Analytic Methods in Concrete Complexity

Li-Yang Tan

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2014

ABSTRACT

Analytic Methods in Concrete Complexity

Li-Yang Tan

This thesis studies computational complexity in concrete models of computation. We draw on a range of mathematical tools to understand the structure of Boolean functions, with analytic methods — Fourier analysis, probability theory, and approximation theory — playing a central role. These structural theorems are leveraged to obtain new computational results, both algorithmic upper bounds and complexity-theoretic lower bounds, in property testing, learning theory, and circuit complexity.

- We establish the best-known upper and lower bounds on the classical problem of testing whether an unknown Boolean function is monotone. We prove an $\tilde{\Omega}(n^{1/5})$ lower bound on the query complexity of non-adaptive testers, an exponential improvement over the previous lower bound of $\Omega(\log n)$ from 2002. We complement this with an $\tilde{O}(n^{5/6})$-query non-adaptive algorithm for the problem.

- We characterize the statistical query complexity of agnostically learning Boolean functions with respect to product distributions. We show that ℓ_1-approximability by low-degree polynomials, known to be sufficient for efficient learning in this setting, is in fact necessary. As an application we establish an optimal lower bound showing that no statistical query algorithm can efficiently agnostically learn monotone k-juntas for any $k = \omega(1)$ and any constant error less than $1/2$.

- We initiate a systematic study of the tradeoffs between accuracy and efficiency in Boolean circuit complexity, focusing on disjunctive normal form formulas, among the most basic types of circuits. A conceptual message that emerges is that the landscape of circuit complexity changes dramatically, both qualitatively and quantitatively, when the formula is only required to approximate a function rather than compute it exactly.

- Finally we consider the Fourier Entropy-Influence Conjecture, a longstanding open problem in the analysis of Boolean functions with significant applications in learning theory, the theory of pseudorandomness, and random graph theory. We prove a composition theorem for the conjecture, broadly expanding the class of functions for which the conjecture is known to be true.

Table of Contents

List of Tables

Acknowledgments

I am unbelievably lucky to have had Rocco Servedio as my advisor. I cannot imagine a more meaningful and enjoyable Ph.D. experience than that one I had, and this is almost entirely thanks to Rocco's inspiration, influence, and generosity. I have far too much to thank you for, Rocco — I will always be indebted to you for all that you have taught me over the years; for your outstanding advice and your faith in me; for the many opportunities I have benefited from and the exciting ones ahead; for the literary recommendations, the Diet Dr. Peppers, and so much more. Thanks for being an amazing advisor.

Next, I would like to express my thanks to Ryan O'Donnell. I have learned a tremendous amount from Ryan, starting from my very first day at Columbia when Rocco pointed me to his survey and course notes, and later through our collaborations. Ryan is a great person on top of being an inspiring researcher and expositor, and it has been a pleasure getting to know him. I had some of my best times in graduate school over the summer I visited CMU, and I am grateful to Ryan for this and many other opportunities.

I thank Johan Håstad for inviting me spend a semester in beautiful Stockholm. Interacting with Johan was always as inspiring as it was humbling, and I benefited greatly from every one of our conversations. Thanks also to Boaz Barak for an enjoyable internship at MSR New England, and for teaching me everything I know about the Lasserre hierarchy.

I wish to also give special thanks to Andrew Wan and Dov Gordon, whose friendship helped make this such a great experience, and to Tal Malkin, for her support and candor.

I thank the members of my thesis committee, Xi Chen, Ryan O'Donnell, Rocco Servedio, Cliff Stein, and Mihalis Yannakakis, for their time and insightful comments. I have had the privilege of collaborating with many brilliant people, and I thank my co-authors for everything they have taught me. Special thanks to Eric Blais, Clément Canonne, Dominik Scheder, Justin Thaler, Andrew Wan, Karl Wimmer, David Witmer, and John Wright for all the good times we shared.

On a more personal note:

I thank my friends in New York City: Jacqui Brown, AJ Cephus, Nigel Cheong, Sonya Chandra, Kofi Edzie, Carl Jeanbart, Bryce Kirschbaum, Robyn Lym, Matteo Malinverno, Alex McCurdy, and Li Yan McCurdy. My memories of this great city are filled with the times I shared with these great people.

My teenage years were mostly spent on the diamond. I thank the team for the countless good times and their steadfast friendship, and our coach Nelson Lim for teaching me lessons that continue to serve me well today. Going back further, much of my early childhood was spent under the care of Peggy Zee, and I thank her for being the first of many great influences in my life. While I do not remember much from those years, I know she has shaped the person I am today.

And above all I thank my family for their love, without which surely none of this would have been possible.

Bibliographic Note

Much of this research has been published already, and all of it was performed jointly with other researchers. Chapter 2 of this thesis is based on the paper "New Algorithms and Lower Bounds for Testing Monotonicity" which is joint work with Xi Chen and Rocco Servedio. Chapter 3 is based on the paper "Approximate Resilience, Monotonicity, and the Complexity of Agnostic Learning" which is joint work with Dana Dachman-Soled, Vitaly Feldman, Andrew Wan, and Karl Wimmer.

Chapter 4 is based on the paper "Approximating Boolean Functions with Depth-2 Circuits", which is joint work with Eric Blais and appeared in the Proceedings of the 28th IEEE Conference on Computational Complexity (CCC 2013). Chapter 5 is based on the paper "On DNF Approximators for Monotone Boolean Functions" which is joint work with Eric Blais, Johan Håstad, and Rocco Servedio and will appear in the Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP 2014). Chapter 6 is based on the paper "A Composition Theorem for the Fourier Entropy–Influence Conjecture" which is joint work with Ryan O'Donnell and appeared in the Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP 2013).

Chapter 1

Introduction

1.1 Concrete complexity

Computational complexity theory studies the nature and limitations of feasible computation. It seeks to understand the role of fundamental computing resources — time, space, and randomness, among others — in the design of efficient algorithms and in computational intractability. Core open problems of the field include:

- What is the relationship between time and space? Does $\mathsf{P} = \mathsf{L}$ or $\mathsf{P} = \mathsf{PSPACE}$?
- Is randomness essential for efficient computation? Does $\mathsf{P} = \mathsf{BPP}$?
- Can every sequential algorithm be efficiently parallelized? Does $\mathsf{P} = \mathsf{NC}$?

Rigorously considering these questions within a mathematical framework necessitates a formal model of computation: what *is* an algorithm for solving a computational problem, and what makes the algorithm efficient?

The Turing machine is the standard model of computation in theoretical computer science. For example, we define the time complexity of a computational problem to be the number of operations necessary for a Turing machine to solve a worst-case instance, and we identify the notion of tractability with the class of problems solvable by polynomial-time Turing machines. The universality and expressiveness of the Turing machine make it a good choice — it is a generally-accepted thesis that every physically realizable computation can

be efficiently simulated on the Turing machine — but these are also the primary reasons why the core open problems of the field remain insurmountable. Strikingly, for example, we are unable to establish even super-linear time lower bounds against NP, a seemingly modest first step towards an eventual separation of P from NP. This is because we remain far from understanding of what a Turing machine can and cannot compute in linear time.

Given this state of affairs it is natural to consider simpler, restricted models of computation where an algorithm is not afforded the full power of a Turing machine. This is the focus of concrete complexity, a subfield of computational complexity theory where concrete models of computation take the place of abstract Turing machines. Consider the following elementary result from concrete complexity:

> *Any deterministic comparison-based algorithm for sorting a list of N integers $A_1, \ldots, A_N$ must perform $\Omega(N \log N)$ operations in the worst case.*

This simple example illustrates a few hallmarks of concrete complexity:

- The algorithm is given limited access to the input and a restricted set of operations it can perform during intermediate computations. In our example above the algorithm is only permitted to make binary comparisons of the form "$A_i \geq A_j$?".

- Computation is captured by concrete mathematical objects, and computational complexity by structural properties of these objects. In our example every comparison-based sorting algorithm has a natural representation as a binary decision tree, and its time complexity corresponds to the depth of this tree.

- Computational lower bounds, while only applying within the restricted model, are unconditional. In particular, they do not rely on unproven complexity-theoretic assumptions such as $\mathsf{P} \neq \mathsf{NP}$ or the Exponential Time Hypothesis.

In the next two sections we take a closer look at a few aspects of concrete complexity — the types of problems considered, the mathematical objects that arise, and the techniques involved — with Boolean functions as our guide.

1.2 Boolean functions

The central mathematical object in this thesis is the Boolean function,

$$f : \{0,1\}^n \to \{0,1\}.$$

Boolean functions play a central role in several well-studied concrete models seeking to capture the complexity of basic algorithmic tasks. A few examples include:

- **Learning [Valiant, 1984]:** Given access to examples $\langle x^1, f(x^1)\rangle, \ldots, \langle x^m, f(x^m)\rangle$ labeled by f, efficiently construct a good approximation h to f.
- **Testing [Rubinfeld and Sudan, 1996]:** Given black-box query access to f, efficiently determine whether f has a certain property $\mathcal{P}$.
- **Communication [Yao, 1979]:** Alice has half of the input x to f, and Bob the other half. How much information do they need to exchange in order to compute $f(x)$?

Beyond their versatility in the modeling of basic algorithmic tasks like those above, Boolean functions are themselves a universal abstraction of computation; a Boolean function simply specifies how one output bit is determined by n input bits. Their computational complexity — the cost of computing $f(x)$ given x — is captured by the structural complexity of mathematical objects that naturally compute them. A few examples include:

- **Circuits:** What is the minimum size and depth of any circuit computing f? How are these measures affected by the types of gates allowed in the circuit?
- **Polynomials:** What is the minimum degree and sparsity of any polynomial computing f? How are these measures affected by the underlying field (e.g. $\mathbb{R}$ or $\mathbb{F}_2$)?
- **Halfspaces:** Can f be computed by a halfspace? That is, can we write $f(x) = \text{sign}(w \cdot x - \theta)$ for some $w \in \mathbb{R}^n$ and $\theta \in \mathbb{R}$? More generally, what is the smallest integer d such that $f(x) = \text{sign}(p(x))$ for a degree-d polynomial p?

In this thesis we study Boolean functions from both vantage points, considering both the computational complexity of algorithmic tasks involving Boolean functions and the

structural complexity of Boolean functions themselves. In Parts I and II we present new algorithms and lower bounds for testing and learning Boolean functions, and in Part III we prove new structural theorems about circuits and polynomials computing Boolean functions. As we will see these two directions are closely intertwined; in particular, the computational results in the first two parts build heavily on an improved understanding of the mathematical structure of Boolean functions.

1.3 Analytic methods

The results in this thesis are obtained by applying a range of tools from different branches of mathematics, with analytic methods — Fourier analysis, probability theory, and approximation theory — playing a central role.

The analysis of Boolean functions is concerned with properties of Boolean functions viewed as real polynomials via their Fourier transform. Introduced into theoretical computer science by the work of Kahn, Kalai, and Linial [Kahn *et al.*, 1988], this simple point of view is by now a mainstay of concrete complexity and has transformed areas such as property testing, learning theory, and the hardness of approximation. Its broad influence is mainly due to the many parallels that have been established between combinatorial properties of a Boolean function — linearity, monotonicity, noise sensitivity, etc. — and analytic properties of its Fourier spectrum. These connections allow us to drawn on a range of analytic tools to tackle computational problems that tend to be intrinsically combinatorial in nature. A few examples include:

- A Boolean function computes a linear $\mathbb{F}_2$-polynomial if and only if its Fourier expansion consists of a single monomial. A generalization of this simple fact underlies the analysis of a constant-query tester for the linearity of Boolean functions [Blum *et al.*, 1993; Bellare *et al.*, 1996].

- A monotone Boolean function has almost all of its Fourier mass concentrated on coefficients of low degree. This is the basis for our best-known algorithms for PAC learning monotone Boolean functions under the uniform distribution [Bshouty and Tamon, 1996; Servedio, 2004b; O'Donnell and Servedio, 2008].

- The circuit complexity of an n-variable Boolean function f composed with n copies of a Boolean function g is determined by the circuit complexity of g and the noise sensitivity of f. This characterization, proved using Fourier-analytic methods, initiated our study of hardness amplification within NP [O'Donnell, 2004].

In this thesis we further develop the analytic toolkit for studying Boolean functions, and we apply these tools to establish new and stronger connections between their combinatorial structure and Fourier spectrum. We leverage these structural theorems to obtain new computational results, both algorithmic upper bounds and complexity-theoretic lower bounds, in property testing, learning theory, and circuit complexity.

1.4 Outline of this thesis

Part I: Property Testing

In **Chapter 2** we consider the problem of *Property Testing* — efficiently determining whether a large, often high-dimensional, mathematical object has a certain property — within the elegant framework of Rubinfeld and Sudan [Rubinfeld and Sudan, 1996]. We study the classical problem of testing whether an unknown Boolean function $f : \{0,1\}^n \to \{0,1\}$ is monotone versus ε-far from every monotone function, and our two main results are a new lower bound and a new algorithm for this well-studied problem. First, we prove an $\tilde{\Omega}(n^{1/5})$ lower bound on the query complexity of any non-adaptive two-sided error algorithm for testing whether an unknown Boolean function f is monotone versus constant-far from monotone. This gives an exponential improvement over the previous lower bound of $\Omega(\log n)$ due to Fischer *et al.* [Fischer *et al.*, 2002]. We also show that the same lower bound holds for monotonicity testing of Boolean-valued functions over hypergrid domains $\{1,\ldots,m\}^n$ for all $m \geq 2$. Second, we give an $\tilde{O}(n^{5/6})\mathrm{poly}(1/\varepsilon)$-query algorithm that tests whether an unknown Boolean function f is monotone versus ε-far from monotone. Our algorithm, which is non-adaptive and makes one-sided error, is a modified version of the algorithm of Chakrabarty and Seshadhri [Chakrabarty and Seshadhri, 2013a], which makes $\tilde{O}(n^{7/8})\mathrm{poly}(1/\varepsilon)$ queries.

Part II: Computational Learning Theory

In **Chapter 3** we consider the problem of *Agnostic Learning* — efficient learning in the presence of harsh adversarial noise — within Kearns' well-studied statistical query variant [Kearns *et al.*, 1994] of Valiant's Probably Approximately Correct model [Valiant, 1984]. Our main result in this chapter is a characterization of the statistical query (SQ) complexity of agnostic learning via the analytic notion of *approximate resilience* of Boolean functions. Roughly speaking, we show that if all functions in a class $\mathcal{C}$ are far from being d-resilient then $\mathcal{C}$ can be learned agnostically in time $n^{O(d)}$, and conversely, if $\mathcal{C}$ contains a function close to being d-resilient then the SQ complexity of agnostically learning $\mathcal{C}$ is at least $n^{\Omega(d)}$. Our characterization implies that ℓ_1-approximability by low-degree polynomials, known to be sufficient for agnostic learning over product distributions [Kalai *et al.*, 2008], is in fact necessary. As an application we give an optimal lower bound showing that no SQ algorithm can efficiently agnostically learn monotone k-juntas for any $k = \omega(1)$ and any constant error less than $1/2$.

Part III: Boolean Function Complexity

In the third part of this thesis we study the structural complexity of mathematical objects — Boolean circuits and real polynomials — that compute Boolean functions.

In **Chapters 4 and 5** we initiate a systematic study of the tradeoffs between accuracy and efficiency in Boolean circuit complexity, focusing on disjunctive normal form (DNF) formulas, which are among the most basic types of circuits. We begin in **Chapter 4** by exploring two main directions: universal bounds on the approximability of all Boolean functions, and the approximability of the parity function. In the first direction, our main positive results are the first non-trivial universal upper bounds on approximability by DNFs, showing that:

- Every Boolean function can be ε-approximated by a DNF of size $O_\varepsilon(2^n/\log n)$.
- Every Boolean function can be ε-approximated by a DNF of width $c_\varepsilon\, n$, where $c_\varepsilon < 1$.

In the second direction our main positive result is the construction of an explicit DNF that approximates the parity function, showing that:

- PAR_n can be ε-approximated by a DNF of size $2^{(1-2\varepsilon)n}$ and width $(1-2\varepsilon)n$.

We continue this study in **Chapter 5**, turning our attention to the complexity of approximating *monotone* Boolean functions with DNF formulas. Our first result in this chapter is an explicit construction of DNF approximators for arbitrary monotone functions achieving one-sided error: we show that every monotone f can be ε-approximated by a DNF g of size $2^{n-\Omega_\varepsilon(\sqrt{n})}$ satisfying $g(x) \leq f(x)$ for all $x \in \{0,1\}^n$. Next we study the power of negations in DNF approximators for monotone functions. We exhibit monotone functions for which non-monotone DNF formulas perform better than monotone ones, giving separations with respect to both DNF size and width.

In **Chapter 6** we study the Fourier Entropy-Influence (FEI) conjecture, a long-standing open problem in the analysis of Booelan functions with significant applications in learning theory, the theory of pseudorandomness, and random graph theory. The conjecture seeks to relate two fundamental measures of Boolean function complexity: spectral entropy and total influence. Our main result in this chapter is a composition theorem for the FEI conjecture. We show that if $g_1, \ldots, g_k$ are functions over disjoint sets of variables satisfying the conjecture, and if the Fourier transform of F taken with respect to the product distribution with biases $\mathbf{E}[g_1], \ldots, \mathbf{E}[g_k]$ satisfies the conjecture, then their composition $F(g_1(x^1), \ldots, g_k(x^k))$ satisfies the conjecture. As an application we show that the FEI conjecture holds for read-once formulas over arbitrary gates of bounded arity, extending a recent result [O'Donnell *et al.*, 2011] which proved it for read-once decision trees. Our techniques also yield an explicit function with the largest known ratio of $C \geq 6.278$ between total influence and spectral entropy, improving on the previous lower bound of 4.615.

Finally in **Chapter 7** we close with a brief discussion of future research directions.

Part I

Property Testing

Chapter 2

New Algorithms and Lower Bounds for Testing Monotonicity

2.1 Background and context

Monotonicity is a basic and natural property of functions. In the field of property testing, the problem of efficiently testing whether an unknown function is monotone has been the focus of a long and fruitful line of research, with many works (see e.g. [Goldreich *et al.*, 1998; Dodis *et al.*, 1999; Goldreich *et al.*, 2000; Ergün *et al.*, 2000; Fischer *et al.*, 2002; Fischer, 2004; Batu *et al.*, 2004; Ailon *et al.*, 2007; Halevy and Kushilevitz, 2008; Rubinfeld and Servedio, 2009; Blais *et al.*, 2012; Briët *et al.*, 2012; Ron *et al.*, 2012; Chakrabarty and Seshadhri, 2013a; Chakrabarty and Seshadhri, 2013b; Chakrabarty and Seshadhri, 2013c; Blais *et al.*, 2013a]) studying this problem for functions with various domains and ranges.

In this work we will be concerned with the classical problem of testing monotonicity of *Boolean functions* $f : \{-1,1\}^n \to \{-1,1\}$, which was first posed and considered explicitly by Goldreich *et al.* [Goldreich *et al.*, 1998]. Recall that a Boolean function f is monotone if $f(x) \leq f(y)$ for all $x \prec y$, where $\prec$ denotes the bitwise partial order on the hypercube. Let $\mathrm{dist}(f,g) := \mathbf{Pr}_{\boldsymbol{x}\in\{-1,1\}^n}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})]$; we say that f is *ε-close to monotone* if $\mathrm{dist}(f,g) \leq \varepsilon$ for some monotone Boolean function g, and that f is *ε-far from monotone* otherwise. We will be interested in query-efficient randomized testing algorithms for the following task:

> *Given as input a distance parameter $\varepsilon > 0$ and oracle access to an unknown Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, output* Yes *with probability at least $2/3$ if f is monotone, and* No *with probability at least $2/3$ if f is ε-far from monotone.*

The work of Goldreich *et al.* [Goldreich *et al.*, 1998] proposed a simple "edge tester" which queries uniform random edges of $\{-1,1\}^n$ hoping to find an edge whose endpoints violate monotonicity. [Goldreich *et al.*, 1998] proved an $O(n^2 \log(1/\varepsilon)/\varepsilon)$ upper bound on the query complexity of the edge tester, which was subsequently improved to $O(n/\varepsilon)$ in the journal version [Goldreich *et al.*, 2000]. Fischer *et al.* [Fischer *et al.*, 2002] established the first lower bounds shortly after, showing that there exists a constant distance parameter $\varepsilon_0 > 0$ such that $\Omega(\log n)$ queries are necessary for any *non-adaptive* tester (one whose queries do not depend on the oracle's responses to prior queries). This directly implies an $\Omega(\log \log n)$ lower bound for adaptive testers, since any q-query adaptive tester can be simulated by a non-adaptive one that simply carries out all 2^q possible executions. These upper and lower bounds were the best known for more than a decade, until the recent work of Chakrabarty and Seshadhri [Chakrabarty and Seshadhri, 2013a] improved on the linear upper bound of Goldreich *et al.* with an $\tilde{O}(n^{7/8}\varepsilon^{-3/2})$-query tester.

Our main contributions in this work are (i) a new lower bound that improves on the [Fischer *et al.*, 2002] lower bound by an exponential factor, and (ii) a new algorithm that improves on the [Chakrabarty and Seshadhri, 2013a] upper bound (in terms of the dependence on n) by a polynomial factor. We now describe these contributions in more detail.

Our lower bound. We give an exponential improvement on the above-mentioned lower bounds of Fischer *et al.*:

Theorem 1. *There exists a universal constant $\varepsilon_0 > 0$ such that any non-adaptive algorithm for testing whether an unknown Boolean function is monotone versus ε_0-far from monotone must make $\Omega(n^{1/5}(\log n)^{-2/5})$ queries. Consequently, any adaptive algorithm must make $\Omega(\log n)$ queries.*

While the aforementioned results of Fischer *et al.* represent the previous best lower bounds on the general testing problem as defined above, additional lower bounds are known

for several restricted versions of the problem. In the same paper Fischer *et al.* gave an $\Omega(\sqrt{n})$ lower bound on the query complexity of any non-adaptive *one-sided* tester, i.e. one that always outputs Yes when f is monotone (again, this directly implies an $\Omega(\log n)$ lower bound for adaptive one-sided testers). Restricting further, a *pair tester* is a non-adaptive one-sided tester that independently draws pairs of comparable points $x \prec y$ from some distribution and rejects if and only if some pair that is drawn violates monotonicity. Briët *et al.* [Briët *et al.*, 2012] proved an $\Omega(n/(\varepsilon \log n))$ lower bound on the query complexity of pair testers whose query complexity can be written as $q(n)/\varepsilon$ for some function q.

In addition to Theorem 1, we show that essentially the same lower bound holds for monotonicity testing of Boolean-valued functions over hypergrid domains $\{1, \dots, m\}^n$ for $m \geq 2$. (Below and throughout this chapter we write $[m]$ to denote $\{1, 2, \dots, m\}$.) Our most general lower bound is the following:

Theorem 2. *There exists a universal constant $\varepsilon_0 > 0$ such that for all $m \geq 2$, any non-adaptive algorithm for testing whether an unknown function $f : [m]^n \to \{-1, 1\}$ is monotone versus ε_0-far from monotone must make $\tilde{\Omega}(n^{1/5})$ queries.*

To the best of our knowledge Theorem 2 is the first lower bound for testing monotonicity of *Boolean-valued* functions over hypergrid domains. Recent papers of Chakrabarty and Seshadhri [Chakrabarty and Seshadhri, 2013b; Chakrabarty and Seshadhri, 2013c] and Blais *et al.* [Blais *et al.*, 2013a] essentially close the problem of testing monotonicity of functions $f : [m]^n \to \mathbb{N}$, showing that $\Theta(n \log m)$ queries are both necessary and sufficient; however, their lower bounds crucially depend on the functions considered having range $\mathbb{N}$ rather than $\{-1, 1\}$.

Our algorithm. We present a new algorithm for monotonicity testing and prove the following result about its performance:

Theorem 3. *There is a $\tilde{O}(n^{5/6}\varepsilon^{-4})$-query one-sided non-adaptive algorithm for testing whether an unknown n-variable Boolean function is monotone versus ε-far from monotone.*

Recall that the one-sided, non-adaptive tester of Chakrabarty and Seshadhri [Chakrabarty and Seshadhri, 2013a] makes $\tilde{O}(n^{7/8}\varepsilon^{-3/2})$ queries. Thus, while the query complexity of our

tester is worse as a function of $1/\varepsilon$ (though still polynomial), its query complexity is polynomially better as a function of n.[1] Like the [Chakrabarty and Seshadhri, 2013a] algorithm, our algorithm is a pair tester, but it evades the $\Omega(n/(\varepsilon \log n))$ lower bound of [Briët *et al.*, 2012] because its query complexity is not of the form $q(n)/\varepsilon$. Our algorithm builds on the tools developed in [Chakrabarty and Seshadhri, 2013a]; its high-level structure is similar to that of the [Chakrabarty and Seshadhri, 2013a] algorithm, but with an important difference that enables an improved analysis. See Section 2.1.2 for more discussion on this point.

2.1.1 The lower bound approach

Our lower bound for testing monotonicity builds on previous lower bounds for testing restricted classes of *linear threshold functions* (LTFs). Recall that $f : \{-1,1\}^n \to \{-1,1\}$ is a linear threshold function if there exist $w_1, \ldots, w_n, \theta \in \mathbb{R}^n$ such that $f(x) = \text{sign}(w \cdot x - \theta)$ for all $x \in \{-1,1\}^n$.

Background. A *signed majority function* is a linear threshold function of the special form $f(x) = \text{sign}(w \cdot x)$ where $w \in \{-1,1\}^n$. While [Matulef *et al.*, 2010] showed that the class of all LTFs is ε-testable using $\text{poly}(1/\varepsilon)$ queries (independent of n), in [Matulef *et al.*, 2009] Matulef *et al.* gave an $\Omega(\log n)$ lower bound for non-adaptive algorithms that ε_0-test whether $f : \{-1,1\}^n \to \{-1,1\}$ is a signed majority function, where $\varepsilon_0 > 0$ is a universal constant. Like many lower bound arguments in property testing, the proof of [Matulef *et al.*, 2009] employs Yao's minimax principle [Yao, 1977], and works by exhibiting two distributions $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$ over LTFs — more precisely, $\mathcal{D}_{yes}$ is the uniform distribution over all 2^n signed majority functions, and $\mathcal{D}_{no}$ is the uniform distribution over a set of LTFs almost all of which are constant-far from every signed majority function — and arguing that for $q = o(\log n)$, any deterministic q-query algorithm cannot distinguish between the two distributions with non-negligible success probability. (We note that a typical function from $\mathcal{D}_{yes}$ is far from being monotone, and that the same holds for a typical LTF drawn

[1] Recall that in property testing the dependence on the size parameter "n" is typically viewed as more important than the dependence on the "closeness" parameter ε. Indeed, ε is often viewed as a constant, so testers with query complexities that are exponential (or worse) as a function of $1/\varepsilon$ but independent of n are commonly referred to as "constant-query testers."

from the $\mathcal{D}_{no}$ distribution of [Matulef *et al.*, 2009].) A key tool in the [Matulef *et al.*, 2009] proof is the Berry–Esséen "central limit theorem (CLT) with error bounds" for sums of independent real-valued random variables.

An *embedded majority function of size* k is an LTF $f : \{-1,1\}^n \to \{-1,1\}$ of the form $f(x) = \mathrm{sign}(w \cdot x)$ where $w \in \{0,1\}^n$ is a vector with exactly k ones. In [Blais and O'Donnell, 2010] Blais and O'Donnell showed that for $k = n/2$, any non-adaptive testing algorithm for the class of all embedded majority functions of size exactly $n/2$ must make $\Omega(n^{1/12})$ queries. Their proof employed a $\mathcal{D}_{yes}$ distribution which is the uniform distribution over all embedded majority functions of size $n/2$, and a $\mathcal{D}_{no}$ distribution which is supported on certain monotone LTFs (which are far from embedded majority functions of size $n/2$). A key technical ingredient in the proofs of [Blais and O'Donnell, 2010] is a multidimensional extension of the Berry–Esséen theorem (to independent sums of $\mathbb{R}^q$-valued random variables) which was essentially established in the work of [Gopalan *et al.*, 2010], building on ingredients from [Mossel, 2008]. Subsequently Ron and Servedio [Ron and Servedio, 2013] adapted the arguments of [Blais and O'Donnell, 2010] to give an improved analysis of the same $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$ distributions from [Matulef *et al.*, 2009] and establish an $\Omega(n^{1/12})$-query lower bound for non-adaptive algorithms that ε_0-test whether $f : \{-1,1\}^n \to \{-1,1\}$ is a signed majority function, thus exponentially improving over the [Matulef *et al.*, 2009] lower bounds for this problem.

This work. Neither the [Blais and O'Donnell, 2010] construction nor the [Matulef *et al.*, 2009; Ron and Servedio, 2013] construction can be used directly to establish a lower bound for monotonicity testing of functions $f : \{-1,1\}^n \to \{-1,1\}$; as described above, in the [Blais and O'Donnell, 2010] construction both the $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$ functions are monotone, and in the [Matulef *et al.*, 2009; Ron and Servedio, 2013] construction a typical function from either distribution is far from monotone. Nevertheless, in this work we show that ingredients from [Blais and O'Donnell, 2010; Ron and Servedio, 2013] can be leveraged to obtain a polynomial lower bound for testing monotonicity of functions $f : \{-1,1\}^n \to \{-1,1\}$. Like these earlier works we employ Yao's principle: we define a $\mathcal{D}_{yes}$ distribution that is supported on monotone LTFs, and a $\mathcal{D}_{no}$ distribution over LTFs that is almost entirely supported on LTFs that are constant-far from every monotone function, and use an analysis which is

fairly similar to that of [Blais and O'Donnell, 2010; Ron and Servedio, 2013], to prove Theorem 1. Using the multidimensional Berry–Esséen theorem of [Gopalan *et al.*, 2010] to analyze our $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$ distributions would result in an $\Omega(n^{1/12})$ lower bound. To obtain our improved $\Omega(n^{1/5}\log^{-2/5} n)$ lower bound, we instead adapt a multidimensional CLT of Valiant and Valiant [Valiant and Valiant, 2011] (for Wasserstein distance) to our context.

2.1.2 The approach of our algorithm

Our algorithm builds on ingredients from [Chakrabarty and Seshadhri, 2013a], so to explain our approach we first recall the necessary ingredients from that work. Fix a Boolean function[2] $f : \{0,1\}^n \to \{0,1\}$, and let us say that a pair of inputs (x, y) with $x \prec y$ is a *violated edge* if $f(x) = 1, f(y) = 0$ and (x, y) is an edge in $\{0,1\}^n$ (i.e. the Hamming distance between them is 1). [Chakrabarty and Seshadhri, 2013a] establishes a very useful "dichotomy theorem" about Boolean functions $f : \{0,1\}^n \to \{0,1\}$ that are ε-far from monotone: for any $s > 0$, any such function either must have $\Omega(\varepsilon s 2^n)$ violated edges, or must have a *matching* (i.e. a vertex-disjoint set) of $\Omega(\varepsilon 2^n/s)$ violated edges.

To use this dichotomy theorem, Chakrabarty and Seshadhri [Chakrabarty and Seshadhri, 2013a] define a "path tester" which works essentially as follows: it selects a random directed path $\boldsymbol{p}$ of n edges from 0^n up to 1^n, draws two uniform random points $\boldsymbol{x} \prec \boldsymbol{y}$ from the "middle layers" of $\boldsymbol{p}$, and rejects if $\boldsymbol{x}$ and $\boldsymbol{y}$ violate monotonicity, i.e. $f(\boldsymbol{x}) = 1$ and $f(\boldsymbol{y}) = 0$.[3] They prove that if f has a matching of $\Omega(\sigma 2^n)$ violated edges, then their path tester will uncover a violation and reject with probability $\tilde{\Omega}(\sigma^3/\sqrt{n})$. (Roughly speaking, they show that about an $\Omega(\sigma)$ fraction of possible outcomes of $\boldsymbol{y}$, corresponding to the $\sigma 2^n$ upper endpoints of the edges in the matching, are such that with probability $\tilde{\Omega}(\sigma^2/\sqrt{n})$ over the random draw of $\boldsymbol{x}$, the pair $\boldsymbol{y}$ and $\boldsymbol{x}$ together constitute a violation.) On the other hand, if f does not have a matching of this size then (by the dichotomy theorem) it must

[2] For our algorithmic result it will be more convenient to view Boolean functions as mapping $\{0,1\}^n$ to $\{0,1\}$.

[3] Here the "middle layers" of $\boldsymbol{p}$ are the points on the path that have $n/2 \pm O_\varepsilon(\sqrt{n})$ many coordinates which are 1; intuitively, at most an ε-fraction of all points in $\{0,1\}^n$ lie outside these "middle layers" of the hypercube. We note that the above description is a slight simplification of the actual [Chakrabarty and Seshadhri, 2013a] path tester, omitting some details which are not necessary at this stage of our description.

have $\Omega((\varepsilon^2/\sigma)2^n)$ violated edges, so the edge tester of [Goldreich *et al.*, 1998] (querying the endpoints of a uniform random edge) will hit a violated edge with probability $\Omega(\varepsilon^2/(\sigma n))$. Their final algorithm runs their path tester with probability $1/2$ and queries a random edge with probability $1/2$. Choosing σ suitably to equalize the two rejection probabilities, this is a two-query algorithm which succeeds in uncovering a violation for any ε-far-from-monotone function f with probability $\tilde{\Omega}(\varepsilon^{3/2}/n^{7/8})$, giving them a one-sided non-adaptive tester which makes $\tilde{O}(n^{7/8}/\varepsilon^{3/2})$ queries overall.

Our algorithm follows the same high-level framework described above, but differs from [Chakrabarty and Seshadhri, 2013a] by employing a different path tester. After selecting a random path $\boldsymbol{p}$, instead of (essentially) drawing two independent uniform points from the middle layers of the path as is done in [Chakrabarty and Seshadhri, 2013a], our path tester draws a *correlated* pair of points from $\boldsymbol{p}$. More precisely, it selects the first point $\boldsymbol{y}$ independently from the middle layers of $\boldsymbol{p}$, and preferentially selects the second point $\boldsymbol{x}$ from $\boldsymbol{p}$ in a way which favors points which are closer to $\boldsymbol{y}$. Via a careful analysis we are able to show that if f has a matching of $\Omega(\sigma 2^n)$ violated edges, then our path tester will uncover a violation and reject with probability $\tilde{\Omega}(\sigma^2/\sqrt{n}) \cdot \mathrm{poly}(\varepsilon)$. Roughly speaking, we show that if $\boldsymbol{y}$ is a uniform random upper endpoint of the $\sigma 2^n$ edges in the matching (which occurs with probability about σ), then the probability that our tester selects a string $\boldsymbol{x}$ which gives a violation with $\boldsymbol{y}$ is $\tilde{\Omega}(\sigma/\sqrt{n}) \cdot \mathrm{poly}(\varepsilon)$. Trading this off against the success probability of the edge tester using the dichotomy theorem, we obtain our improved query bound.

Organization of this chapter. Our lower bound results are established Sections 2.2 through 2.4. The two distributions $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$ are defined at the beginning of Section 2.2. In Section 2.2.1 we show that with high probability an LTF drawn from $\mathcal{D}_{no}$ is constant-far from monotone, and in Section 2.2.2 we show that unless $q = \Omega(n^{1/5}(\log n)^{-2/5})$, any deterministic q-query algorithm cannot distinguish between the two distributions with non-negligible success probability. The key technical ingredient in our proof of the latter is a lemma that adapts the Valiant–Valiant multidimensional CLT for Wasserstein distance to our context; we prove this lemma in Section 2.3. Finally in Section 2.4 we prove Theorem 2, showing that the same lower bound of $\tilde{\Omega}(n^{1/5})$ also applies to the query complexity of testers for monotonicity of functions $f : [m]^n \to \{0,1\}$ over general hypergrid domains; we do so

via a reduction to the $m = 2$ case (Theorem 1).

Our algorithmic result is established in Section 2.5. In Section 2.5.1 we describe two useful distributions over comparable pairs $(\boldsymbol{x}, \boldsymbol{y})$ from the middle layers of $\{0,1\}^n$ and bound the probability of having both points landing in a fixed set A of size $\sigma 2^n$. Then in Section 2.5.2 we define the *score* of a point x with respect to a set A of points, and use the result of Section 2.5.1 to lower bound the sum of score(x, A) over all points $x \in A$. We present our modified path tester as well as the analysis of its success probability in Section 2.5.3. Finally in Section 2.5.4 we combine this tester and the dichotomy theorem of [Chakrabarty and Seshadhri, 2013a] to obtain our improved upper bound.

2.1.3 Preliminaries

All probabilities and expectations are with respect to the uniform distribution unless otherwise stated; we will use boldface letters (e.g. $\boldsymbol{x}$ and $\mathbf{X}$) to denote random variables. For a $q \times n$ matrix $Q \in \mathbb{R}^{q \times n}$, we write $Q_{i*} \in \mathbb{R}^n$ to denote its i-th row, $Q_{*j} \in \mathbb{R}^q$ its j-th column, and $Q_{i,j} \in \mathbb{R}$ its entry in the i-th column and j-th row. We write $\prec$ to denote the coordinate-wise partial order on $\{-1,1\}^n$, where $x \prec y$ iff $x_i \leq y_i$ for all $i \in [n]$ and $x \neq y$. We say that x and y are *comparable* if $x \prec y$, $y \prec x$, or $x = y$. Given two functions $f, g : \{-1,1\}^n \to \{-1,1\}$ we will use dist(f, g) to denote the (normalized Hamming) distance $\mathbf{Pr}_{\boldsymbol{x} \in \{-1,1\}^n}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})]$ between f and g.

Recall that $f : \{-1,1\}^n \to \{-1,1\}$ is monotone if $f(x) \leq f(y)$ for all $x, y \in \{-1,1\}^n$ such that $x \prec y$. We say that f is *ε-close to monotone* if dist$(f, g) \leq \varepsilon$ for some monotone $g : \{-1,1\}^n \to \{-1,1\}$, and *$\varepsilon$-far from monotone* otherwise. A linear threshold function (LTF) over $\{-1,1\}^n$ is a function $f : \{-1,1\}^n \to \{-1,1\}$ that can be expressed as $f(x) = \text{sign}(w \cdot x - \theta)$ for some $w_1, \ldots, w_n$, $\theta \in \mathbb{R}$. Here sign : $\mathbb{R} \to \{-1,1\}$ is the sign function sign$(t) = 1$ if $t \geq 0$ and sign$(t) = -1$ if $t < 0$. For $f(x) = \text{sign}(w \cdot x - \theta)$, an LTF over $\{-1,1\}^n$, it is straightforward to verify that if $w_i \geq 0$ for all $i \in [n]$ then f is monotone.

We will need a few standard facts from probability theory:

Fact 2.1.1 (Gaussian anti-concentration)**.** *Let $\mathcal{G}$ be a Gaussian with variance σ^2. Then for all $\varepsilon > 0$ it holds that* $\sup_{\theta \in \mathbb{R}} \left\{ \mathbf{Pr}\left[|\mathcal{G} - \theta| \leq \varepsilon\sigma\right] \right\} \leq \varepsilon$.

Fact 2.1.2 (Gaussian concentration)**.** *Let $\mathcal{G}$ be a Gaussian with mean 0 and variance σ^2. Then for all $0 < a < 1$ it holds that $\mathbf{Pr}\left[\mathcal{G} \in [0, a\sigma]\right] = \Omega(a)$.*

Theorem 4 (Berry–Esséen)**.** *Let $\mathbf{S} = \mathbf{X}_1 + \cdots + \mathbf{X}_n$ where $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are independent real-valued random variables with $\mathbf{E}[\mathbf{X}_j] = \mu_j$ and $\mathbf{Var}[\mathbf{X}_j] = \sigma_j^2$, and suppose that $|\mathbf{X}_j - \mathbf{E}[\mathbf{X}_j]| \leq \tau$ with probability 1 for all $j \in [n]$. Let $\mathcal{G}$ be a Gaussian with mean $\sum_{j=1}^n \mu_j$ and variance $\sum_{j=1}^n \sigma_j^2$, matching those of $\mathbf{S}$. Then for all $\theta \in \mathbb{R}$, we have*

$$\left|\mathbf{Pr}[\mathbf{S} \leq \theta] - \mathbf{Pr}[\mathcal{G} \leq \theta]\right| \leq \frac{O(\tau)}{\left(\sum_{j=1}^n \sigma_j^2\right)^{1/2}}.$$

Fact 2.1.3. *For all $c > 0$ there exists an $\varepsilon = \varepsilon(c) \in (0, 1]$ such that the following holds. For all even (resp. odd) n,*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \in \{-1,1\}^n}\left[\sum_{i=1}^n \boldsymbol{x}_i = k\right] \geq \frac{\varepsilon}{\sqrt{n}} \quad \text{for all even (resp. odd) integers } k \in [-c\sqrt{n}, c\sqrt{n}].$$

Finally we recall a few basic facts from the Fourier analysis over the hypercube which we require (for a comprehensive treatment of this topic see [O'Donnell, 2014]). Every function $f : \{-1, 1\}^n \to \mathbb{R}$ can be uniquely expressed as a multilinear polynomial

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S) \prod_{i \in S} x_i \quad \text{where } \widehat{f}(S) := \mathop{\mathbf{E}}_{\boldsymbol{x} \in \{-1,1\}^n}\left[f(\boldsymbol{x}) \prod_{i \in S} \boldsymbol{x}_i\right],$$

known as the *Fourier transform* of f. The numbers $\widehat{f}(S) \in \mathbb{R}$ are the *Fourier coefficients* of f; with a slight abuse of notation we will write $\widehat{f}(i)$ instead of $\widehat{f}(\{i\})$ for the degree-1 Fourier coefficients.

Fact 2.1.4 (Parseval's identity)**.** *Let $f : \{-1, 1\}^n \to \mathbb{R}$. Then*

$$\mathop{\mathbf{E}}_{\boldsymbol{x} \in \{-1,1\}^n}[f(\boldsymbol{x})^2] = \sum_{S \subseteq [n]} \widehat{f}(S)^2.$$

For $i \in [n]$, the *influence of coordinate i on f*, denoted $\mathbf{Inf}_i[f]$, is the probability

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

where $\boldsymbol{x}^{\oplus i}$ denotes the string $\boldsymbol{x}$ with its i-th coordinate flipped. The following fact relates the influences of an LTF to its degree-1 Fourier coefficients:

Fact 2.1.5. *Let $f(x) = \mathrm{sign}(w_1 x_1 + \cdots + w_n x_n - \theta)$ be an LTF over $\{-1, 1\}^n$. Then for all $i \in [n]$, $\mathbf{Inf}_i[f] = \widehat{f}(i)$ if $w_i \geq 0$ and $\mathbf{Inf}_i[f] = -\widehat{f}(i)$ if $w_i < 0$.*

2.2 A polynomial lower bound for testing monotonicity

Let $\mathcal{D}_{yes}$ be the following distribution over monotone LTFs on $\{-1,1\}^n$: a draw $\boldsymbol{f}_{yes} \sim \mathcal{D}_{yes}$ is $\boldsymbol{f}_{yes}(x) = \text{sign}(\boldsymbol{\sigma}_1 x_1 + \cdots + \boldsymbol{\sigma}_n x_n)$, where each $\boldsymbol{\sigma}_i$ is independently and uniformly chosen from $\{1,3\}$. The distribution $\mathcal{D}_{no}$ is similarly a distribution over LTFs $\boldsymbol{f}_{no}(x) = \text{sign}(\boldsymbol{\nu}_1 x_1 + \cdots + \boldsymbol{\nu}_n x_n)$, but each $\boldsymbol{\nu}_i$ is independently chosen to be -1 with probability $1/10$, and $7/3$ with probability $9/10$. The following two propositions along with a standard application of Yao's minimax principle [Yao, 1977] yield Theorem 2:

Proposition 2.2.1. *There exists a universal positive constant $\varepsilon_0 > 0$ such that with probability $1 - o_n(1)$, a random LTF $\boldsymbol{f}_{no} \sim \mathcal{D}_{no}$ satisfies $\text{dist}(\boldsymbol{f}_{no}, g) > \varepsilon_0$ for all monotone Boolean functions $g : \{-1,1\}^n \to \{-1,1\}$.*

Proposition 2.2.2. *Let $\mathcal{T}$ be any deterministic non-adaptive two-sided q-query algorithm for testing whether a black-box Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ is monotone. Then*

$$\left| \Pr_{\boldsymbol{f}_{yes} \sim \mathcal{D}_{yes}} \left[\mathcal{T} \text{ outputs Yes on } \boldsymbol{f}_{yes} \right] - \Pr_{\boldsymbol{f}_{no} \sim \mathcal{D}_{no}} \left[\mathcal{T} \text{ outputs Yes on } \boldsymbol{f}_{no} \right] \right| = O\left(\frac{q^{5/4} (\log n)^{1/2}}{n^{1/4}} \right). \tag{2.1}$$

We prove Proposition 2.2.1 in Section 2.2.1, followed by Proposition 2.2.2 in Section 2.2.2.

