previous works were content with the fact that the output of a PRG has circuit complexity at most polynomial in the seed length. Here, we provide a more fine-grained analysis of the local complexity of the Nisan-Wigderson PRG, which depends on the parameters that we choose for the design, and in turn will depend on the "usefulness" of the average-case hard function. This allows us to turn average-case hardness against some circuit class $\mathfrak{C}$ into a lower bound for MCSP against the same class, where such a lower bound is more quantitatively linked to the average-case hardness.

We first review the Nisan-Wigderson framework.

Definition 6.34 (Designs [NW94])**.** *Let N, r, ℓ, α be positive integers. A family of sets $S_1, \ldots, S_N$ is a* (N, r, ℓ, α)-design *if*

- $\forall j \in [N] : S_j \subseteq [r]$,
- $\forall j \in [N] : |S_j| = \ell$, *and*
- $\forall j, k \in [s]$, *such that $j \neq k$, it is the case that $|S_j \cap S_k| \leq \alpha$.*

Lemma 6.35 (Local designs)**.** *For any positive integers N and α, there exists a (N, r, ℓ, α)-design such that $r = N^{2/(\alpha+1)}$ and $\ell = N^{1/(\alpha+1)}$. Moreover, given any $z \in \{0,1\}^r$, and any $j \in \{0,1\}^{\log N}$, $z|_{S_j} \in \{0,1\}^\ell$ can be computed by a circuit of size*

$$O\Big(N^{2/(\alpha+1)}\Big) + N^{1/(\alpha+1)} \cdot \widetilde{O}(\log N).$$

Proof. Consider the field $\mathbb{F}_\ell$ with ℓ elements. We identify the universe $[r]$ with $\mathbb{F}_\ell \times \mathbb{F}_\ell$ of size ℓ^2. Let $\{e_1, \ldots, e_\ell\}$ be the ℓ elements of the field (in lexicographic order). For each $j \in \{0,1\}^{\log N}$, we view j as an element in $[\ell]^{\alpha+1}$ and identify it with a degree-α polynomial $p_j \in \mathbb{F}_\ell[x]$. Let

$$S_j = \{(e_1, p_j(e_1)), \ldots, (e_\ell, p_j(e_\ell))\}.$$

Note that, for all j, the set S_j is a subset of $\mathbb{F}_\ell \times \mathbb{F}_\ell$, the set S_j has size ℓ, and for two different sets S_j and S_k we have that $|S_j \cap S_k| \leq \alpha$, as the difference $p_j - p_k$ is a polynomial of degree at most α, and thus has at most α roots.

Note that we can hard-wire $(e_k, e_k^2, \ldots, e_k^\alpha)$ into some circuit, for all $k \in [\ell]$, by using size $\ell \cdot \alpha \cdot \log \ell = \widetilde{O}(\ell)$. Then computing $p_j(e_k)$, for any k, can be done with a circuit of size $\alpha \cdot \widetilde{O}(\log \ell)$ (using Fact 6.16). As a result, S_j can be computed in size

$$\ell \cdot \alpha \cdot \widetilde{O}(\log \ell) = N^{1/(\alpha+1)} \cdot \widetilde{O}(\log N).$$

Once we have the set S_j, we can divide the input z into ℓ equal-size blocks. For each element (a, b) in S_j, we output the b-th bit of the a-th block, using Lemma 6.8, in $O(\ell)$ size. Then, computing $z|_{S_j}$ takes size $\ell \cdot O(\ell) = O\left(N^{2/(\alpha+1)}\right)$. □

Definition 6.36 (Average-case hardness)**.** *Let $\mathfrak{C}$ be a class of circuits on N variables. We say that a function f is* (s, ε)-hard against $\mathfrak{C}$ *if, for every $C \in \mathfrak{C}$ of size s, it is the case that*

$$\Pr_{x \sim \{0,1\}^N}[f(x) = C(x)] \leq \frac{1}{2} + \varepsilon.$$

Let DNF_α denote the class of DNF circuits on α variables. Note that every α-variate Boolean function can be computed by a DNF of size at most 2^α.

Theorem 6.37 (Nisan-Wigderson generator [NW94])**.** *Let $\mathfrak{C}$ be a class of circuits on N variables of size s. Let $S_1, \ldots, S_N$ be a (N, r, ℓ, α)-design, and let $f : \{0,1\}^\ell \to \{0,1\}$ be a function that is $(s + N \cdot 2^\alpha, \varepsilon/N)$-hard against $\mathfrak{C} \circ \mathsf{DNF}_\alpha$. Then, the Nisan-Wigderson generator $\mathrm{NW}^f : \{0,1\}^r \to \{0,1\}^N$, defined as*

$$\mathrm{NW}^f(z) = \left(f\left(z|_{S_1}\right), \ldots, f\left(z|_{S_N}\right)\right),$$

is a PRG that ε-fools $\mathfrak{C}$.

Combining Theorem 6.37 with the design construction in Lemma 6.35, we immediately get the following.

Theorem 6.38 (Local Nisan-Wigderson generator)**.** *Let $\mathfrak{C}$ be a class of circuits on N variables of size s. For any $\alpha = \alpha(N, s)$, if there exists a function $f\colon \{0,1\}^\ell \to \{0,1\}$, where $\ell = N^{1/(\alpha+1)}$, that is $(s + N \cdot 2^\alpha, 1/(3N))$-hard against $\mathfrak{C} \circ \mathsf{DNF}_\alpha$, then there exists a $(N, s, \lambda(N, s))$-local PRG against $\mathfrak{C}$, with*

$$\lambda(N, s) = O\left(N^{2/(\alpha+1)}\right) + N^{1/(\alpha+1)} \cdot \widetilde{O}(\log N) + \mathtt{CC}(f).$$

We remark that the above local Nisan-Wigderson generator has local complexity that is comparable to its seed length (for this particular local design and modulo the circuit complexity of the hard function).

6.7.2 Applications

Next we demonstrate the use of such local PRGs in obtaining lower bounds for MCSP from average-case hardness results.

One of the restricted circuit classes that have been well studied in circuit complexity is the class of constant-depth circuits augmented with few SYM (symmetric) or THR (linear threshold) gates (see, e.g., [LVW93, Vio07, LS11, ST18]). A SYM gate computes a symmetric function, which is a Boolean function whose output depends only on the sum of its input

variables. A THR gate computes a linear threshold function, which is a Boolean function defined as the sign of some linear form, over Boolean variables, with real coefficients. We will combine the above local Nisan-Wigderson framework with the following average-case lower bounds against the class of constant-depth circuits augmented with a few (sublinearly many) symmetric and linear threshold gates.

Theorem 6.39 ([ST18, Theorem 4])**.** *There exists a constant $\tau > 0$ such that the following hold. For any ℓ, there exists a function $f\colon \{0,1\}^{\ell} \to \{0,1\}$ that is $\left(\ell^{\tau \log \ell}, \exp\left(-\Omega(\ell^{0.499})\right)\right)$-hard against AC^0 circuits of size $\ell^{\tau \log \ell}$ with at most $\ell^{0.249}$ SYM or THR gates. Moreover f can be computed by a circuit of size $O(\ell)$.*

As a result, we get a local PRG against such circuits.

Corollary 6.40. *There exists some constant $\tau > 0$ such that, for any $s \geq N$, there exists a $(N, s, \lambda(N,s))$-local PRG against AC^0 circuits of size $s = \ell^{\tau \log \ell}$, for some $\ell > 0$, with at most $\ell^{0.249}$ SYM or THR gates and $\lambda(N,s) = 2^{O(\sqrt{\log s})}$.*

Proof. Let $\mathfrak{C}$ be the class of AC^0 circuits of size $\ell^{\tau \log \ell}$, for some constant $\tau > 0$, with at most $\ell^{0.249}$ SYM or THR gates. Choose

$$\alpha = \tau' \cdot \frac{\log N}{\sqrt{\log s}},$$

where $\tau' > 0$ is some sufficiently small constant. Then, for $\ell = N^{1/(\alpha+1)}$, if we can show the existence of some efficiently computable function $f\colon \{0,1\}^{\ell} \to \{0,1\}$ that is $(s + N \cdot 2^{\alpha}, 1/(3N))$-hard against $\mathfrak{C} \circ \mathsf{DNF}_{\alpha}$, then the result follows from Theorem 6.38. The existence of such a function is given by Theorem 6.39, by noting that for our choice of α we have

$$\ell^{\tau \log \ell} \geq s + N \cdot 2^{\alpha},$$

and

$$\exp\left(-\Omega(\ell^{0.249})\right) \leq 1/(3N). \qquad \square$$

We remark that the above example does not take advantage of the fact that the local complexity of the Nisan-Wigderson PRG is almost the same as its seed length. This is because, in this case, the seed length has some arbitrary constant in the exponent.

Combining Corollary 6.40 with Theorem 6.13, we get the following.