2.2.1 Proof of Proposition 2.2.1

By the Chernoff bound, with probability $1 - o_n(1)$ a draw $\boldsymbol{f}_{no} = \text{sign}(\boldsymbol{\nu}_1 x_1 + \cdots + \boldsymbol{\nu}_n x_n)$ from $\mathcal{D}_{no}$ satisfies

$$|\{i \in [n] : \boldsymbol{\nu}_i = -1\}| \in \left[0.1n - \sqrt{n \log n}, 0.1n + \sqrt{n \log n} \right].$$

We call any such LTF *nice*, and we will argue that all nice LTFs are constant-far from monotonicity. For the remainder of this proof let f be a nice LTF, which we may without loss of generality express as $f(x) = \text{sign}(\ell(x))$ where

$$\ell(x) := -(x_1 + \cdots + x_m) + \tfrac{7}{3} \cdot (x_{m+1} + \cdots + x_n)$$

and $m \in \left[0.1n - \sqrt{n \log n}, 0.1n + \sqrt{n \log n} \right]$. We assume that m is odd, noting that the case when m is even follows via an identical argument. We first claim that $\mathbf{Inf}_i[f] = \Omega(1/\sqrt{n})$

for all $i \in [m]$; by symmetry it suffices to show this for $i = 1$. Define $\ell'(x) := -(x_2 + \ldots + x_m) + \frac{7}{3}(x_{m+1} + \cdots + x_n)$ and note that $f(x) \neq f(x^{\oplus 1})$ if and only if $\ell'(x) \in [-1, 1)$. Applying Fact 2.1.3 twice, we have

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\in\{-1,1\}^n}\left[\tfrac{7}{3}(\boldsymbol{x}_{m+1} + \cdots + \boldsymbol{x}_n) \in [-\sqrt{n}, \sqrt{n}]\right] = \Omega(1)$$

and

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\in\{-1,1\}^n}[\boldsymbol{x}_2 + \cdots + \boldsymbol{x}_m = k] = \Omega(1/\sqrt{n}) \text{ for all even integers } k \in [-\sqrt{n} - 1, \sqrt{n} + 1],$$

and therefore indeed,

$$\mathbf{Inf}_1[f] = \mathop{\mathbf{Pr}}_{\boldsymbol{x}\in\{-1,1\}^n}[\ell'(\boldsymbol{x}) \in [-1, 1)] = \Omega(1/\sqrt{n}).$$

Since $\mathbf{Inf}_i[f] = \Omega(1/\sqrt{n})$ for all $i \in [m]$, by Fact 2.1.5 we have that $\widehat{f}(i) = -\Omega(1/\sqrt{n})$ for all $i \in [m]$. Hence for all monotone Boolean functions g, we have

$$\begin{aligned} 4 \cdot \mathrm{dist}(f, g) \;=\; \mathop{\mathbf{E}}_{\boldsymbol{x}\in\{-1,1\}^n}\left[(f(\boldsymbol{x}) - g(\boldsymbol{x}))^2\right] \;&=\; \sum_{S\subseteq[n]} \left(\widehat{f}(S) - \widehat{g}(S)\right)^2 \\ &\geq\; \sum_{i=1}^{m} \left(\widehat{f}(i) - \widehat{g}(i)\right)^2 \;=\; m \cdot \Omega(1/n) \;=\; \Omega(1). \end{aligned}$$

Here the second equality is by Parvseval's identity; the penultimate equality uses the fact that $\widehat{g}(i) \geq 0$ for all $i \in [n]$, which in turn holds since g is a monotone Boolean function. This completes the proof of Proposition 2.2.1.

2.2.2 Proof of Proposition 2.2.2

Let $\mathcal{T}$ be any deterministic non-adaptive q-query tester, and view its q queries as a $q \times n$ matrix $Q \in \{-1, 1\}^{q\times n}$. Following the terminology of [Blais and O'Donnell, 2010], we define a "Response Vector" random variable $\mathbf{R}_{yes} \in \{-1, 1\}^q$ which is obtained by drawing $\boldsymbol{f}_{yes} = \mathrm{sign}(\boldsymbol{\sigma}_1 x_1 + \cdots + \boldsymbol{\sigma}_n x_n)$ from $\mathcal{D}_{yes}$ and setting the i-th coordinate of $\mathbf{R}_{yes}$ to be

$$\boldsymbol{f}_{yes}(Q_{i*}) = \mathrm{sign}(\boldsymbol{\sigma}_1 Q_{i,1} + \cdots + \boldsymbol{\sigma}_n Q_{i,n}),$$

and similarly $\mathbf{R}_{no} \in \{-1, 1\}^q$ which is obtained by drawing $\boldsymbol{f}_{no} \sim \mathcal{D}_{no}$ and setting the i-th coordinate of $\mathbf{R}_{no}$ to be $\boldsymbol{f}_{no}(Q_{i*})$. By the definition of total variation distance, the left-hand

side of (2.1) is upper bounded by $d_{\mathrm{TV}}(\mathbf{R}_{yes}, \mathbf{R}_{no})$, and hence we can prove Proposition 2.2.2 by showing that $d_{\mathrm{TV}}(\mathbf{R}_{yes}, \mathbf{R}_{no}) = O(q^{5/4}(\log n)^{1/2}/n^{1/4})$.

Let $\mathbf{S} \in \mathbb{R}^q$ be the random column vector $Q\boldsymbol{\sigma}$ where $\boldsymbol{\sigma}$ is uniform over $\{1,3\}^n$, and $\mathbf{T} \in \mathbb{R}^q$ be the random column vector $Q\boldsymbol{\nu}$ where $\boldsymbol{\nu}$ is drawn from the product distribution over $\{-1, 7/3\}^n$ where $\mathbf{Pr}[\boldsymbol{\nu}_i = -1] = 1/10$ for all $i \in [n]$. The Response Vector $\mathbf{R}_{yes}$ is determined by the orthant of $\mathbb{R}^q$ in which $\mathbf{S}$ lies (as each coordinate of $\mathbf{R}_{yes}$ is simply the sign of the respective coordinate of $\mathbf{S}$), and likewise $\mathbf{R}_{no}$ by the orthant of $\mathbb{R}^q$ in which $\mathbf{T}$ lies. Therefore it suffices for us to prove the following lemma:

Lemma 2.2.3. *Let $\mathbf{S}, \mathbf{T} \in \mathbb{R}^q$ be defined as above. Then for any union $\mathcal{O}$ of orthants in $\mathbb{R}^q$,*

$$\left|\mathbf{Pr}[\mathbf{S} \in \mathcal{O}] - \mathbf{Pr}[\mathbf{T} \in \mathcal{O}]\right| = O\left(\frac{q^{5/4}(\log n)^{1/2}}{n^{1/4}}\right).$$

We will need the following multidimensional Berry–Esséen theorem, the proof of which we defer to Section 2.3.

Theorem 5. *Let $\mathbf{S} = \mathbf{X}^{(1)} + \cdots + \mathbf{X}^{(n)}$ where $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}$ are independent $\mathbb{R}^q$-valued random variables, and suppose that $\left|\mathbf{X}_i^{(j)} - \mathbf{E}[\mathbf{X}_i^{(j)}]\right| \le \tau$ with probability 1 for all $i \in [q]$ and $j \in [n]$. Let $\mathcal{G}$ be the q-dimensional Gaussian with the same mean and covariance matrix as $\mathbf{S}$. Let $\mathcal{O}$ be a union of orthants in $\mathbb{R}^q$. Then for all $r > 0$,*

$$\left|\mathbf{Pr}[\mathbf{S} \in \mathcal{O}] - \mathbf{Pr}[\mathcal{G} \in \mathcal{O}]\right| = O\left(\frac{\tau q^{3/2}\log n}{r} + \sum_{i=1}^{q} \frac{r+\tau}{\left(\sum_{j=1}^{n} \mathbf{Var}\left[\mathbf{X}_i^{(j)}\right]\right)^{1/2}}\right).$$

Proof of Lemma 2.2.3 assuming Theorem 5. We begin by writing $\mathbf{S} = \mathbf{X}^{(1)} + \cdots + \mathbf{X}^{(n)}$, where $\mathbf{X}^{(j)} = \boldsymbol{\sigma}_j \cdot Q_{*j}$ and $\boldsymbol{\sigma}_j$ is uniform over $\{1,3\}$; *i.e.* each $\mathbf{X}^{(j)}$ is independently Q_{*j} with probability $1/2$ and $3 \cdot Q_{*j}$ with probability $1/2$. Likewise we may express $\mathbf{T} = \mathbf{Y}^{(1)} + \cdots + \mathbf{Y}^{(n)}$, where $\mathbf{Y}^{(j)} = \boldsymbol{\nu}_j \cdot Q_{*j}$ and $\boldsymbol{\nu}_j$ is -1 with probability $1/10$ and $7/3$ with probability $9/10$. We claim that the $\mathbf{X}^{(j)}$'s and $\mathbf{Y}^{(j)}$'s have matching means and covariance matrices; it suffices to check this for $\mathbf{X}^{(1)}$ and $\mathbf{Y}^{(1)}$. For means, we see that indeed

$$\mathbf{E}\left[\mathbf{X}^{(1)}\right] = \mathbf{E}[\boldsymbol{\sigma}_1] \cdot Q_{*1} = (\tfrac{1}{2} + \tfrac{3}{2}) \cdot Q_{*1} = 2 \cdot Q_{*1}$$
$$\mathbf{E}\left[\mathbf{Y}^{(1)}\right] = \mathbf{E}[\boldsymbol{\nu}_1] \cdot Q_{*1} = (-\tfrac{1}{10} + \tfrac{9}{10}\tfrac{7}{3}) \cdot Q_{*1} = 2 \cdot Q_{*1}.$$

As for the covariance matrices, we let $i_1, i_2 \in [q]$ and calculate

$$\begin{aligned}
\mathbf{Cov}[\mathbf{X}^{(1)}]_{i_1,i_2} &= \mathbf{E}\left[\left(\mathbf{X}^{(1)}_{i_1} - 2\cdot Q_{i_1,1}\right)\left(\mathbf{X}^{(1)}_{i_2} - 2Q_{i_2,1}\right)\right] \\
&= \mathbf{E}\left[\mathbf{X}^{(1)}_{i_1}\cdot\mathbf{X}^{(1)}_{i_2}\right] - 2\cdot Q_{i_2,1}\,\mathbf{E}\left[\mathbf{X}^{(1)}_{i_1}\right] - 2\cdot Q_{i_1,1}\,\mathbf{E}\left[\mathbf{X}^{(1)}_{i_2}\right] + 4\cdot Q_{i_1,1}Q_{i_2,1} \\
&= \mathbf{E}\left[\mathbf{X}^{(1)}_{i_1}\cdot\mathbf{X}^{(1)}_{i_2}\right] - 4\cdot Q_{i_1,1}Q_{i_2,1} \\
&= \left(\mathbf{E}\left[\boldsymbol{\sigma}_1^2\right] - 4\right)\cdot Q_{i_1,1}Q_{i_2,1} \;=\; (\tfrac{1}{2} + \tfrac{9}{2} - 4)\cdot Q_{i_1,1}Q_{i_2,1} \;= Q_{i_1,1}Q_{i_2,1}.
\end{aligned}$$

Similarly, the corresponding entry of $\mathbf{Cov}[\mathbf{Y}^{(1)}]$ is:

$$\begin{aligned}
\mathbf{Cov}[\mathbf{Y}^{(1)}]_{i_1,i_2} &= \mathbf{E}\left[\left(\mathbf{Y}^{(1)}_{i_1} - 2\cdot Q_{i_1,1}\right)\left(\mathbf{Y}^{(1)}_{i_2} - 2Q_{i_2,1}\right)\right] \\
&= \mathbf{E}\left[\mathbf{Y}^{(1)}_{i_1}\cdot\mathbf{Y}^{(1)}_{i_2}\right] - 2\cdot Q_{i_2,1}\,\mathbf{E}\left[\mathbf{Y}^{(1)}_{i_1}\right] - 2\cdot Q_{i_1,1}\,\mathbf{E}\left[\mathbf{Y}^{(1)}_{i_2}\right] + 4\cdot Q_{i_1,1}Q_{i_2,1} \\
&= \mathbf{E}\left[\mathbf{Y}^{(1)}_{i_1}\cdot\mathbf{Y}^{(1)}_{i_2}\right] - 4\cdot Q_{i_1,1}Q_{i_2,1} \\
&= \left(\mathbf{E}\left[\boldsymbol{\nu}_1^2\right] - 4\right)\cdot Q_{i_1,1}Q_{i_2,1} \;=\; (\tfrac{1}{10} + \tfrac{9}{10}\tfrac{49}{9} - 4)\cdot Q_{i_1,1}Q_{i_2,1} \;= Q_{i_1,1}Q_{i_2,1}.
\end{aligned}$$

Since the $\mathbf{X}^{(j)}$'s and $\mathbf{Y}^{(j)}$'s have matching means and covariance matrices, so do their sums $\mathbf{S}$ and $\mathbf{T}$, and so Theorem 5 gives a bound on the differences $|\,\mathbf{Pr}[\mathbf{S}\in\mathcal{O}] - \mathbf{Pr}[\mathcal{G}\in\mathcal{O}]|$ and $|\,\mathbf{Pr}[\mathbf{T}\in\mathcal{O}] - \mathbf{Pr}[\mathcal{G}\in\mathcal{O}]|$ for the same q-dimensional Gaussian $\mathcal{G}$. Recalling that $\mathbf{X}^{(j)}_i = \boldsymbol{\sigma}_j\cdot Q_{i,j}$ where $Q_{i,j}\in\{-1,1\}^n$, we have that $\mathbf{Var}[\mathbf{X}^{(j)}_i] = 1$, and likewise $\mathbf{Var}[\mathbf{Y}^{(j)}_i] = 1$. Therefore, two applications of Theorem 5 with $\tau := O(1)$ along with the triangle inequality yields the bound

$$\left|\,\mathbf{Pr}[\mathcal{S}\in\mathcal{O}] - \mathbf{Pr}[\mathcal{G}\in\mathcal{O}]\right| = O\left(\frac{q^{3/2}\log n}{r} + \frac{q(r+\tau)}{\sqrt{n}}\right)$$

for all $r > 0$. Choosing $r := (qn)^{1/4}(\log n)^{1/2}$ completes the proof. □

2.3 Multidimensional Berry–Esséen via the Valiant–Valiant CLT

In this section we prove Theorem 5 by adapting a recent multidimensional CLT of Valiant and Valiant [Valiant and Valiant, 2011] which bounds the *Wasserstein distance* between a sum of independent vector-valued random variables and a multidimensional Gaussian.

Definition 6 (Wasserstein distance)**.** The Wasserstein distance between two $\mathbb{R}^q$-valued random variables $\mathbf{S}$ and $\mathbf{T}$, denoted $d_W(\mathbf{S}, \mathbf{T})$, is defined to be:

$$d_W(\mathbf{S}, \mathbf{T}) = \inf_{\mathcal{D}} \left\{ \mathop{\mathbf{E}}_{\mathcal{D}} \left[\|\mathbf{U} - \mathbf{V}\|_2 \right] \right\},$$

where the infimum is taken over all couplings $\mathcal{D}$ of $\mathbf{S}$ and $\mathbf{T}$, i.e., all joint distributions $\mathcal{D}$ of pairs of $\mathbb{R}^q$-valued random variables $(\mathbf{U}, \mathbf{V})$ with marginals distributed according to $\mathbf{S}$ and $\mathbf{T}$ respectively.

Valiant and Valiant [Valiant and Valiant, 2011] recently used Stein's method to prove the following central limit theorem for Wasserstein distance:

Theorem 7 (Valiant–Valiant CLT)**.** *Let $\mathbf{S} = \mathbf{X}^{(1)} + \cdots + \mathbf{X}^{(n)}$ where $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}$ are independent $\mathbb{R}^q$-valued random variables, and suppose $\left\|\mathbf{X}^{(j)} - \mathbf{E}\left[\mathbf{X}^{(j)}\right]\right\|_2 \le \beta$ with probability 1 for any $j \in [n]$. Then*

$$d_W(\mathbf{S}, \mathcal{G}) \le O(\beta q \log n),$$

where $\mathcal{G}$ is the q-dimensional Gaussian with the same mean and covariance matrix as $\mathbf{S}$.

We recall Theorem 5:

Theorem 5. *Let $\mathbf{S} = \mathbf{X}^{(1)} + \cdots + \mathbf{X}^{(n)}$ where $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(n)}$ are independent $\mathbb{R}^q$-valued random variables, and suppose that $\left|\mathbf{X}_i^{(j)} - \mathbf{E}\left[\mathbf{X}_i^{(j)}\right]\right| \le \tau$ with probability 1 for all $i \in [q]$ and $j \in [n]$. Let $\mathcal{G}$ be the q-dimensional Gaussian with the same mean and covariance matrix as $\mathbf{S}$. Let $\mathcal{O}$ be a union of orthants in $\mathbb{R}^q$. Then for all $r > 0$,*

$$\left|\mathbf{Pr}[\mathbf{S} \in \mathcal{O}] - \mathbf{Pr}[\mathcal{G} \in \mathcal{O}]\right| = O\left(\frac{\tau q^{3/2} \log n}{r} + \sum_{i=1}^{q} \frac{r + \tau}{\left(\sum_{j=1}^{n} \mathbf{Var}\left[\mathbf{X}_i^{(j)}\right]\right)^{1/2}}\right).$$

Proof. We define

$$W_r := \left\{x \in \mathbb{R}^q \colon |x_i| \le r \text{ for some } i \in [q]\right\}$$

to be the radius-r region around the orthant boundaries, and partition $\mathcal{O}$ into $\mathcal{O}_{bd} := \mathcal{O} \cap W_r$ (the points in $\mathcal{O}$ that lie close to the orthant boundaries) and $\mathcal{O}_{in} := \mathcal{O} \setminus W_r$ (the points that lie far away from the orthant boundaries). We have

$$\begin{aligned} \left|\mathbf{Pr}[\mathbf{S} \in \mathcal{O}] - \mathbf{Pr}[\mathcal{G} \in \mathcal{O}]\right| &= \left|(\mathbf{Pr}[\mathbf{S} \in \mathcal{O}_{in}] + \mathbf{Pr}[\mathbf{S} \in \mathcal{O}_{bd}]) - (\mathbf{Pr}[\mathcal{G} \in \mathcal{O}_{in}] + \mathbf{Pr}[\mathcal{G} \in \mathcal{O}_{bd}])\right| \\ &\le \underbrace{\left|\mathbf{Pr}[\mathbf{S} \in \mathcal{O}_{in}] - \mathbf{Pr}[\mathcal{G} \in \mathcal{O}_{in}]\right|}_{\Delta} + \underbrace{\mathbf{Pr}[\mathbf{S} \in \mathcal{O}_{bd}] + \mathbf{Pr}[\mathcal{G} \in \mathcal{O}_{bd}]}_{\Gamma} \end{aligned} \quad (2.2)$$

We bound the quantities Δ and Γ separately. For Γ, we have that

$$\begin{aligned}\Gamma &\leq \sum_{i=1}^{q} \mathbf{Pr}\left[\mathbf{S}_i \in [-r,r]\right] + \mathbf{Pr}\left[\mathcal{G}_i \in [-r,r]\right] \qquad (2.3)\\ &\leq \sum_{i=1}^{q} 2\,\mathbf{Pr}\left[\mathcal{G}_i \in [-r,r]\right] + \left|\mathbf{Pr}\left[\mathbf{S}_i \in [-r,r]\right] - \mathbf{Pr}\left[\mathcal{G}_i \in [-r,r]\right]\right| \\ &\leq \sum_{i=1}^{q} \frac{O(r)}{\left(\sum_{j=1}^{n} \mathbf{Var}\left[\mathbf{X}_i^{(j)}\right]\right)^{1/2}} + \frac{O(\tau)}{\left(\sum_{j=1}^{n} \mathbf{Var}\left[\mathbf{X}_i^{(j)}\right]\right)^{1/2}} = \sum_{i=1}^{q} \frac{O(r+\tau)}{\left(\sum_{j=1}^{n} \mathbf{Var}\left[\mathbf{X}_i^{(j)}\right]\right)^{1/2}}\end{aligned}$$

where (2.3) is a union bound over all q dimensions, and the final inequality uses Fact 2.1.1 (Gaussian anti-concentration), the fact that $\mathcal{G}_i$ is a Gaussian with variance $\sum_{j=1}^{n} \mathbf{Var}\left[\mathbf{X}_i^{(j)}\right]$, and Theorem 4 (Berry–Esséen).

For Δ, let us assume without loss of generality (a symmetrical argument works in the other case) that $\mathbf{Pr}[\mathbf{S} \in \mathcal{O}_{in}] \geq \mathbf{Pr}[\mathcal{G} \in \mathcal{O}_{in}]$, so $\Delta = \mathbf{Pr}[\mathbf{S} \in \mathcal{O}_{in}] - \mathbf{Pr}[\mathcal{G} \in \mathcal{O}_{in}]$. Let $\mathcal{D}$ be any coupling of $\mathbf{S}$ and $\mathcal{G}$, so $\mathcal{D}$ is the joint distribution of a pair $(\mathbf{U}, \mathbf{V})$ of $\mathbb{R}^q$-valued random variables with marginals distributed according to $\mathbf{S}$ and $\mathcal{G}$ respectively. Since

$$\int_{\mathcal{O}_{in}} \int_{\mathbb{R}^q} \mathcal{D}(u,v)\,dv\,du = \mathbf{Pr}[\mathbf{S} \in \mathcal{O}_{in}]$$

and

$$\int_{\mathcal{O}_{in}} \int_{\mathcal{O}_{in}} \mathcal{D}(u,v)\,dv\,du \leq \int_{\mathbb{R}^q} \int_{\mathcal{O}_{in}} \mathcal{D}(u,v)\,dv\,du = \mathbf{Pr}[\mathcal{G} \in \mathcal{O}_{in}],$$

it follows that

$$\int_{\mathcal{O}_{in}} \int_{\mathbb{R}^q \setminus \mathcal{O}_{in}} \mathcal{D}(u,v)\,dv\,du = \int_{\mathcal{O}_{in}} \int_{\mathbb{R}^q} \mathcal{D}(u,v)\,dv\,du - \int_{\mathcal{O}_{in}} \int_{\mathcal{O}_{in}} \mathcal{D}(u,v)\,dv\,du \geq \Delta. \qquad (2.4)$$

Next we define the quantities

$$\begin{aligned}\Delta_{near}(\mathcal{D}) &:= \int_{\mathcal{O}_{in}} \int_{\mathcal{O}_{bd}} \mathcal{D}(u,v)\,dv\,du \\ \Delta_{far}(\mathcal{D}) &:= \int_{\mathcal{O}_{in}} \int_{\mathbb{R}^q \setminus \mathcal{O}} \mathcal{D}(u,v)\,dv\,du.\end{aligned}$$

Note that $\Delta_{near}(\mathcal{D})$ and $\Delta_{far}(\mathcal{D})$ sum to the quantity on the left-hand side of (2.4), and so $\Delta_{near}(\mathcal{D}) + \Delta_{far}(\mathcal{D}) \geq \Delta$. (In words, since $\mathbf{S}$ places Δ more mass on $\mathcal{O}_{in}$ than $\mathcal{G}$ does, any scheme $\mathcal{D}$ of moving the mass of $\mathbf{S}$ to obtain $\mathcal{G}$ must move at least Δ amount from within $\mathcal{O}_{in}$ to outside it. $\Delta_{near}(\mathcal{D})$ is the amount moved from within $\mathcal{O}_{in}$ to $\mathcal{O}$'s boundary

$\mathcal{O}_{bd}$, and $\Delta_{far}(\mathcal{D})$ is the rest, moved from within $\mathcal{O}_{in}$ to locations entirely out of $\mathcal{O}$.) Since $\|u - v\|_2 \geq r$ for any pair of points $u \in \mathcal{O}_{in}$ and $y \notin \mathcal{O}$, it follows that

$$d_W(\mathbf{S}, \mathcal{G}) \geq r \cdot \Delta_{far}(\mathcal{D}).$$

We consider two cases, depending on the relative magnitudes of $\Delta_{near}(\mathcal{D})$ and $\Delta_{far}(\mathcal{D})$. If $\Delta_{far}(\mathcal{D}) \geq \Delta_{near}(\mathcal{D})$, we first observe that for all $j \in [n]$ we have $\left\|\mathbf{X}^{(j)} - \mathbf{E}\left[\mathbf{X}^{(j)}\right]\right\|_2 \leq \tau\sqrt{q}$ with probability 1, since each of its q coordinates i satisfies $\left|\mathbf{X}_i^{(j)} - \mathbf{E}\left[\mathbf{X}_i^{(j)}\right]\right| \leq \tau$ with probability 1 by the assumption of the theorem. Therefore we may apply Theorem 7 (Valiant–Valiant CLT) with $\beta := \tau\sqrt{q}$ to get

$$r \cdot \frac{\Delta}{2} \leq r \cdot \Delta_{far}(\mathcal{D}) \leq d_W(\mathbf{S}, \mathcal{G}) = O(\tau q^{3/2} \log n)$$

and hence $\Delta = O((\tau q^{3/2} \log n)/r)$, which along with our upper bound on Γ completes the proof. If on the other hand $\Delta_{near}(\mathcal{D}) > \Delta_{far}(\mathcal{D})$, then

$$\frac{\Delta}{2} \leq \Delta_{near}(\mathcal{D}) \leq \int_{\mathbb{R}^q} \int_{\mathcal{O}_{bd}} \mathcal{D}(u, v)\, dv\, du = \mathbf{Pr}[\mathcal{G} \in \mathcal{O}_{bd}] \leq \Gamma,$$

and again our bound on Γ completes the proof. □

2.4 General hypergrid domains

In this section we prove Theorem 2, showing that for all $m \in \mathbb{N}$ essentially the same lower bound of $\tilde{\Omega}(n^{1/5})$ also applies to the query complexity of testers for monotonicity of functions $f : [m]^n \to \{-1, 1\}$, Boolean-valued functions over general hypergrid domains. The notions of monotonicity and distance to monotonicity of functions generalize to functions $f : [m]^n \to \{-1, 1\}$ the natural way: f is monotone if $f(x) \leq f(y)$ for all $x \prec y$, where $x \prec y$ iff $x_i \leq y_i$ for all $i \in [n]$ and $x \neq y$. We say that f is ε-close to monotone if $\mathbf{Pr}_{x \in [m]^n}[f(x) \neq g(x)] \leq \varepsilon$ for some monotone $g : [m]^n \to \{-1, 1\}$, and ε-far from monotone otherwise.

We prove Theorem 2 via a reduction to the $m = 2$ case (i.e. Theorem 1). The reduction is simpler for even m we first prove it assuming that m is even. In this case Theorem 2 is a direct consequence of Theorem 1 and the following proposition:

Proposition 2.4.1. *For all even $m \in \mathbb{N}$ the mapping*

$$\Phi : \left\{\textit{all functions } f : \{-1, 1\}^n \to \{-1, 1\}\right\} \to \left\{\textit{all functions } f : [m]^n \to \{-1, 1\}\right\}$$

defined by (2.5) below satisfies the following two properties:

1. *If $f : \{-1,1\}^n \to \{-1,1\}$ is monotone then $\Phi[f]$ is monotone as well.*
2. *If $f : \{-1,1\}^n \to \{-1,1\}$ is ε-far from monotone then $\Phi[f]$ is ε-far from monotone as well.*

We will need the following characterization of distance to monotonicity.

Theorem 8 ([Fischer *et al.*, 2002] Lemma 4)**.** *For all $f : [m]^n \to \{-1,1\}$ and $\varepsilon > 0$, we have that f is ε-far from monotone if and only if there exists εm^n many pairwise disjoint ordered pairs of vertices $(x^i, y^i) \in [m]^n \times [m]^n$ such that $x^i \prec y^i$ and $f(x^i) > f(y^i)$. We will call each such pair a* violation with respect to f.

Proof of Proposition 2.4.1. For every $f : \{-1,1\}^n \to \{-1,1\}$, we define $\Phi[f] : [m]^n \to \{-1,1\}$ to be the function

$$\Phi[f](x_1, \ldots, x_n) := f\big(\mathbf{1}[x_1 > m/2], \ldots, \mathbf{1}[x_n > m/2]\big), \tag{2.5}$$

where we use $\mathbf{1}[\cdot]$ to denote the $\{\pm 1\}$-valued indicator where $\mathbf{1}[P] = 1$ if P is true, and -1 otherwise. (Note that $m/2$ is an integer by our assumption that m is even.)

It is straightforward to verify that $\Phi[f]$ is monotone if f is monotone, and so it remains to show that $\Phi[f]$ is ε-far from monotone if f is ε-far from monotone. Since f is ε-far from monotone, we have by Theorem 8 that there exist $\varepsilon 2^n$ many pairwise disjoint pairs $(x^i, y^i) \in \{-1,1\}^n \times \{-1,1\}^n$ that are violations with respect to f; we will exhibit εm^n many pairwise disjoint pairs in $[m]^n$ that are violations with respect to $\Phi[f]$, which along with another application of Theorem 8 completes the proof. Let $S : \{-1,1\} \to \big\{[m/2], \{(m/2)+1, \ldots, m\}\big\}$ be the set-valued function

$$S(b) = \begin{cases} [m/2] & \text{if } b = -1 \\ \{(m/2)+1, \ldots, m\} & \text{if } b = 1, \end{cases}$$

and by a slight abuse of notation, we also define

$$S(x) = S(x_1) \times \cdots \times S(x_n) \subseteq [m]^n$$

to be a function that maps points $x \in \{-1,1\}^n$ to subsets of $[m]^n$. Note that $|S(x)| = (m/2)^n$ for all $x \in \{-1,1\}^n$, and $S(x) \cap S(y) = \emptyset$ if $x \neq y$. Furthermore, $\Phi[f](x') = f(x)$

for all $x \in \{-1,1\}^n$ and $x' \in S(x)$. In words, S maps each 1-input of f to a set of $(m/2)^n$ many 1-inputs of $\Phi[f]$, and likewise each 0-input of f to a set of $(m/2)^n$ many 0-inputs of $\Phi[f]$.

For any pair $(x,y) \in \{-1,1\}^n \times \{-1,1\}^n$ that is a violation with respect to f, consider pairing the $(m/2)^n$ elements of $S(x)$ with the $(m/2)^n$ elements of $S(y)$ in the obvious way (i.e. each $a = (a_1,\ldots,a_n) \in S(x)$ is is paired with the unique element $b = (b_1,\ldots,b_n) \in S(y)$ that has $(a_i \mod m/2) = (b_i \mod m/2)$ for all i). Since $x \prec y$, it follows from the definition of S that every $x' \in S(x)$ is paired with $y' \in S(y)$ where $x' \prec y'$. Furthermore, as noted above $\Phi[f](x') = f(x) = 1$ whereas $\Phi[f](y') = f(y) = 0$, and so every pair $(x',y') \in S(x) \times S(y)$ is a violation with respect to $\Phi[f]$. Therefore each of the $\varepsilon 2^n$ many pairs $(x,y) \in \{-1,1\}^n \times \{-1,1\}^n$ that are violations with respect to f gives rise to $(m/2)^n$ many pairwise disjoint pairs $(x',y') \in S(x) \times S(y)$ that are violations with respect to $\Phi[f]$. Finally recalling that $S(x) \cap S(y) = \emptyset$ if $x \neq y$, we conclude that there are indeed $\varepsilon 2^n \cdot (m/2)^n = \varepsilon m^n$ many pairwise disjoint pairs that are violations with respect to $\Phi[f]$. This finishes the proof. □

2.4.1 Reduction from hypergrid domains $[m]^n$ when m is odd

The proof for odd m is via a similar but more involved version of Proposition 2.4.1. In place of the simple indicator function $\mathbf{1}[x_i > m/2]$ (whose domain is simply $[m]$), we now use an "almost-balanced" monotone function $h : [m]^k \to \{-1,1\}$ where $k = \Theta(\log n)$ and h has some additional properties. The fact that $k = \Theta(\log n)$ incurs an additional logarithmic loss in the parameters but still results in a $\tilde{\Omega}(n^{1/5})$ lower bound.

Lemma 2.4.2. *Let $m \in \mathbb{N}$ be odd. There exists a monotone function $h : [m]^k \to \{-1,1\}$ such that $|\{x \in [m]^k : h(x) = 1\}| = |\{x \in [m]^k : h(x) = -1\}| + 1$, and a one-to-one mapping $\Psi : h^{-1}(-1) \to h^{-1}(1)$ such that $\Psi(x) \succ x$ for all $x \in h^{-1}(-1)$.*

Proof. The function h is defined as follows:

$$h(x_1,\ldots,x_k) = \begin{cases} 1 & \text{if } x = \lceil m/2 \rceil^k \\ \operatorname{sign}(x_i - \lceil m/2 \rceil) & \text{otherwise, where } i := \min\{i \in [k] : x_i \neq \lceil m/2 \rceil\}. \end{cases}$$

The monotonicity of h is straightforward to verify, as is the fact that $|\{x \in [m]^k \colon h(x) = 1\}| = |\{x \in [m]^k \colon h(x) = -1\}| + 1$. The proof is complete by noticing that the mapping

$$\Psi(x_1, \ldots, x_k) = (x_1, \ldots, x_{i-1}, -x_i, x_{i+1}, \ldots, x_k), \quad \text{where } i := \min\{i \in [k] \colon x_i \neq \lceil m/2 \rceil\}$$

is a bijection between $h^{-1}(-1)$ and $h^{-1}(1) \setminus \{\lceil m/2 \rceil^k\}$. □

With Lemma 2.4.2 in hand we are ready to prove the following analogue of Proposition 2.4.1 for hypergrid domains $[m]^n$ when m is odd. Given the monotone function h defined in Lemma 2.4.2, let $h' : [m]^k \to \{-1, 1, \perp\}$ be the partial function where $h'(x) = \perp$ if $x = \lceil m/2 \rceil^k$, and $h'(x) = h(x)$ otherwise (and so $|\{x \in [m]^k \colon h(x) = 1\}| = |\{x \in [m]^k \colon h(x) = -1\}| = (m^k - 1)/2$).

Proposition 2.4.3. *For all odd $m \in \mathbb{N}$ the mapping*

$$\Phi : \big\{\text{all functions } f : \{-1,1\}^n \to \{-1,1\}\big\} \to \big\{\text{all functions } f : [m]^{n\lceil \log n \rceil} \to \{-1,1\}\big\}$$

defined by (2.6) below satisfies the following two properties:

1. *If $f : \{-1,1\}^n \to \{-1,1\}$ is monotone then $\Phi[f]$ is monotone as well.*
2. *If $f : \{-1,1\}^n \to \{-1,1\}$ is ε-far from monotone then $\Phi[f]$ is $\Omega(\varepsilon)$-far from monotone.*

Proof. Fix $k := \lceil \log n \rceil$. For every $f : \{-1,1\}^n \to \{-1,1\}$, we define $\Phi[f] : [m]^{kn} \to \{-1,1\}$ to be the following function: for all $x^1, \ldots, x^n \in [m]^k$

$$\Phi[f](x^1, \ldots, x^n) := f\big(h(x^1), \ldots, h(x^n)\big). \tag{2.6}$$

Since h is monotone it follows that $\Phi[f]$ is monotone if f is monotone, and so it remains to show that $\Phi[f]$ is $\Omega(\varepsilon)$-far from monotone if f is ε-far from monotone. Since f is ε-far from monotone, we have by Theorem 8 that there exist $\varepsilon 2^n$ many pairwise disjoint pairs $(x^i, y^i) \in \{-1,1\}^n \times \{-1,1\}^n$ that are violations with respect to f; we will exhibit $\Omega(\varepsilon\, m^{kn})$ many pairwise disjoint pairs in $[m]^{kn}$ that are violations with respect to $\Phi[f]$, which along with another application of Theorem 8 completes the proof.

Using the same notation as in the proof of Proposition 2.4.1, we define the set-valued function S mapping $x \in \{-1,1\}^n$ to subsets of $[m]^{kn}$ as follows:

$$S(x) = h'^{-1}(x_1) \times \cdots \times h'^{-1}(x_n) \subseteq [m]^{kn}.$$

Note that

$$|S(x)| = \left(\frac{m^k-1}{2}\right)^n = \frac{m^{kn}}{2^n}\left(1-\frac{1}{m^k}\right)^n \geq \frac{m^{kn}}{2^n}\left(1-\frac{1}{n^{\log m}}\right)^n = \Omega\left(\frac{m^{kn}}{2^n}\right)$$

for all $x \in \{-1,1\}^n$ (where we have used our choice of $k = \lceil \log n \rceil$ for the inequality), and $S(x) \cap S(y) = \emptyset$ if $x \neq y$. Furthermore, $\Phi[f](x') = f(x)$ for all $x \in \{-1,1\}^n$ and $x' \in S(x)$. In words, S maps each 1-input of f to a set of $((m^k-1)/2)^n$ many 1-inputs of $\Phi[f]$, and likewise each 0-input of f to a set of $((m^k-1)/2)^n$ many 0-inputs of $\Phi[f]$.

For any pair $(x,y) \in \{-1,1\}^n \times \{-1,1\}^n$, $x \prec y$, that is a violation with respect to f, consider pairing the $((m^k-1)/2)^n$ elements of $S(x)$ with the $((m^k-1)/2)^n$ elements of $S(y)$ via Ψ from Lemma 2.4.2 as follows: each $a \in S(x)$, which we will view as $a = (a_1,\ldots,a_n) \in ([m]^k)^n$, is paired with the unique element $b = (b_1,\ldots,b_n) \in S(y)$ where $b_i = a_i$ if $x_i = y_i$, and $b_i = \Psi(a_i)$ if $x_i < y_i$. Since $x \prec y$, it follows from the definitions of S and Ψ that every $x' \in S(x)$ is paired with $y' \in S(y)$ where $x' \prec y'$. Furthermore, as noted above $\Phi[f](x') = f(x) = 1$ whereas $\Phi[f](y') = f(y) = 0$, and so every pair $(x',y') \in S(x) \times S(y)$ is a violation with respect to $\Phi[f]$. Therefore each of the $\varepsilon 2^n$ many pairs $(x,y) \in \{-1,1\}^n \times \{-1,1\}^n$ that are violations with respect to f gives rise to $\Omega(m^{kn}/2^n)$ many pairwise disjoint pairs $(x',y') \in S(x) \times S(y)$ that are violations with respect to $\Phi[f]$. Finally recalling that $S(x) \cap S(y) = \emptyset$ if $x \neq y$, we conclude that there are indeed $\varepsilon 2^n \cdot \Omega(m^{kn}/2^n) = \Omega(\varepsilon\, m^{kn})$ many pairwise disjoint pairs that are violations with respect to $\Phi[f]$. This finishes the proof. □

Proposition 2.4.3 along with Theorem 1 implies the existence of a universal constant $\varepsilon_0 > 0$ such that any non-adaptive ε_0-tester for the monotonicity of $f : [m]^N \to \{-1,1\}$, where $N := n\lceil \log n \rceil$ and m is odd, must make $\tilde{\Omega}(n^{1/5}) = \tilde{\Omega}(N^{1/5})$ many queries. This along with Proposition 2.4.1 (establishing the same lower bound for hypergrid domains $[m]^n$ where m is even) completes the proof of Theorem 2.

2.5 An improved algorithm for testing monotonicity

Throughout the proof of our upper bound we will assume that $1/n \le \varepsilon \le 1/2$. Note that this is without loss of generality, since if $\varepsilon < 1/n$ then the edge tester alone succeeds with probability $\Omega(\varepsilon/n) = \Omega(\varepsilon^2)$, and if $\varepsilon > 1/2$ then every f is ε-close to one of the two constant functions, both of which are monotone.

For our upper bound it will be more convenient to view Boolean functions as mapping $\{0,1\}^n$ to $\{0,1\}$. For $x, y \in \{0,1\}^n$ we write $\|x\|_1$ to denote $\sum_{i=1}^n x_i$, the number of 1s in x, and $\|x-y\|_1$ to denote $|\{i \in [n] \colon x_i \neq y_i\}|$, the ℓ_1-distance between x and y. Given $1/n \le \varepsilon \le 1/2$, we fix

$$d(n,\varepsilon) := 2\left\lceil \sqrt{2n\ln(100/\varepsilon)} \right\rceil = O\big(\sqrt{n\ln(1/\varepsilon)}\big),$$

and will denote $d(n,\varepsilon)$ simply by d when the distance parameter ε is clear from the context. For each $i \in \{0,1,\dots,n\}$ we let $L_i := \{x \in \{0,1\}^n \colon \|x\|_1 = i\}$ denote the i-th layer, and refer to

$$L_{\mathrm{mid}} := \big\{x \in L_i \colon i \in [(n-d)/2, (n+d)/2]\big\}$$

as the middle layers of the hypercube. A standard Chernoff bound gives $|\{0,1\}^n \setminus L_{\mathrm{mid}}| \le (\varepsilon/50)2^n$. Finally, by a "path" we always mean a directed path of $n+1$ adjacent vertices from 0^n up to 1^n.

2.5.1 Two useful distributions over comparable pairs

Let $\mathcal{D} = \mathcal{D}_{n,\varepsilon}$ denote the following distribution over comparable pairs $(\boldsymbol{x}, \boldsymbol{y}) \in L_{\mathrm{mid}} \times L_{\mathrm{mid}}$:

1. First pick a path $\boldsymbol{p}$ uniformly from the collection of all paths going from 0^n to 1^n.
2. Pick $\boldsymbol{x}$ and $\boldsymbol{y}$ independently and uniformly from $\boldsymbol{p}_{\mathrm{mid}} := \{z \in \boldsymbol{p} \colon z \in L_{\mathrm{mid}}\}$.

This distribution is a slight variant of the one induced by the [Chakrabarty and Seshadhri, 2013a] path tester, which takes a parameter σ as input and disallows pairs (x,y) for which $\|x-y\|_1$ is too small relative to σ. Our new tester will *not* sample from $\mathcal{D}$ (see Section 2.5.3), but we will use $\mathcal{D}$ in our analysis. We remark here that $\boldsymbol{x} = \boldsymbol{y}$ with positive probability under $\mathcal{D}$.

If $\boldsymbol{x}, \boldsymbol{y}$ were chosen independently and uniformly from $\{0,1\}^n$, then the probability that they both land in a fixed set A of $\sigma 2^n$ points, for some $\sigma \in (0,1)$, would be σ^2. The following lemma states that the probability is not much lower for a pair drawn from $\mathcal{D}$:

Lemma 2.5.1. *Let $A \subseteq L_{\text{mid}}$ be a set of $\sigma 2^n$ points. Then $\mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}}[\boldsymbol{x},\boldsymbol{y} \in A] = \Omega\left(\sigma^2 \ln^{-1}(1/\varepsilon)\right)$.*

Proof. Applying Jensen's inequality, we have

$$\begin{aligned}\Pr_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}}[\boldsymbol{x},\boldsymbol{y} \in A] &= \mathop{\mathbf{E}}_{\boldsymbol{p}}\left[\Pr_{\boldsymbol{x},\boldsymbol{y}\in\boldsymbol{p}}[\boldsymbol{x},\boldsymbol{y} \in A]\right] \\ &= \mathop{\mathbf{E}}_{\boldsymbol{p}}\left[\left(\frac{|\boldsymbol{p}_{\text{mid}} \cap A|}{|\boldsymbol{p}_{\text{mid}}|}\right)^2\right] = \Omega\left(\frac{1}{n\ln(1/\varepsilon)}\right) \cdot \mathop{\mathbf{E}}_{\boldsymbol{p}}\left[|\boldsymbol{p}_{\text{mid}} \cap A|\right]^2,\end{aligned}$$

and so it suffices to lower bound $\mathbf{E}_{\boldsymbol{p}}[|\boldsymbol{p}_{\text{mid}} \cap A|]$ by $\Omega(\sigma\sqrt{n})$. This is exactly Claim 2.2.1 of [Chakrabarty and Seshadhri, 2013a]; we repeat the calculation here for the sake of completeness:

$$\begin{aligned}\mathop{\mathbf{E}}_{\boldsymbol{p}}\left[|\boldsymbol{p}_{\text{mid}} \cap A|\right] &= \mathop{\mathbf{E}}_{\boldsymbol{p}}\left[\sum_{i=\frac{1}{2}(n-d)}^{\frac{1}{2}(n+d)} \mathbf{1}\left[(\boldsymbol{p}_{\text{mid}} \cap L_i) \subseteq A\right]\right] \\ &= \sum_{i=\frac{1}{2}(n-d)}^{\frac{1}{2}(n+d)} \mathop{\mathbf{E}}_{\boldsymbol{p}}\left[\mathbf{1}\left[(\boldsymbol{p}_{\text{mid}} \cap L_i) \subseteq A\right]\right] \\ &= \sum_{i=\frac{1}{2}(n-d)}^{\frac{1}{2}(n+d)} \frac{|A \cap L_i|}{|L_i|} \qquad (2.7) \\ &\geq \frac{\sqrt{n}}{2^n} \sum_{i=\frac{1}{2}(n-d)}^{\frac{1}{2}(n+d)} |A \cap L_i| = \frac{|A|\sqrt{n}}{2^n} = \sigma\sqrt{n}, \qquad (2.8)\end{aligned}$$

where we use $\mathbf{1}[\cdot]$ to denote the $\{0,1\}$-valued indicator where $\mathbf{1}[P] = 1$ if P is true, and 0 otherwise. Here (2.7) uses the fact that a uniformly random path $\boldsymbol{p}$ from 0^n to 1^n contains a uniformly random point in layer L_i, and (2.8) holds since $|L_i| \leq 2^n/\sqrt{n}$ for all i. □

We will need a numerical lemma concerning the ratio of binomial coefficients.