Theorem 6.41. *There exists a constant $\gamma > 0$ such that the following hold. Let $\mathfrak{C}$ be the class of constant-depth AC^0 circuits augmented with at most $2^{\gamma\sqrt{\log N}}$ SYM or THR gates. Then, any circuit in $\mathfrak{C}$ computing* MCSP *on truth tables of length N must have size $N^{\Omega(\log N)}$.*

As another application of our framework, combined with the Nisan-Wigderson generator, we show that separating $\mathsf{P/poly}$ (non-uniform circuits of polynomial size) from some

restricted circuit class, such as TC^0 (non-uniform constant-depth polynomial-size circuits with threshold gates) or NC^1 (non-uniform polynomial-size logarithmic-depth circuits), implies MCSP lower bounds against the same class of circuits. More precisely, we show that if there exists some function in $\mathsf{P/poly}$ that is mildly hard against TC^0 (resp. NC^1), then MCSP cannot be computed by TC^0 (resp. NC^1) circuits.

Theorem 6.42. *If there exists a function in* $\mathsf{P/poly}$ *that requires size-*s TC^0 *(resp.* NC^1*) circuits to compute within error* $1/\mathsf{poly}(n)$*, for some superpolynomial size function* s*, then* MCSP *requires superpolynomial size* TC^0 *(resp.* NC^1*) circuits.*

Proof (sketch). Let $s(n) = n^{\omega(1)}$ and let $f = \{f_n\}_n$, with $f_n \colon \{0,1\}^n \to \{0,1\}$, be a function that requires size-$s(n)$ TC^0 circuits to compute with error at most $1/\mathsf{poly}(n)$. Using standard hardness amplification tools, such as the direct product theorem and the XOR lemma (see, e.g., [CIKK16, Section 4]), we can amplify f to a strongly hard on average function within $\mathsf{P/poly}$. By plugging f into the Nisan-Wigderson construction (Theorem 6.37) we get a local PRG against TC^0; this implies that MCSP $\notin \mathsf{TC}^0$ by Theorem 6.13. □

6.8 Open problems

Our De Morgan formula lower bound for MCSP is still slightly weaker than the state-of-the-art De Morgan formula lower bound due to Tal [Tal17a], which is $\Omega(N^3/(\log N \cdot (\log\log N)^2))$. Can the MCSP lower bound be improved? Are there better constructions of local PRGs against formulas? Or, are there alternative proofs that do not rely on local PRGs?

What are other restricted models of computation against which we can show MCSP lower bounds using local PRGs? The recent "random walk PRG" by Chattopadhyay, Hatami, Hosseini, and Lovett [CHHL18] is also local and can be used to get MCSP lower bounds. However, as a general PRG that can be used to fool a variety of restricted models, it has sub-optimal usefulness (which is determined by its seed length) compared to the best-known lower bounds for most of those models.

Bibliography

[AB18] Amir Abboud and Karl Bringmann. Tighter connections between formula-SAT and shaving logs. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming* (ICALP), pages 8:1–8:18, 2018.

[ABK+06] Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006.

[ACR+10] Andris Ambainis, Andrew M. Childs, Ben Reichardt, Robert Spalek, and Shengyu Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM Journal on Computing*, 39(6):2513–2530, 2010.

[ACW16] Josh Alman, Timothy M. Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *Proceedings of the 57th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), 2016.

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.

[Ajt83] Miklós Ajtai. Σ^1_1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983.

[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57(3), 2010.

[All89] Eric Allender. A note on the power of threshold circuits. In *Proceedings of the 30th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 580–584, 1989.

[And87] Alexander E. Andreev. On a method of obtaining more than quadratic effective lower bounds for the complexity of π-schemes. *Vestnik Moskovskogo Universiteta. Matematika*, 42(1):70–73, 1987. English translation in *Moscow University Mathematics Bulletin*.

[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.

[ASWZ96] Roy Armoni, Michael E. Saks, Avi Wigderson, and Shiyu Zhou. Discrepancy sets and pseudorandom generators for combinatorial rectangles. In *Proceedings of the 37th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 412–421, 1996.

[AW89] Miklós Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant depth circuits. *Advances in Computing Research*, 5:199–222, 1989.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001.

[BGH07] Debajyoti Bera, Frederic Green, and Steven Homer. Small depth quantum circuits. *SIGACT News*, 38(2):35–50, June 2007.

[BGS75] Theodore P. Baker, John Gill, and Robert Solovay. Relativizations of the P =? NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: New simple PRF candidates and their applications. In *Proceedings of the 16th International Conference on Theory of Cryptography* (TCC), pages 699–729, 2018.

[BKK+19] Swapnam Bajpai, Vaibhav Krishan, Deepanshu Kush, Nutan Limaye, and Srikanth Srinivasan. A #SAT algorithm for small constant-depth circuits with PTF gates. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference* (ITCS), pages 8:1–8:20, 2019.

[BNRdW07] Harry Buhrman, Ilan Newman, Hein Röhrig, and Ronald de Wolf. Robust polynomials and quantum algorithms. *Theory of Computing Systems*, 40(4):379–395, 2007.

[BNS92] László Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992.

[Bon70] Aline Bonami. Étude des coefficients Fourier des fonctions de $L^p(G)$. *Annales de l'Institut Fourier*, 20(2):335–402, 1970.

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

[CHHL18] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. In *Proceedings of the 33rd Computational Complexity Conference* (CCC), pages 1:1–1:21, 2018.

[Cho61] Chao-Kong Chow. On the characterization of threshold functions. In *Proceedings of the 2nd IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 34—38, 1961.

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *Proceedings of the 31st Computational Complexity Conference* (CCC), pages 10:1–10:24, 2016.

[CJW19] Lijie Chen, Ce Jin, and Ryan Williams. Hardness magnification for all sparse NP languages. In *Proceedings of the 60th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), 2019.

[CKK+15] Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining circuit lower bound proofs for meta-algorithms. *Computational Complexity*, 24(2):333–392, 2015.

[CKLM20] Mahdi Cheraghchi, Valentine Kabanets, Zhenjian Lu, and Dimitrios Myrisiotis. Circuit lower bounds for MCSP from local pseudorandom generators. *ACM Trans. Comput. Theory*, 12(3):21:1–21:27, 2020.

[CM01] Mary Cryan and Peter Bro Miltersen. On pseudorandom generators in NC^0. In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 272–284, 2001.

[Cop82] Don Coppersmith. Rapid multiplication of rectangular matrices. *SIAM Journal on Computing*, 11(3):467–471, 1982.

[CS15a] Ruiwen Chen and Rahul Santhanam. Improved algorithms for sparse MAX-SAT and MAX-k-CSP. In *Proceedings of the 18th International Conference on Theory and Applications of Satisfiability Testing* (SAT), pages 33–45, 2015.

[CS15b] Ruiwen Chen and Rahul Santhanam. Improved algorithms for sparse MAX-SAT and MAX-k-CSP. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:112, 2015.

[CSS18] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-case lower bounds and satisfiability algorithms for small threshold circuits. *Theory of Computing*, 14(1):1–55, 2018.

[CW01] Anthony Carbery and James Wright. Distributional and L^q norm inequalities for polynomials over convex bodies in R^n. *Mathematical Research Letters*, 8(3):233–248, 5 2001.

[CW16] Timothy M. Chan and Ryan Williams. Deterministic APSP, orthogonal vectors, and more: Quickly derandomizing Razborov-Smolensky. In *Proceedings of the 27nd Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), pages 1246–1255, 2016.

[CW19] Lijie Chen and Ruosong Wang. Classical algorithms from quantum and Arthur-Merlin communication protocols. In *Proceedings of the 10th Innovations in Theoretical Computer Science Conference* (ITCS), pages 23:1–23:20, 2019.

[DDS14] Anindya De, Ilias Diakonikolas, and Rocco A. Servedio. Deterministic approximate counting for juntas of degree-2 polynomial threshold functions. In *Proceedings of the 29th Computational Complexity Conference* (CCC), pages 229–240, 2014.

[DETT10] Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. In *Proceedings of the 14th International Workshop on Randomization and Computation* (RANDOM), pages 504–517, 2010.

[DGJ+10] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco A. Servedio, and Emanuele Viola. Bounded independence fools halfspaces. *SIAM Journal on Computing*, 39(8):3441–3462, 2010.

[DH09] Evgeny Dantsin and Edward A Hirsch. Worst-case upper bounds. *Handbook of Satisfiability*, 185:403–424, 2009.

[DKN10] Ilias Diakonikolas, Daniel M. Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. In *Proceedings of the 51th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 11–20, 2010.

[DM18] Irit Dinur and Or Meir. Toward the KRW composition conjecture: Cubic formula lower bounds via communication complexity. *Computational Complexity*, 27(3):375–462, 2018.

[DOSW11] Ilias Diakonikolas, Ryan O'Donnell, Rocco A. Servedio, and Yi Wu. Hardness results for agnostically learning low-degree polynomial threshold functions. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), pages 1590–1606, 2011.

[DRST14] Ilias Diakonikolas, Prasad Raghavendra, Rocco A. Servedio, and Li-Yang Tan. Average sensitivity and noise sensitivity of polynomial threshold functions. *SIAM Journal on Computing*, 43(1):231–253, 2014.

[DS14] Anindya De and Rocco A. Servedio. Efficient deterministic approximate counting for low-degree polynomial threshold functions. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing* (STOC), pages 832–841, 2014.

[DSTW14] Ilias Diakonikolas, Rocco A. Servedio, Li-Yang Tan, and Andrew Wan. A regularity lemma and low-weight approximators for low-degree polynomial threshold functions. *Theory of Computing*, 10:27–53, 2014.

[Erd45] Paul Erdős. On a lemma of Littlewood and Offord. *Bulletin of the American Mathematical Society*, 51:898—-902, 1945.

[FGG08] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum algorithm for the hamiltonian NAND tree. *Theory of Computing*, 4(1):169–190, 2008.

[FK09] Lance Fortnow and Adam R. Klivans. Efficient learning algorithms yield circuit lower bounds. *Journal of Computer and System Sciences*, 75(1):27–36, 2009.

[Fre90] Yoav Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the 3rd Annual Conference on Learning Theory* (COLT), pages 202–216, 1990.

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

[FSUV13] Bill Fefferman, Ronen Shaltiel, Christopher Umans, and Emanuele Viola. On beating the hybrid argument. *Theory of Computing*, 9:809–843, 2013.

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

[GII+19] Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. $AC^0[p]$ lower bounds against MCSP via the coin problem. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming* (ICALP), pages 66:1–66:15, 2019.

[GL94] Craig Gotsman and Nathan Linial. Spectral properties of threshold functions. *Combinatorica*, 14(1):35–50, 1994.

[GOWZ10] Parikshit Gopalan, Ryan O'Donnell, Yi Wu, and David Zuckerman. Fooling functions of halfspaces under product distributions. In *Proceedings of the 25th Computational Complexity Conference* (CCC), pages 223–234, 2010.

[GPW18] Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *Computational Complexity*, 27(2):245–304, 2018.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (STOC), pages 212–219, 1996.

[GS13] Sergey B. Gashkov and Igor S. Sergeev. Complexity of computation in finite fields. *Journal of Mathematical Sciences*, 191(5):661–685, 2013.

[GW14] Oded Goldreich and Avi Wigderson. On derandomizing algorithms that err extremely rarely. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing* (STOC), pages 109–118, 2014.

[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.

[Hås98] Johan Håstad. The shrinkage exponent of de Morgan formulas is 2. *SIAM Journal on Computing*, 27(1):48–64, 1998.

[Hås16] Johan Håstad. An average-case depth hierarchy theorem for higher depth. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:41, 2016.

[Hir18] Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 247–258, 2018.

[HLS07] Peter Høyer, Troy Lee, and Robert Spalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM Symposium on the Theory of Computing* (STOC), pages 526–535, 2007.

[HMP+93] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2):129–154, 1993.

[HS17] Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its variants. In *Proceedings of the 32nd Computational Complexity Conference* (CCC), pages 7:1–7:20, 2017.

[HS19] Prahladh Harsha and Srikanth Srinivasan. On polynomial approximations to AC^0. *Random Structures and Algorithms*, 54(2):289–303, 2019.

[IMZ12a] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. In *Proceedings of the 53rd IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 111–119, 2012.

[IMZ12b] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:57, 2012.

[IMZ19] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. *Journal of the ACM*, 66(2):11:1–11:16, 2019.

[IN93] Russell Impagliazzo and Noam Nisan. The effect of random restrictions on formula size. *Random Structures and Algorithms*, 4(2):121–134, 1993.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing* (STOC), pages 356–364, 1994.

[IPS97] Russell Impagliazzo, Ramamohan Paturi, and Michael E. Saks. Size-depth tradeoffs for threshold circuits. *SIAM Journal on Computing*, 26(3):693–707, 1997.

[IPS13] Russell Impagliazzo, Ramamohan Paturi, and Stefan Schneider. A satisfiability algorithm for sparse depth two threshold circuits. In *Proceedings of the 54th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 479–488, 2013.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing* (STOC), pages 220–229, 1997.

[Juk12] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

[Kab02] Valentine Kabanets. Derandomization: A brief overview. *Current Trends in Theoretical Computer Science*, 1:165–188, 2002.

[Kan11] Daniel M. Kane. A small PRG for polynomial threshold functions of Gaussians. In *Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 257–266, 2011.

[Kan12] Daniel M. Kane. A structure theorem for poorly anticoncentrated Gaussian chaoses and applications to the study of polynomial threshold functions. In *Proceedings of the 53rd IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 91–100, 2012.

[Kan14] Daniel M. Kane. The correct exponent for the Gotsman-Linial conjecture. *Computational Complexity*, 23(2):151–175, 2014.

[Kan15] Daniel M. Kane. A polylogarithmic PRG for degree 2 threshold functions in the Gaussian setting. In *Proceedings of the 30th Computational Complexity Conference* (CCC), pages 567–581, 2015.

[KC00] Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Proceedings of the 32nd Annual ACM Symposium on the Theory of Computing* (STOC), pages 73–79, 2000.

[Khr71] V.M. Khrapchenko. A method of determining lower bounds for the complexity of π-schemes. *Matematicheskie Zametki*, 10(1):83–92, 1971. English translation in *Mathematical Notes of the Academy of Sciences of the USSR.*

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[KKL17] Valentine Kabanets, Daniel M. Kane, and Zhenjian Lu. A polynomial restriction lemma with applications. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing* (STOC), pages 615–628, 2017.

[KKL$^+$20] Valentine Kabanets, Sajin Koroth, Zhenjian Lu, Dimitrios Myrisiotis, and Igor Oliveira. Algorithms and lower bounds for De Morgan formulas of low-communication leaf gates. In Shubhangi Saraf, editor, *Proceedings of the 35th Computational Complexity Conference* (CCC), pages 15:1–15:41, 2020.

[KKMS08] Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[KKO13] Adam R. Klivans, Pravesh Kothari, and Igor Carboni Oliveira. Constructing hard functions using learning algorithms. In *Proceedings of the 28th Computational Complexity Conference* (CCC), pages 86–97, 2013.

[KL18] Valentine Kabanets and Zhenjian Lu. Satisfiability and derandomization for small polynomial threshold circuits. In *Proceedings of the 22nd International Workshop on Randomization and Computation* (RANDOM), pages 46:1–46:19, 2018.

[KMV08] Adam Tauman Kalai, Yishay Mansour, and Elad Verbin. On agnostic boosting and parity learning. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing* (STOC), pages 629–638, 2008.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[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 *Proceedings of the 33rd Computational Complexity Conference* (CCC), 2018.

[KV94] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[KW16] Daniel M. Kane and Ryan Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *Proceedings of the 48th Annual ACM Symposium on the Theory of Computing* (STOC), 2016.

[LLS06] Sophie Laplante, Troy Lee, and Mario Szegedy. The quantum adversary method and classical formula size lower bounds. *Computational Complexity*, 15(2):163–196, 2006.

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[LO43] John E. Littlewood and A. Cyril Offord. On the number of real roots of a random algebraic equation (III). *Rec. Math. (Mat. Sbornik) N.S.*, 12 (54)(3):277—-286, 1943.

[LS11] Shachar Lovett and Srikanth Srinivasan. Correlation bounds for poly-size AC^0 circuits with $n^{1-o(1)}$ symmetric gates. In *Proceedings of the 15th International Workshop on Randomization and Computation* (RANDOM), pages 640–651, 2011.

[LVW93] Michael Luby, Boban Velickovic, and Avi Wigderson. Deterministic approximate counting of depth-2 circuits. In *Proceedings of the 2nd Israel Symposium on Theory of Computing Systems* (ISTCS), pages 18–24, 1993.

[MK61] J. Myhill and W. H. Kautz. On the size of weights required for linear-input switching functions. *IRE Transactions on Electronic Computers*, EC-10(2):288–290, June 1961.

[MNV16] Raghu Meka, Oanh Nguyen, and Van Vu. Anti-concentration for polynomials of independent random variables. *Theory of Computing*, 12(11):1–17, 2016.

[MOO10] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. *Annals of Mathematics*, 171(1):295—-341, 2010.

[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.

[MP69] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, Mass., 1969. (3rd Edition published in 1988).

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

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

[Nec66] E.I. Nechiporuk. On a Boolean function. *Doklady Akademii Nauk SSSR*, 169(4):765–766, 1966. English translation in Soviet Mathematics Doklady.

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

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[OPS19] Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. In *Proceedings of the 34th Computational Complexity Conference* (CCC), pages 27:1–27:29, 2019.