Lemma 2.5.2. *Let $\varepsilon \geq 1/n$, and $a, b \in [(n-d)/2, (n+d)/2]$ be integers where $a > b$. Then*

$$\binom{a}{a-b}\Big/\binom{n-b}{a-b} = O(1/\varepsilon^4) \quad \text{and} \quad \binom{n}{n/2}\Big/\binom{n}{a} = O(1/\varepsilon^4).$$

Proof. We prove the first equation and the second equation is similar. By a routine calculation we verify that the first ratio is maximized when $a = (n+d)/2$ and $b = n/2$, and so

$$\frac{\binom{a}{a-b}}{\binom{n-b}{a-b}} \leq \frac{\frac{1}{2}(n+d)}{\frac{n}{2}} \cdot \frac{\frac{1}{2}(n+d)-1}{\frac{n}{2}-1} \cdots \frac{\frac{n}{2}+1}{\frac{1}{2}(n-d)+1} \leq \exp\left(\sum_{i=0}^{(d/2)-1} \frac{d/2}{(n/2)-i}\right),$$

where we used $(1+t) \leq e^t$ for $t \in \mathbb{R}$. The lemma follows from the definition of d and $\varepsilon \geq 1/n$. □

For our analysis, the following distribution $\mathcal{D}' = \mathcal{D}'_{n,\varepsilon}$ over comparable pairs $(\boldsymbol{x}, \boldsymbol{y}) \in L_{\text{mid}} \times L_{\text{mid}}$ in the middle layers comes in handy:

1. First pick a point $\boldsymbol{x}$ uniformly at random from L_{mid}.
2. Then pick a path $\boldsymbol{p}$ uniformly from the collection of all paths going through 0^n, $\boldsymbol{x}$, and 1^n.
3. Pick $\boldsymbol{y}$ uniformly from $\boldsymbol{p}_{\text{mid}} := \{z \in \boldsymbol{p} \colon z \in L_{\text{mid}}\}$.

We note that $\mathcal{D}'$ is not the same as $\mathcal{D}$, since picking a uniformly random $\boldsymbol{x}$ from the middle layers of a uniformly random path $\boldsymbol{p}$ does not induce a uniform distribution over L_{mid}; however, Lemma 2.5.2 allows us to switch between these essentially-equivalent distributions at the cost of a $O(1/\varepsilon^4)$ factor. (On the other hand the conditional distributions $\mathcal{D}_{\boldsymbol{x}=x}$ and $\mathcal{D}'_{\boldsymbol{x}=x}$ on $\boldsymbol{y}$ are the same for all possible outcomes $x \in L_{\text{mid}}$ of $\boldsymbol{x}$.)

We get the following corollary from Lemmas 2.5.1 and 2.5.2:

Corollary 2.5.3. *Let $A \subseteq L_{\text{mid}}$ be a set of $\sigma 2^n$ points. Then*

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y}) \leftarrow \mathcal{D}'}[\boldsymbol{x}, \boldsymbol{y} \in A] = \Omega\left(\sigma^2 \varepsilon^4 \ln^{-1}(1/\varepsilon)\right).$$

Proof. It is clear from the definition of $\mathcal{D}, \mathcal{D}'$ that the conditional distribution of $\boldsymbol{y}$ induced from $\mathcal{D}$ by conditioning on a particular outcome of $\boldsymbol{x}$ is the same as that induced from $\mathcal{D}'$ under the same conditioning. It follows from the second part of Lemma 2.5.2 that for any $x \in L_{\text{mid}}$ we have

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y}) \leftarrow \mathcal{D}'}[\boldsymbol{x} = x] = \Omega(\varepsilon^4) \cdot \mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y}) \leftarrow \mathcal{D}}[\boldsymbol{x} = x].$$

As a result, we have for every comparable pair (x, y) in the middle layers

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}'}[(\boldsymbol{x},\boldsymbol{y}) = (x, y)] = \Omega(\varepsilon^4) \cdot \mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}}[(\boldsymbol{x},\boldsymbol{y}) = (x, y)].$$

The claim then follows from Lemma 2.5.1. □

2.5.2 Density and score

We need the following definition to give a more detailed analysis on the consequence of Corollary 2.5.3, which is key to the analysis of our monotonicity tester described in Section 2.5.3.

Definition 9 (density and score)**.** Let $A \subseteq \{0,1\}^n$. For all $x \in \{0,1\}^n$ and $k \in \{0, 1, \ldots, n\}$, we define the following quantities:

$$\mathrm{dens}_k^{\downarrow}(x, A) := \mathop{\mathbf{Pr}}_{\substack{\boldsymbol{y}\preceq x \\ \|\boldsymbol{y}-x\|_1=k}}[\boldsymbol{y} \in A] \text{ if } k \le \|x\|_1, \text{ and } \mathrm{dens}_k^{\downarrow}(x, A) := 0 \text{ otherwise,}$$

and similarly

$$\mathrm{dens}_k^{\uparrow}(x, A) := \mathop{\mathbf{Pr}}_{\substack{\boldsymbol{y}\succeq x \\ \|\boldsymbol{y}-x\|_1=k}}[\boldsymbol{y} \in A] \text{ if } k \le n - \|x\|_1, \text{ and } \mathrm{dens}_k^{\uparrow}(x, A) := 0 \text{ otherwise.}$$

We also define

$$\mathrm{score}^{\downarrow}(x, A) := \sum_{k=0}^{n} \mathrm{dens}_k^{\downarrow}(x, A) \quad \text{and} \quad \mathrm{score}^{\uparrow}(x, A) := \sum_{k=1}^{n} \mathrm{dens}_k^{\uparrow}(x, A),$$

and refer to $\mathrm{score}^{\downarrow}(x, A)$ as the *downward A-score of x* and $\mathrm{score}^{\uparrow}(x, A)$ as its *upward A-score*.

We point out the asymmetry between the definitions of $\mathrm{score}^{\downarrow}(x, A)$ and $\mathrm{score}^{\uparrow}(x, A)$: the first is summed over k starting at 0, whereas the second is summed over k starting at 1. (Note that $\mathrm{dens}_0^{\downarrow}(x, A) = \mathrm{dens}_0^{\uparrow}(x, A) = \mathbf{1}[x \in A]$.) We will need the fact that both the upward and downward A-scores of any $x \in \{0,1\}^n$ are at most $d = d(n, \varepsilon)$ when $A \subseteq L_{\mathrm{mid}}$.

The following lemma relates the distribution $\mathcal{D}'$ (more precisely, the distribution over $\boldsymbol{y}$ that is induced by conditioning on a particular outcome of $\boldsymbol{x}$) to the notion of score:

Lemma 2.5.4. *Let $A \subseteq L_{\mathrm{mid}}$ be a set of $\sigma 2^n$ points and fix $x^* \in L_{\mathrm{mid}}$. Then*

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}'}\left[\boldsymbol{y} \in A \mid \boldsymbol{x} = x^*\right] = \frac{1}{\Theta(\sqrt{n \ln(1/\varepsilon)})}\left(\mathrm{score}^{\downarrow}(x^*, A) + \mathrm{score}^{\uparrow}(x^*, A)\right).$$

Proof. This holds since

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}'}\left[\boldsymbol{y}\in A \mid \boldsymbol{x}=x^*\right] &= \mathop{\mathbf{E}}_{\boldsymbol{p}\ni x^*}\left[\frac{|\boldsymbol{p}_{\text{mid}}\cap A|}{|\boldsymbol{p}_{\text{mid}}|}\right] \\
&= \frac{1}{\Theta(d)}\cdot\mathop{\mathbf{E}}_{\boldsymbol{p}\ni x^*}\left[|\boldsymbol{p}_{\text{mid}}\cap A|\right] \\
&= \frac{1}{\Theta(d)}\left(\sum_{k\geq 0}\left(\mathop{\mathbf{E}}_{\substack{\boldsymbol{y}\preceq x^*\\ \|\boldsymbol{y}-x^*\|_1=k}}\left[\mathbf{1}[\boldsymbol{y}\in A]\right]\right)+\sum_{k\geq 1}\left(\mathop{\mathbf{E}}_{\substack{\boldsymbol{y}\succ x^*\\ \|\boldsymbol{y}-x^*\|_1=k}}\left[\mathbf{1}[\boldsymbol{y}\in A]\right]\right)\right) \\
&= \frac{1}{\Theta(d)}\left(\sum_{k\geq 0}\left(\mathop{\mathbf{Pr}}_{\substack{\boldsymbol{y}\preceq x^*\\ \|\boldsymbol{y}-x^*\|_1=k}}\left[\boldsymbol{y}\in A\right]\right)+\sum_{k\geq 1}\left(\mathop{\mathbf{Pr}}_{\substack{\boldsymbol{y}\succ x^*\\ \|\boldsymbol{y}-x^*\|_1=k}}\left[\boldsymbol{y}\in A\right]\right)\right) \\
&= \frac{1}{\Theta(d)}\left(\text{score}^{\downarrow}(x^*,A)+\text{score}^{\uparrow}(x^*,A)\right).
\end{aligned}$$

□

We use the previous two lemmas to lower bound the expected downward A-score of an $\boldsymbol{x}$ drawn uniformly at random from A:

Lemma 2.5.5. *Let $\varepsilon\geq 1/n$ and $A\subseteq L_{\text{mid}}$ be a set of $\sigma 2^n$ points. Then*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\in A}\left[\text{score}^{\downarrow}(\boldsymbol{x},A)\right]=\Omega\left(\frac{\varepsilon^8\sigma\sqrt{n}}{\sqrt{\ln(1/\varepsilon)}}\right).$$

Proof. We begin with the claim that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\in A}\left[\text{score}^{\downarrow}(\boldsymbol{x},A)\right]\geq\Omega(\varepsilon^4)\mathop{\mathbf{E}}_{\boldsymbol{x}\in A}\left[\text{score}^{\uparrow}(\boldsymbol{x},A)\right]+1, \tag{2.9}$$

where the $+1$ is due to $\text{dens}_0^{\downarrow}(\boldsymbol{x},A)=1$. To see (2.9), we rewrite the LHS of the inequality as follows:

$$\begin{aligned}
\mathop{\mathbf{E}}_{\boldsymbol{x}\in A}\left[\text{score}^{\downarrow}(\boldsymbol{x},A)\right]-1 &= \frac{1}{\sigma 2^n}\sum_{x\in A}\sum_{k\geq 1}\sum_{\substack{y\prec x\\ \|y-x\|_1=k}}\frac{\mathbf{1}[y\in A]}{\binom{\|x\|_1}{k}} \\
&= \frac{1}{\sigma 2^n}\sum_{x\in A}\sum_{\substack{y\in A\\ y\prec x}}\frac{1}{\binom{\|x\|_1}{\|x-y\|_1}} \\
&= \frac{1}{\sigma 2^n}\sum_{y\in A}\sum_{\substack{x\in A\\ x\succ y}}\frac{\binom{n-\|y\|_1}{\|x-y\|_1}}{\binom{\|x\|_1}{\|x-y\|_1}}\cdot\frac{1}{\binom{n-\|y\|_1}{\|x-y\|_1}} \\
&\geq \min_{\substack{x\succ y\\ x,y\in L_{\text{mid}}}}\left\{\frac{\binom{n-\|y\|_1}{\|x-y\|_1}}{\binom{\|x\|_1}{\|x-y\|_1}}\right\}\mathop{\mathbf{E}}_{\boldsymbol{y}\in A}\left[\text{score}^{\uparrow}(\boldsymbol{y},A)\right]=\Omega(\varepsilon^4)\mathop{\mathbf{E}}_{\boldsymbol{y}\in A}\left[\text{score}^{\uparrow}(\boldsymbol{y},A)\right],
\end{aligned}$$

where the final equality holds by the first part of Lemma 2.5.2. This proves (2.9), which together with Lemma 2.5.4 gives

$$\begin{aligned}\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}'}\left[\boldsymbol{y}\in A \mid \boldsymbol{x}\in A\right] &= \frac{1}{\Theta(\sqrt{n\ln(1/\varepsilon)})}\mathop{\mathbf{E}}_{\boldsymbol{x}\in A}\left[\mathrm{score}^{\uparrow}(\boldsymbol{x},A)+\mathrm{score}^{\downarrow}(\boldsymbol{x},A)\right]\\ &= \frac{O(\varepsilon^{-4})}{\Theta(\sqrt{n\ln(1/\varepsilon)})}\left(\mathop{\mathbf{E}}_{\boldsymbol{x}\in A}\left[\mathrm{score}^{\downarrow}(\boldsymbol{x},A)\right]\right). \qquad (2.10)\end{aligned}$$

On the other hand, by Corollary 2.5.3 we have

$$\begin{aligned}\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}'}\left[\boldsymbol{y}\in A \mid \boldsymbol{x}\in A\right] &= \frac{\mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}'}[\boldsymbol{x},\boldsymbol{y}\in A]}{\mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}'}[\boldsymbol{x}\in A]}\\ &= \frac{\mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}'}[\boldsymbol{x},\boldsymbol{y}\in A]}{\sigma} \;=\; \Omega\left(\frac{\varepsilon^4\sigma}{\ln(1/\varepsilon)}\right). \qquad (2.11)\end{aligned}$$

Combining (2.10) with (2.11) and rearranging completes the proof. □

Lemma 2.5.5 lower bounds the average downward A-score of points $x\in A$; its conclusion may be equivalently rewritten as the following sum:

$$\sum_{x\in A}\mathrm{score}^{\downarrow}(x,A)=\Omega\left(\frac{\varepsilon^8\sigma^2\sqrt{n}\,2^n}{\sqrt{\ln(1/\varepsilon)}}\right). \qquad (2.12)$$

We may express the downward A-score $\mathrm{score}^{\downarrow}(x,A)$ of a point x as a sum over $m+1$ "buckets" of exponentially increasing size:

$$\mathrm{score}^{\downarrow}(x,A)=\left(\sum_{k\in B_0}\mathrm{dens}^{\downarrow}_k(x,A)\right)+\left(\sum_{k\in B_1}\mathrm{dens}^{\downarrow}_k(x,A)\right)+\cdots+\left(\sum_{k\in B_m}\mathrm{dens}^{\downarrow}_k(x,A)\right), \qquad (2.13)$$

where $B_0=\{0\}$ and $B_i=\{2^{i-1},\dots,2^i-1\}$ for each $i\in[m]$ and $m=\lceil\log(n+1)\rceil$. It will be useful for us to focus on a particular bucket $\ell\in\{0,1,\dots,m\}$ such that the overall sum of $\mathrm{score}^{\downarrow}(x,A)$ in (2.12) has a "large" contribution from the ℓ-th bucket. A straightforward argument, exploiting the fact that there are only logarithmically many buckets, lets us achieve this without losing too much in the sum:

Corollary 2.5.6. *Let $\varepsilon\geq 1/n$ and $A\subseteq L_{\mathrm{mid}}$ be a set of $\sigma 2^n$ points. There exists $\ell\leq m$ such that*

$$\sum_{x\in A}\sum_{k\in B_\ell}\mathrm{dens}^{\downarrow}_k(x,A)=\Omega\left(\frac{\varepsilon^8\sigma^2\sqrt{n}\,2^n}{(\log n)\sqrt{\ln(1/\varepsilon)}}\right). \qquad (2.14)$$

Proof. This follows from (2.12), (2.13), and the fact that there are only $m+1$ many buckets. □

Corollary 2.5.6 gives us a lower bound on the sum of downward A-scores of points $x \in A$ coming from a certain bucket B_ℓ. Our next corollary uses this to give a lower bound on the sum of downward A-scores of points $y \in A_u$ coming from (essentially) the same bucket B_ℓ, where A_u is an "upper vertex boundary" of A in the following sense: there exists an $|A|$-sized matching M of edges (x, y) where $x \prec y$, $x \in A$ and $y \in A_u$.

Corollary 2.5.7. *Let $\varepsilon \geq 1/n$ and M be a matching of $\sigma 2^n$ edges in the middle layers. Let*

$$A := \left\{x \in \{0,1\}^n \colon x \prec y \text{ and } (x,y) \in M\right\} \quad \text{and}$$

$$A_u := \left\{y \in \{0,1\}^n \colon y \succ x \text{ and } (x,y) \in M\right\}$$

be the lower and upper endpoints of edges in M, respectively. For each bucket B_i, $i \in \{0,1,\ldots,m\}$, we let $B'_i := \{j+1 : j \in B_i\}$. Then there exists an integer $\ell \leq m$ such that

$$\sum_{y \in A_u} \sum_{k \in B'_\ell} \mathrm{dens}^{\downarrow}_k(y, A) = \Omega\left(\frac{2^{\ell+n}\varepsilon^8\sigma^2}{(\log n)\sqrt{n\ln(1/\varepsilon)}}\right). \tag{2.15}$$

Proof. By Corollary 2.5.6, there exists an $\ell \leq m$ such that A satisfies (2.14).

Next for every edge $(x, y) \in M$ we have that

$$\mathrm{dens}^{\downarrow}_{k+1}(y, A) = \mathop{\mathbf{Pr}}_{\substack{\boldsymbol{z} \prec y \\ \|\boldsymbol{z}-y\|_1 = k+1}}[\boldsymbol{z} \in A] \geq \frac{\binom{\|x\|_1}{k}}{\binom{\|y\|_1}{k+1}} \mathop{\mathbf{Pr}}_{\substack{\boldsymbol{z} \prec x \\ \|\boldsymbol{z}-x\|_1 = k}}[\boldsymbol{z} \in A] = \frac{(k+1)\cdot \mathrm{dens}^{\downarrow}_k(x, A)}{\|x\|_1 + 1}.$$

Therefore, by (2.14) we have

$$\begin{aligned}
\sum_{y \in A_u} \sum_{k \in B'_\ell} \mathrm{dens}^{\downarrow}_k(y, A) &= \sum_{y \in A_u} \sum_{k \in B_\ell} \mathrm{dens}^{\downarrow}_{k+1}(y, A) \\
&\geq \sum_{x \in A_u} \sum_{k \in B_\ell} \frac{(k+1)\cdot \mathrm{dens}^{\downarrow}_k(x, A)}{\|x\|_1 + 1} \\
&= \Omega\left(\frac{\varepsilon^8\sigma^2\sqrt{n}\,2^n}{(\log n)\sqrt{\ln(1/\varepsilon)}} \cdot \frac{2^\ell}{n}\right).
\end{aligned}$$

This completes the proof. □

2.5.3 The weighted path tester and its analysis

Given a Boolean function f, recall that a pair (x, y) of vertices is a *violated pair with respect to f if $x \prec y$ and $f(x) > f(y)$*. Our algorithm `weighted-path-tester` for monotonicity testing proceeds as follows:

`weighted-path-tester`:

1. Pick a point $\boldsymbol{y}$ uniformly from L_{mid}.
2. Pick $\boldsymbol{\ell} \in \{0, 1, \ldots, m = \lceil \log(n+1) \rceil\}$ uniformly.
3. Pick $\boldsymbol{k} \in B'_{\boldsymbol{\ell}}$ uniformly.
4. Pick a path $\boldsymbol{p}$ uniformly from the collection of all paths going through $0^n, \boldsymbol{y}$ and 1^n, and set $\boldsymbol{x}$ to be the (unique) point on $\boldsymbol{p}$ that has $\boldsymbol{x} \prec \boldsymbol{y}$ and $\|\boldsymbol{x} - \boldsymbol{y}\|_1 = k$.
5. Reject iff $(\boldsymbol{x}, \boldsymbol{y})$ is a violated pair.

We note that an equivalent formulation of step (4) is that $\boldsymbol{x}$ is drawn uniformly from $\{z \in \{0,1\}^n \colon z \prec \boldsymbol{y} \text{ and } \|\boldsymbol{y} - z\|_1 = \boldsymbol{k}\}$. Below we show that if there is a $(\sigma 2^n)$-sized matching M of violated edges of f in the middle layers of the hypercube, then the tester above succeeds in finding a violated pair with probability roughly $\Omega(\sigma^2/\sqrt{n})$.

Proposition 2.5.8. *Let $f : \{0,1\}^n \to \{0,1\}$ and $\varepsilon \geq 1/n$. Suppose there is a $(\sigma 2^n)$-sized matching M of violated edges of f all lying in the middle layers of the hypercube. Then* `weighted-path-tester` *succeeds (i.e. samples $\boldsymbol{x}$ and $\boldsymbol{y}$ that form a violated pair with respect to f) with probability*

$$\Omega\left(\frac{\varepsilon^8 \sigma^2}{(\log^2 n)\sqrt{n \ln(1/\varepsilon)}}\right). \tag{2.16}$$

Proof. Let A be the 1-endpoints of edges in M, and A_u be the 0-endpoints, and note that every pair $(x, y) \in A \times A_u$ satisfying $x \prec y$ is a violated pair with respect to f. Let $\mathcal{D}^w$ denote the distribution over comparable pairs $(\boldsymbol{x}, \boldsymbol{y}) \in L_{\text{mid}} \times L_{\text{mid}}$ induced by `weighted-path-tester`. Applying Corollary 2.5.7, we know there that exists an $\ell^* \in$

$\{0, 1, \ldots, m\}$ such that

$$\sum_{y \in A_u} \sum_{k \in B'_{\ell^*}} \mathrm{dens}^{\downarrow}_k(y, A) = \Omega\left(\frac{2^{\ell^*+n}\varepsilon^8\sigma^2}{(\log n)\sqrt{n\ln(1/\varepsilon)}}\right).$$

Note that conditioning on the event of $\boldsymbol{y} = y$ and $\boldsymbol{k} = k$, the probability of $\boldsymbol{x} \in A$ is $\mathrm{dens}^{\downarrow}_k(y, A)$. Since $\boldsymbol{y}, \boldsymbol{\ell}, \boldsymbol{k}$ are all sampled uniformly, `weighted-path-tester` succeeds with probability at least

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}^w}\left[\boldsymbol{y} \in A_u, \boldsymbol{x} \in A\right] &= \mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}^w}[\boldsymbol{y} \in A_u] \cdot \mathop{\mathbf{Pr}}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\mathcal{D}^w}\left[\boldsymbol{x} \in A \mid \boldsymbol{y} \in A_u\right] \\
&= \frac{|A_u|}{|L_{\mathrm{mid}}|} \cdot \frac{1}{|A_u|} \sum_{y \in A_u} \frac{1}{m+1} \sum_{\ell=0}^{m} \frac{1}{|B'_\ell|} \sum_{k \in B'_\ell} \mathrm{dens}^{\downarrow}_k(y, A) \\
&\geq \frac{1}{(m+1)|L_{\mathrm{mid}}||B'_{\ell^*}|} \cdot \sum_{y \in A_u} \sum_{k \in B'_{\ell^*}} \mathrm{dens}^{\downarrow}_k(y, A) \\
&= \Omega\left(\frac{2^{\ell^*+n}\varepsilon^8\sigma^2}{(\log n)\sqrt{n\ln(1/\varepsilon)}(\log n)2^{\ell^*+n}}\right) = \Omega\left(\frac{\varepsilon^8\sigma^2}{(\log^2 n)\sqrt{n\ln(1/\varepsilon)}}\right)
\end{aligned}$$

and this finishes the proof. □

2.5.4 Proof of Theorem 3

Finally we combine Proposition 2.5.8 with the dichotomy theorem of [Chakrabarty and Seshadhri, 2013a] to prove Theorem 3. To state the latter, we let $v2^n$ denote the total number of violated edges in f. We also let $\sigma 2^n$ denote the size of the largest matching of violated edges in the middle layers. Then we have

Theorem 10 (Theorem 2.4 of [Chakrabarty and Seshadhri, 2013a])**.** *For any f that is ε-far from monotone, $v \cdot \sigma = \Omega(\varepsilon^2)$.*

We now prove Theorem 3.

Proof of Theorem 3. As mentioned at the beginning of Section 2.5, we may assume without loss of generality that $\varepsilon \geq 1/n$ since otherwise the edge tester alone succeeds with probability $\Omega(\varepsilon/n) = \Omega(\varepsilon^2)$. When $\varepsilon \geq 1/n$, our tester flips a coin, runs the edge tester with probability $1/2$, and runs `weighted-path-tester` with probability $1/2$. Given v and σ as defined above, the success probability of the edge tester is $\Omega(v/n)$; the success probability

of `weighted-path-tester` is given in (2.16). It follows from Theorem 10 that the average of these two is at least

$$\Omega\left(\frac{\varepsilon^4}{n^{5/6}(\log^{2/3} n)\,(\ln(1/\varepsilon))^{1/6}}\right).$$

This finishes the proof of Theorem 3. □

Part II

Computational Learning Theory

Chapter 3

Approximate Resilience and the Complexity of Agnostic Learning

3.1 Background and context

Learning in the agnostic learning framework [Haussler, 1992; Kearns *et al.*, 1994], which models learning from examples in the presence of worst-case noise, is notoriously difficult. In this framework the learning algorithm is given random examples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x}$ is chosen from some distribution D and f is an *arbitrary* Boolean function. The goal of the agnostic learning algorithm for a concept class $\mathcal{C}$ is to output a hypothesis h that agrees with f almost as well as the best function in $\mathcal{C}$; that is:

$$\mathbf{Pr}_D[h(\boldsymbol{x}) \neq f(\boldsymbol{x})] \leq \min_{c \in \mathcal{C}} \mathbf{Pr}_D[c(\boldsymbol{x}) \neq f(\boldsymbol{x})] + \varepsilon,$$

where ε is an error parameter given to the algorithm.

Even when D is the uniform distribution, agnostic learning has proven extremely challenging: in the twenty years since the model was first introduced, few non-trivial classes are known to be learnable agnostically. The primary technique used for agnostic learning in this setting is the polynomial ℓ_1 regression algorithm introduced in the influential work of Kalai et al. [Kalai *et al.*, 2008]. This algorithm finds a low-degree polynomial that minimizes the ℓ_1 distance to the target function, and can be applied to agnostically learn classes which are well approximated by polynomials. This approach has lead to the first agnostic learning

algorithm for AC^0 circuits (in quasi-polynomial time) and halfspaces (in $n^{O(1/\varepsilon^2)}$ time) over the uniform distribution [Kalai *et al.*, 2008] and was used in many other agnostic learning results.

In this work we explain why the polynomial ℓ_1 regression algorithm is the best approach known to date for agnostically learning over product distributions. Specifically, we prove that the complexity of agnostic learning $\mathcal{C}$ over a product distribution in the statistical query model is characterized by how well $\mathcal{C}$ can be approximated in the ℓ_1 norm by low-degree polynomials over the same distribution. The statistical query (SQ) model [Kearns, 1998] is a well-studied restriction of the PAC learning model in which the learner relies on approximate expectations of functions of an example rather than examples themselves. With the exception of Gaussian elimination[1] all known techniques used in the theory and practice of machine learning have statistical query analogues. Polynomial ℓ_1 regression is no exception, and therefore to prove our characterization it suffices to establish a lower bound on learning by statistical query algorithms for function classes that are not well-approximated by low-degree polynomials.

A critical parameter in understanding lower bounds on agnostic learning is the value of $\mathsf{OPT}_{\mathcal{C}}(D, f) = \min_{c \in \mathcal{C}} \mathbf{Pr}[c(\boldsymbol{x}) \neq f(\boldsymbol{x})]$ to which the lower bound applies (note that OPT is essentially the noise rate). If a hardness result requires functions f for which $\mathsf{OPT}_{\mathcal{C}}(D, f)$ is close to $1/2$, then learning may still be possible in more practically realistic scenarios when OPT is a small constant close to 0 (or even approaches 0 as n grows). In machine learning literature it is more common to specify the *excess error* which is the difference between $\mathsf{OPT}_{\mathcal{C}}(D, f)$ and the error of the produced hypothesis that an algorithm can achieve. It is easy to see that lower bounds showing that excess error of κ cannot be achieved also imply that the lower bound applies to a setting where $\mathsf{OPT} = 1/2 - \kappa$ (since error of $1/2$ can always be achieved). Most known lower bounds for agnostic learning are in the regime when $\mathsf{OPT}_{\mathcal{C}}(D, f)$ goes to $1/2$ as dimension and other problem parameters grow although there are some notable exceptions in the more challenging distribution-independent setting [Klivans and Sherstov, 2010; Feldman *et al.*, 2012]. In this work we aim to precisely characterize

[1] Note that Gaussian elimination fails in the presence of even minor amounts of random noise and is not applicable in the agnostic framework.

the values of κ and OPT for which agnostic learning becomes hard and therefore will make these parameters explicit in our lower bounds.

3.1.1 Approximate resilience and agnostic learning

Our lower bounds for agnostic learning are based on a formal connection between agnostic learning and a basic structural property of Boolean functions. We say that a function $g : \{-1,1\}^n \to \mathbb{R}$ is d-resilient if $\widehat{g}(S) = 0$ for all $|S| \leq d$, *i.e.* g is uncorrelated with every low-degree parity. Equivalently, g is d-resilient if and only if $\mathbf{E}[g] = 0$ and $\mathbf{E}[g_\rho] = 0$ for any restriction ρ to at most d out of n variables. The structural question we will be interested in is:

How close can a Boolean function be to a highly resilient function with range in $[-1,1]$*?*

More precisely, we say that $f : \{-1,1\}^n \to [-1,1]$ is α-approximately d-resilient if there exists a d-resilient $g : \{-1,1\}^n \to [-1,1]$ such that $\|f-g\|_1 = \mathbf{E}[|f(\boldsymbol{x}) - g(\boldsymbol{x})|] \leq \alpha$, and we will be interested in functions that are α-approximately d-resilient for small values of α and large values of d. We note that for simplicity and convenience the definitions here are for the uniform distribution on the hypercube but can be easily extended to general product distributions over other n-dimensional domains (see Section 3.2.1).

The notion of resilience is well-studied and has applications in cryptography, pseudorandomness, inapproximability, circuit complexity and more (for a few examples, see [Chor *et al.*, 1985; Luby and Wigderson, 1995; Austrin and Mossel, 2009; Austrin and Håstad, 2011; Sherstov, 2011]). However, to the best of our knowledge our notion of approximate resilience does not appear to have been explicitly studied before.

At a high level we show that if a concept class $\mathcal{C}$ contains an α-approximately d-resilient function then the complexity of learning $\mathcal{C}$ agnostically in the SQ model is $n^{\Omega(d)}$. Further, learning is hard even for $\mathsf{OPT} \leq \alpha/2$ (in other words when noise rate is $\alpha/2$). For simplicity the complexity of an SQ algorithm refers to a polynomial upper-bounding both the running time and the inverse of query tolerance. Naturally, the presence of a single α-approximately d-resilient function would not suffice for a hardness result since a concept class with a single function can be easily learned agnostically. We therefore need some assumptions under

which existence of a single α-approximately d-resilient function will imply that there are many of them. One such assumption that we adopt is that the α-approximately d-resilient function c depends on at most $n^{1/3}$ variables (such a function is called a $n^{1/3}$-junta) and the concept class $\mathcal{C}$ is closed under renaming of variables. Alternatively, if we consider an ensemble of concept classes $\{\mathcal{C}_n\}_{n=1}^{\infty}$ parameterized by dimension n it would be sufficient to assume that the ensemble is closed under addition of irrelevant variables. For brevity we omit the closed-ness under renaming since it is satisfied by all commonly-studied concept classes. We now state our lower bound in terms of resilience informally.

Theorem 11. *Let $\mathcal{C}$ be a concept class. Fix d and let $\alpha(d)$ be such that, there exists a $\alpha(d)$-approximately d-resilient $n^{1/3}$-junta $c \in \mathcal{C}$. Then any SQ algorithm for agnostically learning $\mathcal{C}$ with excess error of at most $\frac{1-\alpha(d)}{2} - n^{-o(d)}$ has complexity of at least $n^{\Omega(d)}$.*

Alternatively, this result can be stated as saying that if for every function f satisfying $\mathsf{OPT}_{\mathcal{C}}(D, f) \leq \alpha(d)/2$ the algorithm outputs h such that $\mathbf{Pr}_D[h(\boldsymbol{x}) \neq f(\boldsymbol{x})] \leq 1/2 - n^{-o(d)}$ then its SQ complexity is $n^{\Omega(d)}$. An immediate implication of this theorem is that a concept class containing an $o(1)$-approximately d-resilient function cannot be learned with noise rate larger than $o(1)$ in time $n^{\Omega(d)}$.

The proof of this theorem is based on the simple observation that agnostic learning of $\mathcal{C}$ is at least as hard as weak learning of a class of d-resilient functions which are close to functions in $\mathcal{C}$. From there we rely on hardness of SQ learning of pairwise nearly orthogonal functions to obtain the claim. This result relies crucially on the distribution being a product distribution.

While this might appear to be a relatively limited approach to obtaining lower bounds we show that lower bounds it achieves are essentially optimal. This follows from the duality between approximate resilience and approximation by low-degree polynomials that we establish. More formally, let $\mathcal{P}_d$ be the class of degree at most d real-valued polynomials. For a Boolean function f, let $\Delta_{\mathcal{P}_d}(f) = \min_{p \in \mathcal{P}_d} \mathbf{E}[|f - p|]$.

Theorem 12. *For $f : \{-1,1\}^n \to \{-1,1\}$ and $0 \leq d \leq n$ and $\varepsilon \geq 0$. f is α-approximately d-resilient if and only if $\Delta_{\mathcal{P}_d}(f) \geq 1 - \alpha$.*

The proof of this result is a fairly simple application of a classical result on duality of norms by Ioffe and Tikhomirov [Ioffe and Tikhomirov, 1968].

Now for a concept class $\mathcal{C}$, let $\Delta_{\mathcal{P}_d}(\mathcal{C}) = \max_{f \in \mathcal{C}} \Delta_{\mathcal{P}_d}(f)$. To see how this quantity characterizes agnostic learning in the statistical query model, we state the error and running time achieved by the polynomial ℓ_1 regression algorithm of Kalai *et al.* for agnostic learning [Kalai *et al.*, 2008]. This algorithm is easy to implement in the SQ model[2].

Theorem 13 ([Kalai *et al.*, 2008])**.** *Let $\mathcal{C}$ be a concept class over $\{-1,1\}^n$ and fix d. There exists a SQ algorithm which for any $\varepsilon > 0$ agnostically learns $\mathcal{C}$ with excess error $\Delta_{\mathcal{P}_d}(\mathcal{C})/2 + \varepsilon$ and has complexity* $\mathrm{poly}(n^d, 1/\varepsilon)$.

On the other hand, we may apply Theorems 12 and 11 to show that this is the best any SQ algorithm can do; by Theorem 12 there exists an $\alpha(d)$-approximately d-resilient function in $\mathcal{C}$ with $1 - \alpha(d) = \Delta_{\mathcal{P}_d}(\mathcal{C})$. Therefore Theorem 11 essentially matches the upper bound of Theorem 13 in excess error and complexity, implying the optimality of ℓ_1-regression based algorithms for agnostic learning over the uniform distribution. The extension to other product distributions is fairly straightforward and we discuss it in Sec. 3.2.1.

3.1.2 Learning monotone juntas

With this characterization in hand, we would like to better understand what classes of functions we can hope to agnostically learn on the uniform distribution. Uniform distribution learning is challenging even in the noiseless setting, with efficient algorithms out of reach for natural classes such as polynomial size DNF formulas and decision trees. However, learning monotone functions and their corresponding subclasses seems significantly easier; for example, monotone decision trees [O'Donnell and Servedio, 2008] and monotone DNFs with few terms [Servedio, 2004a] are efficiently learnable in the SQ model (for other examples see [O'Donnell and Wimmer, 2013; Blum *et al.*, 1998; Bshouty and Tamon, 1996]).

This difference is demonstrated most dramatically in the junta learning problem, which is considered by many to be the single most important open problem in uniform distribution

[2]To the best of our knowledge this is not proved anywhere explicitly but follows easily from the fact that LPs can optimized approximately using approximate evaluations of the optimized function (in our case expected ℓ_1 error) [Lovász, 1987]

learning. In this problem, the target function is an unknown k-*junta*, a Boolean function which depends on at most $k \ll n$ variables. The junta problem also lies at the heart of the notorious DNF and decision tree learning problems: Since s-term DNFs and s-leaf decision trees can compute arbitrary $(\log s)$-juntas, learning either of these classes requires that we first be able to efficiently learn $\omega(1)$-juntas. Progress has remained slow in the 20 years since Blum posed the junta problem, with the current fastest algorithm running in time $n^{.60k}$ [Valiant, 2012], improving on the first non-trivial algorithm which runs in time $n^{.704k}$ [Mossel *et al.*, 2004] (the trivial algorithm exhaustively checks all k-subsets of $[n]$ and runs in time $O(n^k)$). In contrast, monotone juntas are easy to learn using an extremely simple algorithm: the relevant variables can be identified by estimating their correlations with the target function $\mathbf{E}[f(\boldsymbol{x})\boldsymbol{x}_i] = \widehat{f}(\{i\})$, and thus monotone k-juntas can be learned in time $O(n + 2^k)$.

Does the advantage of monotonicity hold in the agnostic setting as well? We first consider the simplest problem of agnostic learning monotone juntas. While it appears to be a hard problem, known hardness results for specific monotone functions do not rule out polynomial time algorithms for any constant ε. Specifically, the best known lower bound is $n^{\Omega(1/\varepsilon^2)}$ for majority functions [Kalai *et al.*, 2008] and is based on the assumption that learning sparse noisy parities is hard. Further, this hardness result only applies when $\mathsf{OPT} \geq 1/2 - \varepsilon$ which leaves open the possibility that the problem is solvable efficiently when the noise rate is a constant smaller than $1/2$.

As we saw in Theorem 11, the complexity of agnostic learning of $\mathcal{C}$ is characterized by the approximate resilience of functions in $\mathcal{C}$. Therefore we consider the structural question of how close monotone functions are to bounded resilient functions. The structure of monotone functions over the Boolean hypercube has been investigated in many influential works (see [Blum *et al.*, 1998; Bshouty and Tamon, 1996; Mossel and O'Donnell, 2002; O'Donnell, 2003; O'Donnell and Wimmer, 2013]). While to the best of our knowledge our notion has not been studied before, several works have examined the total spectral weight that monotone functions have on low-degree coefficients [Bshouty and Tamon, 1996; Mossel and O'Donnell, 2002]. Spectral weight indicates the distance to the closest *unbounded* resilient function in ℓ_2 norm. Both differences of bounded/unbounded and ℓ_1/ℓ_2 are significant, but we show

how some of the bounds on low-degree spectral weight can serve as a basis for bounds on our notion of distance to resilience.

It is easy to see that monotone functions cannot be 1-resilient, and prior to our work, it was possible that every monotone function was $\Omega(1)$-far from 1-resilient. Our first structural result rules out this possibility in a very strong way:

Theorem 14. *For every $\alpha > 0$ there exists an α-approximately d-resilient monotone Boolean function where $d = \Omega(\alpha\sqrt{n}/\log n)$.*

Our proof of this result is indirect. We use a lower bound for PAC learning of monotone functions by Blum *et al.* [Blum *et al.*, 1998] to obtain strong lower bounds on ℓ_1-approximation of monotone functions by polynomials. We can then use Theorem 12 to obtain bounds on distance to resilience.

This degree of resilience is essentially optimal: combining basic facts from discrete Fourier analysis, it is straightforward to see that every monotone Boolean function is α-far from any $\Omega(\alpha\sqrt{n})$-resilient function [Bshouty and Tamon, 1996]. Applying our connection between approximate resilience and agnostic learning, we get as a corollary our main application:

Corollary 3.1.1. *Any SQ algorithm for agnostically learning the class of monotone k-juntas with excess error of $1/2 - \alpha$ has complexity of $n^{\Omega(\alpha\sqrt{k}/\log k)}$.*

Qualitatively, Corollary 3.1.1 gives the first super-polynomial lower bound on the complexity of SQ algorithms for agnostically learning monotone k-juntas with constant (and even sub-constant) noise. It also rules out the possibility of efficient SQ algorithms for agnostic learning monotone decision trees and monotone DNFs with few terms (which, as previously mentioned, do have efficient SQ algorithms in the noiseless setting). Quantitatively, our lower bound essentially matches the upper bound of $n^{O(\sqrt{k}/\varepsilon)}$ that follows as a corollary of the low-degree concentration bound of [Bshouty and Tamon, 1996] and the polynomial ℓ_1 regression algorithm [Kalai *et al.*, 2008]. Note that lower bounds on PAC learning of monotone functions [Blum *et al.*, 1998] cannot be translated directly to lower bounds in the junta learning setting since these lower bounds are subexponential in k while junta learning algorithms are allowed to run in time polynomial in 2^k.

While Theorem 14 yields a near-optimal lower bound on the complexity of agnostically learning monotone juntas, the construction is not explicit: it is based on a randomized DNF construction (similar to Talagrand's randomized DNF construction [Talagrand, 1996]), and contains functions of high complexity. Furthermore, for more general classes such as monotone DNFs, the hardness results implied are not optimal. We first show that even the simple Tribes function, a read-once DNF, is close to a resilient function (which gives a stronger hardness result for learning small monotone DNFs).

Theorem 15. *The* Tribes *function is α-approximately d-resilient, where $\alpha = O(n^{-1/3})$ and $d = \Omega(\log n / \log\log n)$.*

The resilience in Theorem 15 can be amplified to $2^{\Omega(\sqrt{\log n})}$ in an explicit way by iteratively composing Tribes with itself (see Section 3.3.4 for details). Our proof technique for Theorem 15 is quite general: we show that the projection of a Boolean function onto its high-degree part can be transformed into a bounded function which is both resilient and close to the original function, as long as it has $o(1)$ spectral weight on low-degree coefficients.

Both Theorems 14 and 15 give monotone Boolean functions which are close to resilient functions, however the resilient functions are not necessarily Boolean-valued. Resilience is often studied specifically for Boolean functions, and therefore we ask if there are such functions that are close to monotone Boolean functions. Using a new function called CycleRun [Wieder, 2002], we show that this is indeed possible, and furthermore we nearly match the resilience of the iterated Tribes construction:

Theorem 16. *There is an explicit α-approximately d-resilient monotone Boolean function f where $\alpha = o_n(1)$ and $d = 2^{\Omega(\sqrt{\log n}/\log\log n)}$. Furthermore, f is α-close to a Boolean d-resilient function.*

We prove Theorem 16 by first showing that CycleRun is $O(\sqrt{\log n/n})$-approximately 1-resilient, where our witness to this approximate resilience is a Boolean function. Our argument crucially relies on four key properties of CycleRun: monotonicity, low influence, oddness, and invariance under cyclic shifts; as far as we know, CycleRun is the only explicit Boolean function known to have all four properties. These properties allow us to use a structured combinatorial argument, unlike our argument for Tribes that relies on properties

of polynomials and produces a witness that is a bounded function (and applying this style of argument to Tribes quickly gets unruly). Having established $O(\sqrt{\log n/n})$-approximate 1-resilience, we then apply the aforementioned general amplification technique to increase the degree of resilience to $2^{\widetilde{\Omega}(\sqrt{\log n})}$.

We remark that while the degrees of resilience obtained in Theorems 16 and 15 are not as strong as that of Theorem 14, both are sufficient to rule out the existence of efficient SQ algorithms for learning monotone k-juntas for any $k = \omega_n(1)$ and subconstant error-rate.

3.1.3 Related work

Lower bounds for statistical query algorithms were first shown by Kearns [Kearns, 1998] who proved that parities cannot be learned by SQ algorithms. Soon after this Blum et al. [Blum *et al.*, 1994] characterized the weak PAC learnability of every function class $\mathcal{C}$ in the SQ model in terms of the *statistical query dimension* of $\mathcal{C}$; roughly speaking, this is the largest number of functions from $\mathcal{C}$ that are pairwise nearly orthogonal to each other (we give a precise definition in Section 3.2). These lower bound techniques were extended to strong PAC learning and agnostic learning in more recent work [Simon, 2007; Feldman, 2012; Szörényi, 2009]. Lower bounds for SQ algorithms were proved for many learning problems including, for example, PAC learning of juntas [Blum *et al.*, 1994], weak-learning of intersections of halfspaces [Klivans and Sherstov, 2007] and learning of monotone depth-3 formulas [Feldman *et al.*, 2011]. These lower bounds are information-theoretic but capture remarkably well the computational hardness of learning problems. In some cases, such as learning juntas over the uniform distribution, this is the only known formal evidence of the hardness of the problem.

Several previously known lower bounds for agnostic learning are based on the reduction to learning of k-sparse noisy parities. This is a notoriously hard problem for which the only non-trivial algorithm is the recent breakthrough result of Valiant that gives an algorithm running in time $n^{0.8k}$ [Valiant, 2012]. Assuming that this problem requires $n^{\Omega(k)}$ time we get that agnostic learning of majorities on the uniform distribution requires $n^{\Omega(1/\varepsilon^2)}$ time [Kalai *et al.*, 2008] and conjunctions require $n^{\Omega(\log(1/\varepsilon))}$ time [Feldman, 2012]. Learning k-sparse parities in the SQ model has complexity of $n^{\Omega(k)}$ and therefore these result also

give unconditional SQ lower bounds. These lower bounds can be interpreted as special cases of our approach. They are based on showing that a parity of high-degree has a significant correlation with a function in $\mathcal{C}$. Clearly a k-sparse parity function is $(k-1)$-resilient and correlation implies that distance to that parity is slightly better than the trivial 1. The main disadvantage of this approach is that in most cases it can only lead to hardness results when the noise rate is close to $1/2$. In particular this approach cannot lead to the strong hardness results we prove here for monotone juntas.

In a recent work Feldman and Kothari [Feldman and Kothari, 2013] show that the equivalence between ℓ_1 approximation by polynomials and agnostic learning does not extend to non-product distributions. They exhibit a distribution D for which any polynomial that is $1/3$-close to the disjunction of all the variables in ℓ_1 (measured relative to D) must have degree $\Omega(\sqrt{n})$. At the same time disjunctions are SQ learnable in time $n^{O(\log(1/\varepsilon))}$ over that distribution.

Our approach to proving lower bounds is closest in spirit and some technical elements to the influential pattern matrix method of Sherstov [Sherstov, 2011]. His method shows that lower bounds on the approximation by polynomials in ℓ_∞ norm of a function f can be translated into lower bounds on randomized communication complexity of a certain communication problem corresponding to evaluation of f on different subsets of variables (which were previously thought as stronger than lower bounds on approximation in ℓ_∞ by polynomials). A crucial step in his result is an application of duality that is in some sense symmetric to ours and shows the existence of an unbounded resilient function g that is correlated with f. Such g then serves to upper bound discrepancy for the communication problem (from which a lower bound on randomized communication complexity follows).

3.1.4 Preliminaries

All probabilities and expectations are with respect to the uniform distribution unless otherwise stated, and we will use boldface (e.g. $\boldsymbol{x}$ and $\boldsymbol{y}$) to denote random variables. Given $f, g : \{-1,1\}^n \to \mathbb{R}$, we say that f and g are ε-close if $\|f-g\|_1 = \mathbf{E}[|f(\boldsymbol{x}) - g(\boldsymbol{x})|] \leq \varepsilon$. We say that g is bounded if it takes values in the interval $[-1,1]$. Note that if f is Boolean valued and g is bounded, then $\|f-g\|_1 = 1 - \mathbf{E}[fg]$. Every function $g : \{-1,1\}^n \to \mathbb{R}$ can

be uniquely written as a multilinear polynomial such that $g(x) = \sum_{S \subseteq [n]} \widehat{g}(S) \prod_{i \in S} x_i$ for all $x \in \{-1,1\}^n$; the coefficients $\widehat{g}(S)$ are called the Fourier coefficients of g. The total influence of a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, denoted $\mathbf{Inf}[f]$, is $\sum_{i=1}^n \mathbf{Pr}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})]$, where $x^{\oplus i}$ denotes x with its i-th coordinate flipped.

Definition 17. A function $g : \{-1,1\}^n \to \mathbb{R}$ is d-resilient if $\widehat{g}(S) = 0$ for all $|S| \leq d$. We say that a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ is α-approximately d-resilient if there exists a d-resilient bounded function g such that $\|f - g\|_1 \leq \alpha$.

Learning background In the agnostic learning framework, the learning algorithm is given labeled examples $(\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{x} \in \{-1,1\}^n$ and $\boldsymbol{y} \in \{-1,1\}$ are drawn from a distribution $\mathcal{D}$ over $\{-1,1\}^n \times \{-1,1\}$. As usual we describe such distributions by a pair (D, g), where D is the marginal distribution on $\{-1,1\}^n$ and $g : \{-1,1\}^n \to [-1,1]$, where $g(x) = \mathbf{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[\, \boldsymbol{y} \mid \boldsymbol{x} = x \,]$ is expectation of the label for each input. Note that for every Boolean function f, if U denotes the uniform distribution then $\mathbf{E}_{(\boldsymbol{x},\boldsymbol{y})\sim(U,g)}[f(\boldsymbol{x}) \neq \boldsymbol{y}] = \|f - g\|_1/2$.

Definition 18. Let $\mathcal{C}$ be a class of Boolean functions on $\{-1,1\}^n$. An algorithm A agnostically learns $\mathcal{C}$ over distribution D on $\{-1,1\}^n$ if for any $g : \{-1,1\}^n \to [-1,1]$ and $\varepsilon > 0$, given examples from distribution $\mathcal{D} = (D, g)$ and ε, it outputs with probability at least $2/3$ hypothesis $h : \{-1,1\}^n \to \{-1,1\}$ such that:

$$\mathbf{Pr}[h(\boldsymbol{x}) \neq \boldsymbol{y}] \leq \mathsf{OPT}_{\mathcal{C}}(D, g) + \varepsilon,$$

where $\mathsf{OPT} = \min_{c \in \mathcal{C}} \mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y})\sim(D,g)}[c(\boldsymbol{x}) \neq \boldsymbol{y}]$. The algorithm is said to learn with *excess error* κ if h instead satisfies

$$\mathbf{Pr}[h(\boldsymbol{x}) \neq \boldsymbol{y}] \leq \mathsf{OPT}_{\mathcal{C}}(D, g) + \kappa.$$

Definition 19. A statistical query is defined by a bounded function of an example $\phi : \{-1,1\}^n \times \{-1,1\} \to [-1,1]$ and positive tolerance τ. A valid reply to such a query relative to a distribution $\mathcal{D}$ over examples is a value v that satisfies:

$$|\, \mathbf{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[\phi(\boldsymbol{x}, \boldsymbol{y})] - v| \leq \tau.$$

A statistical query learning algorithm is an algorithm which relies solely on statistical queries and does not have access to actual examples. We say that an SQ algorithm has **statistical query complexity** T if it makes at most q statistical queries of tolerance at least τ and $T \geq \max\{q, 1/\tau\}$.

3.2 Characterizing the complexity of agnostic learning

In this section we show that approximate resilience implies hardness of agnostic learning for statistical query algorithms (Lemma 3.2.1). We then show that the implication works in the reverse direction as well: if a class does not contain approximately resilient functions, then it can be agnostically learned by SQ algorithms. We prove this equivalence using the duality between approximate resilience and approximation by low-degree polynomials stated in Theorem 12. This simple observation turns out to be surprisingly useful, leading both to a characterization of agnostic learning and to a proof of our first structural result for monotone functions (Theorem 14).

To connect our notion of approximate resilience to the hardness of agnostic learning we will use the following standard notion of designs of sets with small overlap. A (n, k, d)-design of size m is a collection of sets $S_1, \ldots, S_m \subseteq [n]$ such that $|S_i| = k$ and $|S_i \cap S_j| \leq d$ for all $i \neq j$. Let $\mathcal{M}(n, k, d)$ denote the size of the largest (n, k, d)-design. Standard probabilistic/greedy argument implies that

$$\mathcal{M}(n, k, d) \geq \frac{\binom{n}{k}}{\binom{k}{d}\binom{n-d}{k-d}} = \frac{\binom{n}{d}}{\binom{k}{d}^2} \geq \left(\frac{nd}{e^2k^2}\right)^d. \tag{3.1}$$

For a function $f : \{-1, 1\}^k \to \{-1, 1\}$ and set $S \subseteq [n]$ of size k we use $f_S : \{-1, 1\}^n \to \{-1, 1\}$ to denote $f(\boldsymbol{x}_{|S})$ where $\boldsymbol{x}_{|S}$ refers to the restriction of $\boldsymbol{x}$ to coordinates with indices in S (in the usual order).

Lemma 3.2.1. *Let $f : \{-1, 1\}^k \to \{-1, 1\}$ be an α-approximately d-resilient function. Let $S_1, \ldots, S_m$ be a (n, k, d)-design. If $\{f_{S_i}\}_{i=1}^m \subseteq \mathcal{C}$, then any SQ algorithm for agnostically learning $\mathcal{C}$ with excess error of at most $\frac{1-\alpha}{2} - m^{-1/3}$ has complexity of at least $m^{1/3}$.*

To prove Lemma 3.2.1, we will use the following result implicit in [Feldman, 2012] that is a simple generalization of the well-known SQ-DIM bounds from [Blum *et al.*, 1994] and

their strengthening in [Yang, 2005; Szörényi, 2009].

Theorem 20. *Let D be a distribution and let $g_1, \ldots, g_m$ be bounded real-valued functions such that $|\langle g_i, g_j \rangle_D| \leq 1/m$ for $i \neq j$, where $\langle g_i, g_j \rangle_D = \mathbf{E}_D[g_i(\boldsymbol{x}) \cdot g_j(\boldsymbol{x})]$. Then any SQ algorithm that for every i, given access to statistical queries with respect to distribution (D, g_i) outputs a hypothesis h such that $\mathbf{E}_{(\boldsymbol{x},\boldsymbol{y})\sim(D,g_i)}[h(\boldsymbol{x}) \neq \boldsymbol{y}] \leq \frac{1}{2} - \frac{1}{m^{1/3}}$ has complexity of at least $m^{1/3}$.*

We can now prove Lemma 3.2.1.

Proof. By our assumption, the function f is α-close to a d-resilient bounded function $g : \{-1,1\}^k \to [-1,1]$. We first note that each pair of functions g_{S_i} g_{S_j} shares at most d relevant variables. These functions are d-resilient and therefore there is no single set T such that $\widehat{g_{S_i}}(T) \cdot \widehat{g_{S_j}}(T) \neq 0$. This, by linearity of expectation implies that for $i \neq j$, $\mathbf{E}[g_{S_i} g_{S_j}] = 0$.

Let A be an agnostic algorithm for $\mathcal{C}$ with excess error of at most $\frac{1-\alpha}{2} - m^{-1/3}$. For every i, f_{S_i} is α-close to g_{S_i}. Therefore if the input distribution is (U, g_i) then $\mathsf{OPT}_{\mathcal{C}}(U, g_i) \leq \|f_{S_i} - g_{S_i}\|_1/2 = \|f - g\|_1/2 \leq \alpha/2$. This implies that A will output a hypothesis h with error of at most $\alpha/2 + \frac{1-\alpha}{2} - m^{-1/3} = 1/2 - m^{-1/3}$. By Theorem 20 and orthogonality of g_{S_i}s we get that the complexity of A is at least $m^{1/3}$. □

An immediate corollary of Lemma 3.2.1 is the following lower bound that generalizes Theorem 11.

Theorem 21. *Let $\mathcal{C}$ be a concept class closed under renaming of variables and assume that $\mathcal{C}$ contains an α-approximately d-resilient k-junta. Then any SQ algorithm for agnostically learning $\mathcal{C}$ with excess error of at most $\frac{1-\alpha}{2} - m^{-1/3}$ has complexity of at least $m^{1/3}$, where $m = \mathcal{M}(n, k, d)$. In particular, for any constant $\delta > 0$ and $k = n^{1/2+\delta}$, we have $m = n^{\Omega(d)}$.*

To show that Theorem 21 is essentially tight we prove the duality stated in Theorem 12 (which we restate here for convenience).

Theorem 12. *For $f : \{-1,1\}^n \to \{-1,1\}$ and $0 \leq d \leq n$ let α denote the ℓ_1 distance of f to the closest d-resilient bounded function. Then $\Delta_{\mathcal{P}_d}(f) = 1 - \alpha$.*

Proof. Our proof is an adaptation of the general results on duality of norms [Ioffe and Tikhomirov, 1968] to the case where f is Boolean and g is bounded. In this case it is easy to see that $\|f-g\|_1 = 1 - \mathbf{E}[fg]$ and therefore minimization of distance to resilience can be expressed as maximization of $\sum_x f(x)g(x)$ subject to resilience constraints on g. Viewing values of $g(x)$ as variables we get:

$$\begin{aligned} \max \ & \sum_x f(x)g(x) & \\ \text{subject to } & \sum_x g(x)\chi_S(x) = 0 & \forall |S| \leq d \\ \text{and } & |g(x)| \leq 1 & \forall x \in \{-1,1\}^n \end{aligned}$$

The dual LP can be easily verified to be the following program with variables p_S for every $S \subseteq [n]$ of size at most d.

$$\begin{aligned} \min \ & \sum_x |q(x)| & \\ \text{subject to } & q(x) = f(x) - \sum_{S:|S|\leq d} p_S \chi_S(x) & \forall x \in \{-1,1\}^n \end{aligned}$$

Now the claim of the theorem follows from LP duality. By definition the maximum value of the primal is $2^n \cdot \mathbf{E}[fg] = 2^n(1 - \|f-g\|_1) = 2^n(1-\alpha)$. This is therefore also the minimum of the dual program which, by definition, is exactly $2^n \cdot \Delta_{\mathcal{P}_d}(f)$. □

Note that $(1-\alpha)/2$ in the excess error term in the statement of Theorem 21 is equal to $\Delta_{\mathcal{P}_d}(\mathcal{C})/2$ in the excess error term in the statement Theorem 13. Therefore combining the duality with the upper-bounds on polynomial ℓ_1 regression stated in Theorem 13 we get our claimed characterization of the complexity of agnostic learning in terms of $\Delta_{\mathcal{P}_d}(\mathcal{C})$ or, alternatively, distance to d-resilience.

3.2.1 Product distributions

We now outline the extension of our characterization of the SQ complexity of agnostic learning to more general product distributions. Let X be the domain of each individual variable, that is our leaning problem is defined over X^n. We will start with symmetric

product distributions and let Π be a distribution over X. Let $\mathcal{B} = \{B_0(x), B_1(x), \ldots\}$ be the basis obtained via Gram-Schmidt orthonormalization on the basis $1, x, x^2, \ldots$ with respect to the inner product $\langle f, g\rangle_\Pi = \mathbf{E}_\Pi[f(\boldsymbol{x})g(\boldsymbol{x})]$. By definition we obtain that the polynomial degree of B_i is i (for $i \leq |X| - 1$). As special cases this process gives $\{1, \frac{1-\mu\cdot x}{\sqrt{1-\mu^2}}\}$ basis if $X = \{-1, 1\}$ and $\mu = \mathbf{E}_\Pi[\boldsymbol{x}]$; Legendre polynomials when $X = [-1, 1]$ and Π is uniform; and Hermite polynomials when $X = \mathbb{R}$ and Π is the Gaussian $N(0, 1)$ distribution.

For $S \subseteq [n]$ and a function $t : S \to \mathbb{N}$ let $\Phi_{S,t}(x) = \Pi_{i\in S} x_i^{t(i)}$ and $\Psi_{S,t}(x) = \Pi_{i\in S} B_{t(i)}(x_i)$. For a finite X we restrict the range of such t's to $[|X| - 1]$. Clearly, Ψ's are orthonormal functions relative to the inner product $\langle f, g\rangle_{\Pi^n} = \mathbf{E}_\Pi[f(\boldsymbol{x})g(\boldsymbol{x})]$.

We now say that a function g is d-resilient relative to Π^n if for every $S \subseteq [n]$ of size at most d and any function $t : S \to \mathbb{N}$, $\langle g, \Psi_{S,t}\rangle_{\Pi^n} = 0$. Note that equivalently this can be defined as $\langle g, \Phi_{S,t}\rangle_{\Pi^n} = 0$ for all $S \subseteq [n]$ of size at most d and $t : S \to \mathbb{N}$.

We say that a Boolean f is α-approximately d-resilient relative to Π^n if there exists a d-resilient $g : X^n \to [-1, 1]$ such that $\mathbf{E}_{\Pi^n}[|f(\boldsymbol{x}) - g(\boldsymbol{x})|] \leq \alpha$. In the following discussion functions are over X^n and all norms and inner products relative to Π^n.

We now describe generalizations of Theorems 13, 12 and 21. Let $\mathcal{P}_{d,\ell}$ denote the class of polynomials where each monomial has at most d different variables each of degree at most ℓ; let $\mathcal{P}_d = \mathcal{P}_{d,\infty}$. Note that by definition this is the span of $\{\Phi_{S,t}\}_{|S|\leq d, t:S\to[\ell]}$ but is also equal to the span of $\{\Psi_{S,t}\}_{|S|\leq d, t:S\to[\ell]}$. For a function f, let $\Delta_{\mathcal{P}_{d,\ell}}(f) = \min_{p\in\mathcal{P}_{d,\ell}} \mathbf{E}_{\Pi^n}[|f(\boldsymbol{x}) - p(\boldsymbol{x})|]$ and for a concept class $\mathcal{C}$, let $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C}) = \max_{f\in\mathcal{C}} \Delta_{\mathcal{P}_{d,\ell}}(f)$.

The polynomial ℓ_1 regression algorithm of Kalai *et al.* for agnostic learning [Kalai *et al.*, 2008] applies to this general setting and gives the following bound.

Theorem 22 ([Kalai *et al.*, 2008])**.** *Let $\mathcal{C}$ be a concept class over X^n and fix d and ℓ. There exists a SQ algorithm which for any $\varepsilon > 0$ agnostically learns $\mathcal{C}$ over Π^n with excess error $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C})/2 + \varepsilon$ and has complexity* $\mathrm{poly}((n\ell)^d, 1/\varepsilon)$.

Our SQ lower bound can be easily seen to generalize to the following statement.

Theorem 23. *Let $\mathcal{C}$ be a concept class over X^n closed under renaming of variables and assume that $\mathcal{C}$ contains a k-junta which is α-approximately d-resilient over Π^n. Then any SQ algorithm for agnostically learning $\mathcal{C}$ over Π^n with excess error of at most $\frac{1-\alpha}{2} - m^{-1/3}$*

has complexity of at least $m^{1/3}$*, where* $m = \mathcal{M}(n, k, d)$*. In particular, for any constant* $\delta > 0$ *and* $k = n^{1/2+\delta}$*, we have* $m = n^{\Omega(d)}$*.*

Finally, the duality is also easy to verify in this case.

Theorem 24. *For* $f : X^n \to \{-1, 1\}$ *and* $0 \leq d \leq n$ *let* α *denote the* ℓ_1 *distance of* f *to the closest* d*-resilient bounded function. Then* $\Delta_{\mathcal{P}_d}(f) = 1 - \alpha$*.*

Now the upper bound is $(n\ell)^{O(d)}$ with excess error $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C})/2$ and the lower bound is $n^{\Omega(d)}$ with excess error of $\Delta_{\mathcal{P}_d}(\mathcal{C})/2$ (if k is not too large). Therefore tightness depends on how fast $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C})$ approaches $\Delta_{\mathcal{P}_d}(\mathcal{C})$ as ℓ grows. Note that if $\mathcal{C}$ contains only functions that depend on at most k-variables then convergence of $\Delta_{\mathcal{P}_{d,\ell}}(\mathcal{C})$ to $\Delta_{\mathcal{P}_d}(\mathcal{C})$ depends only on k (and not on n) and also as long as $\ell = n^{O(1)}$ the bounds are still within a polynomial factor.

Non-symmetric product distributions. Now let the domain be $X_1 \times X_2 \times \cdots \times X_n$ and the product distribution be $\Pi = \Pi_1 \times \Pi_2 \times \cdots \times \Pi_n$. We first note that the upper bound in Thm. 22 and the duality hold even if the distribution is not symmetric (that is different variables might have different marginal distributions). Therefore we only need to adapt Thm. 23 to this setting.

Our lower-bound construction requires closed-ness with respect to renaming of variables. That would not suffice if different variables have different marginal distributions. For example ℓ_1 distance to polynomials clearly depends on the marginal distributions of variables and therefore we can no longer claim that the analogue of $\|f_{S_i} - g_{S_i}\|_1 = \|f - g\|_1$ holds in this setting (as we did in the proof of Lemma 3.2.1). Therefore we will need an additional assumption. Let S be the set of variables of the optimal (in terms of distance to d-resilience) k-junta. We will assume that for every variable $i \in S$, there are many other variables that have the same marginal distribution as variable i. Specifically, there exists a set $I_i \subseteq [n]$, such that for $j_1, j_2 \in I_i$, $\Pi_{j_1} = \Pi_{j_2}$ and the size of I_i is at least s. In addition, we need $\mathcal{C}$ to be closed under renaming of variables, where a variable that is in I_i is renamed to another variable in I_i.

Now we can construct a family of ordered sets $S_1, \ldots, S_m$ (each of size k) such that the intersection of any two sets is at most d, and the i'th element of each set S_j (recall that we

think of S_j as an ordered set) is from I_i. This means that X and Π restricted to variables in S_j (ordered in the same way as they are in S_j) are exactly the same as X and Π restricted to variables in S. This means that the proof of the lower bound in Lemma 3.2.1 applies to this setting, as before essentially verbatim. The complexity is now determined by the size of the largest family of sets with the property we described. By the same argument as in eq.(3.1) there exists a family of size:

$$\frac{s^k}{\binom{k}{d}s^{k-d}} = \Omega\left(\left(\frac{sd}{k}\right)^d\right).$$

This family has size $n^{\Omega(d)}$ for $s = n^{\Omega(1)}$ and a large range of parameters k and d (e.g. $d = k^{1-\Omega(1)}$).

3.3 Monotonicity and approximate resilience

In this section we prove bounds on the approximate resilience of monotone functions. First, we give a bound for general monotone functions (Theorem 14) in Section 3.3.1. In Sections 3.3.2 and 3.3.3 we show that Tribes and CycleRun are approximately resilient (Theorems 15 and 16). Finally, in Section 3.3.4 we show how these functions can be used in an iterated construction to yield explicit functions with high approximate resilience.

3.3.1 A monotone function with nearly-optimal approximate resilience

Our characterization suggests an approach for proving Theorem 14: since the ℓ_1-minimization algorithm characterizes SQ agnostic learning, we seek monotone functions where the ℓ_1-minimization algorithm will badly fail. In other words, our first step will be to move to the dual problem: Theorem 12 tells us that we may equivalently show the existence of a monotone function f which is far from from every low-degree polynomial p. Strangely, to show that no dual solution exists, we will use the fact that if every monotone function had a weak approximation by some low-degree polynomial, then the ℓ_1-minimization algorithm would learn monotone functions, contradicting known information-theoretic lower bounds [Blum *et al.*, 1998]. Note that while the ℓ_1-minimization algorithm is presented as an agnostic learning algorithm, we may apply it directly to the class of monotone functions.

We now prove Theorem 14:

Theorem 14. *For every $\alpha > 0$, there is a monotone function that is α-approximately d-resilient for $d = \Omega(\alpha\sqrt{n}/\log n)$.*

Proof. We show the existence of a monotone function f such that $\mathbf{E}[|f(\boldsymbol{x}) - p(\boldsymbol{x})|] > 1 - \alpha$ for every degree-d polynomial p and then apply Theorem 12. Suppose that every monotone f satisfies $\mathbf{E}[|f(\boldsymbol{x}) - p(\boldsymbol{x})|] \leq 1 - \alpha$. Then for $\varepsilon = \alpha/4$, Theorem 13 gives an algorithm for learning monotone functions which uses $s = \mathrm{poly}(n^d/\alpha)$ examples and has error $1/2 - \alpha/2 + \alpha/4 = 1/2 - \alpha/4$. We now use an information-theoretic lower bound on the number of random examples needed to weakly learn monotone functions; the proof in [Blum *et al.*, 1998] uses a randomized construction of DNF formulas:

Theorem 25 ([Blum *et al.*, 1998])**.** *Let A be a any learning algorithm that uses s random examples and outputs a hypothesis h. Then there is some monotone $f : \{-1,1\}^n \to \{-1,1\}$ such that*

$$\mathbf{Pr}[f(\boldsymbol{x}) = h(\boldsymbol{x})] \leq \frac{1}{2} + O\left(\frac{\log sn}{\sqrt{n}}\right).$$

Theorem 25 tells us that $\alpha = O\left(\frac{d\log n + \log 1/\alpha}{\sqrt{n}}\right)$, which completes the proof. □

The function from Theorem 14 gives us a k-junta that is α-approximately d-resilient for $d = \Omega(\alpha\sqrt{k}/\log k)$. Plugging this into Theorem 21 and using eq.(3.1) (assuming $k \leq n^{1/2}$) we obtain the proof of Corollary 3.1.1.

While the degree of resilience in Theorem 14 is nearly optimal, the proof is non-constructive and relies crucially on the fact that monotone functions can have high complexity. In the following sections we show that even simple, explicit monotone functions can exhibit high approximate resilience.

3.3.2 Tribes is approximately resilient: Proof of Theorem 15

The $\mathsf{Tribes}_{w,s} : \{-1,1\}^{sw} \to \{-1,1\}$ function is the disjunction of s disjoint monotone conjunctions, each of width w; i.e. a read-once width-w DNF. For notational brevity we write Tribes to denote $\mathsf{Tribes}_{w,s}$ with $s = (\ln 2)2^w$ (so $w \approx \log n - \log\ln n$ and $s \approx n/(\log n)$).

Our construction of a highly resilient function close to Tribes is based on the fact that the Tribes function has small Fourier weight on its low-degree terms.

Proposition 3.3.1. *For any $d \leq w$ the Fourier weight of* Tribes *on degree d and below is at most*

$$\sum_{|S|\leq d} \widehat{\mathsf{Tribes}}(S)^2 \leq 2\frac{(2\ln n)^{2d+4}}{n}.$$

Proof. The proof follows Ryan O'Donnell's thesis, pages $66-67$ [O'Donnell, 2003]. For a set $T \subseteq [n]$ let T_i denote the intersection of T with the variables in the i-th conjunction. We use the following expressions proved in [Mansour, 1995]:

$$\widehat{\mathsf{Tribes}_{w,s}}(T) = \begin{cases} 2(1-2^{-w})^s - 1 & T = \emptyset \\ 2(-1)^{k+|T|}2^{-kw}(1-2^{-w})^{s-k} & k = \#\{i : T_i \neq \emptyset\} > 0. \end{cases} \tag{3.2}$$

Using the calculations above, we have that for any $T \subseteq [n]$ with $k = \#\{i : T_i \neq \emptyset\}$:

$$\widehat{\mathsf{Tribes}}(T)^2 \leq \left(\frac{2\ln n}{n}\right)^{2k}.$$

For any k, the number of coefficients that have degree at most d and intersect k conjunctions is at most

$$\sum_{j=0}^{d} \binom{s}{k}\binom{kw}{j} \leq (d+1)s^k(kw+1)^d \leq n^k w^{2d+2}.$$

The last inequality holds because $s \leq n$ and $k \leq d$ (and we assume that $d \leq w$). Summing over $1 \leq k \leq d$, we obtain:

$$\begin{aligned} \sum_{|T|\leq d} \widehat{\mathsf{Tribes}}(T)^2 &\leq \sum_{k=1}^{d} n^k w^{2d+2}\left(\frac{2\ln n}{n}\right)^{2k} \\ &\leq w^{2d+2}\sum_{k=1}^{d}\left(\frac{(2\ln n)^2}{n}\right)^k \\ &\leq 2w^{2d+2}\frac{(2\ln n)^2}{n} \\ &\leq 2\frac{(2\ln n)^{2d+4}}{n}, \end{aligned}$$

where we used $w \leq 2\ln n$ in the last step. □

We begin our construction with the Fourier polynomial for Tribes and discard the low-degree terms. That we may do so and hope to arrive at a bounded, resilient function comes from hypercontractivity: since the discarded polynomial has low-degree, it will by highly

concentrated around its mean. The following Chernoff-type concentration inequality for low-degree polynomials over independent Rademacher random variables follows from the hypercontractivity inequalities of Bonami and Beckner [Bonami, 1970; Beckner, 1975] (see for example [O'Donnell, 2014]).

Theorem 26 ("concentration of degree-d polynomials")**.** *There exists a universal constant $K > 0$ such that for every degree-d polynomial $\{-1,1\}^n \to \mathbb{R}$ and $t > e^d$, we have*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}}[|p(\boldsymbol{x})| \geq t \cdot \|p\|_2] \leq \exp\left(-Kt^{2/d}\right).$$

We now begin the proof of Theorem 15. Let

$$\ell(x) = \sum_{|S|\leq d} \widehat{\mathsf{Tribes}}(S)\chi_S(x), \quad \text{and} \quad h(x) = \mathsf{Tribes}(x) - \ell(x).$$

Our final resilient, bounded function p will be based on h, the high-degree part of Tribes. Note that while h is d-resilient by definition, it may not be uniformly bounded. However, the degree-d Chernoff bound applied to ℓ (the low-degree part), together with Proposition 3.3.1 which bounds the variance of ℓ, tell us that ℓ does not attain large values very often. Therefore, while h may not be uniformly bounded, we have that h is bounded on almost all inputs x since $h(x) + \ell(x) = \mathsf{Tribes}(x) \in \{-1,1\}$.

More formally, we set $t = \frac{\tau}{2} \cdot \frac{\sqrt{n}}{(2\ln n)^{d+2}}$ in Theorem 26 (where τ will be specified later) and use the bound $\|\ell\|_2 \leq 2\frac{(2\ln n)^{d+2}}{\sqrt{n}}$ from Proposition 3.3.1 to obtain:

$$\mathbf{Pr}_{\boldsymbol{x}}[|\ell(\boldsymbol{x})| \geq \tau] \leq \exp\left(\frac{-K(\tau\sqrt{n})^{2/d}}{(2\ln n)^3}\right) := \delta.$$

Next, we define $q : \{-1,1\}^n \to \mathbb{R}$ to be such that

$$q(x) = \begin{cases} 0 & \text{if } |\ell(x)| > \tau \\ h(x) & \text{if } |\ell(x)| \leq \tau. \end{cases}$$

Since $h = \mathsf{Tribes} - \ell$ and Tribes is $\{-1,1\}$-valued, the range of q is $[-1-\tau, 1+\tau]$. While q is bounded, it may now have correlations with low-degree terms (i.e. q is no longer resilient like h is). However, we may also write q as $q(x) = h(x) - h(x) \cdot \mathbf{1}_{[\ell>\tau]}(x)$, where h is

d-resilient and $\mathbf{1}_{[\ell>\tau]}$ has very small support. Thus, we will show that we may discard the low-degree terms of q and the effect on boundedness will be uniformly small.

Let $q_{>d}(x) = \sum_{|S|\geq d+1} \widehat{q}(S)\chi_S(x)$, $q_{\leq d} = q - q_{>d}$ and $p(x) = \frac{q_{>d}(x)}{\|q_{>d}\|_\infty}$. Certainly, the range of p is $[-1, 1]$; it remains to bound the correlation of p with Tribes.

We have that:

$$\begin{aligned}\mathbf{E}[p \cdot \mathsf{Tribes}] &= \mathbf{E}\left[\frac{(q - q_{\leq d})}{\|q_{>d}\|_\infty} \cdot \mathsf{Tribes}\right] \\ &\geq \frac{1}{\|q\|_\infty + \|q_{\leq d}\|_\infty} \cdot (\mathbf{E}[q \cdot \mathsf{Tribes}] - \|q_{\leq d}\|_\infty) \qquad (3.3)\end{aligned}$$

The correlation of Tribes with q is large:

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[q(\boldsymbol{x}) \cdot \mathsf{Tribes}(\boldsymbol{x})] \geq (1-\tau)(1-\delta) \geq 1 - \tau - \delta. \qquad (3.4)$$

The above holds because the contribution to the correlation is 0 when $q(x) = 0$, which happens on at most a δ fraction of the inputs. On the remaining inputs, $q(x) = h(x) = \mathsf{Tribes}(x) - \ell(x)$, and we assumed $|\ell(x)| \leq \tau$. Thus the contribution on such x is

$$q(x) \cdot \mathsf{Tribes}(x) = (\mathsf{Tribes}(x) - \ell(x)) \cdot \mathsf{Tribes}(x) = 1 - \ell(x) \cdot \mathsf{Tribes}(x) \geq 1 - |\ell(x)| \geq 1 - \tau.$$

Thus, it only remains to bound the maximum value of the low-degree part of q:

Claim 3.3.2.

$$\|q_{\leq d}\|_\infty \leq \delta n^{2d+2}$$

Proof. We will show that $|\widehat{q}(S)| < \delta n^{d+1}$ holds for any $|S| \leq d$. Recalling that $q(x) = h(x) - \mathbf{1}_{|\ell|>\tau} \cdot h(x)$, we have:

$$\begin{aligned}\widehat{q}(S) &= \widehat{h}(S) - \widehat{\mathbf{1}_{|\ell|>\tau} \cdot h}(S) \\ |\widehat{q}(S)| &\leq |\widehat{h}(S)| + \mathbf{E}[|\mathbf{1}_{|\ell|>\tau} \cdot h|] \\ &\leq 0 + \delta \cdot \|h\|_\infty \\ &\leq \delta(\|\ell\|_\infty + 1),\end{aligned}$$

where the second inequality holds when $|S| \leq d$ because h is d-resilient, and the last inequality holds because $|h(x)| \leq |\ell(x)| + 1$ for all x. Tribes is a Boolean function and therefore each of the non-zero Fourier coefficients of ℓ is at most 1 in magnitude. The rough

bound of n^{d+1} on the number of non-zero coefficients of ℓ gives a bound of n^{d+1} on $\|\ell\|_\infty$; summing over at most n^{d+1} terms of degree at most d gives the claim. □

Let $\kappa = \delta n^{2d+2}$. Substituting into Equations (3.3) and (3.4), we have that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[p(\boldsymbol{x}) \cdot \mathsf{Tribes}(\boldsymbol{x})] \geq \frac{1-\tau-\delta-\kappa}{1+\tau+\kappa} \geq 1-\delta-2\tau-2\kappa,$$

using the fact that $1/(1+x) \geq 1-x$ for $x \geq 0$. We now set $\tau = \frac{(2\ln n)^{3d}}{n^{2/5}}$, so that $t \geq n^{1/10}$. Now there exists a small constant $c > 0$ such that for $d = c\log n/\log\log n$ and large enough n, we have that $\tau = O(n^{-1/3})$, $t > e^d$ and $t^{2/d} \geq n^{1/(5d)} \geq \frac{3}{K}(\log n)^2 \geq \frac{(2d+3)}{K}\ln n$. This implies that $\delta = \exp\left(-Kt^{2/d}\right) \leq n^{-2d-3}$ and $\kappa = \delta n^{2d+2} \leq 1/n$.

Therefore

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[p(\boldsymbol{x}) \cdot \mathsf{Tribes}(\boldsymbol{x})] \geq 1-\delta-2\tau-2\kappa \geq 1-O(n^{-1/3}),$$

which completes the proof of Theorem 15.

3.3.3 CycleRun is approximately resilient: Proof of Theorem 16

Definition 27. For every n, the CycleRun Boolean function $\mathsf{CycleRun} : \{-1,1\}^n \to \{-1,1\}$ is defined as follows: Call a consecutive sequence of 1's a 1-run. Similarly, a consecutive sequence of -1's is a -1-run. We allow runs to wrap around, so if a run reaches x_n it may continue with x_1. The value of CycleRun is the winner (1 for 1-player or -1 for -1-player) from the following procedure:

1. Check which player has the longest run.
2. In case of tie check which player has a larger number of maximum-length runs.
3. In case of tie check the total length of segments between maximum-length runs, where a segment starting from a 1-run clockwise is counted for the 1-player and a segment starting at a -1-run clockwise is counted for the -1-player. The player that has a larger total count is declared the winner.

We will need that fact that CycleRun has influence $O(\log n)$. Since the proof of this fact has not appeared in the literature before, we include a proof in Section 3.4.1 for completeness.

Theorem 28. *There exist universal constants c_1, c_2 such that for every $n \geq c_2$, there exists a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ such that:*

1. *For all $S \subseteq [n]$ such that $|S| \leq 1$, $\widehat{f}(S) = 0$, and*
2. $\mathbf{E}_{\boldsymbol{x}}[f(\boldsymbol{x}) \cdot \mathsf{CycleRun}(\boldsymbol{x})] \geq 1 - c_1\sqrt{(\log n)/n}$.

Our proof of Theorem 28 relies on four key properties of CycleRun: monotonicity, low influence, oddness, and invariance under cyclic shifts; as far as we know, CycleRun is the only explicit Boolean function known to have all four properties. First, as CycleRun is monotone and transitive, we note that

$$\widehat{\mathsf{CycleRun}}(\{i\}) = \widehat{\mathsf{CycleRun}}(\{j\}) = O\left(\frac{\log n}{n}\right) \quad \text{for all } i \neq j \in [n].$$

The high level intuition behind our proof is simple: we show that by flipping the values of CycleRun from the top of the hypercube downwards and bottom upwards simultaneously, we obtain a balanced function with no Fourier weight at the first level. This can be done without changing too many points because CycleRun has small influence; we are able to do it in a controlled way because it is additionally odd and invariant under cyclic shifts. We defer the proof of Theorem 28 to Section 3.4.

It is natural to wonder how close a monotone function can be to a 1-resilient Boolean function. We show in Section 3.4.2 that Theorem 28 is tight:

Theorem 29. *For every monotone function $f : \{-1,1\}^n \to \{-1,1\}$ and 1-resilient $g : \{-1,1\}^n \to \{-1,1\}$, we have $\mathbf{Pr}_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] \geq \Omega\left(\sqrt{\frac{\log n}{n}}\right)$.*

3.3.4 Resilience amplification

In this section we prove a general amplification lemma for resilience. Given a value $t \in [-1,1]$, we write $\boldsymbol{b}(t)$ to denote a random ± 1 bit with expected value t:

$$\boldsymbol{b}(t) = \begin{cases} 1 & \text{with probability } (1+t)/2 \\ -1 & \text{with probability } (1-t)/2. \end{cases}$$

(In particular, $\boldsymbol{b}(1)$ is the constant 1 and $\boldsymbol{b}(-1)$ is the constant -1). Given bounded functions $G : \{-1,1\}^m \to [-1,1]$ and $g : \{-1,1\}^n \to [-1,1]$, we define their (disjoint) composition $G \circ g : \{-1,1\}^{mn} \to [-1,1]$ to be $(G \circ g)(x^1, \ldots, x^m) := \mathbf{E}[G(\boldsymbol{b}(g(x^1)), \ldots, \boldsymbol{b}(g(x^m))]$.

Note that if $\mathbf{E}[g(\boldsymbol{x})] = 0$, then $\mathbf{E}[\boldsymbol{b}(g(\boldsymbol{x}))] = 0$ as well. Throughout this section we write $\mathrm{dist}(f, g)$ to denote $\frac{1}{2}\mathbf{E}[|f(\boldsymbol{x}) - g(\boldsymbol{x})|]$ for notational brevity (this is simply the fractional Hamming distance $\mathbf{Pr}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})]$ when f and g are $\{\pm 1\}$-valued).

The main result in this section is the following amplification lemma:

Theorem 30. *Let $f : \{-1,1\}^n \to \{-1,1\}$ and $g : \{-1,1\}^n \to [-1,1]$ where $\mathbf{E}[f(\boldsymbol{x})] = \mathbf{E}[g(\boldsymbol{x})] = 0$, and suppose g is d-resilient. Consider the recursively-defined functions where $f_k = f \circ f_{k-1}$ and $g_k = g \circ g_{k-1}$ for all $k \in \mathbb{N}$, and $f_0 = f$ and $g_0 = g$. Then for $k \geq 1$:*

1. *f_k and g_k are functions over n^{k+1} variables,*
2. *g_k is $((d+1)^{k+1} - 1)$-resilient,*
3. *$\mathrm{dist}(f_k, g_k) \leq \mathrm{dist}(f, g) \sum_{t=0}^{k} \mathbf{Inf}[f]^t$.*

The first claim is straightforward to verify, and so we focus on the second and third claims. For a Boolean-valued function $F : \{-1,1\}^m \to \{-1,1\}$ and $\delta \in [0,1]$, recall that the *noise-sensitivity of F at noise rate δ* is defined as $\mathbf{NS}_\delta[F] := \mathbf{Pr}_{\boldsymbol{y},\boldsymbol{z}}[F(\boldsymbol{y}) \neq F(\boldsymbol{z})]$, where $\boldsymbol{y}$ is uniform in $\{-1,1\}^m$ and $\boldsymbol{z}$ is obtained from $\boldsymbol{y}$ by independently flipping each of its coordinates with probability δ.

Lemma 3.3.3. *Given $F, f : \{-1,1\}^m \to \{-1,1\}$ and $G, g : \{-1,1\}^m \to [-1,1]$ where $\mathbf{E}[f(\boldsymbol{x})] = \mathbf{E}[g(\boldsymbol{x})] = 0$, we have*

$$\mathrm{dist}(F \circ f, G \circ g) \leq \mathrm{dist}(F, G) + \mathbf{NS}_\delta[F],$$

where $\delta := \mathrm{dist}(f, g)$.

Proof. We first apply the triangle inequality and note that

$$\mathrm{dist}(F \circ f, G \circ g) \leq \mathrm{dist}(F \circ f, F \circ g) + \mathrm{dist}(F \circ g, G \circ g).$$

Since $\mathbf{E}[g(\boldsymbol{x})] = 0$, we have that $\langle \boldsymbol{b}(g(\boldsymbol{x}^1)), \ldots, \boldsymbol{b}(g(\boldsymbol{x}^m)) \rangle$ is uniformly distributed on $\{-1,1\}^m$ when $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m$ are independently and uniformly distributed on $\{-1,1\}^n$, and therefore the second distance on the right hand side is exactly $\mathrm{dist}(F, G)$. Since $\mathbf{Pr}[\boldsymbol{b}(f(x)) \neq$

$\boldsymbol{b}(g(x))] = \mathbf{Pr}[f(x) \neq \boldsymbol{b}(g(x))] = \frac{1}{2}|f(x) - g(x)|$ for all $x \in \{-1,1\}^n$, it follows that $\mathbf{Pr}[\boldsymbol{b}(f(\boldsymbol{x})) \neq \boldsymbol{b}(g(\boldsymbol{x}))] = \frac{1}{2}\,\mathbf{E}[|f(\boldsymbol{x}) - g(\boldsymbol{x})|] = \delta$ and so

$$\mathrm{dist}(F \circ f, F \circ g) = \mathop{\mathbf{Pr}}_{\boldsymbol{y},\boldsymbol{z}}[F(\boldsymbol{y}) \neq F(\boldsymbol{z})],$$

where $\boldsymbol{y}$ is uniform in $\{-1,1\}^m$ and $\boldsymbol{z}$ is obtained from $\boldsymbol{y}$ by independently flipping each of its coordinates with probability δ. This completes the proof, since the probability on the right hand side is precisely $\mathbf{NS}_\delta[F]$. □

Using the union bound, we have

$$\mathbf{NS}_\delta[F] \leq \delta \sum_{i=1}^{n} \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[F(\boldsymbol{x}) \neq F(\boldsymbol{x}^{\oplus i})] = \delta \cdot \mathbf{Inf}[F] = \mathrm{dist}(f,g) \cdot \mathbf{Inf}[F],$$

where $\boldsymbol{x}^{\oplus i}$ is the string $\boldsymbol{x}$ with the i-th bit flipped, and $\delta = \mathrm{dist}(f,g)$ as in the previous lemma. This, along with a straightforward recursion, yields the following corollary.

Corollary 3.3.4. *Let $f : \{-1,1\}^n \to \{-1,1\}$ and $g : \{-1,1\}^n \to [-1,1]$ where $\mathbf{E}[f(\boldsymbol{x})] = \mathbf{E}[g(\boldsymbol{x})] = 0$, and suppose g is d-resilient. Consider the recursively-defined functions where $f_k = f \circ f_{k-1}$ and $g_k = g \circ g_{k-1}$ for all $k \in \mathbb{N}$, and $f_0 = f$ and $g_0 = g$. Then for $k \geq 1$:*

$$\mathrm{dist}(f_k, g_k) \leq \mathrm{dist}(f,g) \sum_{t=0}^{k} \mathbf{Inf}[f]^t.$$

Lemma 3.3.5. *If $G : \{-1,1\}^m \to [-1,1]$ is d_1-resilient and $g : \{-1,1\}^n \to [-1,1]$ is d_2-resilient, then $G \circ g$ is $(d_1 d_2)$-resilient.*

Proof. By linearity of the Fourier transform it suffices to prove this claim when $G(x_1, \ldots, x_m) = \prod_{i \in T} x_i$ and $|T| > d_1$, the parity function over $d_1 + 1$ or more variables. We begin by noting that

$$\begin{aligned}
(G \circ g)(x^1, \ldots, x^m) &= \mathbf{E}\left[\prod_{i \in T} \boldsymbol{b}(g(x^i))\right] \\
&= \prod_{i \in T} \mathbf{E}[\boldsymbol{b}(g(x^i))] \\
&= \prod_{i \in T} \left[\frac{1 + g(x^i)}{2} - \frac{1 - g(x^i)}{2}\right] = \prod_{i \in T} g(x^i).
\end{aligned}$$

We view the mn coordinates of the composed function $G \circ g$ as the disjoint union of $A_1 \cup \cdots \cup A_m$, where each A_i has size n. With this notation in hand, every subset S of the mn

coordinates may be viewed as the disjoint union $S_1 \cup \cdots \cup S_m$, where $A_j \subseteq S_j$ for all $j \in [m]$. Fix $S = S_1 \cup \cdots \cup S_m$ of cardinality at most $d_1 d_2$, and recall that our goal is to show that $\widehat{(G \circ g)}(S) = 0$. There exists at least one set S_j where $|S_j| \leq d_2$, and we assume without loss of generality that $|S_1| \leq d_2$. Since g is d_2-resilient (in particular, $\widehat{g}(S_1) = 0$), we see that indeed

$$\widehat{(G \circ g)}(S) = \mathbf{E}\left[\prod_{i \in T} g(\boldsymbol{x}^i) \prod_{j \in [m]} \prod_{\ell \in S_j} \boldsymbol{x}_\ell^j\right] = \prod_{i \in T} \widehat{g}(S_i) \prod_{j \notin T} \prod_{\ell \in S_j} \mathbf{E}[\boldsymbol{x}_\ell^j] = 0,$$

and the proof is complete. □

Combining Corollary 3.3.4 and Lemma 3.3.5 yields Theorem 30.

3.3.4.1 Amplifying Tribes and CycleRun

We now apply Theorem 30 to Tribes and CycleRun.