[OS08] Ryan O'Donnell and Rocco A. Servedio. Extremal properties of polynomial threshold functions. *Journal of Computer and System Sciences*, 74(3):298–312, 2008.

[OS17] Igor Carboni Oliveira and Rahul Santhanam. Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness. In *Proceedings of the 32nd Computational Complexity Conference* (CCC), pages 18:1–18:49, 2017.

[OS18] Igor Carboni Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In *Proceedings of the 59th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 65–76, 2018.

[OST19] Ryan O'Donnell, Rocco A. Servedio, and Li-Yang Tan. Fooling polytopes. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing* (STOC), pages 614–625, 2019.

[Per04] Yuval Peres. Noise Stability of Weighted Majority. *arXiv.math/0412377*, 2004.

[PRS88] Pavel Pudlák, Vojtech Rödl, and Petr Savický. Graph complexity. *Acta Informatica*, 25(5):515–535, 1988.

[PS94] Ramamohan Paturi and Michael E. Saks. Approximating threshold circuits by rational functions. *Information and Computation*, 112(2):257–272, 1994.

[PW10] Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), pages 1065–1075, 2010.

[PZ93] Mike Paterson and Uri Zwick. Shrinkage of de Morgan formulae under restriction. *Random Structures and Algorithms*, 4(2):135–150, 1993.

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

[Raz92] Alexander A. Razborov. On small depth threshold circuits. In Otto Nurmi and Esko Ukkonen, editors, *Algorithm Theory — SWAT '92: Third Scandinavian Workshop on Algorithm Theory Helsinki, Finland, July 8–10, 1992 Proceedings*, pages 42–52, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

[Rei09] Ben Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proceedings of the 50th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 544–551, 2009.

[Rei11a] Ben Reichardt. Faster quantum algorithm for evaluating game trees. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), pages 546–559, 2011.

[Rei11b] Ben Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA), pages 560–569, 2011.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[RS12] Ben Reichardt and Robert Spalek. Span-program-based quantum algorithm for evaluating formulas. *Theory of Computing*, 8(1):291–319, 2012.

[RSO94] Vwani Roychowdhury, Kai-Yeung Siu, and Alon Orlitsky. *Theoretical Advances in Neural Computation and Learning*, chapter Neural Models and Spectral Methods, pages 3–36. Springer US, Boston, MA, 1994.

[RST15] Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. In *Proceedings of the 56th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 1030–1048, 2015.

[RT19] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing* (STOC), pages 13–23, 2019.

[Sak93] Michael E. Saks. Slicing the hypercube. In K. Walker, editor, *Surveys in Combinatorics, 1993*, pages 211–256. Cambridge University Press, 1993.

[SB91] Kai-Yeung Siu and Jehoshua Bruck. On the power of threshold circuits with small weights. *SIAM Journal on Discrete Mathematics*, 4(3):423–435, 1991.

[Ser17] Igor S Sergeev. Upper bounds for the size and the depth of formulae for MOD-functions. *Discrete Mathematics and Applications*, 27(1):15–22, 2017.

[Sha49] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell Systems Technical Journal*, 28:59–98, 1949.

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

[Spi71] Philip M. Spira. Theory and applications of trapdoor functions. In *Proceedings of the 4th Hawaii International Symposium on SystemSciences*, pages 525–527, 1971.

[Sri13] Srikanth Srinivasan. On improved degree lower bounds for polynomial approximation. In *Proceedings of the 33rd Annual Conference on Foundations of Software Technology and Theoretical Computer Science* (FSTTCS), pages 201–212, 2013.

[SRK94] Kai-Yeung Siu, Vwani P. Roychowdhury, and Thomas Kailath. Rational approximation techniques for analysis of neural networks. *IEEE Transactions on Information Theory*, 40(2):455–466, 1994.

[SSS95] Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-Hoeffding bounds for applications with limited independence. *SIAM Journal on Discrete Mathematics*, 8(2):223–250, 1995.

[SSTT16] Takayuki Sakai, Kazuhisa Seto, Suguru Tamaki, and Junichi Teruyama. Bounded depth circuits with weighted symmetric gates: Satisfiability, lower bounds and compression. In *Proceedings of the 41st International Symposium on. Mathematical Foundations of Computer Science* (MFCS), pages 82:1–82:16, 2016.

[ST17] Rocco A. Servedio and Li-Yang Tan. What circuit classes can be learned with non-trivial savings? In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference* (ITCS), pages 30:1–30:21, 2017.

[ST18] Rocco A. Servedio and Li-Yang Tan. Luby-Velickovic-Wigderson revisited: Improved correlation bounds and pseudorandom generators for depth-two circuits. In *Proceedings of the 22nd International Workshop on Randomization and Computation* (RANDOM), pages 56:1–56:20, 2018.

[ST19] Rocco A. Servedio and Li-Yang Tan. Improved pseudorandom generators from pseudorandom multi-switching lemmas. In *Proceedings of the 23rd International Workshop on Randomization and Computation* (RANDOM), pages 45:1–45:23, 2019.

[STT12] Rocco A. Servedio, Li-Yang Tan, and Justin Thaler. Attribute-efficient learning and weight-degree tradeoffs for polynomial threshold functions. In *Proceedings of the 25th Annual Conference on Learning Theory* (COLT), pages 14.1–14.19, 2012.

[Sub61] Bella A. Subbotovskaya. Realizations of linear function by formulas using $\vee$, $\&$, $^-$. *Doklady Akademii Nauk SSSR*, 136(3):553–555, 1961. English translation in *Soviet Mathematics Doklady*.

[Tal14] Avishay Tal. Shrinkage of De Morgan formulae by spectral techniques. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 551–560, 2014.

[Tal15] Avishay Tal. #SAT algorithms from shrinkage. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:114, 2015.

[Tal16] Avishay Tal. The bipartite formula complexity of inner-product is quadratic. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:181, 2016.

[Tal17a] Avishay Tal. Formula lower bounds via the quantum method. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing* (STOC), pages 1256–1268, 2017.

[Tal17b] Avishay Tal. Tight bounds on the Fourier spectrum of AC^0. In *Proceedings of the 32nd Computational Complexity Conference* (CCC), pages 15:1–15:31, 2017.

[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.

[Tel17a] Roei Tell. Improved bounds for quantified derandomization of constant-depth circuits and polynomials. In *Proceedings of the 32nd Computational Complexity Conference* (CCC), pages 13:1–13:48, 2017.

[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.

[Tel18] Roei Tell. Quantified derandomization of linear threshold circuits. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing* (STOC), pages 855–865, 2018.

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

[Tra84] Boris A. Trakhtenbrot. A survey of Russian approaches to perebor (brute-force searches) algorithms. *IEEE Annals of the History of Computing*, 6(4):384–400, 1984.

[TX13] Luca Trevisan and Tongke Xue. A derandomized switching lemma and an improved derandomization of AC^0. In *Proceedings of the 27th Computational Complexity Conference* (CCC), pages 242–247, 2013.

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

[Val84] Leslie G. Valiant. A theory of the learnable. In *Proceedings of the 16th Annual ACM Symposium on the Theory of Computing* (STOC), pages 436–445, 1984.

[Vio07] Emanuele Viola. Pseudorandom bits for constant-depth circuits with few arbitrary symmetric gates. *SIAM Journal on Computing*, 36(5):1387–1403, 2007.

[Vio15] Emanuele Viola. The communication complexity of addition. *Combinatorica*, 35(6):703–747, 2015.

[vzGG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2013.

[Weg87] Ingo Wegener. *The complexity of Boolean functions*. Wiley-Teubner, 1987.

[Wil13] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal on Computing*, 42(3):1218–1244, 2013.

[Wil14a] Ryan Williams. New algorithms and lower bounds for circuits with linear threshold gates. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing* (STOC), pages 194–202, 2014.

[Wil14b] Ryan Williams. Nonuniform ACC circuit lower bounds. *Journal of the ACM*, 61(1):2:1–2:32, 2014.

[Win61] Robert O. Winder. Single stage threshold logic. In *Proceedings of the 2nd IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 321–332, Oct 1961.

[WKST19] Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing* (STOC), pages 515–526, 2019.

[Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23rd IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 80–91, 1982.

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

[Yao90] Andrew Chi-Chih Yao. On ACC and threshold circuits. In *Proceedings of the 31st IEEE Annual Symposium on Foundations of Computer Science* (FOCS), pages 619–627, 1990.

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

Appendix A

Omitted proofs for Chapter 5

A.1 Useful lemmas for formulas

The proofs in this section are essentially the same as that of [Tal16].