Theorem 31. *There is an explicit α-approximately d-resilient monotone Boolean function F where $\alpha = o_n(1)$ and $d = 2^{\Omega(\sqrt{\log n})}$.*

Proof. We apply Theorem 30 with f being Tribes and g the bounded resilient function that results from applying Theorem 15. Since $\mathbf{Inf}[\mathsf{Tribes}] = \Theta(\log n)$ (see e.g. [Kahn *et al.*, 1988]), taking $k := c \log n / \log\log n$ where $c > 0$ is a sufficiently small universal constant gives functions f_k, g_k over $N := n^k = 2^{O(\log^2 n / \log\log n)}$ variables, where

$$\mathrm{dist}(f_k, g_k) = O(\mathbf{Inf}[\mathsf{Tribes}]^{k+1} \cdot n^{-1/3}) = n^{-\Omega(1)} = o_N(1),$$

and g_k is d-resilient for $d = \Omega((\log n / \log\log n)^{k+1}) = 2^{\Omega(\sqrt{\log N})}$. □

Analogous calculations for CycleRun yield the following:

Theorem 16. *There is an explicit α-approximately d-resilient monotone Boolean function F where $\alpha = o_n(1)$ and $d = 2^{\Omega(\sqrt{\log n / \log\log n})}$. Furthermore, F is α-close to a d-resilient function that is Boolean-valued as well.*

Proof. We apply Theorem 30 with f being CycleRun and g the Boolean-valued resilient function that results from applying Theorem 28. Since $\mathbf{Inf}[\mathsf{CycleRun}] = O(\log n)$ (The-

orem 33), we again take $k = c \log n / \log\log n$ where $c > 0$ is a sufficiently small universal constant to get Boolean-valued functions f_k, g_k over $N = 2^{O(\log^2 n/\log\log n)}$ variables, where $\mathbf{Pr}[f_k(\boldsymbol{x}) \neq g_k(\boldsymbol{x})] = \mathrm{dist}(f_k, g_k) = n^{-\Omega(1)} = o_N(1)$, and g_k is d-resilient for $d = n^{\Omega(1/\log\log n)} = 2^{O\left(\sqrt{\log N}/\log\log N\right)}$. □

3.4 The CycleRun function

To aid us in proving properties of CycleRun, we will require several bounds involving Gaussian approximations. Specifically, we will make use of the functions $f_t : \{-1,1\}^n \to \{-1,0,1\}$ that appear in [O'Donnell and Wimmer, 2013]. We define $|x| = \sum_{i=1}^n x_i$ for a string $x \in \{-1,1\}^n$. These functions f_t are defined so that

$$f_t(x) = \begin{cases} 1 & \text{if } |x| > t\sqrt{n} \\ 0 & \text{if } -t\sqrt{n} \leq |x| \leq t\sqrt{n} \\ -1 & \text{if } |x| < -t\sqrt{n} \end{cases}$$

We use three properties (implicitly) appearing in [O'Donnell and Wimmer, 2013] that follow from error estimates for the Central Limit Theorem [Feller, 1968]: for large enough n and $\sqrt{\log n}/100 < t < n^{1/10}$, we have

$$\phi(t)\sqrt{n}/3 \leq \mathbf{Inf}(f_t) \leq 3\phi(t)\sqrt{n} \tag{3.5}$$

$$\phi(t)/(3t) \leq \mathbf{Pr}_{\boldsymbol{x}}[f_t(\boldsymbol{x}) \neq 0] \leq 3\phi(t)/t \tag{3.6}$$

$$\mathbf{Pr}[|\boldsymbol{x}| = t] \leq 4\phi(t)/\sqrt{n}. \tag{3.7}$$

where ϕ is the probability density function of the standard Gaussian distribution: $\phi(u) = \frac{1}{2\pi}\exp(-u^2/2)$; and $\mathbf{Inf}(f_t) = \mathbf{E}_{\boldsymbol{x}}[f_t(\boldsymbol{x}) \cdot |\boldsymbol{x}|] = \sum_{i\in[n]} \widehat{f_t}(\{i\})$. We note that $\mathbf{Inf}(g) = \mathbf{E}_{\boldsymbol{x}}[g(\boldsymbol{x}) \cdot |\boldsymbol{x}|] = \sum_{i\in[n]} \widehat{g}(\{i\})$ for a monotone Boolean function $g : \{-1,1\}^n \to \{-1,1\}$.

Definition 32. For every $x \in \{-1,1\}^n$, define the set Shift_x to contain the following:

- $x^\alpha = x_{(1+\alpha \mod n)} \cdots x_{(n+\alpha \mod n)}$, for $0 \leq \alpha \leq n-1$.
- $-x^\alpha = -x_{(1+\alpha \mod n)} \cdots - x_{(n+\alpha \mod n)}$, for $0 \leq \alpha \leq n-1$.

Note that $|\text{Shift}_x|$ always divides $2n$, and if the Hamming weight of x is relatively prime to n, then $|\text{Shift}_x| = 2n$. Because CycleRun is odd and invariant under cyclic shifts, CycleRun is 1 on exactly half the points of Shift_x.

Theorem 28. *There exist universal constants c_1, c_2 such that for every $n \geq c_2$, there exists a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ such that:*

1. *For all $S \subseteq [n]$ such that $|S| \leq 1$, $\widehat{f}(S) = 0$, and*
2. $\mathbf{E}_{\boldsymbol{x}}[f(\boldsymbol{x}) \cdot \mathsf{CycleRun}(\boldsymbol{x})] \geq 1 - 2c_1 \cdot \sqrt{\frac{\log(n)}{n}}$, *which implies* $\mathbf{Pr}_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq \mathsf{CycleRun}(\boldsymbol{x})] \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}}$.

Proof. Given $\mathsf{CycleRun} : \{-1,1\}^n \to \{-1,1\}$, we construct a set $\overline{S} \subseteq \{-1,1\}^n$ using the following greedy algorithm $\text{Const}_{\overline{S}}(\mathsf{CycleRun}, n)$:

$\text{Const}_{\overline{S}}(\mathsf{CycleRun}, n)$

1. Initialize $\overline{S} = \emptyset$, $\overline{S}' = \emptyset$.
2. Initialize $\sigma = 2^n \cdot \sum_{i \in [n]} \widehat{\mathsf{CycleRun}}(\{i\})$.
3. While $|\overline{S}| \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$, do the following:
   - 3*a*. Find some x with maximal value of $|x|$ such that $\mathsf{CycleRun}(x) = 1$ and such that $x \notin \overline{S}$.
   - 3*b*. If $\sigma - 2|\text{Shift}_x| \cdot |x| < 0$, then find an $x^* \notin \overline{S}$ such that $|x^*| = 1$ and $\mathsf{CycleRun}(x^*) = 1$ (if no such x^* exists, exit loop and output "Fail."). Then set $\overline{S} := \overline{S} \cup \text{Shift}_{x^*}$, set $\overline{S}' = \overline{S}' \cup \text{Shift}_{x^*}$, and set $\sigma := \sigma - 4n$. If $\sigma = 0$, exit the loop.
   - 3*c*. If $\sigma - 2|\text{Shift}_x| \cdot |x| > 0$, set $\overline{S} := \overline{S} \cup \text{Shift}_x$ and set $\sigma := \sigma - 2|\text{Shift}_x| \cdot |x|$.
4. Return $\overline{S}$.

Given the set $\overline{S}$ outputted by $\text{Const}_{\overline{S}}(\mathsf{CycleRun}, n)$, the function $f : \{-1,1\}^n \to \{-1,1\}$

is defined in the following way:

$$f(x) = \begin{cases} \mathsf{CycleRun}(x) & \text{if } x \notin \overline{S} \\ -\mathsf{CycleRun}(x) & \text{if } x \in \overline{S}. \end{cases}$$

Clearly, $\mathbf{E}_{\boldsymbol{x}}[f(\boldsymbol{x}) \cdot \mathsf{CycleRun}(\boldsymbol{x})] \geq 1 - 2c_1 \cdot \sqrt{\frac{\log(n)}{n}}$, since the set $\overline{S}$ satisfies $|\overline{S}| \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$. Additionally, f is clearly balanced due to the structure of the set Shift_x of modified points in each iteration of $\mathrm{Const}_{\overline{S}}$ and the fact that $\mathsf{CycleRun}$ is odd. Thus, it remains to show that $\widehat{f}(S) = 0$ for all $S \subseteq [n]$ such that $|S| \leq 1$.

Claim 3.4.1. *Consider an execution of* $\mathrm{Const}_{\overline{S}}$*. At the end of the* i*-th iteration,* $1 \leq i \leq c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$*, if* $\mathrm{Const}_{\overline{S}}$ *has not terminated, let* $\overline{S}^i$ *denote the current set of points in* $\overline{S}$*, let* σ^i *denote the current setting of the variable* σ *and let* f^i *denote the following Boolean function:*

$$f^i(x) = \begin{cases} \mathsf{CycleRun}(\boldsymbol{x}) & \text{if } \boldsymbol{x} \notin \overline{S}^i \\ -\mathsf{CycleRun}(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in \overline{S}^i. \end{cases}$$

Additionally, we define $\overline{S}^0 = \emptyset$*,* $\sigma^0 = 2^n \cdot \sum_{i \in [n]} \widehat{\mathsf{CycleRun}}(\{i\})$*, and* $f^0 = \mathsf{CycleRun}$*.*

For every $0 \leq i \leq c_1 \cdot \frac{\log(n)}{2n\sqrt{n}} \cdot 2^n$ *the following invariants hold:*

1. $\widehat{f^i}(\{1\}) = \widehat{f^i}(\{2\}) = \cdots = \widehat{f^i}(\{n\})$.
2. $\sigma^i = 2^n \cdot \sum_{j \in [n]} \widehat{f^i}(\{j\})$.
3. $\sigma^i = 4nw \geq 0$*, for some integer* w*.*

Proof. Proof by induction.

Base Case: The base case follows trivially from the definition of $\mathsf{CycleRun}$ and the definition of $\overline{S}^0$, σ^0, f^0.

Inductive Case: Assume the invariants hold for all $0 \leq j \leq i < c_1 \cdot \sqrt{\frac{\log(n)}{n}} \cdot 2^n$, we show that the invariants must also hold for $i + 1$.

For every $j \in [n]$, let us consider the quantity $2^n \left(\widehat{f^i}(\{j\}) - \widehat{f^{i+1}}(\{j\}) \right)$. Note that by flipping the value of f^i on the points in the set Shift_x, $\widehat{f^i}(\{j\})$ is reduced by exactly

$1/2^n \cdot 4 \cdot \frac{|\text{Shift}_x| \cdot |x|}{2n}$ for each $j \in [n]$ and so we have that $\widehat{f^{i+1}}(\{1\}) = \widehat{f^{i+1}}(\{2\}) = \cdots = \widehat{f^{i+1}}(\{n\})$. Moreover, $2^n \left(\sum_{j \in [n]} \widehat{f^i}(\{j\}) - \sum_{j \in [n]} \widehat{f^{i+1}}(\{j\}) \right) = 2|\text{Shift}_x| \cdot |x|$ and so we have that

$$\begin{aligned} \sigma^{i+1} &= \sigma^i - 2|\text{Shift}_x| \cdot |x| \\ &= 2^n \cdot \sum_{j \in [n]} \widehat{f^i}(\{j\}) - 2|\text{Shift}_x| \cdot |x| \\ &= 2^n \cdot \sum_{j \in [n]} \widehat{f^{i+1}}(\{j\}), \end{aligned}$$

where the second equality holds by the induction hypothesis.

Finally, since $\sigma^{i+1} = 2^n \cdot \sum_{j \in [n]} \widehat{f^{i+1}}(\{j\})$ and f^{i+1} is an odd $\{-1, 1\}$-valued function, we have that $\sigma^{i+1} = 4nw$ for some integer $w \geq 0$.

□

We proceed to show that $\text{Const}_{\overline{S}}$ terminates. Our goal is to show that at the termination of the algorithm, we have $\sigma = 0$.

Claim 3.4.2. *The algorithm* $\text{Const}_{\overline{S}}$ *always reaches a point where the condition in line* 3*b is true.*

Proof. We use the functions f_t from the beginning of this section. Take

$$t' = \sqrt{\log n - 2 \log \log n - C}$$

for a constant C to be determined later. Then $\phi(t') = \frac{1}{2\pi} e^{C/2} (\log n)/\sqrt{n}$, so $\mathbf{Inf}(f_{t'}) \geq \frac{1}{6\pi} e^{C/2} \log n$ and $\mathbf{Pr}_x[f_{t'}(x) \neq 0] \leq \frac{3}{2\pi} e^{C/2}/t' \leq \frac{3}{\pi} e^{C/2} \sqrt{\log n / n}$ by Equations 3.5 and 3.6 respectively. We choose C so that $\mathbf{Inf}(f_{t'}) \geq 3 \cdot \mathbf{Inf}(\mathsf{CycleRun})$, which can be done since $\mathbf{Inf}(\mathsf{CycleRun}) = O(\log n)$.

We claim that $\text{Const}_{\overline{S}}$ does not include any strings x in $\overline{S}$ with $3 \leq |x| < t'$ (and thus none with $-t' < |x| \leq -3$). Suppose that this claim is false. Because the algorithm is greedy, then every string x where $\mathsf{CycleRun}(x) = 1$ with $t' \leq |x| \leq n$ is corrupted and in $\overline{S}$. Since $\mathsf{CycleRun}$ is odd and monotone, at least half of the strings where $|x| = k$ are corrupted for $t' \leq k \leq n$. The contribution to be reduction in the first-order Fourier coefficients when

we flip the value on these strings from 1 to -1 is at least $(1/2)\mathbf{Inf}(f'_t) \geq (3/2)\mathbf{Inf}(\mathsf{CycleRun})$. But this implies that the sum of first-order Fourier coefficients for the corrupted function is at most $-(1/2)\mathbf{Inf}(\mathsf{CycleRun}) < 0$. This implies that $\sigma < 0$ in the execution of $\mathrm{Const}_{\overline{S}}$, which is a contradiction since σ stays nonnegative during the execution of the algorithm.

It remains to show that the condition in line 3 is satisfied throughout the execution of $\mathrm{Const}_{\overline{S}}$. Because no strings with $3 \leq |x| < t'$ or $t' < |x| \leq -3$ are corrupted, the fraction of strings corrupted is at most $\mathbf{Pr}_{\boldsymbol{x}}[f_{t'}(\boldsymbol{x}) \neq 0] + \mathbf{Pr}_{\boldsymbol{x}}[|\boldsymbol{x}| = \pm 1] = O(\sqrt{\log n/n})$. Thus at most $c_1\sqrt{\frac{\log n}{n}}2^n$ strings are in $\overline{S}$, so the condition in line 3 holds. □

Next, we argue that when $\mathrm{Const}_{\overline{S}}$ reaches the point where the condition in line $3b$ evaluates true, there always exists a point $x^* \notin \overline{S}$ such that $\mathsf{CycleRun}(x^*) = 1$ and $|x^*| = 1$. We first prove two lemmas.

Lemma 3.4.3. *Let S_1^1 be the set of $x \in \{-1,1\}^n$ such that $|x| = 1$ and $\mathsf{CycleRun}(x) = 1$. Then $|S_1^1| \geq 2n^2$.*

Proof. Note that since $\mathsf{CycleRun}$ is odd, we have that $\sum_{x:|x|=\pm 1} \mathsf{CycleRun}(x) = 0$. Moreover, since $\mathsf{CycleRun}$ is monotone, we must have that $\sum_{x:|x|=1} \mathsf{CycleRun}(x) \geq \sum_{x:|x|=-1} \mathsf{CycleRun}(x)$. Therefore, we must have that $\sum_{x:|x|=1} \mathsf{CycleRun}(x) \geq 0$. Since $\mathsf{CycleRun}$ is $\{-1,1\}$-valued, this immediately implies that at least half of the points x where $|x| = 1$ are such that $\mathsf{CycleRun}(x) = 1$. There are $\binom{n}{(n-1)/2} \geq 4n^2$ such strings where $|x| = 1$, so we have that $|S_1^1| \geq 2n^2$. This concludes the proof of Lemma 3.4.3. □

Lemma 3.4.4. $|\overline{S}'| \leq 2n^2$.

Proof. Consider the first time the condition in line $3b$ evaluates to true. Then there is some some x such that $\mathsf{CycleRun}(x) = 1$ and such that $\sigma - 2|\mathrm{Shift}_x| \cdot |x| < 0$. Since $|x| \leq n$, this implies that $\sigma \leq 4n^2$. Moreover, in each iteration $2n$ points are added to $\overline{S}'$, and σ is reduced by $4n$. Thus, after at most n iterations, σ is reduced to 0. These iterations are the only iterations that contribute to $\overline{S}'$, so $|\overline{S}'| \leq n \cdot 2n = 2n^2$ as claimed. □

We proceed to show that the when the condition in line $3b$ is true, there is an $x^* \notin \overline{S}$ such that $\mathsf{CycleRun}(x^*) = 1$ and $|x^*| = 1$. By Lemma 3.4.3, there exist at least $2n^2$ number of points x^* such that $\mathsf{CycleRun}(x^*) = 1$ and $|x^*| = 1$. Thus, if $\mathrm{Const}_{\overline{S}}$ reaches a point

where the condition in line $3b$ evaluates to true and there is no point $x^* \notin \overline{S}$ such that $\mathsf{CycleRun}(x^*) = 1$ and $|x^*| = 1$, then it must be the case that all such x^* are already contained in $\overline{S}$. But since we have by Lemma 3.4.4 that $|\overline{S}'| \leq 2n^2$ then we must have that some point y such that $\mathsf{CycleRun}(y) = 1$ and $|y| = 1$ was added to $\overline{S}$ before the first time the condition in line $3b$ evaluates to true. But the first time the condition in line $3b$ evaluates to true, we must have that $|x| > 1$, and since $\mathrm{Const}_{\overline{S}}$ always chooses to add points y with maximal $|y| \geq |x| > 1$ to the set $\overline{S}$, this is impossible.

We have now argued that $\mathrm{Const}_{\overline{S}}$ always reaches a point where the condition in line $3b$ is true, and that whenever this occurs there always exists a point $x^* \notin \overline{S}$ such that $\mathsf{CycleRun}(x^*) = 1$ and $|x^*| = 1$. This immediately implies that when $\mathrm{Const}_{\overline{S}}$ completes, we have $\sigma = 0$ and $|\overline{S}| \leq c_1\sqrt{\frac{\log n}{n}}2^n$. As in the beginning of the proof, we take f to be function to be the function such that

$$f(x) = \begin{cases} \mathsf{CycleRun}(x) & \text{if } x \notin \overline{S} \\ -\mathsf{CycleRun}(x) & \text{if } x \in \overline{S}. \end{cases}$$

Clearly, $\mathbf{Pr}_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq \mathsf{CycleRun}(\boldsymbol{x})] = |\overline{S}| \leq c_1\sqrt{\frac{\log n}{n}}2^n$, and applying the invariants of Claim 3.4.1 shows that f is 1-resilient, concluding the proof of Theorem 28. □

This analysis almost works for any balanced monotone function with influence $O(\log n)$, such as Tribes. While the above could be adapted in a straightforward matter to show that there is a Boolean function close to Tribes with very small constant and first-order Fourier coefficients, showing that all of these Fourier coefficients can be made *exactly* zero seems challenging. Since we are applying these results to juntas, our proofs can not tolerate even exponentially small Fourier coefficients. The structure of $\mathsf{CycleRun}$ is quite amenable to "local" changes while retaining structure.

3.4.1 Influence bound for Cycle Run

The main result of this section is the following:

Theorem 33. $\mathbf{Inf}(\mathsf{CycleRun}) = O(\log n)$.

The condition on CycleRun given in Definition 27 implies that for every influential edge $(x, x^{\oplus i})$, at least one of the endpoints is in the first two cases in Definition 27, and the pivotal coordinate i occurs in a maximum length run. Thus $\mathbf{Inf}(\mathsf{CycleRun}) \leq 2\,\mathbf{E}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x}) \cdot (r_{\ell(\boldsymbol{x})}(\boldsymbol{x}) + 1)]$, where $\ell(x)$ is the maximum length run in the string x, $r_i(x)$ is the number of maximal runs of length exactly i in x, and $\mathcal{U}$ is the uniform distribution on $\{-1, 1\}^n$. In this section, we will not consider the runs wrapping around, and the $+1$ here takes care of the case that we "split" the cycle in a maximum length run to lay out the bits in a line.

We make use of a result from [Schilling, 1990]:

Theorem 34. $\mathbf{E}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})] = O(\log n)$

Thus $\mathbf{Inf}(\mathsf{CycleRun}) \leq 2\,\mathbf{E}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x}) \cdot r_{\ell(\boldsymbol{x})}(\boldsymbol{x})] + O(\log n)$, so the remainder of the section is devoted to showing $\mathbf{E}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x}) \cdot r_{\ell(\boldsymbol{x})}(\boldsymbol{x})] = O(\log n)$. To aid in our analysis, we will consider different distributions over binary strings. Consider the following method of generating a string $\boldsymbol{x} \sim \mathcal{U}$:

1. Initialize $\boldsymbol{x}$ to the empty string, and set b to a uniform ± 1 random bit $\boldsymbol{b}$.
2. (Iterative step) Assuming there are still $j > 0$ bits of $\boldsymbol{x}$ to determine, then draw $\boldsymbol{g} \sim$ Geometric$(1/2)$ conditioned on $\boldsymbol{g}$ being at most j, and set the next $\boldsymbol{g}$ bits of $\boldsymbol{x}$ to b.
3. If not all n bits of $\boldsymbol{x}$ are set, set b to $-b$ and return to step 2.
4. If all bits of $\boldsymbol{x}$ are set, then $\boldsymbol{x}$ is a uniformly random string in $\{-1, 1\}^n$.

Further, if we want to condition on the maximum run in $\boldsymbol{x}$ being at most some value t, we can replace the conditioning in step 2 from "being at most j" to "being at most $\min\{t, j\}$".

Lemma 3.4.5. *For $\boldsymbol{g} \sim$ Geometric$(1/2)$, and $1 \leq g \leq t$, we have $\mathbf{Pr}[\boldsymbol{g} = g \mid \boldsymbol{g} \leq t] \leq 2\,\mathbf{Pr}[\boldsymbol{g} = g]$.*

Proof. Follows directly from conditional probability and the fact that $\mathbf{Pr}[\boldsymbol{g} \leq t] \geq 1/2$ for all $t \geq 1$. □

For an integer $k > 0$, we define the distribution $\mathcal{G}_k$ on binary strings of varying length such that a draw from $\mathcal{G}_k$ is $\boldsymbol{b}^{\boldsymbol{g}_1}(-\boldsymbol{b})^{\boldsymbol{g}_2}\boldsymbol{b}^{\boldsymbol{g}_3}\cdots\boldsymbol{b}^{\boldsymbol{g}_k}$ if k is odd and $\boldsymbol{b}^{\boldsymbol{g}_1}(-\boldsymbol{b})^{\boldsymbol{g}_2}\boldsymbol{b}^{\boldsymbol{g}_3}\cdots(-\boldsymbol{b})^{\boldsymbol{g}_k}$ if k is even. Here, the $\boldsymbol{g}_i$'s are independent Geometric(1/2) variables, and $\boldsymbol{b}$ is a uniform ± 1 bit.

Lemma 3.4.6.

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})\cdot r_{\ell(\boldsymbol{x})}(\boldsymbol{x})|\ell(\boldsymbol{x})=t]\leq t(2^{1-t}n+1)$$

Proof. We first claim that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})\cdot r_{\ell(\boldsymbol{x})}(\boldsymbol{x})|\ell(\boldsymbol{x})=t]\leq t+\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})\cdot r_t(\boldsymbol{x})|\ell(\boldsymbol{x})\leq t]$$

To see this, note that if we further condition on the first run of length t selected, this expectation is maximized when the first run is of length t. Also, the expectation can only increase if we allow all n more bits to be set rather than $n-t$. Since the first run is of length t, we only need the maximum length run to be at most t in the rest of the string.

Now we have

$$t+\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})\cdot r_t(\boldsymbol{x})|\ell(\boldsymbol{x})\leq t]\leq t+t\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{U}}[r_t(\boldsymbol{x})|\ell(\boldsymbol{x})\leq t]\leq t+t\mathop{\mathbf{E}}_{\boldsymbol{y}\sim\mathcal{G}_n}[r_t(\boldsymbol{y})|\ell(\boldsymbol{y})\leq t]$$

where the second inequality comes from the fact that $\boldsymbol{x}$ is generated by at most n runs, and not bounding the length of the string only increases the possible number of runs of length t, conditioned on the maximum length run being at most t. By Lemma 3.4.5, the probability of a single run being of length t is at most 2^{1-t}, so we have

$$t+t\mathop{\mathbf{E}}_{\boldsymbol{y}\sim\mathcal{G}_n}[r_t(\boldsymbol{y})|\ell(\boldsymbol{y})\leq t]\leq t+t(2^{1-t}n)=t(2^{1-t}n+1)$$

completing the proof. □

Lemma 3.4.7.

$$\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})\leq t]\leq(1-2^{-t})^{n/8}+\exp(-n/32)$$

Proof. For $\boldsymbol{x}\in\{-1,1\}^n$, let runs($\boldsymbol{x}$) be the number of runs in $\boldsymbol{x}$. We first show that with probability at least $1-\exp(-n/32)$, a string $\boldsymbol{x}\sim\mathcal{U}$ has runs($\boldsymbol{x}$) $\geq n/8$. To do this, we prove that with probability $1-\exp(-n/32)$, the first $n/8$ runs of $\boldsymbol{x}$ contain at most $n/2$

bits. Note that we may instead bound the number of bits in $\boldsymbol{y} \sim \mathcal{G}_{n/8}$, since each run of $\mathcal{G}_{n/8}$ can only be longer.

The expected number of bits in $\mathcal{G}_{n/8}$ generated is $n/4$, and this number of bits is concentrated around its mean; the number of bits has a negative binomial distribution. We have

$$\mathbf{Pr}_{\boldsymbol{y} \sim \mathcal{G}_{n/8}}[\text{bits}(\boldsymbol{y}) > 2(n/4)] \leq \exp(-n/32),$$

since the number of runs does not increase the probability of getting a longer run, and the distributions of the lengths of each run in $\boldsymbol{x}$ are identical to (or conditioned on being shorter than) the lengths of the runs in $\mathcal{G}_{n/8}$. We then have:

$$\begin{aligned}\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{U}}[\ell(\boldsymbol{x}) \leq t] &\leq \mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{U}}[\ell(\boldsymbol{x}) \leq t, \text{runs}(\boldsymbol{x}) \geq n/8] + \exp(-n/32) \\ &\leq \mathbf{Pr}_{\boldsymbol{y} \sim \mathcal{G}_{n/8}}[\ell(\boldsymbol{y}) \leq t] + \exp(-n/32)\end{aligned}$$

where the second inequality holds because the length of each run of $\boldsymbol{x}$ is distributed identically (or conditioned to be shorter) to each run of $\boldsymbol{y}$, and considering fewer runs only decreases the chances of obtaining a run longer than t. It is then straightforward to calculate $\mathbf{Pr}_{\boldsymbol{y} \sim \mathcal{G}_{n/8}}[\ell(\boldsymbol{y}) \leq t] = (1 - 2^{-t})^{n/8}$, since $\mathbf{Pr}[\boldsymbol{g} \leq t] = 1 - 2^{-t}$ for $\boldsymbol{g} \sim \text{Geometric}(1/2)$. □

We now proceed to show $\mathbf{E}_{\boldsymbol{x} \sim \mathcal{U}}[\ell(\boldsymbol{x}) \cdot r_{\ell(\boldsymbol{x})}] = O(\log n)$, starting by applying total expectation and applying Lemma 3.4.6:

$$\begin{aligned}
\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})\cdot r_{\ell(\boldsymbol{x})}(\boldsymbol{x})] &= \sum_{t=1}^{n}\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})=t]\,\mathbf{E}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})\cdot r_\ell|\ell(\boldsymbol{x})=t] \\
&\le \sum_{t=1}^{n}\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})=t]t(2^{1-t}n+1) \\
&\le \mathbf{E}_{x\sim U}[\ell(\boldsymbol{x})] + \sum_{t=1}^{n}\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{U}}[\ell(\boldsymbol{x})=t]t2^{1-t}n \\
&\le O(\log n) + \sum_{t=1}^{n}((1-2^{-t})^{n/8}+\exp(-n/32))t2^{1-t}n \\
&\le O(\log n) + \sum_{t=1}^{n}(1-2^{-t})^{n/8})t2^{1-t}n \\
&\le O(\log n) + \sum_{t=1}^{n}tn2^{1-t}\exp(-2^{-t}n/8)
\end{aligned}$$

Letting $a_t = tn2^{1-t}\exp(-2^{-t}n/8)$, we see that $a_{t-1}/a_t < 3/4$ when $2 \le t \le \log n - 10$, and $a_{t+1}/a_t < 3/4$ when $\log n + 10 \le t \le n$. Also, $a_t \le O(\log n)$ for each term where $\log n - 10 \le t \le \log n + 10$. So the proof is completed by noting the above is at most

$$\begin{aligned}
&O(\log n) + \sum_{t=2}^{\log n-10} a_{\log n-10}(3/4)^{\log n-10-t} + \sum_{t=\log n-9}^{t=\log n+9} a_t + \sum_{t=\log n+10}^{n} a_{\log n+10}(3/4)^{t-(\log n+10)} \\
&\le O(\log n)\left(\sum_{t=2}^{\log n-10}(3/4)^{\log n-10-t} + \sum_{t=\log n-9}^{t=\log n+9} 1 + \sum_{t=\log n+10}^{n}(3/4)^{t-(\log n+10)}\right) = O(\log n).
\end{aligned}$$

3.4.2 Lower bound for monotonicity-resiliency distance

We give a lower bound for distance between monotonicity and resiliency that matches the bound for CycleRun up to constant factors.

Theorem 35. *For every monotone function $f : \{-1,1\}^n \to \{-1,1\}$ and 1-resilient $g : \{-1,1\}^n \to \{-1,1\}$, we have $\mathbf{Pr}_{\boldsymbol{x}}[f(\boldsymbol{x}) \ne g(\boldsymbol{x})] \ge \Omega(\sqrt{\frac{\log n}{n}})$.*

Proof. If $\mathbf{Var}[f] < 1/2$, then $\widehat{f}(\emptyset)^2 > 1/2$, and $\mathbf{Pr}[f \neq g] \geq \frac{1}{4}E[(f-g)^2] \geq 1/8$ for any balanced (hence 1-resilient) Boolean function g. If $\widehat{f}(\{i\}) > n^{-0.49}$ for some i, then f is $\Omega(n^{-0.49})$-far from every Boolean function g where $\widehat{g}(\{i\}) = 0$.

We assume $\mathbf{Var}[f] \geq 1/2$ and $\widehat{f}(\{i\}) \leq n^{-0.49}$ for all $i \in [n]$. Since f is monotone, $\mathbf{Inf}_i(f) \leq n^{-0.49}$ for all $i \in [n]$, and by (Talagrand's strengthening of) the KKL Theorem [Talagrand, 1993; Kahn *et al.*, 1988], $\mathbf{Inf}(f) \geq K \log n$ for some constant K, and $\sum_{i\in[n]} \widehat{f}(\{i\}) \geq K \log n$. Let $g : \{-1,1\}^n \to \{-1,1\}$ be a 1-resilient Boolean function; we will show that $\mathbf{Pr}_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] = \Omega(\sqrt{\frac{\log n}{n}})$.

Recall the functions f_t defined earlier:

$$f_t(x) = \begin{cases} 1 & \text{if } |x| > t\sqrt{n} \\ 0 & \text{if } -t\sqrt{n} \leq |x| \leq t\sqrt{n} \\ -1 & \text{if } |x| < -t\sqrt{n} \end{cases}$$

Select t to be the largest t such that f_t satisfies $\mathbf{Pr}[f_t(\boldsymbol{x}) \neq 0] \geq \mathbf{Pr}[(f-g)(\boldsymbol{x}) \neq 0] = \mathbf{Pr}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})]$. We then have $K \log n \leq \sum_{i\in[n]} \widehat{f-g}(\{i\}) \leq \sum_{i\in[n]} \widehat{f}_t(\{i\})$, where the second inequality holds because f_t maximizes the sum of the linear coefficients for any function with support size $\mathbf{Pr}[f_t(\boldsymbol{x}) \neq 0]$, and the support size of f_t is at least the support size of $f - g$.

Again, because f_t is monotone, $\mathbf{Inf}(f_t) = \sum_{i\in[n]} \widehat{f}_t(\{i\})$. Equation 3.5 implies that $(3K \log n)/\sqrt{n} \geq \phi(t) \geq (K \log n)/(3\sqrt{n})$, and it follows that $t \leq 4\sqrt{\log n}$. From Equation 3.6, we have $\mathbf{Pr}_{\boldsymbol{x}}[f_t(\boldsymbol{x}) \neq 0] \geq (4K/3)\sqrt{\frac{\log n}{n}}$. By the choice of t, we have

$$\begin{aligned} \mathbf{Pr}_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] &> \mathbf{Pr}_{\boldsymbol{x}}[f_{t+1}(\boldsymbol{x}) \neq 0] \\ &\geq \mathbf{Pr}_{\boldsymbol{x}}[f_t(\boldsymbol{x}) \neq 0] - 2\,\mathbf{Pr}_{\boldsymbol{x}}[|\boldsymbol{x}| = t] \\ &\geq \frac{4K}{3}\sqrt{\frac{\log n}{n}} - 24K\frac{\log n}{n} = \Omega\left(\sqrt{\frac{\log n}{n}}\right), \end{aligned}$$

where the first inequality is an application of the union bound, and the second is an application of Equation 3.7. □

Part III

Boolean Function Complexity

Chapter 4

Approximating Boolean Functions by Small-Depth Circuits

4.1 Background and context

The study of the DNF complexity of Boolean functions is one of the great success stories in complexity theory. Among the many remarkably precise results in this area, let us highlight three:

Lupanov's Theorem ([Lupanov, 1961]). *Any DNF computing the parity function* PAR_n *has size* 2^{n-1} *and width* n. *Furthermore, every Boolean function can be computed by a DNF of size* 2^{n-1} *and width* n *so the parity function has the largest DNF size and width complexity of all Boolean functions.*

Quine's Theorem ([Quine, 1954]). *Any DNF computing the majority function* MAJ_n *has size at least* $\binom{n}{n/2}$ *and width at least* $n/2$. *Furthermore,* every *monotone Boolean function can be computed by a DNF of size* $\binom{n}{n/2}$ *so* MAJ_n *is the hardest monotone function to compute with respect to DNF size.*

Korshunov–Kuznetsov Theorem ([Korshunov, 1983; Kuznetsov, 1983][1]). *The optimal DNF size for a random Boolean function is* $(K + o(1))(2^n/\log n \log\log n)$, *where* $1 \leq K \leq$

[1]The Korshunov–Kuznetsov theorem is the culmination of a long line of research [Glagolev, 1967; Korshunov, 1969; Sapozhenko, 1972; Korshunov, 1981; Korshunov, 1983; Kuznetsov, 1983; Pippenger, 2003].

1.54169.

Our understanding of the DNF complexity of Boolean functions extends beyond the minimum size of the DNFs computing specific functions or classes of functions. Notably, we have a good understanding of the maximum possible correlation of small-size DNFs with the parity function. Building on a long line of work originally motivated by the goal of showing that PAR_n is not in AC^0 [Furst *et al.*, 1984; Ajtai, 1983; Yao, 1985; Håstad, 1986; Cai, 1989; Linial *et al.*, 1989], and improving on a recent result of Beame *et al.* [Beame *et al.*, 2012], Impagliazzo *et al.* and Håstad recently showed that any DNF of size s has correlation at most $2^{-\Omega(n/\log s)}$ with PAR_n [Impagliazzo *et al.*, 2012; Håstad, 2012].

In this work we are interested in the DNF complexity of *approximating* Boolean functions. Specifically, we say that a DNF *ε-approximates* the function $f : \{0,1\}^n \to \{0,1\}$ if the function $g : \{0,1\}^n \to \{0,1\}$ computed by the DNF satisfies $f(x) = g(x)$ for all but an ε fraction of the inputs $x \in \{0,1\}^n$. The explicit study of the DNF complexity of approximating Boolean functions was initiated by O'Donnell and Wimmer [O'Donnell and Wimmer, 2007], who showed that for any constant $\varepsilon > 0$ there is a DNF of size $2^{O(\sqrt{n})}$ that ε-approximates MAJ_n. They also showed that there exists a monotone function for which every DNF that $\frac{1}{100}$-approximates it must have size $2^{\Omega(n/\log n)}$. A comparison of these results with Quine's theorem shows that the DNF complexity of Boolean functions is strikingly different in the exact and approximate computation models: quantitatively, we see that it is possible to approximate the majority function with much smaller DNFs than those that compute majority exactly, and qualitatively we see that, unlike in the exact computation model, the majority function is far from the hardest monotone function to approximate in terms of DNF size.

We continue the study of the DNF complexity of approximating Boolean functions, focusing especially on the analogues of Lupanov's theorem and the Korshunov–Kuznetsov theorem in the approximation model. As we discuss next, our results further illustrate how different DNF complexity can be in exact and approximate computation models.

For more discussion of the history of this theorem and elegant proofs of its components, we highly recommend Pippenger's article [Pippenger, 2003].

Universal bounds on approximability We begin with a simple question: is there a non-trivial universal upper bound on the size or width of DNFs for approximating *any* Boolean function? To the best of our knowledge, this question has not been considered explicitly before and a large gap exists between the trivial upper bounds and the known limitations on any potential universal upper bound. In fact, we believe that it is not even known whether there is a Boolean function for which every ε-approximating DNF has size[2] $\Omega_\varepsilon(2^n)$, whether every function can be ε-approximated by a DNF of size $2^{c_\varepsilon n}$ for some $c_\varepsilon < 1$ (*i.e.* whether every function can be approximated by a DNF that is *exponentially smaller* than the trivial DNF), or whether the best universal upper bound lies somewhere in the middle. Likewise, it is unknown whether there is a Boolean function for which every ε-approximating DNF has width $n - O_\varepsilon(1)$, whether every function can be ε-approximated by a DNF of width $c_\varepsilon n$ for some $c_\varepsilon < 1$ (*i.e.* whether every function can be approximated by a union of subcubes that each have dimension *linear* in n) or, once again, whether the best universal upper bound on the width of approximating DNFs is somewhere in between.

We answer these questions in the first part of this chapter. Our main positive results are the first non-trivial universal upper bounds on the approximability of all Boolean functions with respect to both DNF size and width. Our universal bound on DNF width is asymptotically optimal, and we accomplish this via a connection between approximating DNFs and low-density coverings of the Boolean hypercube by Hamming balls. We show that this technique extends rather broadly to give universal bounds on approximability by various generalizations of DNFs, including the intersection of halfspaces, low-degree PTFs, and unate functions. We complement these upper bounds with near-matching information-theoretic lower bounds against a random function.

Approximating the parity function In the second part of this chapter we turn our attention to the parity function. Despite decades of intensive study of the circuit complexity of this function, large gaps remain between the minimal size and width of DNFs that ε-

[2]For clarity of presentation in informal discussions, we use the notation $O_\varepsilon(\cdot)$ and $\Omega_\varepsilon(\cdot)$ to represent asymptotic behaviors when ε is a fixed constant. For the dependencies on ε in the bounds, see the corresponding theorem statements in the main body of this chapter.

approximate PAR_n for constant values of $\varepsilon > 0$. For instance, while Håstad's correlation bounds between PAR_n and AC^0 [Håstad, 2012] imply that any DNF ε-approximating PAR_n must have size $2^{\Omega_\varepsilon(n)}$, the precise dependence on ε in this bound is unclear, leaving open a wide range of possibilities. On one extreme, it is possible that the true lower bound is $\Omega_\varepsilon(2^n)$ (so that we only gain a linear savings on the size of DNFs by requiring them to approximate parity rather than compute it directly) and on the other it is possible that the true bound is $2^{c_\varepsilon n}$ for some $c_\varepsilon < 1$ (so that DNFs that approximate PAR_n are *exponentially* smaller than those that compute the same function exactly).

We resolve this question by showing, perhaps somewhat surprisingly, that the right answer falls in the latter extreme. Our construction of an explicit DNF approximator for PAR_n, when combined with information-theoretic lower bounds against a random function, also shows that the landscape of DNF complexity changes dramatically when we move from exact to approximate computation: while PAR_n is the hardest function to compute exactly by DNFs, it can be approximated by DNFs of size exponentially smaller than that required for almost every other function. The remainder of this chapter is then devoted to proving the optimality of our DNF approximator for PAR_n. Using Fourier analytic tools, we show that the dependence on ε in our DNF construction is essentially the best possible, and furthermore, our construction is near-optimal even within the far more expressive classes of the intersection of halfspaces and the intersection of unate functions.

4.1.1 Our results

Universal bounds Our first result is the first non-trivial universal upper bound on the size of DNF approximators for all Boolean functions.

Theorem 36. *Every Boolean function can be ε-approximated by a DNF of size $O_\varepsilon(2^n/\log n)$.*

The proof of Theorem 36, presented in Section 4.2, is obtained with a randomized algorithm that constructs an explicit approximating DNF. In Section 4.3 we complement Theorem 36 with an asymptotically optimal universal upper bound on the width of DNFs required to approximate any function.

Complexity measure	**Upper bound for all functions**	**Lower bound for almost all functions**
DNF width	$c_\varepsilon\, n \quad (c_\varepsilon < 1)$	$\Omega(n)$
DNF size	$O\left(2^n/\log n\right)$	$\Omega\left(2^n/n\right)$
AND of halfspaces	$(1+o(1))\cdot 2^n/n$	$\Omega\left(2^n/n^2\right)$
AND of unate functions	$2^{c_\varepsilon n} \quad (c_\varepsilon < 1)$	$2^{\Omega(n)}$ for $\varepsilon < \frac{1}{16}$
AND of degree-d PTFs	$O\left(2^n/n^d\right)$	$\Omega\left(2^n/n^{d+1}\right)$

Table 4.1: Approximating any Boolean function to constant accuracy.

Theorem 37. *Every Boolean function can be ε-approximated by a DNF of width $c_\varepsilon\, n$, where $c_\varepsilon < 1$ depends only on ε.*

Theorem 37 highlights an interesting (and stark) contrast between exact and approximate computation with respect to DNF width: not only is the DNF width of a random Boolean function at least $n - \log(3n)$, *every* term in any DNF computing it has to have width at least $n - \log(3n)$ (see *e.g.* Theorem 3.21 of [Crama and Hammer, 2011]). Therefore, while every 1-monochromatic subcube in a random function has dimension at most $\log(3n)$, Theorem 37 shows that every Boolean function can be ε-approximated by the union of 1-monochromatic subcubes *all* of which have dimension $\Omega_\varepsilon(n)$.

Theorem 37 is obtained by exploiting a connection between approximating DNFs and low-density coverings of the Boolean cube by Hamming balls. This technique extends rather broadly to give strong universal bounds on approximability by the intersection of unate functions and low-degree PTFs.

Theorem 38. *Every Boolean function can be ε-approximated by the intersection of $2^{c_\varepsilon n}$ unate functions, where $c_\varepsilon < 1$ depends only on ε.*

Theorem 39. *For every positive integer d, every Boolean function can be $O(1/n)$-approximated by the intersection of $O(2^n/n^d)$ degree-d PTFs.*

Using a theorem of Kabatyanski and Pachenko on the existence of asymptotically perfect covering codes of radius 1 [Kabatyanski and Panchenko, 1988], we obtain improvements on

Complexity measure	Upper bound	Lower bound
DNF width	$(1-2\varepsilon)n$	$(1-2\varepsilon)n$
DNF size	$2^{(1-2\varepsilon)n}$	$\max_{\delta>0}\{\delta\, 2^{(1-2\varepsilon-2\delta)n},\ (\frac{1}{2}-\varepsilon)2^{\frac{1-2\varepsilon}{1+2\varepsilon}n}\}$
AND of halfspaces	$2^{(1-2\varepsilon)n}$	$2^{\Omega_\varepsilon(n)}$ assuming Klivans *et al.*'s conjecture
AND of unate functions	$2^{(1-2\varepsilon)n}$	$2^{\Omega(n)}$ for $\varepsilon < \frac{1}{16}$

Table 4.2: Approximating PAR_n to accuracy ε.

both the accuracy and size of our approximators in Theorem 39 when $d = 1$ (*i.e.* the intersection of LTFs). See Section 4.3 for the details.

In Section 4.4 we turn our attention to lower bounds, giving a lower bound on the size complexity of approximating DNFs for almost all functions that nearly matches the universal upper bound of Theorem 36.

Theorem 40. *For almost every Boolean function f, any DNF that ε-approximates f has size $\Omega_\varepsilon(2^n/n)$.*

The proof of Theorem 40 is obtained by extending Pippenger's elegant information-theoretic proof [Pippenger, 2003] of Kuznetsov's theorem showing that the DNF size of a random Boolean function is at least $(1+o(1))(2^n/\log n \log\log n)$ [Kuznetsov, 1983]. Notably, our extension is rather general, and also implies strong bounds on the inapproximability of a random function by other types of depth-2 circuits; due to space considerations we do not list the associated corollaries here. We remark that our construction of small-width DNF approximators in Theorem 37 is asymptotically optimal. We prove in the second part of this chapter that any ε-approximator for PAR_n must have width at least $(1-2\varepsilon)n$, and also show that the same proof extends easily to show that almost every function has ε-approximating DNF width $\Omega_\varepsilon(n)$.

Approximating the parity function In Section 4.5 we turn our focus to the complexity of approximating the parity function. We begin with a deterministic construction of a DNF that approximates PAR_n with one-sided error.

Theorem 41. *The parity function can be ε-approximated by a DNF f of width $(1-2\varepsilon)n$ and size $2^{(1-2\varepsilon)n}$. Furthermore, f has one-sided error: if $\mathsf{PAR}_n(x) = 1$ then $f(x) = 1$.*

We point out the interesting contrast between this upper bound on size and the lower bound of Theorem 40: although PAR_n is the hardest function to compute exactly with respect to DNF size, Theorems 41 and 40 together show that it is in fact *exponentially* easier to approximate than *almost every* other function. We prove the optimality of our construction by giving matching lower bounds on the size and width of DNF approximators for PAR_n.

Theorem 42. *Any DNF that ε-approximates PAR_n has width at least $(1-2\varepsilon)n$.*

Theorem 43. *Any DNF that ε-approximates PAR_n has size at least*

$$s \geq \max\left\{ \max_{\delta>0} \delta\, 2^{(1-2\varepsilon-2\delta)n}, \left(\tfrac{1}{2}-\varepsilon\right) 2^{\frac{1-2\varepsilon}{1+2\varepsilon}n} \right\}.$$

The width lower bound is obtained by applying Amano's bound on the total influence of small-width DNFs [Amano, 2011], and the first size lower bound is obtained by combining Amano's theorem with an elementary truncation argument. The second lower bound on size uses a sharpening of Boppana's bound on the total influence of small-size DNFs [Boppana, 1997], obtained in concurrent work by the present authors using the entropy method [Blais and Tan, 2013].

In Section 4.6 we provide further evidence of the optimality of our DNF approximators for PAR_n. Assuming a noise sensitivity conjecture of Klivans *et al.* [Klivans *et al.*, 2004], we prove that ε-approximating PAR_n even with the intersection of halfspaces requires size $2^{\Omega_\varepsilon(n)}$, matching the size of our DNF approximators in Theorem 41.[3]

Theorem 44. *Assume the KOS conjecture holds. Let f be computed by the intersection of k halfspaces, and suppose f ε-approximates PAR_n. Then $k = 2^{\Omega_\varepsilon(n)}$.*

[3]Since the class of halfspaces is closed under negation, universal bounds on approximability by the intersection of halfspaces immediately imply identical universal bounds for the disjunction of halfspaces, a strict superclass of DNFs. Likewise for the intersection of unate functions, a further generalization of the intersection of halfspaces.

Naturally we would like an unconditional proof of Theorem 44. We are able to accomplish this for all $\varepsilon < \frac{1}{16}$, and in fact, our proof holds against the more expressive class of the intersection of unate functions.

Theorem 45. *Fix $\varepsilon < \frac{1}{16}$. Let f be computed by the intersection of k unate functions and suppose f ε-approximates PAR_n. Then $k = 2^{\Omega(n)}$.*

4.1.2 Preliminaries

All probabilities and expectations are with respect to the uniform distribution and logarithms are base 2 unless otherwise stated. For strings $x, y \in \{0,1\}^n$, we write $\mathrm{dist}(x,y)$ to denote the Hamming distance between x and y. We write $\mathrm{Vol}(d)$ to denote the quantity $\sum_{i=0}^{d} \binom{n}{d}$, the volume of a Hamming ball of radius d.

A DNF is the OR of ANDs, where each AND is referred to as a *term*. The size of a DNF is the number of terms in the DNF. The width of a term is the number of literals it contains, and the width of a DNF is the maximum width of any term in the DNF. It is straightforward to check that every Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a DNF with at most 2^{n-1} terms of width at most n.

We say that a Boolean function f is *sensitive* at coordinate $i \in [n]$ on input x if $f(x^{i=0}) \neq f(x^{i=1})$, where $x^{i=b}$ is x with its i-th coordinate set to b. We write $s(f,x,i)$ to denote the indicator for this event, and $s(f,x)$ to denote $\sum_{i=1}^{n} s(f,x,i)$. We say that f is *monotone* in direction i if $f(x^{i=0}) \leq f(x^{i=1})$ for all x, and *anti-monotone* in direction i if $f(x^{i=0}) \geq f(x^{i=1})$ for all x. A Boolean function f is *unate* if for all $i \in [n]$, f is either monotone or anti-monotone in direction i.

The *subcube* $C \subseteq \{0,1\}^n$ corresponding to the pair of disjoint sets $S_0, S_1 \subseteq [n]$ is the set of elements $x \in \{0,1\}^n$ for which $x_i = 0$ for every $i \in S_0$ and $x_i = 1$ for every $i \in S_1$. The *free coordinates* of C are the coordinates in $[n] \setminus (S_0 \cup S_1)$. The subcube C is 1-*monochromatic* with respect to f if $f(x) = 1$ for every x in C. Each term in a DNF corresponds to a 1-monochromatic subcube, and so a DNF may be viewed geometrically as a union of 1-monochromatic subcubes.

A degree-d *polynomial threshold function* (PTF) is a Boolean function $f(x) = \mathrm{sgn}(p(x))$, where $p : \{0,1\}^n \to \mathbb{R}$ is a degree-d polynomial. If $d = 1$ we refer to f as a *linear threshold*

function (LTF), or a *halfspace*. It is straightforward to verify that halfspaces are unate.

4.1.2.1 Fourier analysis over the Boolean hypercube

Every Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ has a Fourier expansion

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x),$$

where the numbers $\widehat{f}(S) \in [-1,1]$ are the Fourier coefficients of f. We write $\mathbf{W}_k[f] = \sum_{|S|=k} \widehat{f}(S)^2$ to denote the total Fourier weight of f at level k.

Definition 46. The influence of coordinate $i \in [n]$ on a Boolean function f, denoted $\mathbf{Inf}_i[f]$, is the probability $\mathbf{Pr}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})]$, where $x^{\oplus i}$ denotes x with its i-th bit flipped. The total influence of f is $\mathbf{Inf}[f] = \sum_{i=1}^n \mathbf{Inf}_i[f]$.

Definition 47. The noise sensitivity of a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ at noise rate δ, denoted $\mathbf{NS}_\delta[f]$, is the probability $\mathbf{Pr}[f(\boldsymbol{x}) \neq f(\boldsymbol{y})]$, where $\boldsymbol{x} \sim \{-1,1\}^n$ is uniformly random and $\boldsymbol{y}$ is obtained from $\boldsymbol{x}$ by flipping each coordinate independently with probability δ.

Both total influence and noise sensitivity have well-known Fourier formulas:

$$\begin{aligned} \mathbf{Inf}[f] &= \sum_{S \subseteq [n]} |S| \cdot \widehat{f}(S)^2. \\ \mathbf{NS}_\delta[f] &= \frac{1}{2}\sum_{k=0}^{n} (1 - (1-2\delta)^k) \cdot \mathbf{W}_k[f]. \end{aligned}$$

4.2 Universal upper bound on DNF size

In this section we prove that every Boolean function can be ε-approximated by a DNF of size $O_\varepsilon(2^n/\log n)$. The proof of this result is obtained via a randomized construction that, given a function f, constructs an explicit approximating DNF for f of the required size.

Before presenting the construction, let us first informally describe the intuition behind it. In order to build a good approximator for f, the construction must identify a small family of subcubes that (i) cover almost all of the inputs $x \in \{0,1\}^n$ for which $f(x) = 1$, and (ii) cover almost none of the inputs x for which $f(x) = 0$. For most functions f, these

constraints are roughly equivalent to the requirement that the subcubes in the family should have relatively small overlap with each other over $f^{-1}(1)$ while having large overlap with each other over $f^{-1}(0)$.

Our construction meets these apparently conflicting requirements with a two-stage process. In the first stage, the algorithm selects a (small) random subset S of $f^{-1}(0)$ and defines the random function $\boldsymbol{g}$ to take the value 1 on every input in $f^{-1}(1) \cup S$. The second stage selects a random subset of the large subcubes that are 1-monochromatic in $\boldsymbol{g}$. The union of those subcubes corresponds to a small DNF that computes a function $\boldsymbol{h}$ that is close to f provided that S is small enough (in which case constraint (ii) is satisfied) and that most elements in $f^{-1}(1)$ are covered by many large subcubes that are 1-monochromatic in $\boldsymbol{g}$ (in which case constraint (i) is also satisfied). As we see below, with the right parameter settings, we can guarantee that both those events happen with large probability.

Theorem 36. *Let $\varepsilon \geq 10/n$.*[4] *Every Boolean function $f : \{0,1\}^n \to \{0,1\}$ can be ε-approximated by a DNF of size $4\ln(4/\varepsilon) \cdot 2^{n-d}$ and width $n-d$, where*

$$d = \log\log_{2/\varepsilon}\left(\frac{n}{\ln(4/\varepsilon)\log\log_{2/\varepsilon} n}\right).$$

That is, every f can be ε-approximated by a DNF of size $O_\varepsilon(2^n/\log n)$ and width $n - \Omega_\varepsilon(\log\log n)$.

Proof. We may assume that $\min\{\mathbf{Pr}[f(\boldsymbol{x}) = 0], \mathbf{Pr}[f(\boldsymbol{x}) = 1]\} \geq \varepsilon$, since otherwise f is ε-close to constant and the claim is trivially true. Let $\boldsymbol{g} : \{0,1\}^n \to \{0,1\}$ be the random function obtained by setting $\boldsymbol{g}(x) = 1$ for all $x \in f^{-1}(1)$, and for each $x \in f^{-1}(0)$, we independently set $\boldsymbol{g}(x) = 1$ with probability $\varepsilon/2$ and $\boldsymbol{g}(x) = 0$ otherwise. Let $\mathcal{G}$ denote the induced distribution over all Boolean functions. Since $\mathbf{E}_{\mathcal{G}}[\mathbf{Pr}_{x \in f^{-1}(0)}[\boldsymbol{g}(x) = 1]] = \varepsilon/2$, we apply the Chernoff bound to deduce that

$$\mathop{\mathbf{Pr}}_{\mathcal{G}}\left[\mathop{\mathbf{Pr}}_{f^{-1}(0)}[f(x) \neq \boldsymbol{g}(x)] \geq \varepsilon\right] \leq e^{-\varepsilon^2 \cdot 2^n/3}. \tag{4.1}$$

Let us call a subcube C *special* if C has dimension exactly d and the d free coordinates of C are $\{dk+1, \ldots, dk+d\}$ for some $k = 0, \ldots, \lfloor n/d \rfloor - 1$. There are $\lfloor n/d \rfloor \cdot 2^{n-d}$ special

[4] Our lower bounds in Section 4.5 imply that any DNF that ε-approximates PAR_n has size $\Omega(2^n)$ and width $n - O(1)$ when $\varepsilon = O(1/n)$, and so the universal upper bounds are only of interest for $\varepsilon = \omega(1/n)$.

subcubes in total, and every $x \in \{0,1\}^n$ is contained in exactly $\lfloor n/d \rfloor$ special subcubes. Let $\boldsymbol{h} : \{0,1\}^n \to \{0,1\}$ be the union of a random subset of the 1-monochromatic special subcubes in $\boldsymbol{g}$ where each 1-monochromatic special subcube C in $\boldsymbol{g}$ is included in $\boldsymbol{h}$ with probability $(\varepsilon/2)^{\#\{x \in C\,:\, f(x)=1\}}$. The probability, over the randomness of $\boldsymbol{g}$ and $\boldsymbol{h}$, that a fixed special subcube C is included in $\boldsymbol{h}$ is therefore exactly $(\varepsilon/2)^{2^d}$, since the probability that C is 1-monochromatic in $\boldsymbol{g}$ is $(\varepsilon/2)^{\#\{x \in C\,:\, f(x)=0\}}$, and the probability that C is then included in $\boldsymbol{h}$ is $(\varepsilon/2)^{\#\{x \in C\,:\, f(x)=1\}}$. Note that $\boldsymbol{h}$ is a DNF of width $n-d$, and $\boldsymbol{h}^{-1}(1) \subseteq \boldsymbol{g}^{-1}(1)$; in particular, the error of $\boldsymbol{h}$ on the 0-inputs of f is at most that of $\boldsymbol{g}$, and (4.1) remains true with $\boldsymbol{h}$ in place of $\boldsymbol{g}$. Next we argue that

$$\mathop{\mathbf{Pr}}_{\mathcal{G}}\left[\mathop{\mathbf{Pr}}_{f^{-1}(1)}[f(x) \neq \boldsymbol{h}(x)] \geq \varepsilon\right] \leq 1/4. \tag{4.2}$$

Fix $x \in f^{-1}(1)$. The probability that $\boldsymbol{h}(x) = 0$ (*i.e.* $\boldsymbol{h}(x) \neq f(x)$) is the probability that none of the $\lfloor n/d \rfloor$ special subcubes containing x are included in $\boldsymbol{h}$. Since any two of these $\lfloor n/d \rfloor$ special subcubes intersect only at x, their inclusion probability are independent and so we have

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\mathcal{G}}[\boldsymbol{h}(x) = 0] &= \left(1 - \left(\frac{\varepsilon}{2}\right)^{2^d}\right)^{\lfloor n/d \rfloor} \\ &\leq e^{-(\varepsilon/2)^{2^d} n/d} < \varepsilon/4,\end{aligned}$$

where we have used our choice of d in the final inequality. This gives us

$$\mathop{\mathbf{E}}_{\mathcal{G}}\left[\mathop{\mathbf{Pr}}_{f^{-1}(1)}[f(x) \neq \boldsymbol{h}(x)]\right] < \varepsilon/4,$$

and so

$$\mathop{\mathbf{Pr}}_{\mathcal{G}}\left[\mathop{\mathbf{Pr}}_{f^{-1}(1)}[f(x) \neq \boldsymbol{h}(x)] \geq \varepsilon\right] \leq 1/4,$$

matching the claimed bound in (4.2) above. It remains to bound the DNF size of $\boldsymbol{h}$. Since there are $\lfloor n/d \rfloor \cdot 2^{n-d}$ special subcubes, and each is included with probability $(\varepsilon/2)^{2^d}$, we have

$$\begin{aligned}\mathop{\mathbf{E}}_{\mathcal{G}}[\text{DNF-size}[\boldsymbol{h}]] &= \left(\frac{\varepsilon}{2}\right)^{2^d} \lfloor \frac{n}{d} \rfloor 2^{n-d} \\ &\leq \frac{\ln(4/\varepsilon) \log \log_{2/\varepsilon} n}{d} \cdot 2^{n-d} \\ &\leq 2 \ln(4/\varepsilon) \cdot 2^{n-d}.\end{aligned}$$

Here in the final inequality we use the fact that $d \geq (\log \log_{2/\varepsilon} n) - 1$ for n sufficiently large (which in turn holds since $\ln(4/\varepsilon) \log \log_{2/\varepsilon} n < \sqrt{n}$). Again, we apply Markov's inequality to say that

$$\mathop{\mathbf{Pr}}_{\mathcal{G}}\left[\text{DNF-size}[\boldsymbol{h}] \geq 4\ln(4/\varepsilon) \cdot 2^{n-d}\right] \leq 1/2. \tag{4.3}$$

Taking a union bound over (4.1), (4.2), and (4.3), we conclude that there must exist some h such that DNF-size$[h] \leq 4\ln(4/\varepsilon) \cdot 2^{n-d}$, DNF-width$[h] = n - d$, and $\mathbf{Pr}[f(x) \neq h(x)] \leq \varepsilon$ and this completes the proof. □

4.3 Approximation via Hamming ball covers of the hypercube

In this section we introduce a general method for constructing small depth-2 circuits with top gate OR that approximate an arbitrary Boolean function f,[5] to which there are three main components. We first show that for any radius d, all but an ε fraction of $\{0,1\}^n$ can be covered with $O_\varepsilon(2^n/\text{Vol}(d))$ Hamming balls of radius d. Next, we approximate f restricted to each Hamming ball with the desired second layer gate (*e.g.* an LTF, degree-d PTF, or unate function[6]); these sub-approximators approximate f to high accuracy within the ball, and label all points outside the ball 0. Finally, our overall approximator for f is simply the $O_\varepsilon(2^n/\text{Vol}(d))$-wise disjunction of these sub-approximators. We begin with a short proof of the first claim:

Lemma 4.3.1. *For every $\varepsilon > 0$ and $d > 0$ there is a collection of $\ln(1/\varepsilon)(2^n/\text{Vol}(d))$ Hamming balls of radius d that covers all but an ε fraction of $\{0,1\}^n$.*

[5] As alluded to in the introduction, since the classes of LTFs, degree-d PTFs, and unate functions are each closed under negation, if f can be approximated by a disjunction of k of them then $\neg f$ can be approximated by an intersection (*i.e.* conjunction) of k of them. We focus our exposition on depth-2 approximators with top gate OR, noting that they immediately imply the existence of universal approximators with top gate AND of the same size.

[6] For our construction of DNF approximators the sub-approximators are themselves DNFs instead of conjunctions; we may do this since the disjunction of DNFs is itself a DNF.

Proof. Let $\mathcal{C}$ be a random collection of $\ln(1/\varepsilon)(2^n/\mathrm{Vol}(d))$ Hamming balls of radius d with centers picked uniformly at random with replacement from $\{0,1\}^n$. For any $x \in \{0,1\}^n$, the probability that x is not covered by $\mathcal{C}$ is $(1 - \mathrm{Vol}(d) \cdot 2^{-n})^{\ln(1/\varepsilon)(2^n/\mathrm{Vol}(d))} \leq \exp(-\ln(1/\varepsilon)) = \varepsilon$. Therefore $\mathcal{C}$ covers all but an ε fraction of points in $\{0,1\}^n$ in expectation, and so there is a collection of $\ln(1/\varepsilon)(2^n/\mathrm{Vol}(d))$ many Hamming balls of radius d that covers all but an ε fraction of $\{0,1\}^n$. □

4.3.1 Approximating Boolean functions restricted to Hamming balls

Next we construct sub-approximators for Boolean functions restricted to Hamming balls. To be precise, when we write "f restricted to a Hamming ball B of radius d", we mean the function f_B that agrees with f on all points within B and takes value 0 on all points outside B. Since the bulk of the points in B lie on its surface, our sub-approximators may err on all the points in the interior of B (*i.e.* the points at distance $< d$ from the center of B). We begin with sub-approximators that are small-width DNFs:

Proposition 4.3.2. *Let $z \in \{0,1\}^n$ and $d \in [n]$. Let $f : \{0,1\}^n \to \{0,1\}$ be the characteristic function of a subset of the Hamming ball of radius d centered at z. There is a DNF g of width $n - d$ satisfying $g(y) = f(y)$ for all y such that $\mathrm{dist}(y,z) = d$, and $g(y) = f(y) = 0$ for all y such that $\mathrm{dist}(y,z) > d$.*

Proof. For each y such that $\mathrm{dist}(y,z) = d$ and $f(y) = 1$ we include a term T_y in the DNF g defined as follows: T_y is the conjunction of literals ℓ_i for each $i \in [n]$ such that $y_i \neq z_i$, where $\ell_i = x_i$ if $z_i = 1$, and $\neg\, x_i$ otherwise. The key property of T_y is that it accepts only y among all $\binom{n}{d}$ inputs at distance exactly d from the center z, and it rejects any input at distance greater than d from z. Since each term T_y has width exactly $n - d$, it follows that g is a DNF of width $n - d$ that satisfies the claimed properties. □

Note that the DNFs g constructed in Proposition 4.3.2 are unate since no variable occurs both positively and negated in them; if the literal ℓ_i occurs in g then $\neg\, \ell_i$ does not occur in g. This observation yields the following corollary of Proposition 4.3.2:

Corollary 4.3.3. *Let $z \in \{0,1\}^n$ and $d \in [n]$. Let $f : \{0,1\}^n \to \{0,1\}$ be the characteristic function of a subset of the Hamming ball of radius d centered at z. There is a unate function*

g satisfying $g(y) = f(y)$ for all y such that $\mathrm{dist}(y,z) = d$*, and $g(y) = f(y) = 0$ for all y such that* $\mathrm{dist}(y,z) > d$*.*

Finally we construct sub-approximators that are low-degree PTFs. These sub-approximators have the nice feature that they have one-sided error.

Proposition 4.3.4. *Let $z \in \{0,1\}^n$ and $d \in [n]$. Let $f : \{0,1\}^n \to \{0,1\}$ be the characteristic function of a subset of the Hamming ball of radius d centered at z. There is a degree-d PTF $g(x) = \mathrm{sgn}(p(x))$ satisfying $g(y) = f(y)$ for all y such that* $\mathrm{dist}(y,z) = d$*, and $g(y) = f(y) = 0$ for all y such that* $\mathrm{dist}(y,z) > d$*. Furthermore, $g(y) = 1$ for all y such that* $\mathrm{dist}(y,z) < d$*.*

Proof. The polynomial $p(x)$ will be $L(x) + D(x) - \theta$, where $\theta = n - d + \frac{1}{2\binom{n}{d}}$,

$$L(x) = \sum_{i \in [n] \,:\, z_i = 1} x_i + \sum_{i \in [n] \,:\, z_i = 0} (1 - x_i),$$

and

$$D(x) = \frac{1}{2\binom{n}{d}} \sum_{\substack{y \,:\, \mathrm{dist}(y,z) = d \\ f(y) = 1}} \mathbf{1}_{\mathrm{differ}(y,z)}(x),$$

where $\mathbf{1}_{\mathrm{differ}(y,z)}(x) = 1$ iff x agrees with y on the coordinates that y and z differ on (and 0 otherwise). For any y such that $\mathrm{dist}(y,z) = d$, the indicator $\mathbf{1}_{\mathrm{differ}(y,z)}$ is a function of d variables and hence is computed by a degree-d polynomial. Note that $L(y) = n - \mathrm{dist}(y,z)$ for all $y \in \{0,1\}^n$. Since $D(y) \in [0, 1/2]$ for all $y \in \{0,1\}^n$ and the threshold θ is set to be $n - d + \frac{1}{2\binom{n}{d}}$, it follows that $p(y) > 0$ for all y such that $\mathrm{dist}(y,z) > d$, and $p(y) < 0$ for all y such that $\mathrm{dist}(y,z) < d$. Finally for all y such that $\mathrm{dist}(y,z) = d$, we have $D(y) = 0$ if $f(y) = 0$ and $D(y) = \frac{1}{2\binom{n}{d}}$ otherwise (since $\mathbf{1}_{\mathrm{differ}(y,z)}(y) = 1$), and so p takes the correct sign on these inputs. □

4.3.2 Combining the sub-approximators

To combine the sub-approximators from the last section, we use the following bound on the surface-to-volume ratio of Hamming balls.

Lemma 4.3.5. *Fix $1 \le d \le n$. Then* $\mathrm{Vol}(d-1) \le \frac{d}{n-d+1}\mathrm{Vol}(d)$*. In particular, if $d = \rho n$ for some $\rho \le 1/2$, then* $\mathrm{Vol}(d-1) \le 2\rho \cdot \mathrm{Vol}(d)$*.*

Proof. Let $\boldsymbol{x} \in \{0,1\}^n$ be drawn uniformly at random from the Hamming ball of radius $d-1$ around 0, let $\mathbf{i} \in [n]$ be drawn uniformly at random from the coordinates j for which $x_j = 0$, and let $\boldsymbol{y} \in \{0,1\}^n$ be the input obtained by flipping the $\mathbf{i}$-th coordinate of $\boldsymbol{x}$.

By the chain rule, the joint entropy of $\boldsymbol{x}$, $\boldsymbol{y}$, and $\mathbf{i}$ is

$$\begin{aligned} H(\boldsymbol{x},\boldsymbol{y},\mathbf{i}) &= H(\boldsymbol{x}) + H(\mathbf{i} \mid \boldsymbol{x}) + H(\boldsymbol{y} \mid \boldsymbol{x},\mathbf{i}) \\ &\geq \log \mathrm{Vol}(d-1) + \log(n-d+1) \end{aligned} \tag{4.4}$$

since $\boldsymbol{x}$ is drawn uniformly at random from a set of size $\mathrm{Vol}(d-1)$, $\mathbf{i}$ is drawn uniformly from a set of size at least $n-d+1$, and $\boldsymbol{y}$ is completely determined by $\boldsymbol{x}$ and $\mathbf{i}$. A different application of the chain rule also yields

$$\begin{aligned} H(\boldsymbol{x},\boldsymbol{y},\mathbf{i}) &= H(\boldsymbol{y}) + H(\mathbf{i} \mid \boldsymbol{y}) + H(\boldsymbol{x} \mid \boldsymbol{y},\mathbf{i}) \\ &\leq \log \mathrm{Vol}(d) + \log d \end{aligned} \tag{4.5}$$

since the support of $\boldsymbol{y}$ has size $\mathrm{Vol}(d)$, the support of $\mathbf{i}$ has size at most d, and $\boldsymbol{x}$ is completely determined by $\boldsymbol{y}$ and $\mathbf{i}$. Combining (4.4) and (4.5) and re-arranging the terms completes the proof. □

Theorem 37. *Every Boolean function f can be ε-approximated by a DNF of width $(1-\rho)n$, where $\rho = \varepsilon/(4\ln(2/\varepsilon))$. In particular, every Boolean function can be 0.01-approximated by a DNF of width $c \cdot n$, where $c < 1$ is an absolute constant.*

Proof. Let $d = \rho n$ where $\rho \leq 1/2$ will be chosen later. By Lemma 4.3.1, there is a collection of $\ln(2/\varepsilon)(2^n/\mathrm{Vol}(d))$ Hamming balls of radius d that cover all but an $\varepsilon/2$ fraction of $\{0,1\}^n$. We approximate f restricted to each Hamming ball with the DNF of width $n-d$ given by Proposition 4.3.2. Note that the disjunction g of these DNFs is itself a DNF of width $n-d$, and

$$\begin{aligned} \mathbf{Pr}[f(x) \neq g(x)] &\leq \frac{\varepsilon}{2} + \ln(2/\varepsilon)\frac{\mathrm{Vol}(d-1)}{\mathrm{Vol}(d)} \\ &\leq \frac{\varepsilon}{2} + 2\rho\ln(2/\varepsilon), \end{aligned}$$

where the final inequality is by Lemma 4.3.5. Here the first inequality is a union bound over the $\varepsilon/2$ fraction of uncovered points and the error of the $\ln(2/\varepsilon)(2^n/\mathrm{Vol}(d))$ sub-

approximators, each of which errs on at most $\mathrm{Vol}(d-1)$ points. It suffices to ensure that $2\rho\ln(2/\varepsilon) \le \varepsilon/2$, and so we may take $\rho = \varepsilon/(4\ln(2/\varepsilon))$. □

As noted in Corollary 4.3.3 the DNF sub-approximators in Theorem 37 are unate. Viewing our overall approximator as a disjunction of unate functions instead of a disjunction of DNFs gives us the following:

Theorem 38. *Every Boolean function f can be ε-approximated by the disjunction (equivalently, the intersection) of $\ln(2/\varepsilon)\,2^{(1-H(\rho))n}$ unate functions, where $\rho = \varepsilon/(4\ln(2/\varepsilon))$. In particular, every Boolean function can be 0.01-approximated by the intersection of $O(2^{cn})$ unate functions, where $c < 1$ is an absolute constant.*

Using the PTF sub-approximators of Proposition 4.3.4 in place of the DNF sub-approximators of Proposition 4.3.2, an identical proof to the one for Theorem 37 yields approximators that are the intersection of degree-d PTFs:

Theorem 39. *Let $d = \rho n$ where $\rho \le 1/2$. Every Boolean function f can be $(\varepsilon + O(\ln(1/\varepsilon)\rho))$-approximated by the disjunction (equivalently, the intersection) of $\ln(1/\varepsilon)(2^n/\mathrm{Vol}(d))$ degree-d PTFs. In particular, for any constant d every Boolean function can be $O(1/n)$-approximated by the intersection of $O(2^n/n^d)$ degree-d PTFs.*

Improvements via covering codes. It is natural to ask if Lemma 4.3.1 can be improved. The strongest possible improvement is a covering of all of $\{0,1\}^n$ with $(1+o(1))(2^n/\mathrm{Vol}(d))$ Hamming balls of radius d, a covering code with efficiency asymptotically approaching that of a perfect code. This is a longstanding open problem in the field of covering codes [Cohen *et al.*, 2005], and even the $d = 2$ case remains open. The $d = 1$ case was resolved in the affirmative by Kabatyanski and Pachenko [Kabatyanski and Panchenko, 1988]:

Theorem 48 (Kabatyanski and Pachenko)**.** *All of $\{0,1\}^n$ can be covered by $(1+o(1))(2^n/n)$ Hamming balls of radius 1.*

Using Theorem 48 in place of the approximate cover given by Lemma 4.3.1 allows us to sharpen the parameters of Theorem 39 in the case of $d = 1$ (*i.e.* intersection of LTFs). As an added bonus, since all of $\{0,1\}^n$ is covered and our LTF sub-approximators in Proposition

4.3.4 have one-sided error within each Hamming ball, our overall approximator has one-sided error as well.

Theorem 49. *For every Boolean function $f : \{0,1\}^n \to \{0,1\}$ there is a Boolean function g computed by the disjunction (equivalently, the intersection) of $(1+o(1))(2^n/n)$ halfspaces that $((1+o(1))/n)$-approximates f. Furthermore, these approximators have one sided error: $g(x) = 1$ whenever $f(x) = 1$.*

A generalization of Theorem 48 for Hamming balls of larger radii $d > 1$ would imply analogous improvements of Theorem 39 for the intersection of degree-d PTFs. The current best bound for general d is due to Krivelevich *et al.* who show that all of $\{0,1\}^n$ can be covered with $O(d \log d)(2^n/\mathrm{Vol}(d))$ many Hamming balls of radius d [Krivelevich *et al.*, 2003]; however, using this in place of Lemma 4.3.1 does not yield any improvements on the parameters of our constructions.

4.4 Inapproximability of a random function

We write $\boldsymbol{f}_n$ to denote a uniformly random Boolean function with arity n.

Theorem 50. *Let $F_1, F_2, F_3, \ldots$ be an infinite sequence of Boolean functions indexed by $\mathbb{N}$, where F_k has arity k. Let $\mathcal{C}$ be a class of Boolean functions. For any $\varepsilon > 0$ and $f : \{0,1\}^n \to \{0,1\}$, let $\mathrm{opt}_\varepsilon[f]$ denote the smallest k such that there exists $g_1, \ldots, g_k \in \mathcal{C}$ and $F_k(g_1, \ldots, g_k)$ is an ε-approximator for f. Then*

$$\mathbf{E}[\mathrm{opt}_\varepsilon[\boldsymbol{f}_n]] \geq (1 - H(\varepsilon)) \cdot \frac{2^n}{\log(e \cdot |\mathcal{C}|)}.$$

Proof. Let $t = |\mathcal{C}|$, and fix a numbering $g_1, \ldots, g_t$ of the t Boolean functions in $\mathcal{C}$. Let $\mathbf{X} = \langle \mathbf{X}_1, \ldots, \mathbf{X}_t \rangle$ be a vector of indicator random variables where $\mathbf{X}_i = 1$ iff g_i occurs in the optimal ε-approximator $F_k(g_1, \ldots, g_k)$ for $\boldsymbol{f}_n$ (if there are multiple ε-approximators achieving minimal size we fix an arbitrary one to be the optimal).

Since $\boldsymbol{f}_n$ determines $\mathbf{X}$ and $H(\boldsymbol{f}_n) = 2^n$, we have $H(\boldsymbol{f}_n, \mathbf{X}) = 2^n$ as well. Furthermore, since $\mathbf{X}$ uniquely determines a Boolean function and there are $2^{H(\varepsilon)\cdot 2^n}$ Boolean functions that are ε-close to it, we have $H(\boldsymbol{f}_n \,|\, \mathbf{X}) \leq H(\varepsilon) \cdot 2^n$. Applying the chain rule, we see that

$$H(\mathbf{X}) = H(\boldsymbol{f}_n, \mathbf{X}) - H(\boldsymbol{f}_n \,|\, \mathbf{X}) \geq (1 - H(\varepsilon)) \cdot 2^n. \tag{4.6}$$

For each $i \in [t]$, we write $p_i = \mathbf{E}[\mathbf{X}_i]$ to denote that probability that g_i occurs in the optimal ε-approximator for $\boldsymbol{f}_n$, and so $\mathbf{E}_t[p_t] = \mathbf{E}[\mathrm{opt}_\varepsilon[\boldsymbol{f}_n]]/t$. and note that

$$\begin{aligned} H(\mathbf{X}) \leq \sum_{i=1}^{t} H(\mathbf{X}_i) &= \sum_{i=1}^{t} H(p_i) = t \cdot \mathop{\mathbf{E}}_{t}[H(p_t)] \\ &\leq t \cdot H(\mathop{\mathbf{E}}_{t}[p_t]) \\ &= t \cdot H\left(\frac{\mathbf{E}[\mathrm{opt}_\varepsilon[\boldsymbol{f}_n]]}{t}\right) \end{aligned}$$

where we have used the concavity of the binary entropy function. Finally, using the inequality $H(p) \leq p \log(e/p)$, we get

$$\begin{aligned} H(\mathbf{X}) &\leq t \cdot H\left(\frac{\mathbf{E}[\mathrm{opt}_\varepsilon[\boldsymbol{f}_n]]}{t}\right) \\ &\leq \mathbf{E}[\mathrm{opt}_\varepsilon[\boldsymbol{f}_n]] \cdot \log\left(\frac{e \cdot t}{\mathbf{E}[\mathrm{opt}_\varepsilon[\boldsymbol{f}_n]]}\right) \\ &\leq \mathbf{E}[\mathrm{opt}_\varepsilon[\boldsymbol{f}_n]] \cdot \log(e \cdot t). \end{aligned}$$

Combining this upper bound with the lower bound of (4.6) completes the proof. □

Since there are 3^n conjunctions over $\{0,1\}^n$ and $2^{n^{d+1}+O(n)}$ degree-d PTFs over $\{0,1\}^n$ [Chow, 1961], applying Theorem 50 immediately implies strong bounds on the inapproximability of a random function by DNFs and the intersection of low-degree PTFs, respectively.

Theorem 40. *Suppose $\boldsymbol{f}_n$ is ε-approximated by a size-s DNF. Then $s = \Omega(1-H(\varepsilon)) \cdot 2^n/n$.*

Theorem 51. *Suppose $\boldsymbol{f}_n$ is ε-approximated by the intersection of k degree-d PTFs. Then $k = \Omega(1 - H(\varepsilon)) \cdot 2^n/n^{d+1}$.*

The two remaining lower bounds in Table 4.1 are both witnessed explicitly by the parity function. We show in Section 4.5 that any DNF that ε-approximates PAR_n must have width $(1-2\varepsilon)n$ (Theorem 42), and in Section 4.6 that the intersection of $2^{\Omega_\varepsilon(n)}$ unate functions is required to ε-approximate PAR_n for any $\varepsilon < \frac{1}{16}$ (Theorem 45). In fact, we will see that our proof of the former extends easily to show that DNFs that ε-approximate a random function have width $\Omega_\varepsilon(n)$.

4.5 Approximating the parity function

We begin this section, and the second part of this chapter, with a deterministic construction of a small-size small-width DNF that ε-approximates PAR_n with one-sided error.

Lemma 4.5.1. *Let $B_1, B_2, \ldots, B_\ell \subseteq [n]$ be linearly independent [7] sets of coordinates. Define $k = |B_1 \cup B_2 \cup \cdots \cup B_\ell|$. There is a DNF of size $2^{k-\ell}$ and width k that accepts exactly all $x \in \{0,1\}^n$ such that $\bigoplus_{i \in B_1} x_i = \bigoplus_{i \in B_2} x_i = \cdots = \bigoplus_{i \in B_\ell} x_i = 0$ (i.e. the set of all strings with even Hamming weight within B_1, B_2, ..., and B_ℓ). We write $\mathrm{even}(B_1, \ldots, B_\ell)$ to denote this DNF.*

Proof. By linear independence, there are exactly $2^{-\ell} \cdot 2^{|B_1 \cup \cdots \cup B_\ell|} = 2^{k-\ell}$ possible settings of the k coordinates in $B_1 \cup \cdots \cup B_\ell$ that satisfy $\bigoplus_{i \in B_1} x_i = \cdots = \bigoplus_{i \in B_\ell} x_i = 0$. The DNF will be a disjunction of all $2^{k-\ell}$ such settings, each of which is computed by a conjunction of k literals. □

Theorem 41. *For every $\varepsilon > 0$ there is a DNF f of width $(1-2\varepsilon)n$ and size $2^{(1-2\varepsilon)n}$ that ε-approximates PAR_n. Furthermore f has one-sided error: if $\mathsf{PAR}_n(x) = 1$ then $f(x) = 1$.*

Proof. Let $k = \log(\frac{1}{2\varepsilon})$. For $i = 1, \ldots, k$, let $B_{i,0} = \{j \in [n] : j_i = 0\}$ and $B_{i,1} = \{j \in [n] : j_i = 1\} = [n] \setminus B_{i,0}$.[8] Consider the function $f = \bigvee_{z \in \{0,1\}^k} \mathrm{even}(B_{1,z_1}, \ldots, B_{k,z_k})$. For any $z \in \{0,1\}^n$, the union of the blocks $B_{1,z_1}, \ldots, B_{k,z_k}$ has size $(1-2^{-k})n = (1-2\varepsilon)n$. So by Lemma 4.5.1, f is a DNF of width $(1-2\varepsilon)n$ and size $2^k \cdot 2^{(1-2\varepsilon)n-k} = 2^{(1-2\varepsilon)n}$.

For any $x \in \{0,1\}^n$ such that $\mathsf{PAR}_n(x) = 1$ and every $i \in [k]$, either $\mathrm{even}(B_{i,0})$ or $\mathrm{even}(B_{i,1})$ is true. Thus, for each such x there is some $z \in \{0,1\}^k$ for which $\mathrm{even}(B_{1,z_1}, \ldots, B_{k,z_k}) = 1$ and we have $f(x) = 1$. Finally, when $\mathsf{PAR}_n(x) = 0$, then the probability that it has even Hamming weight in all of the blocks B_{i,z_i} is $2^{-k} = 2\varepsilon$. Therefore, the probability that $f(x) \neq \mathsf{PAR}_n(x)$ is $\frac{1}{2} \cdot 2\varepsilon = \varepsilon$, as we wanted to show. □

[7] More formally, let these sets correspond to linearly independent vectors under the usual correspondence between subsets of $[n]$ and vectors in $\mathbb{F}_2^n$.

[8] Here, the notation j_i represents the ith bit of the binary representation of j.

4.5.1 Lower bounds on DNF width and size

We begin with a simple lemma relating the total influence of close Boolean functions.

Lemma 4.5.2. *For any functions $f, g : \{0,1\}^n \to \{0,1\}$, the total influence of f and g satisfies $|\mathbf{Inf}[f] - \mathbf{Inf}[g]| \leq 2\,\mathbf{Pr}[f(x) \neq g(x)] \cdot n$. In particular, if f is ε-close to PAR_n then $\mathbf{Inf}[f] \geq (1-2\varepsilon)n$.*

Proof. We think of g as being obtained from f by flipping the values of an $\varepsilon = \mathbf{Pr}[f(x) \neq g(x)]$ fraction of inputs. Let f^* denote f with the value of a single input x^* flipped, and note that $|s(f,x^*) - s(f^*,x^*)| \leq n$, that $|s(f,y) - s(f^*,y)| \leq 1$ for all y such that $\mathrm{dist}(x^*,y) = 1$, and that $s(f,z) = s(f^*,z)$ for all other z such that $\mathrm{dist}(x^*,z) \geq 2$. It follows that $|\mathbf{Inf}[f] - \mathbf{Inf}[f^*]| \leq n + n = 2n$, and taking a union bound over all x^* such that $f(x^*) \neq g(x^*)$ completes the proof. □

Building on the bounds on Boppana and Traxler [Boppana, 1997; Traxler, 2009] and resolving an open problem of O'Donnell, Amano proved that the total influence of a Boolean function is at most its DNF width [Amano, 2011].

Theorem 52 (Amano)**.** *Let f be a width-w DNF. Then $\mathbf{Inf}[f] \leq w$.*

Combining Lemma 4.5.2 and Theorem 52 yields a lower bound on the width of any DNF that ε-approximates PAR_n, matching the width of our construction in Theorem 41 exactly.

Theorem 42. *Let f be a width-w DNF that ε-approximates PAR_n. Then $w \geq (1-2\varepsilon)n$.*

Straightforward Fourier-analytic computations show that the total influence of a random Boolean function is $n/2$ in expectation (see *e.g.* Theorem 6 of [Bernasconi *et al.*, 1997]), and so by the same reasoning used to establish Theorem 42, we see that DNFs that ε-approximate a random function have width at least $(\frac{1}{2} - 2\varepsilon)n = \Omega_\varepsilon(n)$, as had been claimed at the end of Section 4.4. Next we turn to lower bounds on size of DNFs that ε-approximate PAR_n, giving two incomparable bounds. The first simply combines Amano's theorem with an elementary truncation argument.

Theorem 43. *(first lower bound) Let f be a size-s DNF that ε-approximates PAR_n. Then $s \geq \delta\, 2^{(1-2\varepsilon-2\delta)n}$ for all $\delta > 0$.*

Proof. We use the folklore observation that dropping all terms of width greater than $\log(s/\delta)$ in f yields a DNF g that is δ-close to f (to see this, note that each dropped term is satisfied with probability less than $2^{-\log(s/\delta)} = \delta/s$, and taking a union bound yields an approximation error that is less than δ). Since g is an s-term DNF of width $\log(s/\delta)$ that is $(\varepsilon+\delta)$-close to PAR_n, and we may apply Theorem 42 to get $\log(s/\delta) \geq (1-2(\varepsilon+\delta))n$; rearranging completes the proof. □

The second lower bound on size uses a sharpening of the Boppana's $O(\log s)$ bound on the total influence of size-s DNFs [Boppana, 1997], obtained in concurrent work by the present authors via the entropy method [Blais *et al.*, 2013b]. For the sake of completeness, we include its short proof here.

Theorem 53. *Let $f : \{0,1\}^n \to \{0,1\}$ be a size-s DNF with $\mathbf{E}[f] = \mu$. Then $\mathbf{Inf}[f] \leq 2\mu \log(s/\mu)$.*

Proof. Consider a fixed ordering of the terms in the DNF for f, and define three random variables: $\mathbf{X}$ is a uniform random $x \in f^{-1}(1)$, $\mathbf{Y}$ is the indicator of uniform random subset of sensitive coordinates of $\mathbf{X}$, and $\mathbf{T}$ is the first term in the DNF satisfied by $\mathbf{X}$. We consider the joint entropy of these three random variables. Applying the chain rule, we see that

$$\begin{aligned} H(\mathbf{X},\mathbf{Y},\mathbf{T}) &= H(\mathbf{X}) + H(\mathbf{Y}\,|\,\mathbf{X}) + H(\mathbf{T}\,|\,\mathbf{X},\mathbf{Y}) \\ &= H(\mathbf{X}) + H(\mathbf{Y}\,|\,\mathbf{X}) + 0 \\ &= n - \log(1/\mu) + \mathbf{Inf}[f]/(2\mu). \end{aligned}$$

We claim that $H(\mathbf{X},\mathbf{Y},\mathbf{T}) \leq n + \log s$, noting that this implies the claimed upper bound by rearranging the terms. Again applying the chain rule, we have

$$\begin{aligned} H(\mathbf{X},\mathbf{Y},\mathbf{T}) &= H(\mathbf{T}) + H(\mathbf{X}\,|\,\mathbf{T}) + H(\mathbf{Y}\,|\,\mathbf{X},\mathbf{T}) \\ &\leq \log s + \mathop{\mathbf{E}}_{\mathbf{T}}[n - |\mathbf{T}|] + \mathop{\mathbf{E}}_{\mathbf{T}}[|\mathbf{T}|] \\ &= n + \log s. \end{aligned}$$

Here the expectations are with respect to the distribution where the weight of a term T is the probability that a uniformly random $x \in f^{-1}(1)$ satisfies T. The inequality uses that

fact that the number of inputs satisfying a term T is $2^{n-|T|}$, and if x satisfies T it can only be sensitive on the coordinates fixed by T. □

Theorem 43. *(second lower bound) Let f be a size-s DNF that ε-approximates PAR_n. Then $s \geq (\frac{1}{2} - \varepsilon) \cdot 2^{\frac{(1-2\varepsilon)n}{(1+2\varepsilon)}}$.*

Proof. Let $\mathbf{E}[f] = \mu$. Since PAR_n is balanced and f is ε-close to PAR_n, we have $\mu \in [\frac{1}{2}-\varepsilon, \frac{1}{2}+\varepsilon]$. Combining Lemma 4.5.2 and Theorem 53, we get $2\mu \log(s/\mu) \geq \mathbf{Inf}[f] \geq (1-2\varepsilon)n$, and rearranging yields the claimed lower bound: $s \geq \mu \cdot 2^{(1-2\varepsilon)n/2\mu} \geq (\frac{1}{2} - \varepsilon) \cdot 2^{(1-2\varepsilon)n/(1+2\varepsilon)}$. □

4.6 Lower bounds for intersection of LTFs and unate functions

In this section we provide further evidence of the optimality of our DNF approximators for PAR_n. We begin by showing that a noise sensitivity conjecture of Klivans *et al.* [Klivans *et al.*, 2004] implies that ε-approximating PAR_n even with the intersection of halfspaces requires size $2^{\Omega_\varepsilon(n)}$, matching the size of our DNF approximators in Theorem 41.

Conjecture 1 (Klivans-O'Donnell-Servedio)**.** *Let f be a Boolean function computed by the intersection of k halfspaces. Then $\mathbf{NS}_\delta[f] \leq O(\sqrt{\log k}\sqrt{\delta})$.*

Theorem 44. *Assume the KOS conjecture and let $\varepsilon < 1/2$. Let f be a Boolean function computed by the intersection of k halfspaces and suppose f ε-approximates PAR_n. Then $k = 2^{\Omega_\varepsilon(n)}$.*

Proof. Since f is ε-close to PAR, we have $\mathbf{W}_n[f] = \widehat{f}([n])^2 \geq (1-2\varepsilon)^2$, and so by the Fourier expression for noise sensitivity at noise rate δ we get

$$\mathbf{NS}_\delta[f] \geq \tfrac{1}{2} - \tfrac{1}{2}\left((1-2\varepsilon)^2(1-2\delta)^n + (1-(1-2\varepsilon)^2)\right).$$

Assuming upper bound on $\mathbf{NS}_\delta[f]$ given by Conjecture 1 and taking $\delta = 1/n$, we have $O(\sqrt{\log(k)/n}) \geq \mathbf{NS}_{1/n}[f]$ and

$$\mathbf{NS}_{1/n}[f] \geq \frac{1}{2} - \frac{1}{2}\left(\frac{(1-2\varepsilon)^2}{e} + (1-(1-2\varepsilon)^2)\right).$$

The quantity on the RHS is positive for any $\varepsilon < 1/2$, and so rearranging yields the claimed lower bound on k. □

Naturally, we would like an unconditional proof of Theorem 44. We are able to accomplish this for any fixed $\varepsilon < \frac{1}{16}$, and in fact, our lower bound holds against the more expressive class of intersection of unate functions.