Lemma A.1 ([Tal16], Lemma 5.18 restated)**.** *Let $\mathcal{D}$ be a distribution over $\{-1,1\}^n$, and let $f, C\colon \{-1,1\}^n \to \{-1,1\}$ be such that*

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[C(x) = f(x)] \geq 1/2 + \varepsilon.$$

Let $\tilde{C}\colon \{-1,1\}^n \to \mathbb{R}$ be a ε-approximating function of C, i.e., for every $x \in \{-1,1\}^n$, $|C(x) - \tilde{C}(x)| \leq \varepsilon$. Then,

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\tilde{C}(x)\cdot f(x)] \geq \varepsilon.$$

Proof. Note that since $\tilde{C}$ ε-approximate C, we have for every $x \in \{-1,1\}^n$

$$\tilde{C}\cdot C(x) \geq 1-\varepsilon,$$

and

$$\tilde{C}\cdot (1-C(x)) \geq -1-\varepsilon.$$

Then,

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\tilde{C}(x)\cdot f(x)] &= \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\tilde{C}(x)\cdot f(x) \mid C(x)=f(x)]\cdot \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[C(x)=f(x)] \\
&\quad + \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\tilde{C}(x)\cdot f(x) \mid C(x)\neq f(x)]\cdot \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[C(x)\neq f(x)] \\
&\geq (1-\varepsilon)\cdot \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[C(x)=f(x)] + (-1-\varepsilon)\cdot\left(1-\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[C(x)=f(x)]\right) \\
&= 2\cdot \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[C(x)=f(x)] - 1 - \varepsilon \\
&\geq 2\cdot(1/2+\varepsilon) - 1 - \varepsilon \geq \varepsilon,
\end{aligned}$$

as desired. □

Lemma A.2 ([Tal16], Lemma 5.19 restated)**.** *Let $\mathcal{D}$ be a distribution over $\{-1,1\}^n$ and let $\mathcal{G}$ be a class of functions. For $f\colon \{-1,1\}^n \to \{-1,1\}$, suppose that $D\colon \{-1,1\}^n \to \{-1,1\} \in \mathsf{FORMULA}[s] \circ \mathcal{G}$ is such that*

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[D(x) = f(x)] \geq 1/2 + \varepsilon_0.$$

Then there exists some $h\colon \{-1,1\}^n \to \{-1,1\} \in \mathsf{XOR}_{O\left(\sqrt{s}\cdot\log(1/\varepsilon_0)\right)} \circ \mathcal{G}$ such that

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[h(x)\cdot f(x)] \geq \frac{1}{s^{O\left(\sqrt{s}\cdot\log(1/\varepsilon_0)\right)}}.$$

Proof. Let

$$D = F(g_1, g_2 \ldots, g_s)$$

be a device in $\mathsf{FORMULA} \circ \mathcal{G}$ where F is a formula and $g_1, g_2, \ldots, g_s$ are function from $\mathcal{G}$.

Let $p\colon \{-1,1\}^s \to \mathbb{R}$ be a ε_0-approximating polynomial for F of degree $d = O(\sqrt{s}\cdot \log(1/\varepsilon_0))$. Note that we can write

$$p(z) = \sum_{\substack{S\subseteq[s]:\\ |S|\leq d}} \hat{p}(S)\cdot\prod_{i\in S} z_i.$$

Also, for each $S \subseteq [s]$, we have

$$|\hat{p}(S)| = \left|\mathop{\mathbf{E}}_{z\in\{-1,1\}^s}[p(z)\cdot\prod_{i\in S} z_i]\right| \leq 1 + \varepsilon_0.$$

Now let

$$\tilde{D} := p(g_1, g_2 \ldots, g_s).$$

Note that $\tilde{D}$ is a ε_0-approximating function for D. Therefore, by Lemma A.1, we have

$$\begin{aligned}
\varepsilon_0 &\leq \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[D(x)\cdot f(x)] \\
&= \mathop{\mathbf{E}}_{x\sim\mathcal{D}}\left[\left(\sum_{\substack{S\subseteq[s]:\\ |S|\leq d}} \hat{p}(S)\cdot\prod_{i\in S} g_i\right)\cdot f(x)\right] \\
&= \sum_{\substack{S\subseteq[s]:\\ |S|\leq d}} \hat{p}(S)\cdot\mathop{\mathbf{E}}_{x\sim\mathcal{D}}\left[\prod_{i\in S} g_i\cdot f(x)\right] \\
&\leq \sum_{\substack{S\subseteq[s]:\\ |S|\leq d}} (1+\varepsilon_0)\cdot\left|\mathop{\mathbf{E}}_{x\sim\mathcal{D}}\left[\prod_{i\in S} g_i\cdot f(x)\right]\right|.
\end{aligned}$$

The above equation is the sum of at most $s^{O(d)}$ summands. Therefore, there exists some $S \subseteq [s]$ such that

$$\left| \mathop{\mathbf{E}}_{x \sim \mathcal{D}} \left[\prod_{i \in S} g_i \cdot f(x) \right] \right| \geq \frac{\varepsilon_0}{(1+\varepsilon_0) \cdot s^{O(d)}} \geq \frac{1}{s^{O\left(\sqrt{s} \cdot \log(1/\varepsilon_0)\right)}},$$

which implies that there exists some h, such that either $h = \prod_{i \in S} g_i$ or $h = -\prod_{i \in S} g_i$, and

$$\mathop{\mathbf{E}}_{x \sim \mathcal{D}} [h(x) \cdot f(x)] \geq \frac{1}{s^{O\left(\sqrt{s} \cdot \log(1/\varepsilon_0)\right)}}.$$

Finally, note that such h can be expressed as the XOR of at most d functions from $\mathcal{G}$. □

A.2 PRG for low-communication functions in the number-in-hand setting

In this subsection, we show how to fool functions with low communication complexity in the number-in-hand model.

Theorem A.3 ([ASWZ96, INW94], Theorem 5.26 restated)**.** *For any $k \geq 2$, there exists a PRG that δ-fools any n-bits functions with k-party number-in-hand deterministic communication complexity at most D', with seed length*

$$n/k + O\left(D' + \log(1/\delta) + \log(k)\right) \cdot \log(k).$$

The PRG in Theorem A.3 is based on the PRG by Impagliazzo, Nisan and Wigderson [INW94] that is used to derandomize "network algorithms" and space-bounded computation. We will need to use randomness extractors, which we review below.

Definition A.4 (Min-entropy)**.** *Let X be a random variable. The* min-entropy *of X, denoted by $H_\infty(X)$, is the largest real number k such that $\mathbf{Pr}[X = x] \leq 2^{-k}$ for every x in the range of X. If X is a distribution over $\{-1,1\}^{\aleph}$ with $H_\infty(X) \geq k$, then X is called a* $(\aleph, k)$-source.

Definition A.5 (Extractors)**.** *A function* $\mathrm{Ext}\colon \{-1,1\}^{\aleph} \times \{-1,1\}^d \to \{-1,1\}^m$ *is an* (k,ε)-extractor *if, for any $(\aleph, k)$-source X, and any test $T\colon \{-1,1\}^m \to \{-1,1\}$, it is the case that*

$$|\mathbf{Pr}[T(\mathrm{Ext}(X, U_d)\, X) = 1] - \mathbf{Pr}[T(U_m) = 1]| \leq \varepsilon.$$

Theorem A.6 ([Vad12, Theorem 6.22])**.** *For any integer $m, \kappa > 0$ and $0 < \delta' < 0$, there exists an explicit (κ, δ') extractor* $\mathrm{Ext}\colon \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^m$ *with $d = O(m - k + \log(1/\delta'))$.*

We are now ready to show Theorem A.3.

Proof of Theorem A.3. We first describe the construction of the PRG. In fact, we will construct a sequence of PRGs $G_0, G_1, \ldots, G_{\log(k)}$. We begin by specifying the parameters of

these PRGs. Let $t = \log(k)$, and let

$$d = O\left(D' + \log(1/\delta) + t\right).$$

For $i = 0, 1, \ldots, t$, let

- $r_0 = n/k$,
- $r_i = r_{i-1} + d$.

Note that we have $r_i = n/k + i \cdot d$. Also, let

$$\mathrm{Ext}_i \colon \{0,1\}^{r_i} \times \{0,1\}^d \to \{0,1\}^{r_i}$$

be a (κ_i, δ')-extractor from Theorem A.6, where

$$\kappa_i = r_i - D' - 2t - \log(1/\delta)$$

and

$$\delta' = \delta / \left(3^t \cdot 2^{D'}\right).$$

Note that the seed length of the extractors is $d = O\left(D' + \log(1/\delta) + t\right)$. Finally, define $G_i \colon \{0,1\}^{r_i} \to \{0,1\}^{n/2^{t-i}}$ recursively as follows

- $G_0(a) = a$, where $a \in \{0,1\}^{n/k}$.
- $G_i(a, z) = G_{i-1}(a) \circ G_{i-1}(\mathrm{Ext}_{i-1}(a, z))$, where $a \in \{0,1\}^{r_{i-1}}$ and $z \in \{0,1\}^d$.

We will show that $G_t \colon \{0,1\}^{r_t = n/k + t \cdot d} \to \{0,1\}^n$ fools any functions f with k-party number-in-hand deterministic communication complexity at most D'. First, note that such f can be written as

$$f(x_1, x_2, \ldots, x_k) = \sum_{i=1}^{2^{D'}} h_1^{(i)}(x_1) \cdot h_2^{(i)}(x_2) \cdot \ldots \cdot h_k^{(i)}(x_k),$$

for some $h_j^{(i)} \colon \{0,1\}^{n/k} \to \{0,1\}$ $(i \in \left[2^{D'}\right], j \in [k])$. Therefore, to show that the PRG G_t δ-fool f, it suffices to show that G_t $\left(\delta/2^{D'}\right)$-fools every function g of the form

$$g(x_1, x_2, \ldots, x_k) = h_1(x_1) \cdot h_2(x_2) \cdot \ldots \cdot h_k(x_k).$$

More specifically we show the following.

Claim A.7. *For every $k \geq 2$ and $0 \leq i \leq t$, the generator G_i defined above $\left(3^i \cdot \delta'\right)$-fools every function $g_i \colon \{0,1\}^{n/2^{t-i}} \to \{0,1\}$ of the form*

$$g_i(x_1, x_2, \ldots, x_{k/2^{t-i}}) = h_1(x_1) \cdot h_2(x_2) \cdot \ldots \cdot h_{k/2^{t-i}}(x_{k/2^{t-i}}),$$

where $x_1, x_2, \ldots, x_{k/2^{t-i}} \in \{0,1\}^{n/k}$.

Proof of Claim A.7. The proof is by induction on i. The base case is $i = 0$, which is trivial given the definition of G_0. Now suppose the claim holds for $i-1$, we show the case for i. This is done using a hybrid argument. Consider the following four distributions

- $\mathcal{D}_1 = U_{n/2^{t-i}}$,
- $\mathcal{D}_2 = U_{n/2^{t-i+1}} \circ G_{i-1}(U_{r_{i-1}})$,
- $\mathcal{D}_3 = G_{i-1}(U_{r_{i-1}}) \circ G_{i-1}(U'_{r_{i-1}})$ (U and U' are two independent uniform distributions),
- $\mathcal{D}_4 = G_i(U_{r_i})$.

We want show show that

$$|\mathbf{E}[g_i(\mathcal{D}_1)] - \mathbf{E}[g_i(\mathcal{D}_4)]| \leq 3^i \cdot \delta'.$$

By the triangle inequality, it suffices to show that

$$|\mathbf{E}[g_i(\mathcal{D}_1)] - \mathbf{E}[g_i(\mathcal{D}_2)]| + |\mathbf{E}[g_i(\mathcal{D}_2)] - \mathbf{E}[g_i(\mathcal{D}_3)]| + |\mathbf{E}[g_i(\mathcal{D}_3)] - \mathbf{E}[g_i(\mathcal{D}_4)]| \leq 3^i \cdot \delta'. \quad \text{(A.1)}$$

We show Equation (A.1) by upper bounding each of the three summands.

First summand. We show that

$$|\mathbf{E}[g_i(\mathcal{D}_1)] - \mathbf{E}[g_i(\mathcal{D}_2)]| \leq 3^{i-1} \cdot \delta'. \quad \text{(A.2)}$$

Let us re-write g_i as

$$g_i(x_1, x_2, \ldots, x_{k/2^{t-i}}) = h^{\mathrm{L}}(x_1, x_2, \ldots, x_{k/2^{t-i+1}}) \cdot h^{\mathrm{R}}(x_{k/2^{t-i+1}+1}, x_{k/2^{t-i+1}+2}, \ldots, x_{k/2^{t-i}}),$$

where

$$h^{\mathrm{L}}(y) := \prod_{j=1}^{k/2^{t-i+1}} h_i(y) \quad \text{and} \quad h^{\mathrm{R}}(y) := \prod_{j=k/2^{t-i+1}}^{k/2^{t-1}} h_i(y).$$

Then,

$$\begin{aligned}
\mathbf{E}[g_i(\mathcal{D}_2)] &= \mathbf{E}\left[h^{\mathrm{L}}(U_{n/2^{t-i+1}}) \cdot h^{\mathrm{R}}(G_{i-1}(U_{r_{i-1}}))\right] \\
&= \mathbf{E}\left[h^{\mathrm{L}}(U_{n/2^{t-i+1}})\right] \cdot \mathbf{E}\left[h^{\mathrm{R}}(G_{i-1}(U_{r_{i-1}}))\right] \\
&= \mathbf{E}\left[h^{\mathrm{L}}(U_{n/2^{t-i+1}})\right] \cdot \left(\mathbf{E}\left[h^{\mathrm{R}}(U_{n/2^{t-i+1}})\right] \pm 3^{i-1} \cdot \delta'\right) \\
&\qquad\qquad \text{(By the induction hypothesis)} \\
&= \mathbf{E}\left[h^{\mathrm{L}}(U_{n/2^{t-i+1}})\right] \cdot \mathbf{E}\left[h^{\mathrm{R}}(U_{n/2^{t-i+1}})\right] \pm 3^{i-1} \cdot \delta' \\
&= \mathbf{E}[g_i(\mathcal{D}_1)] \pm 3^{i-1} \cdot \delta',
\end{aligned}$$

as desired.

Second summand. By a similar argument, it can be shown that

$$|\mathbf{E}[g_i(\mathcal{D}_2)] - \mathbf{E}[g_i(\mathcal{D}_3)]| \leq 3^{i-1} \cdot \delta'. \tag{A.3}$$

We omit the details here.

Third summand. We show that

$$|\mathbf{E}[g_i(\mathcal{D}_3)] - \mathbf{E}[g_i(\mathcal{D}_4)]| \leq \delta'. \tag{A.4}$$

We have

$$\begin{aligned}
\mathbf{E}[g_i(\mathcal{D}_4)] &= \mathbf{E}[g_i(G_i(U_{r_i}))] \\
&= \mathbf{E}\left[h^{\mathrm{L}}(G_{i-1}(X)) \cdot h^{\mathrm{R}}(G_{i-1}(\mathrm{Ext}_{i-1}(X,Z)))\right] \\
&\qquad\qquad \text{(where } X \sim \{0,1\}^{r_{i-1}} \text{ and } Z \sim \{0,1\}^d\text{)} \\
&= \mathbf{E}[A(X) \cdot B(\mathrm{Ext}_{i-1}(X,Z))] \\
&\qquad\qquad \text{(where } A(\cdot) = h^{\mathrm{L}}(G_{i-1}(\cdot)) \text{ and } B(\cdot) = h^{\mathrm{R}}(G_{i-1}(\cdot))\text{)} \\
&= \mathbf{E}[B(\mathrm{Ext}_{i-1}(X,Z)) \mid A(X) = 1] \cdot \mathbf{Pr}[A(X) = 1].
\end{aligned}$$

Similarly, we get

$$\mathbf{E}[g_i(\mathcal{D}_3)] = \mathbf{E}[B(U_{r_{i-1}}) \mid A(X) = 1] \cdot \mathbf{Pr}[A(X) = 1].$$

As a result, we have

$$\begin{aligned}
&|\mathbf{E}[g_i(\mathcal{D}_4)] - \mathbf{E}[g_i(\mathcal{D}_3)]| \\
&= \left|(\mathbf{E}[B(\mathrm{Ext}_{i-1}(X,Z)) \mid A(X) = 1] - \mathbf{E}[B(U_{r_{i-1}}) \mid A(X) = 1]) \cdot \mathbf{Pr}[A(X) = 1]\right|. \qquad \text{(A.5)}
\end{aligned}$$

On the one hand, if $\mathbf{Pr}[A(X) = 1] \leq \delta'$, then Equation (A.5) is at most δ'. On the other hand, if $\mathbf{Pr}[A(X) = 1] > \delta'$, then

$$H_\infty(X \mid A(X) = 1) > r_{i-1} - \log(1/\delta') > r_{i-1} - D' - 2t - \log(1/\delta) = \kappa_{i-1}.$$

Then by the fact that Ext_{i-1} is a (κ_{i-1}, δ')-extractor, we have

$$\left|\mathbf{E}[B(\mathrm{Ext}_{i-1}(X,Z)) \mid A(X) = 1] - \mathbf{E}[B(U_{r_{i-1}}) \mid A(X) = 1]\right| \leq \delta'.$$

Therefore, Equation (A.5) is at most δ' and this complete the proof of Equation (A.4). Finally, note that Equation (A.1) follows from Equation (A.2), Equation (A.3) and Equation (A.4). This completes the proof of Claim A.7. □ (Claim A.7)

Given Claim A.7, Theorem 5.26 now follows by letting $i = t$. □ (Theorem A.3)

Appendix B

Omitted proofs for Chapter 6

B.1 Circuit complexity of the Nisan-Zuckerman extractor: Proof of Lemma 6.21

In this section, we will describe the construction of the Nisan-Zuckerman extractor [NZ96], and show that it can be computed by a circuit of almost-linear size.