Lemma 4.6.1. *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function computed by the intersection of k unate functions. Suppose $S^* \subseteq \{0,1\}^n$ is a set of 0-inputs of f such that for all pairs $x, y \in S^*$, there exists a coordinate $i \in [n]$ such that $x_i \neq y_i$ and $s(f,x,i) = s(f,y,i) = 1$. Then $k \geq |S^*|$.*

Proof. Let f be computed by the intersection of k unate functions $g_1, \ldots, g_k$. We will show that for any $x \in S^*$ there must be some g_j such that $g_j(x) = 0$ and $g_j(y) = 1$ for all other $y \in S^*$, noting that the claimed lower bound $k \geq |S^*|$ follows immediately.

Fix $x \in S^*$. Since $f(x) = 0$ and f is computed by the intersection of $g_1, \ldots, g_k$, certainly there must be some g_j such that $g_j(x) = 0$. We claim that $g_j(y) = 0$ for all other $y \in S^*$. Seeking a contradiction, suppose there is some $y \in S^*$, $y \neq x$ such that $g_j(y) = 0$. Since $x, y \in S^*$, there is some coordinate $i \in [n]$ such that $x_i \neq y_i$ and $s(f,x,i) = s(f,y,i) = 1$. Without loss of generality, suppose g_j is monotone in the i-th direction, and that $x_i = 0$ whereas $y_i = 1$. It follows that $g_j(y^{\oplus i}) = 0$ and hence $f(y^{\oplus i}) = 0$, contradicting the assumption that $s(f,y,i) = 1$. □

Note that Lemma 4.6.1 can be used to lower bound the number of unate functions whose intersection computes PAR_n exactly: taking S^* to be the set of all 0-inputs of PAR_n, we get an optimal lower bound of $k \geq |S^*| = 2^{n-1}$. Next we use Lemma 4.6.1 to show that even functions that approximate PAR_n require exponentially many unate functions.

Theorem 45. *Fix $\varepsilon < \frac{1}{16}$. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function that is computed by the intersection of k unate functions and suppose f ε-approximates PAR_n. Then $k = 2^{\Omega_\varepsilon(n)}$.*

Proof. Let $\varepsilon = \frac{1}{16} - \gamma$, where $\gamma > 0$. We first note that since f is ε-close to PAR_n, the

expected 0-sensitivity of f must be large:

$$\begin{aligned}\mathbf{E}[s(f,x) \mid f(x)=0] &= \frac{\mathbf{Inf}[f]}{\mathbf{Pr}[f(x)=0]} \\ &\geq \frac{(1-2\varepsilon)n}{(1+2\varepsilon)} \\ &\geq (1-4\varepsilon)n = \left(\frac{3}{4}+4\gamma\right)n.\end{aligned}$$

Therefore a constant fraction of the 0-inputs of f have sensitivity at least $(\frac{3}{4}+\gamma)n$. Since $\mathbf{Pr}[f(x)=0] \geq \frac{1}{2}-\varepsilon$, it follows that there is a set $S \subseteq \{0,1\}^n$ of $\Omega(2^n)$ inputs such that $f(x)=0$ and $s(f,x) \geq (\frac{3}{4}+\gamma)n$ for all $x \in S$. Note that for every pair $x, y \in S$ share at least $(\frac{1}{2}+2\gamma)n$ sensitive coordinates. Next we note that for every $x \in S$, there are at most $2^{H(\frac{1}{2}+\gamma)n}$ inputs $y \in S$ that are at distance $(\frac{1}{2}+\gamma)n$ from x. It follows that there is a subset $S^* \subseteq S$ of size at least $|S|/2^{H(\frac{1}{2}+\gamma)n} = 2^{\Omega(n)}$ such that for all $x, y \in S^*$, there exists some $i \in [n]$ such that $x_i \neq y_i$ and $s(f,x,i) = s(f,y,i) = 1$. Applying Lemma 4.6.1 yields the claimed lower bound. □

4.7 Conclusion

Having obtained asymptotically matching universal bounds on the approximability of all Boolean functions with respect to DNF width in this work, the natural next step would be to do likewise for DNF size, closing the current gap between $\Omega_\varepsilon(2^n/n)$ and $O_\varepsilon(2^n/\log n)$.

Open Problem 1. *Obtain matching universal bounds on the approximability of all Boolean functions with respect to DNF size. That is, determine the function $\varphi(n)$ such that every Boolean function f can be ε-approximated by a DNF of size $O_\varepsilon(2^n/\varphi(n))$, and there exists an f such that any ε-approximator for f has size $\Omega_\varepsilon(2^n/\varphi(n))$.*

Another open problem is to prove that the size of our DNF approximators for PAR_n in Theorem 41 is optimal even up to the exact dependence on ε in the exponent, closing the current gap between $(1-2\varepsilon)n$ and $\frac{(1-2\varepsilon)}{(1+2\varepsilon)}n$. One way to accomplish this is to further improve on the sharpening of Boppana's influence bound on small-size DNFs we obtained in [Blais *et al.*, 2013b]; we believe that understanding this basic complexity measure of DNFs is a fundamental question in its own right. We restate here the conjectured bound from

[Blais *et al.*, 2013b], which would be tight by considering the parity function on $\log(s) + 1$ variables.

Conjecture 2. *Let f be computed by a size-s DNF. Then* $\mathbf{Inf}[f] \leq \log(s) + 1$.

Chapter 5

On DNF Approximators for Monotone Boolean Functions

5.1 Background and context

Monotone Boolean functions constitute a rich and complex class of functions, and their structural and combinatorial properties have been intensively studied for decades; see e.g. the monograph [Korshunov, 2003] for an in-depth survey. In complexity theory monotone functions play an especially important role in circuit complexity, where Razborov's celebrated result [Razborov, 1985] has led to a significant body of work centered around monotone functions and the circuits that compute them [Alon and Boppana, 1987; Ajtai and Grevich, 1987; Karchmer and Wigderson, 1988; Tardos, 1988; Raz and Wigderson, 1990; Karchmer *et al.*, 1991; Grigni and Sipser, 1995; Razborov and Rudich, 1997; Goldmann and Håstad, 1998; Raz and McKenzie, 1999; Potechin, 2010; Chan and Potechin, 2012; Filmus *et al.*, 2013].

In this chapter we study the circuit complexity of *approximating* monotone functions, focusing on DNF formulas, one of the simplest and most basic types of circuits. We say that a DNF ε-approximates a function $f : \{0,1\}^n \to \{0,1\}$ if the function g computed by the DNF satisfies $f(x) = g(x)$ on at least a $1 - \varepsilon$ fraction of inputs x in $\{0,1\}^n$. Recent works [O'Donnell and Wimmer, 2007; Blais and Tan, 2013] have highlighted interesting qualitative and quantitative differences in the landscape of DNF complexity when the formula is only

required to approximate f rather than compute it exactly, and while the DNF complexity of exact computation is fairly well-understood, these papers have also pointed to significant gaps in our understanding of seemingly basic questions regarding the DNF complexity of approximate computation.

We continue this study and explore two main directions. In the first direction we seek a non-trivial upper bound on the DNF complexity of approximating an arbitrary monotone function to high accuracy, in the spirit of the positive results of [Blais and Tan, 2013]. In the second direction, in the spirit of Razborov's theorem [Razborov, 1985] we seek a separation between the relative powers of monotone and non-monotone DNF that approximate monotone functions. As we describe below, our results further illustrate how different DNF complexity can be in the settings of exact versus approximate computation.

Universal bounds on approximability Recent work of [Blais and Tan, 2013] established the first non-trivial universal upper bound on the DNF complexity of approximating an arbitrary Boolean function, achieving logarithmic savings over the worst-case cost of $\Omega(2^n)$ necessary for exact computation:

Theorem 1 of [Blais and Tan, 2013]. *Every Boolean function can be ε-approximated by a DNF of size $O_\varepsilon(2^n/\log n)$.*

We begin with the simple observation that this result does not say anything meaningful about the approximation of monotone functions. Since the minimal satisfying assignments of a monotone function form a Sperner family, Sperner's classical theorem readily translates into an upper bound on the DNF complexity of *exactly* computing monotone functions that is polynomially stronger:

Fact 5.1.1. *Every monotone function can be computed exactly by a DNF of size $\binom{n}{\lceil n/2 \rceil} = \Theta(2^n/\sqrt{n})$.*

This bound is exactly tight by considering the n-variable majority function, and in fact an elementary combinatorial argument establishes that a $1 - o_n(1)$ fraction of monotone functions do actually require DNFs of size $\Omega(2^n/\sqrt{n})$ to compute. Fact 5.1.1, taken together with the result of [Blais and Tan, 2013], raises a basic qualitative question: are there monotone functions that require DNFs of size $\Omega(2^n/\sqrt{n})$ to approximate, or can every monotone

function be approximated by a DNF of size $o(2^n/\sqrt{n})$? Despite the vast literature on monotone functions and Sperner families, this question does not appear to have been explicitly studied before. We answer this question in the first half of the chapter, constructing DNF approximators for arbitrary monotone functions that achieve exponential savings over the size necessary for exact computation. Our DNF approximators only make one-sided error, and our construction is based on a new structural decomposition of monotone functions.

Power of negations in approximating monotone functions In the second half of the chapter we turn our attention to the role of *negations* in the DNF complexity of approximating monotone functions. Recall that a circuit is said to be monotone if it does not contain any NOT gates, and non-monotone otherwise. While every monotone function can be computed by a monotone circuit, there is a body of results showing the remarkable fact that for various circuit classes, the optimal circuit computing a monotone function must be non-monotone. The most prominent example is perhaps Razborov's celebrated lower bound:

Razborov's Theorem [Razborov, 1985]. *There is a polynomial-time computable monotone function that requires monotone circuits of quasi-polynomial size.*

This separation of monotone NP from monotone P/poly was subsequently improved from quasi-polynomial to exponential by E. Tardos [Tardos, 1988]. An analogue of Razborov's result in the setting of bounded-depth circuits was established by Okol'nishnikova, Ajtai, and Gurevich:

Okol'nishnikova–Ajtai–Gurevich Theorem [Okol'nishnikova, 1982; Ajtai and Gurevich, 1987]. *There is a monotone function in* AC^0 *that is not in monotone* AC^0.

For the class of DNFs, however, it is well-known (and straightforward to verify) that the analogue these separations does not hold:

Quine's Theorem [Quine, 1954]. *The optimal DNF, with respect to both size and width, computing a monotone function is monotone as well.*

In the second half of this chapter we investigate the question: does Quine's theorem hold for *approximation* by DNFs? In other words, is the optimal DNF approximator for

a monotone function monotone as well, or do negations buy us power in the setting of approximation? We show that the answer is the latter, giving separations with respect to both DNF size and width. Our results, taken in contrast with Quine's theorem, highlight an interesting qualitative difference between the DNF complexity of exact and approximate computation. More broadly, we believe that the role of negations in the circuit complexity of approximating monotone functions is a topic of intrinsic interest, and we view our separations as the first steps in its systematic study.

5.1.1 Our results

Universal bounds on approximability Our first result is the construction of DNF approximators for arbitrary monotone Boolean functions that achieve one-sided error:

Theorem 54. *Every monotone function f can be ε-approximated by a monotone function g of DNF size $2^{n-\Omega_\varepsilon(\sqrt{n})}$, satisfying $g(x) \leq f(x)$ for all $x \in \{0,1\}^n$.*

Prior to our work the only known universal upper bound, even for approximators incurring two-sided error, was the trivial one of $\binom{n}{\lceil n/2 \rceil} = \Theta(2^n/\sqrt{n})$, the size sufficient for exact computation. A standard information-theoretic argument (see [Blais and Tan, 2013] for proof) shows that any ε-approximator for a random Boolean function has DNF size $\Omega_\varepsilon(2^n/n)$; Theorem 54 therefore shows that the structure of monotonicity can be leveraged to obtain DNF approximators with complexity exponentially smaller than that required for almost all other functions. Our construction relies on a new structural fact about monotone functions which we believe may be of independent interest:

Lemma 5.1.2. *Let f be a monotone function and $\varepsilon > 0$. There is a function $g = g_1 \vee \cdots \vee g_t$ that ε-approximates f, where $t = O_\varepsilon(1)$ and each g_i is a monotone DNF with terms of width exactly k_i and size at least $(\varepsilon/2)\binom{n}{k_i}$. Furthermore, $g(x) \leq f(x)$ for all $x \in \{0,1\}^n$.*

Since $g(x) \leq f(x)$ for all $x \in \{0,1\}^n$, we say that g is a lower ε-approximator for f. We prove Lemma 5.1.2 in Section 5.2, and with this structural fact in hand, the task of constructing lower approximators for an arbitrary monotone function reduces to that of constructing lower approximators for the g_i's. Since g comprises only a constant number of these g_i's, taking a naive union bound incurs no more than a constant factor in terms of error

and DNF size of the overall approximator. Our lower approximators for the g_i's, presented in Section 5.3, are obtained via a randomized algorithm that constructs an approximating DNF. We complement our positive result with a lower bound showing that Theorem 54 is essentially optimal:

Theorem 55. *Let g be a $\frac{1}{10}$-approximator for the majority function MAJ_n satisfying $g(x) \leq \mathsf{MAJ}_n(x)$ for all $x \in \{0,1\}^n$. Then g has DNF size $2^{n-O(\sqrt{n}\log n)}$.*

Power of negations in approximating monotone functions The proof of Quine's classical theorem mentioned in the introduction is simple: given a DNF g that computes a monotone function f, if g contains a term T with a negated variable $\bar{x}_i$, it is easy to check that g still computes the same monotone function f if $\bar{x}_i$ is removed from T. Therefore, by removing all occurrences of negated variables in g, we obtain a monotone DNF h computing the same function f, where the size and width of h are at most those of g.

It is natural to suspect that the same would be true for DNF approximators, that the optimal DNF approximator for a monotone function is always monotone as well; indeed, we note that the universal DNF approximators we construct in Theorem 54 are in fact monotone. To be precise, we consider the following question:

Question 1. *Let f be a monotone function that is ε-approximated by a DNF g of size s (resp. width w). Can f be ε-approximated by a* monotone *DNF h of size s (resp. width w)?*

The simple proof of Quine's theorem does not extend to answer this question in the affirmative. In fact, for all three natural ways of "locally monotonizing" the DNF approximator g — removing $\bar{x}_i$ in T (as is done in the proof of Quine's theorem); replacing $\bar{x}_i$ with x_i in T; and removing T from f entirely — it is possible to construct examples showing that these operations increase the distance of g from f (i.e. worsens the quality of approximation).

In the second half of the chapter we resolve Question 1 by showing, perhaps somewhat surprisingly, that the answer is "No" for both complexity measures of DNF size and DNF width. In Section 5.4 we prove the following two theorems:

Theorem 56 (Separation for DNF size)**.** *For all sufficiently large n, there exists an n-variable monotone function f and a value $\varepsilon = \varepsilon(n) > 0$ such that f can be ε-approximated*

by a DNF of size $O(n)$, but any monotone function that ε-approximates f has DNF size $\Omega(n^2)$.

Theorem 57 (Separation for DNF width)**.** *For all sufficiently large n, and for all $k = o(n)$, there exists an n-variable monotone function f and a value $\varepsilon = \varepsilon(n) > 0$ such that f can be ε-approximated by a DNF of width $k + \log k$, but any monotone function that ε-approximates f has DNF width at least $2k - 1 - o_n(1)$.*

We view these separations as the first steps in quantifying just how powerful negations can be in the approximation of monotone functions, a question that does not appear to have been explicitly studied before (despite a significant body of results on the power of negations in the computation of monotone functions, as discussed above). We conclude the chapter by listing a few interesting questions for future work in this direction.

5.1.2 Previous work

The explicit study of the DNF complexity of approximating Boolean functions was initiated by O'Donnell and Wimmer [O'Donnell and Wimmer, 2007]. They showed that DNF size $2^{O_\varepsilon(\sqrt{n})}$ is both necessary and sufficient for ε-approximating the n-variable majority function, and constructed an explicit n-variable monotone function for which any 0.01-approximating DNF must have size $2^{\Omega(n/\log n)}$. As mentioned above, Blais and Tan [Blais and Tan, 2013] gave universal upper bounds on DNF size for approximating arbitrary Boolean functions, but [Blais and Tan, 2013] does not consider monotone functions.

We also note that the earlier work of Bshouty and Tamon [Bshouty and Tamon, 1996], which established Fourier concentration bounds for monotone Boolean functions, implies that every n-variable monotone function is ε-close to a depth-2 circuit of size $2^{O(\sqrt{n}\log(n)/\varepsilon)}$ in which the bottom-level gates are parity gates and the top gate is a threshold gate (with unbounded weights). Recall that while threshold-of-parity circuits can simulate DNF formulas with only a polynomial size increase [Jackson, 1997; Krause and Pudlák, 1997], the converse is not true (indeed, even a single parity gate requires exponential DNF size). Thus the [Bshouty and Tamon, 1996] results do not imply the existence of nontrivial DNF approximators for monotone functions.

5.1.3 Preliminaries

Throughout this chapter all probabilities and expectations are with respect to the uniform distribution unless otherwise stated; we will use boldface (*e.g.* $\boldsymbol{x}$ and $\mathbf{X}$) to denote random variables. For strings $x, y \in \{0,1\}^n$ we write $\|x\|$ to denote the Hamming weight $\#\{i \in [n] \colon x_i = 1\}$ of x, and $x \preceq y$ if $x_i \leq y_i$ for all $i \in [n]$, and $x \prec y$ if $x \preceq y$ and $x \neq y$. For $0 \leq k \leq n$, we write $\mathrm{Vol}(n,k) := \sum_{i=0}^{k} \binom{n}{i}$ to denote the volume of the n-dimensional Hamming ball of radius k.

A monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$ is one that satisfies $f(x) \leq f(y)$ whenever $x \preceq y$. A DNF formula is the logical OR of logical ANDs, where we refer to each AND as a *term*. The *size* of a DNF is the number of terms it contains, and the *width* of a DNF is the maximum width of any term. For a term T, we write $|T|$ to denote the width of T, the number of literals occurring in it. For any $x \in \{0,1\}^n$, we write T_x to denote the monotone conjunction that accepts all and only those $y \in \{0,1\}^n$ such that $y \succeq x$. That is, $T_x(y) = 1$ iff $y_i = 1$ for all $i \in [n]$ such that $x_i = 1$. We say that x *defines a minterm in a monotone function* f if T_x is a minterm in the canonical DNF computing f, and we write $\mathrm{minterm}(x, f)$ to denote the indicator for this event.

Definition 58 (canonical DNF)**.** Let f be a monotone Boolean function. The *canonical DNF* for f is the unique monotone DNF whose terms correspond precisely to the minterms of f.

Definition 59 (ε-approximator)**.** Let $f, g : \{0,1\}^n \to \{0,1\}$ be Boolean functions and $\varepsilon \in [0,1]$. We say that g is an ε-approximator for f, or that f and g are ε-close, if $\mathbf{Pr}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] \leq \varepsilon$. We say that g is a lower approximator for f if $g(x) \leq f(x)$ for all $x \in \{0,1\}^n$, and an upper approximator for f if $f(x) \leq g(x)$ for all $x \in \{0,1\}^n$.

Definition 60 (density)**.** Let $f : \{0,1\}^n \to \{0,1\}$ and $k \in \{0,1,\ldots,n\}$. The density of f at level k, denoted $\mu_k(f)$, is defined to be

$$\mu_k(f) = \mathop{\mathbf{Pr}}_{\|x\|=k}[f(\boldsymbol{x}) = 1] = \#\{x \in \{0,1\}^n \colon \|x\| = k \text{ and } f(x) = 1\} \cdot \binom{n}{k}^{-1}.$$

Fact 5.1.3. *Let f be a monotone function. Then $\mu_k(f) \geq \mu_{k-1}(f)$ for all $k \in [n]$.*

We recall two basic facts from probability theory.

Fact 5.1.4 (Chernoff bound)**.** *Let* $\mathbf{X} \sim \text{Binomial}(n, 1/2)$*. Then for any* $0 \le t \le \sqrt{n}$*, we have* $\mathbf{Pr}\left[\mathbf{X} \ge \frac{n}{2} + t\frac{\sqrt{n}}{2}\right] \le e^{-t^2/2}$ *and* $\mathbf{Pr}\left[\mathbf{X} \le \frac{n}{2} - t\frac{\sqrt{n}}{2}\right] \le e^{-t^2/2}$.

Fact 5.1.5 (anti-concentration of the Binomial)**.** *For every* $\varepsilon \ge 1/\sqrt{n}$ *and interval* $I \subseteq [0, n]$ *of width at most* $\varepsilon\sqrt{n}$*, we have* $\mathbf{Pr}_{\boldsymbol{x}\in\{0,1\}^n}[\|\boldsymbol{x}\| \in I] \le 2\varepsilon$.

5.2 A regularity lemma for monotone DNFs

We begin with a new structural fact about monotone functions, which states that every monotone DNF f is lower approximated by the disjunction g of a constant number of monotone DNFs that are "dense" and "regular." Here a "regular" DNF is one in which all terms have the same width k, and a "dense" regular DNF is one that contains a constant fraction of the $\binom{n}{k}$ many possible terms of width k. This structural decomposition is useful as it reduces the task of (lower) approximating an arbitrary monotone DNF f to that of (lower) approximating a dense regular one. Since g is the disjunction of only a constant number of dense regular DNFs, taking a naive union bound incurs only a constant factor in terms of error and DNF size of the overall approximator.

Definition 61 (regular and dense DNFs)**.** Let $k \in [n]$. We say that a monotone DNF f is k-regular if all its terms have width exactly k, and regular if it is k-regular for some k. Additionally, we say that f is (ε, k)-regular if it is a k-regular DNF with at least $\varepsilon\binom{n}{k}$ many terms.

Our structural result says that every monotone function is lower ε-approximated by the disjunction of $O_\varepsilon(1)$ many $(\varepsilon/2, k_i)$-regular DNFs, where each $k_i = (n/2) \pm O(\sqrt{n})$. More precisely:

Lemma 5.1.2. *For any* $\varepsilon > 0$*,every monotone function* f *is* ε*-close to the disjunction* g *of monotone DNFs,* $g(x) = g_1(x) \vee \cdots \vee g_t(x)$*, where*

1. $t \le 2/\varepsilon$*,*
2. *each* g_i *is* k_i*-regular for some* $k_i \in \left[(n/2) - \sqrt{n\ln(4/\varepsilon)/2}, (n/2) + \sqrt{n\ln(4/\varepsilon)/2}\right]$*,*
3. *the DNF size of* g_i *is at least* $(\varepsilon/2)\binom{n}{k_i}$ *(i.e.,* $\mu_{k_i}(g_i) \ge \varepsilon/2$*).*

4. $g(x) \leq f(x)$ *for all* $x \in \{0,1\}^n$.

Proof. Fix $\ell := \sqrt{n \ln(4/\varepsilon)/2}$. For each $k = \{0, 1, \ldots, n\}$ we define

$$f_k(x) := \bigvee \{T_x \colon \|x\| = k \text{ and } \mathrm{minterm}(x, f) = \mathsf{true}\},$$

where we recall that T_x is the monotone term that accepts all y such that $y \succeq x$. By the Chernoff bound $\mathbf{Pr}_{\boldsymbol{x} \in \{0,1\}^n}[|\|\boldsymbol{x}\| - n/2| \geq \ell] \leq \varepsilon/2$, and so f is $(\varepsilon/2)$-close to

$$f^*(x) = f_{(n/2)-\ell}(x) \vee \cdots \vee f_{(n/2)+\ell}(x).$$

Furthermore, $f^*(x)$ is an lower approximator for f. By the triangle inequality, it suffices to prove that f^* is $(\varepsilon/2)$-close to g satisfying the four claims in the lemma statement.

Consider the algorithm below, and let g be the resulting function when the algorithm terminates. First, since the algorithm only sets $f^*(x) = 0$ for inputs x that define a minterm in f^*, we have that g is a monotone lower approximator for f^*. Second, since the algorithm corrupts less than an $(\varepsilon/2)$-fraction of any layer, g is $(\varepsilon/2)$-close to f^*.

```
regularize(f*):
  1. for k = (n/2) − ℓ, ..., (n/2) + ℓ:
  2.     if Pr_{‖x‖=k}[minterm(x, f*)] < ε/2
  3.         set f*(x) = 0 for all x s.t. ‖x‖ = k and minterm(x, f*) = true.
```

We will argue that g is the disjunction of regular monotone DNFs satisfying the first three claims in the lemma. The algorithm ensures that for every $k \in [(n/2) - \ell, (n/2) + \ell]$, the fraction of inputs at layer k that define a minterm in g is either 0 at least $\varepsilon/2$ — if this fraction is in the range $[0, \varepsilon/2)$, the predicate in Line 2 of the algorithm is satisfied and the fraction is set to 0 by Line 3. Furthermore, since all the minterms of f^* have between $(n/2) - \ell$ and $(n/2) + \ell$ variables, the same is true for g and so

$$\mathop{\mathbf{Pr}}_{\|\boldsymbol{x}\|=k}[\mathrm{minterm}(\boldsymbol{x}, g)] = \begin{cases} 0 & \text{if } k \notin [(n/2) - \ell, (n/2) + \ell] \\ 0 \text{ or } \geq \varepsilon/2 & \text{if } k \in [(n/2) - \ell, (n/2) + \ell]. \end{cases}$$

Each layer $k_i \in [(n/2) - \ell, (n/2) + \ell]$ such that this probability is at least $\varepsilon/2$ naturally defines a k_i-regular monotone DNF g_i satisfying the second and third claims of the lemma: g_i is simply the DNF

$$g_i(x) := \bigvee \{T_x \colon \|x\| = k_i \text{ and } \mathrm{minterm}(x, g)\}.$$

Therefore $g(x) = g_1(x) \vee \cdots \vee g_t(x)$ where each g_i is k_i-regular and $\mu_{k_i}(g_i) \geq \varepsilon/2$, and so it remains to justify the first claim of the lemma, that $t \leq 2/\varepsilon$. We assume without loss of generality that $k_1 < k_2 < \ldots < k_t$, and claim that

$$\mu_{k_i}(g_1 \vee \cdots \vee g_i) \geq i \cdot \frac{\varepsilon}{2} \quad \text{for all } i \in [t]. \tag{5.1}$$

Note that this implies the first claim of the lemma since $\mu_{k_t}(g_1 \vee \ldots \vee g_t) \leq 1$ holds trivially, and so $t \leq 2/\varepsilon$. We prove (5.1) by induction on i, noting that the base case holds since $\mu_{k_i}(g_i) \geq \varepsilon/2$ for all i by construction, and in particular when $i = 1$. Suppose $\mu_{k_i}(g_1 \vee \ldots \vee g_i) \geq i \cdot \varepsilon/2$ for some $i < t$. By Fact 5.1.3, we have

$$\mu_{k_{i+1}}(g_1 \vee \cdots \vee g_i) \geq \mu_{k_i}(g_1 \vee \cdots \vee g_i) \geq i \cdot \frac{\varepsilon}{2}.$$

Since the terms of g_{i+1} are the width-(k_{i+1}) minterms of g, the sets

$$\begin{aligned} A &= \{x \in \{0,1\}^n \colon \|x\| = k_{i+1} \text{ and } g_1(x) \vee \cdots \vee g_i(x) = 1\} \\ B &= \{x \in \{0,1\}^n \colon \|x\| = k_{i+1} \text{ and } g_{i+1}(x) = 1\} \\ &\equiv \{x \in \{0,1\}^n \colon \|x\| = k_{i+1} \text{ and } \mathrm{minterm}(x, g)\} \end{aligned}$$

are disjoint, and so

$$\mu_{k_{i+1}}(g_1 \vee \cdots \vee g_{i+1}) = \mu_{k_{i+1}}(g_1 \vee \ldots \vee g_i) + \mu_{k_{i+1}}(g_{i+1}) = \left(i \cdot \frac{\varepsilon}{2}\right) + \frac{\varepsilon}{2} = (i+1) \cdot \frac{\varepsilon}{2}.$$

This completes the proof. □

5.3 Lower approximators for regular DNFs

With Lemma 5.1.2 in hand it suffices to construct lower approximators for regular DNFs:

Proposition 5.3.1. *Let f be a regular monotone function. For every $\varepsilon > 0$ there exists a monotone DNF g of size $2^{n - \Omega(\varepsilon\sqrt{n} - \log(n))}$ that is a lower ε-approximator for f.*

Proof of Theorem 54 assuming Proposition 5.3.1. By Lemma 5.1.2 every monotone f has a lower $(\varepsilon/2)$-approximator $g(x) = g_1(x) \vee \cdots \vee g_t(x)$ where $t \leq 4/\varepsilon$ and each $g_i(x)$ is a regular monotone function. Next, by Proposition 5.3.1 each regular $g_i(x)$ has a lower $(\varepsilon/2t)$-approximator $h_i(x)$ of size $2^{n-\Omega((\varepsilon\sqrt{n}/t)-\log(n))}$. Finally, by the union bound and the triangle inequality, we conclude that $h(x) = h_1(x) \vee \cdots \vee h_t(x)$ is a lower ε-approximator for f of size at most $t \cdot 2^{n-\Omega((\varepsilon\sqrt{n}/t)-\log(n))} = 2^{n-\Omega_\varepsilon(\sqrt{n})}$. □

Proof of Proposition 5.3.1. We may assume that $\varepsilon \geq (C \log n)/\sqrt{n}$ (for some constant $C > 0$ which we will specify below), since otherwise the claimed bound on monotone DNF size is trivial. Let f be a k-regular monotone function for some $k \in [n]$. The minterms of our monotone approximator g will be conjunctions of the form T_y where $y \in f^{-1}(1)$, which guarantees that g will be a lower approximator for f. Furthermore, since $\mathbf{Pr}_{\boldsymbol{x}\in\{0,1\}^n}\left[\|\boldsymbol{x}\| \geq (n/2) + \sqrt{n\ln(3/\varepsilon)/2}\right] \leq \frac{\varepsilon}{3}$, and $\mathbf{Pr}_{\boldsymbol{x}\in\{0,1\}^n}\left[\|\boldsymbol{x}\| \in [k, k+\varepsilon\sqrt{n}/6]\right] \leq \frac{\varepsilon}{3}$, by the Chernoff bound and Fact 5.1.5 respectively, it suffices to ensure that the monotone DNF g we construct additionally satisfies:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\in A}\left[g(\boldsymbol{x}) \neq f(\boldsymbol{x})\right]] \leq \frac{\varepsilon}{3}, \quad A := \left\{x \in \{0,1\}^n \colon \|x\| \in \left[k + \varepsilon\sqrt{n}/6, (n/2) + \sqrt{n\ln(3/\varepsilon)/2}\right]\right\}. \tag{5.2}$$

Note that if $k + \varepsilon\sqrt{n}/6 > (n/2) + \sqrt{n\ln(3/\varepsilon)/2}$ (*i.e.* the interval in the definition of A is empty) then f is $(2\varepsilon/3)$-close to the constant 0 function and the proposition is trivially true.

For every $\ell \in \{0, 1, \ldots, n-k\}$, we write S_ℓ to denote the 1-inputs of f with Hamming weight exactly $k+\ell$; that is, $S_\ell := \{x \in \{0,1\}^n \colon f(x) = 1 \text{ and } \|x\| = k+\ell\}$. The remainder of this proof will be devoted to showing that for each $\ell \geq \varepsilon\sqrt{n}/6$, there exists a monotone DNF g_ℓ satisfying:

i. The minterms of g_ℓ are of the form T_y for some $y \in S_{\ell/2}$ (and hence $g_\ell \leq f$),

ii. DNF-size$[g_\ell] = O(2^{n-\ell/2}) \leq 2^{n-\Omega(\varepsilon\sqrt{n})}$,

iii. $\mathbf{Pr}_{\boldsymbol{x}\in S_\ell}[g_\ell(\boldsymbol{x}) = 0] \leq \varepsilon/3$.

Indeed, taking g to be the disjunction of all g_ℓ where $k+\ell \in \left[k + \varepsilon\sqrt{n}/3, (n/2) + \sqrt{n\ln(3/\varepsilon)/2}\right]$, we obtain a monotone DNF of size at most $n \cdot 2^{n-\Omega(\varepsilon\sqrt{n})} \leq 2^{n-\Omega(\varepsilon\sqrt{n}-\log(n))}$ satisfying (5.2), which completes the proof.

Consider a random monotone DNF $\boldsymbol{g}_\ell$ sampled according to the following distribution $\mathcal{D}$: for each $y \in S_{\ell/2}$, independently include T_y as a minterm of $\boldsymbol{g}_\ell$ with probability $p := 2^{-\ell/2}$. By definition, every DNF in the support of this distribution satisfies (i), and so it remains to argue that with positive probability, both (ii) and (iii) are satisfied as well. For (ii), we observe that $\mathbf{E}_{\mathcal{D}}\left[\text{DNF-size}[\boldsymbol{g}_\ell]\right] = p \cdot |S_\ell| < p \cdot 2^n = 2^{n-\ell/2}$, and so by Markov's inequality,

$$\mathop{\mathbf{Pr}}_{\mathcal{D}}\left[\text{DNF-size}[\boldsymbol{g}_\ell] \leq 3 \cdot 2^{n-\ell/2}\right] \geq \frac{2}{3}. \tag{5.3}$$

For (iii), consider any fixed $x \in S_\ell$. Since f is k-regular, there must exist some $z \in S_0$ such that $z \prec x$, and therefore $\binom{\ell}{\ell/2} = \Theta(2^\ell/\sqrt{\ell})$ many $y \in S_{\ell/2}$ such that $z \prec y \prec x$. By the definition of $\mathcal{D}$, for each such y the term T_y is independently included as a minterm of $\boldsymbol{g}_\ell$ with probability $p = 2^{-\ell/2}$, and so

$$\mathop{\mathbf{Pr}}_{\mathcal{D}}[\boldsymbol{g}_\ell(x) = 0] \leq (1-p)^{\Theta(2^\ell/\sqrt{\ell})} = \exp\left(-\Omega(2^{\ell/2}/\sqrt{\ell})\right) < \exp\left(-\Omega(2^{\varepsilon\sqrt{n}/12})/\sqrt{n}\right) < \frac{\varepsilon}{9},$$

where we have used $\varepsilon \geq (C \log n)/\sqrt{n}$ for the final inequality. Therefore

$$\mathop{\mathbf{E}}_{\mathcal{D}}\left[\mathop{\mathbf{Pr}}_{\boldsymbol{x}\in S_\ell}[\boldsymbol{g}_\ell(\boldsymbol{x}) = 0]\right] \leq \frac{\varepsilon}{9}, \quad \text{and} \quad \mathop{\mathbf{Pr}}_{\mathcal{D}}\left[\mathop{\mathbf{Pr}}_{\boldsymbol{x}\in S_\ell}[\boldsymbol{g}_\ell(\boldsymbol{x}) = 0] \leq \frac{\varepsilon}{3}\right] \geq \frac{2}{3}. \tag{5.4}$$

Applying a union bound to the failure probabilities of (5.3) and (5.4), we conclude that there is indeed a positive probability that $\boldsymbol{g}_\ell \sim \mathcal{D}$ satisfies all three properties (i), (ii), and (iii), and this completes the proof. □

5.3.1 Near-Matching Lower Bound

In this section we show that our upper bound in Theorem 54 is essentially tight.

Theorem 55. *Let $\varepsilon \leq \frac{1}{10}$ and g be an s-term DNF that is a lower ε-approximator for the majority function MAJ_n. Then $s \geq 2^{n-O(\sqrt{n}\log n)}$.*

Proof. First we claim that we may assume without loss of generality that g is an $\lceil n/2 \rceil$-regular monotone function. To see this, fix a DNF representation of g and consider any term

$$T(x) = \left(\bigwedge_{i\in S^+} x_i\right) \wedge \left(\bigwedge_{j\in S^-} \overline{x}_j\right), \quad S^+, S^- \subseteq [n]$$

in the DNF. Note that $|S^+| \geq \lceil n/2 \rceil$, since otherwise $g(y) = 1$ and $\mathsf{MAJ}_n(y) = 0$ on the input y where $y_i = 1$ iff $i \in S^+$ (of Hamming weight $\|y\| = |S^+| < \lceil n/2 \rceil$), contradicting our assumption that $g(x) \leq \mathsf{MAJ}_n(x)$ for all $x \in \{0,1\}^n$. Replacing $T(x)$ in g by $T'(x) = \bigwedge_{i \in S} x_i$, where S is an arbitrary subset of S^+ of cardinality exactly $\lceil n/2 \rceil$, we get a function g^* satisfying $g^{-1}(1) \subseteq (g^*)^{-1}(1) \subseteq \mathsf{MAJ}_n^{-1}(1)$. Performing this replacement for every term in g, we obtain an $\lceil n/2 \rceil$-regular monotone DNF of size at most s that lower ε-approximates MAJ_n.

Next we claim that if g is an $\lceil n/2 \rceil$-regular monotone DNF that ε-approximates MAJ_n, then $\mu_\ell(g) \geq 1 - 4\varepsilon$ for $\ell := \lceil n/2 \rceil + \sqrt{n \ln 2}$. If $\mu_\ell(g) < 1 - 4\varepsilon$, then by Fact 5.1.3 by have $\mu_k(g) \leq \mu_\ell(g) < 1 - 4\varepsilon$ for all $k \leq \ell$, and since

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \in \{0,1\}^n} \left[\|\boldsymbol{x}\| \in \left[\lceil n/2 \rceil, \ell \right] \right] \geq \frac{1}{4},$$

by the Chernoff bound, we get $\mathbf{Pr}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] > \varepsilon$, a contradiction.

Having established that $\mu_\ell(g) \geq 1 - 4\varepsilon$, we complete the proof with a simple counting argument. For every x of weight $\lceil n/2 \rceil$, we have $|\{y \in \{0,1\}^n \colon \|y\| = \ell \text{ and } y \succ x\}| = \binom{\lceil n/2 \rceil}{\sqrt{n \ln 2}}$. Since $\mu_\ell(g) \geq 1 - 4\varepsilon = \Omega(1)$ and there are $\binom{n}{\ell}$ strings of Hamming weight ℓ, we conclude that the number of terms in the canonical DNF for g is at least

$$\mu_\ell(g) \binom{n}{\ell} \cdot \binom{\lceil n/2 \rceil}{\sqrt{n \ln 2}}^{-1} = \Omega(1) \cdot \binom{n}{\lceil n/2 \rceil + \sqrt{n \ln 2}} \cdot \binom{\lceil n/2 \rceil}{\sqrt{n \ln 2}}^{-1} = 2^{n - O(\sqrt{n} \log n)}$$

as claimed. □

5.4 Power of negations in approximating monotone functions

In this section we present our constructions showing that non-monotone DNFs can asymptotically outperform monotone ones in the approximation of monotone functions. We present our separation for DNF size in Section 5.4.1, followed by our separation for DNF width in Section 5.4.2.

5.4.1 Separation for DNF size

Theorem 56. *Let $f : \{0,1\}^n \times \{0,1\}^{5n} \to \{0,1\}$ be the monotone function:*

$$f(x,y) = (x_1 \vee \ldots \vee x_n) \wedge (y_1 \vee \ldots \vee y_{5n}) = \bigvee_{\substack{i\in[n]\\ j\in[5n]}} (x_i \wedge y_j),$$

and $\varepsilon = (2^{n-1}-1)\cdot 2^{-6n}$. There exists a DNF of size $6n-1$ that ε-approximates f, but any monotone function that ε-approximates f has DNF size at least n^2.

Proof. Consider the function $g = g(x,y)$ defined as

$$g = (x_1 \wedge (y_1 \vee \ldots \vee y_{5n})) \vee (\overline{x}_1 \wedge (x_2 \vee \ldots \vee x_n)) = \left(\bigvee_{j\in[5n]} (x_1 \wedge y_j)\right) \vee \left(\bigvee_{2\le i\le n} (\overline{x}_1 \wedge x_i)\right).$$

This is a non-monotone DNF with $6n-1$ terms that ε-approximates f, since $g(x,y)$ differs from $f(x,y)$ exactly on the $2^{n-1}-1$ inputs satisfying $x_1 = 0$, $y = \mathbf{0}$, and $x_2 \vee \ldots \vee x_n = 1$.

The rest of the proof will be devoted to showing that any monotone function that ε-approximates f has to have more than n^2 terms, asymptotically as many as the canonical DNF for f which has $5n^2$ terms. We will prove the contrapositive: any monotone DNF h with at most n^2 terms differs from f on strictly more than an ε-fraction of inputs.

We group the terms of h into three types: terms with only x-variables, which we call "pure-x"; terms with only y-variables, which we call "pure-y"; and terms with both x- and y-variables, which we call "mixed". We first observe that we may assume that all mixed terms have width exactly two, comprising one x-variable and one y-variable. Indeed, replacing a mixed term $\left(\bigwedge_{i\in S_1} x_i\right) \wedge \left(\bigwedge_{j\in S_2} y_j\right)$, $S_1 \subseteq [n]$ and $S_2 \subseteq [5n]$, in h with $(x_i \wedge y_j)$ for any $i \in S_1$ and $j \in S_2$ yields a DNF h' such that $h'(x,y) \neq h(x,y)$ only on inputs (x,y) such that $h(x,y) = 0$ and $f(x,y) = 1$.

Furthermore, we claim that we may assume all pure-y terms have width greater than $2n$. Indeed, if h contains a term $T(y) = \bigwedge_{i\in S} y_i$ for some $S \subseteq [5n]$ where $|S| \le 2n$, then $f(x,y) = 0$ and $h(x,y) = 1$ on at least $2^{3n} > \varepsilon \cdot 2^{6n}$ inputs (x,y) satisfying $x = \mathbf{0}$ and $T(y) = 1$.

We proceed by considering two cases, depending on the number of x_i's that occur as a singleton term in h. First suppose at least half of the x_i's occur as a singleton term in h, so

there is some $S \subseteq [n]$ where $|S| \geq n/2$ such that if $\mathsf{OR}_S(x) = \bigvee_{i \in S} x_i = 1$ then $h(x,y) = 1$. In this case $f(x,y) = 0$ and $h(x,y) = 1$ on at least $2^n - 2^{n/2} > \varepsilon \cdot 2^{6n}$ inputs satisfying $y = \mathbf{0}$ and $\mathsf{OR}_S(x) = 1$. Finally, suppose less than half of the x_i's occur as singleton terms in h. By our first assumption that all mixed terms have width two (in particular, no mixed term contains more than one x-variable), there must be an x_i that does not occur as a singleton term and participates in at most $2n$ mixed terms (since otherwise h would have more than n^2 terms); without loss of generality suppose x_1 is one such variable. Let $S \subseteq [5n]$ be the set of all $j \in [5n]$ such that $(x_1 \wedge y_j)$ is a mixed term in h, and consider the set of inputs

$$E = \{(x,y) \colon x_1 = 1,\ x_i = 0 \text{ for all } i \geq 2, \text{ and } y_j = 0 \text{ for all } j \in S, \text{ and } \|y\| = (3n)/2\}.$$

Note that $f(x,y) = 1$ for all $(x,y) \in E$, and we claim that $h(x,y) = 0$ on these inputs. To see this, consider the restriction h^* of h obtained by setting $x_1 \leftarrow 1$, $x_i \leftarrow 0$ for all $i \geq 2$, and $y_j \leftarrow 0$ for all $j \in S$. Since x_1 does not occur as a singleton term in h, this partial assignment does not satisfy any terms and the canonical DNF for h^* comprises only of pure-y terms. Since the pure-y terms of h have width greater than $2n$ (by our second assumption), the same is true for h^* and so h^* cannot be satisfied by any assignment of weight $(3n)/2$; hence $h(x,y) = h^*(y) = 0$ for all $(x,y) \in E$. Lastly, we check that $|E| \geq \binom{3n}{(3n)/2} = \Theta\big(2^{3n}/\sqrt{3n}\big) > \varepsilon \cdot 2^{6n}$ and this completes the proof. □

Remark 62. We note that the non-monotone approximator g is actually computed by a $O(n)$-size decision tree. Recall that every size-s decision tree is a size-s DNF, but not vice versa: there are polynomial-size DNFs that require exponential-size decision trees. Therefore the proof of Theorem 56 in fact establishes a stronger statement: f is a monotone function that can be ε-approximated by a $O(n)$-size decision tree, and yet any monotone function that ε-approximates f has DNF size $\Omega(n^2)$.

5.4.2 Separation for DNF width

We will need the following standard result, proved here for completeness, concerning shadows in the hypercube.