Lemma B.1 (Lemma 6.21, restated)**.** *There exists an extractor* $E\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *that is an* $(n/2, \varepsilon)$*-extractor with* $m = \Omega(n)$ *and* $d = \mathsf{polylog}(n/\varepsilon)$*. Moreover,* E *can be computed by a circuit of size* $n \cdot \mathsf{polylog}(n/\varepsilon)$*.*

In proving Lemma 6.21, we start with some definitions. The extractor works for sources of high min-entropy.

Definition B.2 (Dense source)**.** *We say that a distribution over* $\{0,1\}^n$ *is a* δ-source *if it has min-entropy at least* $\delta \cdot n$*.*

Definition B.3 (Block-wise source)**.** *A distribution* $X = (X_1, \ldots, X_s)$ *over* $\{0,1\}^{\ell_1} \times \cdots \times \{0,1\}^{\ell_s}$ *is called a* block-wise δ-source *if, for every* $x_1, \ldots, x_{i-1}$*,* $X_i|_{X_1 = x_1, \ldots, X_{i-1} = x_{i-1}}$ *is a* δ*-source (i.e., has min-entropy at least* $\delta \cdot \ell_i$*).*

The extractor will make use of universal hashing, which we define below.

Definition B.4 (k-wise independent hashing)**.** *A family of hash functions* $\mathcal{H} = \{h\colon \{0,1\}^n \to \{0,1\}^m\}$ *is called* k-wise independent *if, for any* $x_1, \ldots, x_k \in \{0,1\}^n$*, where* $x_1, \ldots, x_k$ *are distinct, and* $y_1, \ldots, y_k \in \{0,1\}^m$*, we have*

$$\Pr_{h \sim \mathcal{H}}[h(x_1) = y_1 \wedge \cdots \wedge h(x_k) = y_k] = (1/2^m)^k\,.$$

$\mathcal{H}$ *is also called a* universal hash family *if it is* 2*-wise independent.*

It is easy to see that any k-wise independent hashing family can be defined using some k-wise independent distribution. As a result, by Lemma 6.17, we have the following construction of k-wise independent hash families.

Lemma B.5. *There exists a k-wise independent hash family $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ such that, given any $h \in \mathcal{H}$, as a kn-bit string, the function h can be computed by a circuit of size $k \cdot \widetilde{O}(\max\{n, m\})$.*

The Nisan-Zuckerman extractor consists of two parts. The first part, block-wise source conversion, takes the source of high min-entropy and converts it into an almost block-wise source by building a list of "blocks". The second part, block-wise source extraction, takes the resulting block-wise source of the previous part and extracts the randomness "block-by-block", using some hash-based extractor. Next, we describe some basic component functions as well as how they are combined to perform the respective task of each part. The main focus here is around the circuit complexity of these procedures and we will not get into details about their correctness. Interested readers are referred to [NZ96, Section 5] for details on the correctness.

In the following, we only work with δ-sources and block-wise δ'-sources where δ and δ' are constants.

Block-wise source converter D. This function has the following parameters:

- n, the size of the original input;
- δ, the quality of the input source;
- $\ell_1 \leq \cdots \leq \ell_s \leq n$, the size of each block; and
- k, the amount of independence used.

We first describe how to build one block using a function that we call B. To build the i-th block, on input $x \in \{0,1\}^n$ and $y_i \in \{0,1\}^{k \log n}$, the function B first divides x into ℓ_i contiguous disjoint sets $A_1, \ldots, A_{\ell_i}$, each of size $m_i = n/\ell_i$. It then uses the $(k \log n)$-bit string y_i to pick, k-wise independently, $j_1, \ldots, j_{\ell_i}$, where $j_q \in [m_i]$, for each $q \in [\ell_i]$, and outputs the ℓ_i-bit vector

$$\left((A_1)_{j_1}, \ldots, (A_{\ell_i})_{j_{\ell_i}}\right).$$

The block-wise source converter D works as follows.

1. **Input:** $x \in \{0,1\}^n$ and $y_1, \ldots, y_s \in \{0,1\}^{k \log n}$.
2. **Output:** $(B(x, y_1), \ldots, B(x, y_s)) \in \{0,1\}^{\ell_1} \times \cdots \times \{0,1\}^{\ell_s}$.

Nisan and Zuckerman [NZ96] showed that if the input x is from a δ-source and $k = O(\log(1/\varepsilon))$, then, for all but at most a $\varepsilon/4$ fraction of the seeds $y_1, \ldots, y_s$, the output of the function D is $(\varepsilon/4)$-close to a block-wise δ'-source, where $\delta' = \Omega(\delta/\log(1/\delta))$.

Claim B.6. *The function D can be computed using a circuit of size $s \cdot k \cdot \widetilde{O}(n)$.*

Proof. It is sufficient to show that outputting the i-th block takes a circuit of size $k \cdot \widetilde{O}(n)$. On input $y_i \in \{0,1\}^{k \cdot \log n}$, we can compute, using Lemma 6.17, $(j_1, \ldots, j_{\ell_i}) \in [m_i]^{\ell_i}$ with a circuit of size

$$\ell_i \cdot k \cdot \widetilde{O}(\log(m_i \cdot \ell_i)) = k \cdot \widetilde{O}(n).$$

Then, for each index j_q, with $q \in [\ell_i]$, we can compute $(A_q)_{j_q}$ using a circuit of size $O(m_i)$ (by Lemma 6.8). □

Block-wise source extractor C. This function has $s+1$ parameters:

- δ', the quality of the block source, and
- $\ell_1, \ldots, \ell_s$, the block sizes. Here, $\frac{\ell_{i-1}}{\ell_i} = 1 + \frac{\delta'}{4}$, for all $1 < i \leq s$.

The way the block-wise source extractor C works is described below.

1. **Input:** $x_1 \in \{0,1\}^{\ell_1}, \ldots, x_s \in \{0,1\}^{\ell_s}$ and $y_0 \in \{0,1\}^{2\ell_s}$.
2. For each i, we consider a universal family of hash functions,

   $$\mathcal{H}_i = \left\{h \colon \{0,1\}^{\ell_i} \to \{0,1\}^{\delta' \ell_i / 2}\right\}_h,$$

   given by Lemma B.5, and each function in $\mathcal{H}_i$ is by $2\ell_i$ bits.
3. $h_s \leftarrow y_0$.
4. For $i \leftarrow s$ down to 1: $h_{i-1} \leftarrow h_i \circ h_i(x_i)$, where "$\circ$" denotes string concatenation.
5. **Output:** h_0, excluding the bits in h_s. Note that this output is a string in $\{0,1\}^m$.

It was shown in [NZ96] that if $x_1, \ldots, x_s$ are chosen from a block-wise δ'-source and y_0 is uniform, then the output of the function C is $\left(2 \cdot 2^{-\delta' \ell_s / 4}\right)$-close to uniform.

Claim B.7. *The function C can be computed using a circuit of size $s \cdot \widetilde{O}(\ell_1)$.*

Proof. Note that, given $h_i \in \{0,1\}^{2\ell_i}$ and $x_i \in \{0,1\}^{\ell_i}$, we can compute $h_i(x_i)$ using a circuit of size $\widetilde{O}(\ell_i)$ (by Lemma B.5). Then, to compute h_0, we need to compute h_i for $i = s-1, \ldots, 0$, which takes a circuit of size

$$\sum_{i=1}^{s} \widetilde{O}(\ell_i).$$

The above is at most $s \cdot \widetilde{O}(\ell_1)$, since ℓ_1 is the largest among $\ell_1, \ldots, \ell_s$. □

The final extractor E. The parameters are:

- n, the size of the input source;
- δ, where $1/n \leq \delta \leq 1/2$, the quality of the input source;
- ε, where $2^{-\delta n} \leq \varepsilon \leq 1/n$, the quality of the output distribution;
- $\delta' = \Theta(\delta/\log(1/\delta))$;
- $\ell_0 = \Theta(\delta^2 n/\log(1/\delta))$; $\ell_i = \ell_{i-1}/(1+\delta'/4)$ for each $0 < i < s$, with $s = O(\log(n)\log(1/\delta)/\delta)$; therefore, $\ell_s = \log(1/\varepsilon)\log(1/\delta)/\delta$;
- $k = O(\log(1/\varepsilon))$.

The following is a description of the extractor E:

1. **Input:** $x_1 \in \{0,1\}^n$, $y_1, \ldots, y_s \in \{0,1\}^{k \log n}$, $y_0 \in \{0,1\}^{2\ell_s}$.
2. **Output:** $C(D(x, y_1, \ldots, y_s), y_0)$. (Here, D and C are used with the parameters specified above.)

It was shown in [NZ96] that if x is from a δ-source and the y's are uniform, then the output of the function E is ε-close to a uniform m-bit string, where $m = \Omega(\delta^2 n/\log(1/\delta))$.

Claim B.8. *The function E can be computed using a circuit of size $n \cdot \mathsf{polylog}(n/\varepsilon)$.*

Proof. This follows easily from Claim B.6 and Claim B.7. □

B.2 The IMZ PRG is "almost strongly local"

Here, we show that the IMZ PRG [IMZ19] is "almost strongly local", in the sense that, for most of its seeds, the output of the PRG can be computed by some circuit of size comparable to its seed length.

Lemma B.9. *For any $s \geq N$, there exists a PRG $G\colon \{0,1\}^r \to \{0,1\}^N$ that $1/\mathsf{poly}(N)$-fools De Morgan formulas in N variables of size s, where $r = s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$. Moreover, for at least a fraction of $1 - 1/\mathsf{poly}(N)$ of the seeds $z \in \{0,1\}^r$, the function defined as*

$$g_z(j) = G(z)_j$$

can be computed by a circuit of size $s^{1/3} \cdot 2^{O\left(\log^{2/3} s\right)}$.

It is easy to see that such a PRG is sufficient to obtain MCSP lower bounds using our framework (see Theorem 6.13).

We first need a version of the pseudorandom shrinkage lemma, in which we select and fix the variables both in a pseudorandom manner (note that in Lemma 6.22 we select the variables pseudorandomly and then fix the variables in a truly-random manner). Such a pseudorandom shrinkage lemma is provided in [IMZ19].

Lemma B.10 (Pseudorandom shrinkage lemma, [IMZ12b, Lemma 4.8])**.** *There exists a constant $c_0 > 0$ such that the following hold. For any constant $c > c_0$, any $s \geq N$, $p \geq s^{-1/2}$, and any De Morgan formula F on N variables of size s, there exists a p-regular pseudorandom restriction $\mathcal{D}$ over $\{0,1,*\}^N$, that is samplable using $r = 2^{O(\log^{2/3} s)}$ random bits, such that*

$$\mathop{\mathbf{Pr}}_{\rho\sim\mathcal{D}}[L(F_\rho) \geq 2^{3\cdot c\cdot \log^{2/3} s} \cdot p^2 \cdot s] \leq s^{-c}.$$

Moreover, there exists a circuit of size $2^{O(\log^{2/3} s)}$ such that, given $j \in \{0,1\}^{\log N}$ and a seed $z \in \{0,1\}^r$, the circuit computes the j-th coordinate of $\mathcal{D}(z)$.

We are now ready to show Lemma B.9.

Proof of Lemma B.9. The construction is essentially that of [IMZ19]. We use the same parameters as those in the proof of Lemma 6.15.

The PRG first samples t independent pseudorandom restrictions using Lemma B.10. For each of the restrictions, the PRG replaces the $*$ coordinates with the output of some extractor (in fact, it is the output of some limited-independence generator that takes the output of the extractor as a seed). After the $*$ coordinates are replaced in each restriction, the PRG XORs, coordinate-wisely, the t binary strings.

More formally, the PRG takes as input a seed

$$(X, Y_1, \ldots, Y_t, \gamma_1, \ldots, \gamma_t) \in \{0,1\}^r,$$

where

- $X \in \{0,1\}^{\aleph}$ is the min-entropy source sample of an extractor,
- $Y_i \in \{0,1\}^{\mathsf{polylog}(N)}$, for each $i \in [t]$, is the seed of an extractor, and
- $\gamma_i \in \{0,1\}^{\ell}$, for each $i \in [t]$, is the seed for sampling a pseudorandom restriction.

Then, the j-th bit of the PRG is the XOR of a sequence of bits $(U_1)_j, \ldots, (U_t)_j$, where for each $i \in [t]$ the value of $(U_i)_j$ depends on the value of $(\rho_i)_j$, where ρ_i is a p-regular pseudorandom restriction sampled from Lemma B.10 with seed γ_i. Specifically,

$$(U_i)_j = \begin{cases} (\rho_i)_j, & \text{if } (\rho_i)_j \neq *, \text{ and} \\ (Z_i)_j = G_k(E(X, Y_i))_j, & \text{if } (\rho_i)_j = *, \end{cases}$$

where $E\colon \{0,1\}^{\aleph} \times \{0,1\}^d \to \{0,1\}^{\Omega(\aleph)}$ is an $(\aleph/2, \varepsilon)$-extractor and $G_k\colon \{0,1\}^{r_k} \to \{0,1\}^N$ is a k-independent generator. It was shown in [IMZ19] that the PRG constructed as above ε-fools De Morgan formulas of size s.

Note that, for each $i \in [t]$ and $j \in [N]$, $(\rho_i)_j$ can be computed by a circuit of size $M_1 = 2^{O(\log^{2/3} s)}$ (Lemma B.10). Also, using Lemma 6.21 and Lemma 6.17, $(Z_i)_j$ can be computed by a circuit of size $M_2 = \widetilde{O}(\aleph) = s^{1/3+o(1)}$.

To compute the j-th bit of the PRG, we need to have the values $(U_1)_j, \ldots, (U_t)_j$. It seems that we need to compute both $(\rho_i)_j$ (which is cheap to compute) and $(Z_i)_j$ (which is expensive to compute) for *all* $i \in [t]$, which seems to require size at least $t \cdot M_2 \geq s^{2/3}$. However, we want to compute this with a circuit of size $s^{1/3}$. The key observation here is that we do not need to compute $(Z_i)_j$ for all the i values; we only need to compute $(Z_i)_j$ for those i's such that the j-th coordinate of the i-th pseudorandom restriction is a star (i.e., $(\rho_i)_j = *$). Since the j-th coordinate is a star with probability p, we can expect to see only $p \cdot t \leq O(\log N)$ stars in the sequence $((\rho_i)_j)_{i\in[t]}$. In fact, since the t pseudorandom restrictions are independently sampled, by a standard concentration bound, with very high probability, we only see $\mathsf{polylog}(N)$ stars in the sequence. Then, a union bound over the N coordinates yields that, with high probability over the ρ's, we only have $\mathsf{polylog}(N)$ stars in $((\rho_i)_j)_{i\in[t]}$, for all $j \in [N]$. Therefore, for each of these "good" seeds, to compute the j-th bit of the PRG, we can first compute the sequence $((\rho_i)_j)_{i\in[t]}$ (which can be done with a circuit of size $t \cdot M_1$). Then, for each i such that the j-th coordinate of the i-th restriction is a star (there are only $\mathsf{polylog}(N)$ such i values), we compute $(Z_i)_j$. This can be done by a circuit of size $\mathsf{polylog}(N) \cdot M_2$.

We provide a sketch of how to implement a circuit performing the above task. First, we need to compute the sequence $((\rho_i)_j)_{i\in[t]}$, which can be done by a circuit of size $t \cdot 2^{O\left(\log^{2/3} s\right)}$. Then, we need to find the i's for which we need to compute $(Z_i)_j$ and select the corresponding Y_i's. This can be done by using divide and conquer and t "bins" with fixed $\mathsf{polylog}(N)$ "slots", each of size $\log t$, to store those indices i. Here, each slot is a set of gates that hold the bits of an index i and each bin is a set of slots.

More specifically, we will look through $(\rho_i)_j$ for $i \in [t]$ and "copy" to the bin those i's for which $(\rho_i)_j = *$. For the first step, we store the index "1" to the leftmost slot of the bin if and only if $(\rho_1)_j = *$. At the next step, we look at the current bin and the next index, say i'. We then create a new bin which holds all the indices in the previous bin and also i' if and only if $(\rho_1)_j = *$; here, the indices are stored in the leftmost slots and the rest of the slots are marked as "empty". Since each bin is of size $\mathsf{polylog}(N)$ and merging the current bin with a new index can be done in polynomial time (which implies that it can be done by a $\mathsf{polylog}(N)$-size circuit), each step can be done by a circuit of size at most $\mathsf{polylog}(N)$. After t steps, we will have a bin that stores all of the star indices (some of the slots in the bin can be empty). Therefore, the whole procedure can be done by a circuit of size $O(t) \cdot \mathsf{polylog}(N)$.

Once we have the indices, we retrieve the corresponding Y_i's (using Lemma 6.8). We then compute the extractor on each of these Y_i's (with the same min-entropy source sample X) and apply the limited-independence generator on the output of the extractor to get the j-th bit for each of those i's. We also need to make sure that we produce only 0 for those i's that come from the "empty" slots, in the bin where the indices are stored. Once we have those bits, we XOR them and then we XOR the resulting bit with the non-star values in $((\rho_i)_j)_{i\in[t]}$. The XOR of the non-star values can be obtained by taking the XOR of the values in $((\rho_i)_j)_{i\in[t]}$ and by treating the stars as 0's. □