Lemma 5.4.1. *Let $k \in [n]$ and $\delta \in (0,1)$, and f be a monotone DNF of width δk. Then $\mu_{k-1}(f) \geq (1-\delta) \cdot \mu_k(f)$.*

Proof. Let $\mathcal{C}$ be the collection of all pairs (y, x) satisfying $\|y\| = k-1, \|x\| = k$, $y \prec x$, and $T(y) = T(x) = 1$ for some term T in f. We first note that every x such that $f(x) = 1$ and $\|x\| = k$ must satisfy some term T of width at most $\delta k < k$, and hence some term of length at most $k-1$, so

$$|\{x \in \{0,1\}^n : \text{ there exists some } y \in \{0,1\}^n \text{ such that } (y,x) \in \mathcal{C}\}| = \mu_k(f) \cdot \binom{n}{k}.$$

Consider any $x^* \in f^{-1}(1)$ where $\|x^*\| = k$, and let T be a term in f such that $T(x) = 1$. Since $|T| \leq \delta k$, there are at least $(1-\delta) \cdot k$ many y such that $(y, x^*) \in \mathcal{C}$. On the other hand, for any y^* where $\|y^*\| = k-1$, there are exactly $n-k+1$ many x such that $\|x\| = k$ and $x \succ y^*$. By double counting, we conclude that

$$\mu_{k-1}(f) \cdot \binom{n}{k-1} = |\{y \in \{0,1\}^n : \text{there exists some } x \in \{0,1\}^n \text{ such that } (y,x) \in \mathcal{C}\}|$$

$$\geq \frac{\mu_k(f) \cdot \binom{n}{k} \cdot (1-\delta) \cdot k}{n-k+1} = \mu_k(f) \cdot (1-\delta) \cdot \binom{n}{k-1}.$$

Equivalently, $\mu_{k-1}(f) \geq (1-\delta) \cdot \mu_k(f)$. □

Theorem 57. *Let $f : \{0,1\}^n \times \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}$ be the monotone function:*

$$f(x,y,z) = \begin{cases} \mathbf{1}[\|x\| \geq k \text{ and } y = 1^k] & \text{if } \|z\| = 0 \\ \mathbf{1}[\|x\| \geq k] & \text{otherwise,} \end{cases}$$

and $\varepsilon = \mathrm{Vol}(n, k-1) \cdot 2^{-(n+k+\ell)}$. There exists a DNF of width $k+\ell$ that ε-approximates f, but for all $k = o(n)$ any monotone function that ε-approximates f has width at least $(2 - 2^{-\ell}(1 + o_n(1))) \cdot k$. In particular, taking $\ell = \log k$ yields a gap of $k + \log k$ versus $2k - 1 - o_n(1)$.

Proof. Consider the function

$$g(x,y,z) = \begin{cases} \mathbf{1}[y = 1^k] & \text{if } \|z\| = 0 \\ \mathbf{1}[\|x\| \geq k] & \text{otherwise.} \end{cases}$$

This is a non-monotone function that is computed by a DNF of width $k+\ell$:

$$g(x,y,z) = \left(\bigwedge_{i \in [\ell]} \bar{z}_i \wedge \bigwedge_{j \in [k]} y_j \right) \vee \bigvee_{\substack{i \in [\ell] \\ S \subseteq [n] : |S| = k}} \left(z_i \wedge \bigwedge_{j \in S} x_j \right),$$

and we observe that g is indeed an ε-approximator for f since f and g differ on the $\mathrm{Vol}(n, k-1)$ inputs (x, y, z) where $z = \mathbf{0}$, $y = \mathbf{1}$, and $\|x\| \leq k-1$.

The rest of this proof will be devoted to showing that for all $k = o(n)$, any monotone DNF h that ε-approximates f has width at least $(2 - 2^{-\ell}(1 + o_n(1))) \cdot k$. Consider the monotone DNF h^* obtained by restricting $z_i \leftarrow 0$ for all $i \in [\ell]$ in h. We claim that *every* term in h^* has to contain *all* of $y_1, \ldots, y_k$. Suppose not, and suppose without loss of generality that there exists a term T in h^* that does not contain y_1. If $|T| \geq 2k$ the overall claimed lower bound on $\mathrm{width}(h)$ is true; otherwise h errs on at least

$$2^{n+(k-1)-|T|} \geq 2^{n+(k-1)-2k} = 2^{\Omega(n)} \gg \mathrm{Vol}(n, k-1)$$

many inputs (x, y, z) where $z = \mathbf{0}, y_1 = 0$ and $T(x, y, z) = 1$, since $h(x, y, z) = h^*(x, y, z) = 1$ and $f(x, y, z) = 0$ on these inputs.

Let $h^\dagger$ be h^* with $y_i \leftarrow 1$ for all $i \in [k]$. Since every term in h^* contains all of $y_1, \ldots, y_k$, it follows that

$$\mathrm{width}(h) \geq \mathrm{width}(h^*) \geq k + \mathrm{width}(h^\dagger), \tag{5.5}$$

and so it suffices to prove that $\mathrm{width}(h^\dagger) \geq (1 - 2^{-\ell}(1 + o_n(1))) \cdot k$. First, since $f(x, y, z) = 1$ on the $\binom{n}{k}$ inputs satisfying $z = \mathbf{0}, y = \mathbf{1}$, and $\|x\| = k$, we have that

$$(1 - \mu_k(h^\dagger))\binom{n}{k} \leq \mathrm{Vol}(n, k-1). \tag{5.6}$$

Next, since $h(x, \mathbf{1}, \mathbf{0}) \leq h(x, \mathbf{1}, z)$ for all $z \in \{0,1\}^\ell$ by monotonicity, every error h incurs on an input $(x, \mathbf{1}, \mathbf{0})$ where $\|x\| = k-1$ implies an error on $(x, \mathbf{1}, z)$ for every $z \in \{0,1\}^\ell$, and so

$$2^\ell \cdot \mu_{k-1}(h^\dagger)\binom{n}{k-1} \leq \mathrm{Vol}(n, k-1). \tag{5.7}$$

Using our assumption that $k = o(n)$, the bound of (5.6) implies that $\mu_k(h^\dagger) \geq 1 - o_n(1)$, and (5.7) that $\mu_{k-1}(h^\dagger) \leq 2^{-\ell}(1 + o_n(1))$. Applying Lemma 5.4.1 to $h^\dagger$ we conclude that $\mathrm{width}(h^\dagger) \geq (1 - 2^{-\ell}(1 + o_n(1))) \cdot k$, which along with (5.5) completes the proof. □

5.4.3 Upper bounds

Given the separations between monotone and non-monotone DNFs established in the previous subsections, it is natural to explore bounds in the other direction which show that the

existence of (non-monotone) DNF approximators implies the existence of monotone DNF approximators of related size, width, and accuracy. We first recall a few standard definitions and useful facts from the analysis of Boolean functions:

Definition 63 (influence)**.** The *total influence* of a Boolean function f, denoted $\mathbf{Inf}[f]$, is defined to be

$$\mathbf{Inf}[f] = \sum_{i=1}^{n} \mathbf{Inf}_i[f] \quad \text{where} \quad \mathbf{Inf}_i[f] = \mathop{\mathbf{Pr}}_{\boldsymbol{x}\in\{0,1\}^n}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})],$$

and $x^{\oplus i}$ denotes x with its i-th coordinate flipped.

Definition 64. A coordinate $i \in [n]$ is *relevant* in a Boolean function f if $\mathbf{Inf}_i[f] > 0$. For $k \in \{0, 1, \ldots, n\}$, we say that f is a *k-junta* if it has at most k relevant coordinates.

Friedgut's Junta Theorem [Friedgut, 1998]. *For every $\delta > 0$, every Boolean function f is δ-close to a $2^{O(\mathbf{Inf}[f]/\delta)}$-junta.*

Amano's Influence Bound [Amano, 2011]. *Let f be computed by a width-w DNF. Then $\mathbf{Inf}[f] \leq w$.*

Gopalan–Meka–Reingold Junta Bound [Gopalan *et al.*, 2013]. *Let f be computed by a width-w DNF. Then f is δ-close to a $(w \log(1/\delta))^{O(w)}$-junta.*

Folklore Junta Bound. *Let f be computed by a size-s DNF. Then f is δ-close to a $(s \log(s/\delta))$-junta.*

Perhaps the most common way to obtain a monotone function from a non-monotone one is via the combinatorial shifting operators introduced by Kleitman:

Definition 65 (combinatorial shifting)**.** For every $i \in [n]$, the i-th shifting operator κ_i acts on Boolean functions as follows:

$$(\kappa_i f)(x) = \begin{cases} f(x) & \text{if } f(x) = f(x^{\oplus i}) \\ x_i & \text{otherwise.} \end{cases}$$

It is straightforward to verify that $\text{shift}(f) := \kappa_1 \kappa_2 \cdots \kappa_n f$ is a monotone function. We will use additional basic facts concerning the shifting operators. The first is that they can only improve approximation with respect to a monotone function, and the second is that they do not increase the number of relevant coordinates.

Fact 5.4.2. *Let f be a monotone function and suppose f is ε-close to g. Then for all coordinates $i \in [n]$*

$$\Pr_{\boldsymbol{x}\in\{0,1\}^n}[f(\boldsymbol{x}) \neq \mathrm{shift}(g)(\boldsymbol{x})] \leq \Pr_{\boldsymbol{x}\in\{0,1\}^n}[f(\boldsymbol{x}) \neq (\kappa_i g)(\boldsymbol{x})] \leq \varepsilon.$$

Fact 5.4.3. *For every Boolean function f and coordinate $i \in [n]$, the number of relevant coordinates in $\kappa_i f$ is at most that in f. Consequently, the number of relevant coordinates in* $\mathrm{shift}(f)$ *is at most that in f.*

With these facts in hand we are now ready to prove our upper bounds showing that the existence of (non-monotone) DNF approximators implies the existence of monotone DNF approximators of related size, width, and accuracy.

Theorem 66. *Let f be a monotone function and suppose f is ε-approximated by a width-w DNF. For every $\delta > 0$ there is a monotone DNF of width* $\min\{2^{O(w/\delta)}, (w\log(1/\delta))^{O(w)}\}$ *that $(\varepsilon+\delta)$-approximates f.*

Proof. Let f^* be the width-w DNF that ε-approximates f. Combining Amano's influence bound and Friedgut's junta theorem, we know that f^* is δ-close to a $2^{O(w/\delta)}$-junta g. Next, by Facts 5.4.2 and 5.4.3, along with the triangle inequality, we get that $\mathrm{shift}(g)$ is a monotone $2^{O(w/\delta)}$-junta that $(\varepsilon+\delta)$-approximates f. This yields the first bound of $2^{O(w/\delta)}$ since every monotone k-junta is trivially computed by a monotone DNF of width at most k. A similar argument, using the Gopalan–Meka–Reingold junta bound in place of Amano's influence bound and Friedgut's junta theorem, yields the incomparable second bound of $(w\log(1/\delta))^{O(w)}$. □

A similar argument using the folklore junta bound in place of Amano's influence bound and Friedgut's junta theorem establishes an analogous result for DNF size:

Theorem 67. *Let f be a monotone function and suppose f is ε-approximated by a size-s DNF. For every $\delta > 0$ there is a monotone DNF of size $2^{s\log(s/\delta)}$ that $(\varepsilon+\delta)$-approximates f.*

5.5 Conclusion

Having obtained near-matching upper and lower bounds on the size of universal lower approximators in this work, the natural next step is to consider *upper* approximators and

approximators incurring error on both sides. The task of constructing universal upper approximators appears to be qualitatively different from that of lower approximators, and we are not aware of any construction achieving size better than the trivial one of $O(2^n/\sqrt{n})$ sufficient for exact computation. For approximators incurring two-sided error, our universal lower approximators of size $2^{n-\Omega_\varepsilon(\sqrt{n})}$ represent the current best upper bound. The strongest known lower bound for two-sided approximators is the $2^{\Omega(n/\log n)}$ lower bound of [O'Donnell and Wimmer, 2007]; it would be interesting to find out whether this or the current $2^{n-\Omega_\varepsilon(\sqrt{n})}$ upper bound is closer to the truth.

As for the power of negations in the approximation of monotone functions, we believe that our results in Section 5.4 suggest a number of interesting avenues for further exploration. We suspect that the separations we presented in Sections 5.4.1 and 5.4.2 can be improved, perhaps even to super-polynomial for DNF size and super-constant for DNF width, and likewise our upper bounds in Section 5.4.3. We remark that in addition to the complexity measures of DNF size and width, the quantitative difference between the accuracy of monotone versus general DNFs is also an aspect in which our separations can be strengthened. In other words, we may view our separations as instantiations of the following general template:

> *There exists a monotone function f and a value $\varepsilon = \varepsilon(n) > 0$ such that f can be ε-approximated by a DNF of size s (resp. width w), but any monotone function that $\varphi(\varepsilon)$-approximates f requires DNF size $\Psi(s)$ (resp. width $\Psi(w)$).*

In Theorems 56 and 57, φ is simply the identity function, but one can consider the possibility of stronger statements where $\varphi(\varepsilon) \gg \varepsilon$.

Chapter 6

A Composition Theorem for the Fourier Entropy-Influence Conjecture

6.1 Background and context

A longstanding and important open problem in the field of Analysis of Boolean Functions is the Fourier Entropy-Influence conjecture made by Ehud Friedgut and Gil Kalai in 1996 [Friedgut and Kalai, 1996; Kalai, 2007]. The conjecture seeks to relate two fundamental analytic measures of Boolean function complexity, the spectral entropy and total influence:

Fourier Entropy-Influence (FEI) Conjecture. *There exists a universal constant $C > 0$ such that for every Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, it holds that $\mathbf{H}[f] \leq C \cdot \mathbf{Inf}[f]$. That is,*

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 \log_2 \left(\frac{1}{\widehat{f}(S)^2} \right) \leq C \sum_{S \subseteq [n]} |S| \cdot \widehat{f}(S)^2.$$

Applying Parseval's identity to a Boolean function f we get $\sum_{S \subseteq [n]} \widehat{f}(S)^2 = \mathbf{E}[f(\boldsymbol{x})^2] = 1$, and so the Fourier coefficients of f induce a probability distribution $\mathcal{S}_f$ over the 2^n subsets of $[n]$ wherein $S \subseteq [n]$ has "weight" (probability mass) $\widehat{f}(S)^2$. The *spectral entropy* of f,

denoted $\mathbf{H}[f]$, is the Shannon entropy of $\mathcal{S}_f$, quantifying how spread out the Fourier weight of f is across all 2^n monomials. The influence of a coordinate $i \in [n]$ on f is $\mathbf{Inf}_i[f] = \mathbf{Pr}[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{\oplus i})]$,[1] where $\boldsymbol{x}^{\oplus i}$ denotes $\boldsymbol{x}$ with its i-th bit flipped, and the *total influence* of f is simply $\mathbf{Inf}[f] = \sum_{i=1}^{n} \mathbf{Inf}_i[f]$. Straightforward Fourier-analytic calculations show that this combinatorial definition is equivalent to the quantity $\mathbf{E}_{\mathbf{S}\sim\mathcal{S}_f}[|\mathbf{S}|] = \sum_{S\subseteq[n]} |S| \cdot \widehat{f}(S)^2$, and so total influence measures the degree distribution of the monomials of f, weighted by the squared-magnitude of its coefficients. Roughly speaking then, the FEI conjecture states that a Boolean function whose Fourier weight is well "spread out" (*i.e.* has high spectral entropy) must have a significant portion of its Fourier weight lying on high degree monomials (*i.e.* have high total influence).[2]

In addition to being a natural question concerning the Fourier spectrum of Boolean functions, the FEI conjecture also has important connections to several areas of theoretical computer science and mathematics. Friedgut and Kalai's original motivation was to understand general conditions under which monotone graph properties exhibit sharp thresholds, and the FEI conjecture captures the intuition that having significant symmetry, hence high spectral entropy, is one such condition. Besides its applications in the study of random graphs, the FEI conjecture is known to imply the celebrated Kahn-Kalai-Linial theorem [Kahn *et al.*, 1988]:

KKL Theorem. *For every Boolean function f there exists an $i \in [n]$ such that* $\mathbf{Inf}_i[f] = \mathbf{Var}[f] \cdot \Omega(\frac{\log n}{n})$.

The FEI conjecture also implies Mansour's conjecture [Mansour, 1994]:

Mansour's Conjecture. *Let f be a Boolean function computed by a t-term DNF formula. For any constant $\varepsilon > 0$ there exists a collection $\mathcal{S} \subseteq 2^{[n]}$ of cardinality* $\mathrm{poly}(t)$ *such that* $\sum_{S\in\mathcal{S}} \widehat{f}(S)^2 \geq 1 - \varepsilon$.

Combined with recent work of Gopalan *et al.* [Gopalan *et al.*, 2008a], Mansour's con-

[1] All probabilities and expectations are with respect to the uniform distribution unless otherwise stated.

[2] The assumption that f is Boolean-valued is crucial here, as the same conjecture is false for functions $f : \{-1,1\}^n \to \mathbb{R}$ satisfying $\sum_{S\subseteq[n]} \widehat{f}(S)^2 = 1$. The canonical counterexample is $f(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_i$ which has total influence 1 and spectral entropy $\log_2 n$.

jecture yields an efficient algorithm for agnostically learning the class of poly(n)-term DNF formulas from queries. This would resolve a central open problem in computational learning theory [Gopalan *et al.*, 2008b]. De *et al.* also noted that sufficiently strong versions of Mansour's conjecture would yield improved pseudorandom generators for depth-2 AC^0 circuits [De *et al.*, 2010]. More generally, the FEI conjecture implies the existence of sparse L_2-approximators for Boolean functions with small total influence:

Sparse L_2-approximators. Assume the FEI conjecture holds. Then for every Boolean function f there exists a $2^{O(\mathbf{Inf}[f]/\varepsilon)}$-sparse polynomial $p : \mathbb{R}^n \to \mathbb{R}$ such that $\mathbf{E}[(f(\boldsymbol{x}) - p(\boldsymbol{x}))^2] \leq \varepsilon$.

By Friedgut's junta theorem [Friedgut, 1998], the above holds unconditionally with a weaker bound of $2^{O(\mathbf{Inf}[f]^2/\varepsilon^2)}$. This is the main technical ingredient underlying several of the best known uniform-distribution learning algorithms [Servedio, 2004a; O'Donnell and Servedio, 2008].

For more on the FEI conjecture we refer the reader to Kalai's blog post [Kalai, 2007].

6.1.1 Our results

Our research is motivated by the following question:

Question 2. *Let $F : \{-1,1\}^k \to \{-1,1\}$ and $g_1, \ldots, g_k : \{-1,1\}^\ell \to \{-1,1\}$. What properties do F and $g_1, \ldots, g_k$ have to satisfy for the FEI conjecture to hold for the disjoint composition $f(x^1, \ldots, x^k) = F(g_1(x^1), \ldots, g_k(x^k))$?*

Despite its simplicity this question has not been well understood. For example, prior to our work the FEI conjecture was open even for read-once DNFs; these are the disjoint compositions of $F = \mathsf{OR}$ and $g_1, \ldots, g_k = \mathsf{AND}$, perhaps two of the most basic Boolean functions with extremely simple Fourier spectra. Indeed, Mansour's conjecture, a weaker conjecture than FEI, was only recently shown to hold for read-once DNFs [Klivans *et al.*, 2010; De *et al.*, 2010]. Besides being a fundamental question concerning the behavior of spectral entropy and total influence under composition, Question 2 (and our answer to it) also has implications for a natural approach towards disproving the FEI conjecture; we elaborate on this at the end of this section.

A particularly appealing and general answer to Question 2 that one may hope for would be the following: "for all $C > 0$, if $\mathbf{H}[F] \leq C \cdot \mathbf{Inf}[F]$ and $\mathbf{H}[g_i] \leq C \cdot \mathbf{Inf}[g_i]$ for all $i \in [k]$, then $\mathbf{H}[f] \leq C \cdot \mathbf{Inf}[f]$." While this is easily seen to be false[3], our main result shows that this proposed answer to Question 2 *is* in fact true for a carefully chosen sharpening of the FEI conjecture. To arrive at a formulation that bootstraps itself, we first consider a slight strengthening of the FEI conjecture which we call FEI$^+$, and then work with a generalization of FEI$^+$ that concerns the Fourier spectrum of f not just with respect to the uniform distribution, but an arbitrary product distribution over $\{-1,1\}^n$:

Conjecture 3 (FEI$^+$ for product distributions)**.** *There is a universal constant $C > 0$ such that the following holds. Let $\mu = \langle \mu_1, \ldots, \mu_n \rangle$ be any sequence of biases and $f : \{-1,1\}^n_\mu \to \{-1,1\}$. Here the notation $\{-1,1\}^n_\mu$ means that we think of $\{-1,1\}^n$ as being endowed with the μ-biased product probability distribution in which $\mathbf{E}_\mu[x_i] = \mu_i$ for all $i \in [n]$. Let $\{\widetilde{f}(S)\}_{S \subseteq [n]}$ be the μ-biased Fourier coefficients of f. Then*

$$\sum_{S \neq \emptyset} \widetilde{f}(S)^2 \log \left(\frac{\prod_{i \in S}(1-\mu_i^2)}{\widetilde{f}(S)^2} \right) \leq C \cdot (\mathbf{Inf}^\mu[f] - \mathop{\mathbf{Var}}_{\mu}[f]).$$

We write $\mathbf{H}^\mu[f]$ to denote the quantity $\sum_{S \subseteq [n]} \widetilde{f}(S)^2 \log \left(\prod_{i \in S}(1-\mu_i^2)/\widetilde{f}(S)^2 \right)$, and so the inequality of Conjecture 3 can be equivalently stated as $\mathbf{H}^\mu[f^{\geq 1}] \leq C \cdot (\mathbf{Inf}^\mu[f] - \mathbf{Var}_\mu[f])$.

In Proposition 6.1.1 we show that Conjecture 3 with $\mu = \langle 0, \ldots, 0 \rangle$ (the uniform distribution) implies the FEI conjecture. We say that a Boolean function f "satisfies μ-biased FEI$^+$ with factor C" if the μ-biased Fourier transform of f satisfies the inequality of Conjecture 3. Our main result, which we prove in Section 6.2, is a composition theorem for FEI$^+$:

Theorem 68. *Let $f(x^1, \ldots, x^k) = F(g_1(x^1), \ldots, g_k(x^k))$, where the domain of f is endowed with a product distribution μ. Suppose $g_1, \ldots, g_k$ satisfy μ-biased FEI$^+$ with factor C_1 and F satisfies η-biased FEI$^+$ with factor C_2, where $\eta = \langle \mathbf{E}_\mu[g_1], \ldots, \mathbf{E}_\mu[g_k] \rangle$. Then f satisfies μ-biased FEI$^+$ with factor $\max\{C_1, C_2\}$.*

Theorem 68 suggests an inductive approach towards proving the FEI conjecture for read-once de Morgan formulas: since the dictators $\pm x_i$ trivially satisfy uniform-distribution

[3]For example, by considering $F = \mathsf{OR}_2$, the 2-bit disjunction, and $g_1, g_2 = \mathsf{AND}_2$, the 2-bit conjunction.

FEI^+ with factor 1, it suffices to prove that both AND_2 and OR_2 satisfy μ-biased FEI^+ with some constant *independent of* $\mu \in [-1,1]^2$. In Section 6.3 we prove that in fact *every* $F : \{-1,1\}^k \to \{-1,1\}$ satisfies μ-biased FEI^+ with a factor depending only on its arity k and not the biases $\mu_1, \ldots, \mu_k$.

Theorem 69. *Every $F : \{-1,1\}^k \to \{-1,1\}$ satisfies μ-biased FEI^+ with factor $C = 2^{O(k)}$ for any product distribution $\mu = \langle \mu_1, \ldots, \mu_k \rangle$.*

Together, Theorems 68 and 69 imply:

Theorem 70. *Let f be computed by a read-once formula over the basis $\mathcal{B}$ and μ be any sequences of biases. Then f satisfies μ-biased FEI^+ with factor C, where C depends only on the arity of the gates in $\mathcal{B}$.*

Since uniform-distribution FEI^+ is a strengthening of the FEI conjecture, Theorem 70 implies that the FEI conjecture holds for read-once formulas over arbitrary gates of bounded arity. As mentioned above, prior to our work the FEI conjecture was open even for the class of read-once DNFs, a small subclass of read-once formulas over the de Morgan basis $\{\mathsf{AND}_2, \mathsf{OR}_2, \mathsf{NOT}\}$ of arity 2. Read-once formulas over a rich basis $\mathcal{B}$ are a natural generalization of read-once de Morgan formulas, and have seen previous study in concrete complexity (see *e.g.* [Heiman *et al.*, 1993]).

Improved lower bound on the FEI constant. Iterated disjoint composition is commonly used to achieve separations between complexity measures for Boolean functions, and represents a natural approach towards disproving the FEI conjecture. For example, one may seek a function F such that iterated compositions of F with itself achieves a super-constant amplification of the ratio between $\mathbf{H}[F]$ and $\mathbf{Inf}[F]$, or consider variants such as iterating F with a different combining function G. Theorem 70 rules out as potential counterexamples all such constructions based on iterated composition.

However, the tools we develop to prove Theorem 70 also yield an explicit function f achieving the best-known separation between $\mathbf{H}[f]$ and $\mathbf{Inf}[f]$ (*i.e.* the constant C in the statement of the FEI conjecture). In Section 6.4 we prove:

Theorem 71. *There exists an explicit family of functions* $f_n : \{-1,1\}^n \to \{-1,1\}$ *such that*

$$\lim_{n\to\infty} \frac{\mathbf{H}[f_n]}{\mathbf{Inf}[f_n]} \geq 6.278.$$

This improves on the previous lower bound of $C \geq 60/13 \approx 4.615$ [O'Donnell *et al.*, 2011].

Previous work. The first published progress on the FEI conjecture was by Klivans *et al.* who proved the conjecture for random poly(n)-term DNF formulas [Klivans *et al.*, 2010]. This was followed by the work of O'Donnell *et al.* who proved the conjecture for the class of symmetric functions and read-once decision trees [O'Donnell *et al.*, 2011].

The FEI conjecture for product distributions was studied in the recent work of Keller *et al.* [Keller *et al.*, 2012], where they consider the case of all the biases being the same. They introduce the following generalization of the FEI conjecture to these measures, and show via a reduction to the uniform distribution that it is equivalent to the FEI conjecture:

Conjecture 4 (Keller-Mossel-Schlank)**.** *There is a universal constant* C *such that the following holds. Let* $0 < p < 1$ *and* $f : \{-1,1\}^n \to \{-1,1\}$*, where the domain of* f *is endowed with the product distribution where* $\mathbf{Pr}[x_i = -1] = p$ *for all* $i \in [n]$*. Let* $\{\widetilde{f}(S)\}_{S\subseteq[n]}$ *be the Fourier coefficients of* f *with respect to this distribution. Then*

$$\sum_{S\subseteq[n]} \widetilde{f}(S)^2 \log_2\left(\frac{1}{\widetilde{f}(S)^2}\right) \leq C \cdot \frac{\log(1/p)}{1-p} \sum_{S\subseteq[n]} |S| \cdot \widetilde{f}(S)^2.$$

Notice that in this conjecture, the constant on the right-hand side, $C \cdot \frac{\log(1/p)}{1-p}$, depends on p. By way of contrast, in our Conjecture 3 the right-hand side constant has no dependence on p; instead, the dependence on the biases is built into the definition of spectral entropy. We view our generalization of the FEI conjecture to arbitrary product distributions (where the biases are not necessarily identical) as a key contribution of this work, and point to our composition theorem as evidence in favor of Conjecture 3 being a good statement to work with.

6.1.2 Biased Fourier analysis

We will be concerned with functions $f : \{-1,1\}^n_\mu \to \mathbb{R}$ where $\mu = \langle \mu_1, \ldots, \mu_n \rangle \in [0,1]^n$ is a sequence of biases. Here the notation $\{-1,1\}^n_\mu$ means that we think of $\{-1,1\}^n$ as

being endowed with the μ-biased product probability distribution in which $\mathbf{E}_\mu[x_i] = \mu_i$ for all $i \in [n]$. We write σ_i^2 to denote variance of the i-th coordinate $\mathbf{Var}_\mu[x_i] = 1 - \mu_i^2$, and $\varphi : \mathbb{R} \to \mathbb{R}$ as shorthand for the function $t \mapsto t^2 \log_2(1/t^2)$, adopting the convention that $\varphi(0) = 0$. All logarithms in this chapter are in base 2 unless otherwise stated.

In this section we briefly review the basics of Fourier analysis with respect to product distributions over $\{-1,1\}^n$.

Theorem 72 (Fourier expansion)**.** *Let $\mu = \langle \mu_1, \ldots, \mu_n \rangle$ be a sequence of biases. The μ-biased Fourier expansion of $f : \{-1,1\}^n \to \mathbb{R}$ is*

$$f(x) = \sum_{S \subseteq [n]} \widetilde{f}(S) \phi_S^\mu(x),$$

where

$$\phi_S^\mu(x) = \prod_{i \in S} \frac{x_i - \mu_i}{\sigma_i} \quad \text{and} \quad \widetilde{f}(S) = \mathop{\mathbf{E}}_{\mu}[f(\boldsymbol{x}) \phi_S^\mu(\boldsymbol{x})],$$

and $\sigma_i^2 = \mathbf{Var}_\mu[x_i] = 1 - \mu_i^2$.

The μ-biased spectral support of f is the collection $\mathcal{S} \subseteq 2^{[n]}$ of subsets $S \subseteq [n]$ such that $\widetilde{f}(S) \neq 0$. We write $f^{\geq k}$ to denote $\sum_{|S| \geq k} \widetilde{f}(S) \phi_S^\mu(x)$, the projection of f onto its monomials of degree at least k.

Theorem 73 (Parseval's identity)**.** *Let $f : \{-1,1\}_\mu^n \to \mathbb{R}$. Then $\sum_{S \subseteq [n]} \widetilde{f}(S)^2 = \mathbf{E}_\mu[f(\boldsymbol{x})^2]$. In particular, if the range of f is $\{-1,1\}$ then $\sum_{S \subseteq [n]} \widetilde{f}(S)^2 = 1$.*

Definition 74 (Influence)**.** Let $f : \{-1,1\}_\mu^n \to \mathbb{R}$. The influence of variable $i \in [n]$ on f is $\mathbf{Inf}_i^\mu[f] = \mathbf{E}_\rho[\mathbf{Var}_{\mu_i}[f_\rho]]$, where ρ is a μ-biased random restriction to the coordinates in $[n] \backslash \{i\}$. The total influence of f, denoted $\mathbf{Inf}^\mu[f]$, is $\sum_{i=1}^n \mathbf{Inf}_i^\mu[f]$.

We recall a few basic Fourier formulas. The expectation of f is given by $\mathbf{E}_\mu[f] = \widetilde{f}(\emptyset)$ and its variance $\mathbf{Var}_\mu[f] = \sum_{S \neq \emptyset} \widetilde{f}(S)^2$. For each $i \in [n]$, $\mathbf{Inf}_i^\mu[f] = \sum_{S \ni i} \widetilde{f}(S)^2$ and so $\mathbf{Inf}^\mu[f] = \sum_{S \subseteq [n]} |S| \cdot \widetilde{f}(S)^2$. We omit the sub- and superscripts when $\mu = \langle 0, \ldots, 0 \rangle$ is the uniform distribution. Comparing the Fourier formulas for variance and total influence yields the Poincaré inequality for functions $f : \{-1,1\}_\mu^n \to \mathbb{R}$:

Theorem 75 (Poincaré inequality)**.** *Let $f : \{-1,1\}_\mu^n \to \mathbb{R}$. Then $\mathbf{Var}_\mu[f] \leq \mathbf{Inf}^\mu[f]$.*

Recall that the i-th discrete derivative operator for $f : \{-1,1\}^n \to \{-1,1\}$ is defined to be

$$D_{x_i}(x) = \tfrac{1}{2}\left(f(x^{i\leftarrow 1}) - f(x^{i\leftarrow -1})\right),$$

and for $S \subseteq [n]$ we write $D_{x^S} f$ to denote $\circ_{i\in S} D_{x_i} f$.

Definition 76 (Discrete derivative)**.** The i-th discrete derivative operator $D_{\phi_i^\mu}$ with respect to the μ-biased product distribution on $\{-1,1\}^n$ is defined by $D_{\phi_i^\mu} f(x) = \sigma_i D_{x_i} f(x)$.

With respect to the μ-biased Fourier expansion of $f : \{-1,1\}_\mu^n \to \mathbb{R}$ the operator $D_{\phi_i^\mu}$ satisfies

$$D_{\phi_i^\mu} f = \sum_{S \ni i} \widetilde{f}(S)\phi^\mu_{S\setminus\{i\}},$$

and so for any $S \subseteq [n]$ we have $\widetilde{f}(S) = \mathbf{E}[\circ_{i\in S} D_{\phi_i^\mu} f] = \prod_{i\in S} \sigma_i \, \mathbf{E}_\mu[D_{x^S} f]$.

6.1.3 Uniform-distribution FEI$^+$ implies FEI

Proposition 6.1.1. *Suppose f satisfies uniform-distribution FEI$^+$ with factor C. Then f satisfies the FEI conjecture with factor* $\max\{C, 1/\ln 2\}$.

Proof. Let $\widehat{f}(\emptyset)^2 = 1-\varepsilon$, where $\varepsilon = \mathbf{Var}[f]$ by Parseval's identity. By our assumption that f satisfies uniform-distribution FEI$^+$ with factor C, we have

$$\begin{aligned}
\sum_{S\subseteq[n]} \widehat{f}(S)^2 \log\left(\frac{1}{\widehat{f}(S)^2}\right) &\le C\cdot(\mathbf{Inf}[f] - \mathbf{Var}[f]) + (1-\varepsilon)\log\frac{1}{(1-\varepsilon)} \\
&\le C\cdot(\mathbf{Inf}[f] - \mathbf{Var}[f]) + \frac{\varepsilon}{\ln 2} \\
&= C\cdot\mathbf{Inf}[f] + \left(\frac{1}{\ln 2} - C\right)\cdot\mathbf{Var}[f].
\end{aligned}$$

If $C > 1/\ln 2$ then the RHS is at most $C\cdot\mathbf{Inf}[f]$ since $(\frac{1}{\ln 2} - C)\cdot\mathbf{Var}[f]$ is negative. Otherwise we apply the Poincaré inequality $\mathbf{Inf}[f] \ge \mathbf{Var}[f]$ to conclude that the RHS is at most $C\cdot\mathbf{Inf}[f] + (\frac{1}{\ln 2} - C)\cdot\mathbf{Inf}[f] = \frac{1}{\ln 2}\cdot\mathbf{Inf}[f]$. □

6.2 A composition theorem for FEI$^+$

We will be concerned with compositions of functions $f = F(g_1(x^1), \ldots, g_k(x^k))$ where $g_1, \ldots, g_k$ are over disjoint sets of variables each of size ℓ. The domain of each g_i is endowed

with a product distribution $\mu^i = \langle \mu^i_1, \ldots, \mu^i_\ell \rangle$, which induces an overall product distribution $\mu = \langle \mu^1_1, \ldots, \mu^1_\ell, \ldots, \mu^k_1, \ldots, \mu^k_\ell \rangle$ over the domain of $f : \{-1,1\}^{k\ell} \to \{-1,1\}$. For notational clarity we will adopt the equivalent view of $g_1, \ldots, g_k$ as functions over the same domain $\{-1,1\}^{k\ell}_\mu$ endowed with the same product distribution μ, with each g_i depending only on ℓ out of $k\ell$ variables.

Our first lemma gives formulas for the spectral entropy and total influence of the product of functions $\Phi_1, \ldots, \Phi_k$ over disjoint sets of variables. The lemma holds for real-valued functions Φ_i; we require this level of generality as we will not be applying the lemma directly to the Boolean-valued functions $g_1, \ldots, g_k$ in the composition $F(g_1(x^1), \ldots, g_k(x^k))$, but instead to their normalized variants $\Phi(g_i) = (g_i - \mathbf{E}[g_i]) / \mathbf{Var}[g_i]^{1/2}$.

Lemma 6.2.1. *Let $\Phi_1, \ldots, \Phi_k : \{-1,1\}^{k\ell}_\mu \to \mathbb{R}$ where each Φ_i depends only on the ℓ coordinates in $\{(i-1)\ell + 1, \ldots, i\ell\}$. Then*

$$\mathbf{H}^\mu[\Phi_1 \cdots \Phi_k] = \sum_{i=1}^k \mathbf{H}^\mu[\Phi_i] \prod_{j \neq i} \mathop{\mathbf{E}}_\mu[\Phi_j^2] \text{ and } \mathbf{Inf}^\mu[\Phi_1 \cdots \Phi_k] = \sum_{i=1}^k \mathbf{Inf}^\mu[\Phi_i] \prod_{j \neq i} \mathop{\mathbf{E}}_\mu[\Phi_j^2].$$

Proof. We prove both formulas by induction on k, noting that the bases cases are trivially true. For the inductive step, we define $h(x) = \prod_{i \in [k-1]} \Phi_i(x)$ and see that

$$\begin{aligned}
\mathbf{H}^\mu[h \cdot \Phi_k] &= \sum_{\substack{S \subseteq [(k-1)\ell] \\ T \subseteq \{(k-1)\ell+1, \ldots k\ell\}}} \widetilde{h}(S)^2 \widetilde{\Phi_k}(T)^2 \log \left(\frac{\prod_{i \in S \cup T} \sigma_i^2}{\widetilde{h}(S)^2 \widetilde{\Phi_k}(T)^2} \right) \\
&= \sum_{S,T} \widetilde{h}(S)^2 \widetilde{\Phi_k}(T)^2 \left[\log \left(\frac{\prod_{i \in S} \sigma_i^2}{\widetilde{h}(S)^2} \right) + \log \left(\frac{\prod_{i \in T} \sigma_i^2}{\widetilde{\Phi_k}(T)^2} \right) \right] \\
&= \mathop{\mathbf{E}}_\mu[h^2] \cdot \mathbf{H}^\mu[\Phi_k] + \mathop{\mathbf{E}}_\mu[\Phi_k^2] \cdot \mathbf{H}^\mu[h] \\
&= \prod_{i \in [k-1]} \mathop{\mathbf{E}}_\mu[\Phi_i^2] \cdot \mathbf{H}^\mu[\Phi_k] + \mathop{\mathbf{E}}_\mu[\Phi_k^2] \left(\sum_{i=1}^{k-1} \mathbf{H}^\mu[\Phi_i] \prod_{j \neq i} \mathop{\mathbf{E}}_\mu[\Phi_j^2] \right) \\
&= \sum_{i=1}^k \mathbf{H}^\mu[\Phi_i] \prod_{j \neq i} \mathop{\mathbf{E}}_\mu[\Phi_j^2].
\end{aligned}$$

Here in the first equality we use the fact that if $f : \{-1,1\}^n_\mu \to \mathbb{R}$ does not depend on coordinate $i \in [n]$ then $\widetilde{f}(S) = 0$ for all $S \ni i$ (*i.e.* the Fourier spectrum of f is supported on sets containing only its relevant variables). The third equality is by Parseval's, and the fourth by the induction hypothesis applied to h.

The formula for influence follows from a similar derivation:

$$\begin{aligned}
\mathbf{Inf}^{\mu}[h \cdot \Phi_k] &= \sum_{\substack{S \subseteq [(k-1)\ell] \\ T \subseteq \{(k-1)\ell+1, \ldots k\ell\}}} |S \cup T| \cdot \widetilde{h}(S)^2 \widetilde{\Phi_k}(T)^2 \\
&= \sum_{S,T} |T| \cdot \widetilde{h}(S)^2 \widetilde{\Phi_k}(T)^2 + \sum_{S,T} |S| \cdot \widetilde{h}(S)^2 \widetilde{\Phi_k}(T)^2 \\
&= \mathop{\mathbf{E}}_{\mu}[h^2] \cdot \mathbf{Inf}^{\mu}[\Phi_k] + \mathop{\mathbf{E}}_{\mu}[\Phi_k^2] \cdot \mathbf{Inf}^{\mu}[h] \\
&= \prod_{i \in [k-1]} \mathop{\mathbf{E}}_{\mu}[\Phi_i^2] \cdot \mathbf{Inf}^{\mu}[\Phi_k] + \mathop{\mathbf{E}}_{\mu}[\Phi_k^2] \left(\sum_{i=1}^{k-1} \mathbf{Inf}^{\mu}[\Phi_i] \prod_{j \neq i} \mathop{\mathbf{E}}_{\mu}[\Phi_j^2] \right) \\
&= \sum_{i=1}^{k} \mathbf{Inf}^{\mu}[\Phi_i] \prod_{j \neq i} \mathop{\mathbf{E}}_{\mu}[\Phi_j^2],
\end{aligned}$$

and this completes the proof. □

We note that this lemma recovers as a special case the folklore observation that the FEI conjecture "tensorizes": for any f if we define $f^{\oplus k}(x^1, \ldots, x^k) = f(x^1) \cdots f(x^k)$ then $\mathbf{H}[f^{\oplus k}] = k \cdot \mathbf{H}[f]$ and $\mathbf{Inf}[f^{\oplus k}] = k \cdot \mathbf{Inf}[f]$. Therefore $\mathbf{H}[f] \leq C \cdot \mathbf{Inf}[f]$ if and only if $\mathbf{H}[f^{\oplus k}] \leq C \cdot \mathbf{Inf}[f^{\oplus k}]$.

Our next proposition relates the basic analytic measures – spectral entropy, total influence, and variance – of a composition $f = F(g_1(x^1), \ldots, g_k(x^k))$ to the corresponding quantities of the combining function F and base functions $g_1, \ldots, g_k$. As alluded to above, we accomplish this by considering f as a linear combination of the normalized functions $\Phi(g_i) = (g_i - \mathbf{E}[g_i])/\mathbf{Var}[g_i]^{1/2}$ and applying Lemma 6.2.1 to each term in the sum. We mention that this proposition is also the crux of our new lower bound of $C \geq 6.278$ on the constant of the FEI conjecture, which we present in Section 6.4.

Proposition 6.2.2. *Let $F : \{-1,1\}^k \to \mathbb{R}$, and $g_1, \ldots, g_k : \{-1,1\}_{\mu}^{k\ell} \to \{-1,1\}$ where each g_i depends only on the ℓ coordinates in $\{(i-1)\ell+1, \ldots, i\ell\}$. Let $f(x) = F(g_1(x), \ldots, g_k(x))$ and $\{\widetilde{F}(S)\}_{S \subseteq [k]}$ be the η-biased Fourier coefficients of F where $\eta = \langle \mathbf{E}_{\mu}[g_1], \ldots, \mathbf{E}_{\mu}[g_k] \rangle$.*

Then

$$\mathbf{H}^{\mu}[f^{\geq 1}] = \mathbf{H}^{\eta}[F^{\geq 1}] + \sum_{S\neq\emptyset} \widetilde{F}(S)^2 \sum_{i\in S} \frac{\mathbf{H}^{\mu}[g_i^{\geq 1}]}{\mathbf{Var}_{\mu}[g_i]}, \tag{6.1}$$

$$\mathbf{Inf}^{\mu}[f] = \sum_{S\neq\emptyset} \widetilde{F}(S)^2 \sum_{i\in S} \frac{\mathbf{Inf}^{\mu}[g_i]}{\mathbf{Var}_{\mu}[g_i]}, \quad \textit{and} \tag{6.2}$$

$$\mathop{\mathbf{Var}}_{\mu}[f] = \sum_{S\neq\emptyset} \widetilde{F}(S)^2 = \mathop{\mathbf{Var}}_{\eta}[F]. \tag{6.3}$$

Proof. By the η-biased Fourier expansion of $F : \{-1,1\}^k_{\eta} \to \mathbb{R}$ and the definition of η we have

$$F(y_1,\ldots,y_k) = \sum_{S\subseteq[k]} \widetilde{F}(S) \prod_{i\in S} \frac{y_i-\eta_i}{\sqrt{1-\eta_i^2}} = \sum_{S\subseteq[k]} \widetilde{F}(S) \prod_{i\in S} \frac{y_i - \mathbf{E}_{\mu}[g_i]}{\mathbf{Var}_{\mu}[g_i]^{1/2}},$$

and so $F(g_1(x),\ldots,g_k(x)) = \sum_{S\subseteq[k]} \widetilde{F}(S) \prod_{i\in S} \Phi(g_i(x))$, where

$$\Phi(g_i(x)) = (g_i(x) - \mathop{\mathbf{E}}_{\mu}[g_i])/\,\mathbf{Var}_{\mu}[g_i]^{1/2}.$$

Note that Φ normalizes g_i such that $\mathbf{E}_{\mu}[\Phi(g_i)] = 0$ and $\mathbf{E}_{\mu}[\Phi(g_i)^2] = 1$. First we claim that

$$\mathbf{H}^{\mu}[f^{\geq 1}] = \mathbf{H}^{\mu}\Big[\sum_{S\neq\emptyset} \widetilde{F}(S) \prod_{i\in S} \Phi(g_i)\Big] = \sum_{S\neq\emptyset} \mathbf{H}^{\mu}\Big[\widetilde{F}(S) \prod_{i\in S} \Phi(g_i)\Big].$$

It suffices to show that for any two distinct non-empty sets $S, T \subseteq [k]$, no monomial ϕ^{μ}_U occurs in the μ-biased spectral support of both $\widetilde{F}(S)\prod_{i\in S}\Phi(g_i)$ and $\widetilde{F}(T)\prod_{i\in T}\Phi(g_i)$. To see this recall that $\Phi(g_i)$ is balanced with respect to μ (*i.e.* $\mathbf{E}_{\mu}[\Phi(g_i)] = \mathbf{E}_{\mu}[\Phi(g_i)\phi^{\mu}_{\emptyset}] = 0$), and so every monomial ϕ^{μ}_U in the support of $\widetilde{F}(S)\prod_{i\in S}\Phi(g_i)$ is of the form $\prod_{i\in S}\phi^{\mu}_{U_i}$ where U_i is a non-empty subset of the relevant variables of g_i (*i.e.* $\{(i-1)\ell+1,\ldots,i\ell\}$); likewise for monomials in the support of $\widetilde{F}(T)\prod_{i\in T}\Phi(g_i)$. In other words the non-empty subsets of $[k]$ induce a partition of the μ-biased Fourier support of f, where ϕ^{μ}_U is mapped to $\emptyset \neq S \subseteq [k]$ if and only if U contains a relevant variable of g_i for every $i \in S$ and none of the relevant variables of g_j for any $j \notin S$.

With this identity in hand we have

$$\begin{aligned}
\mathbf{H}^{\mu}[f^{\geq 1}] &= \sum_{S\neq\emptyset} \mathbf{H}^{\mu}\Big[\widetilde{F}(S)\prod_{i\in S}\Phi(g_i)\Big] \\
&= \sum_{S\neq\emptyset} \varphi(\widetilde{F}(S)) + \widetilde{F}(S)^2 \sum_{i\in S} \mathbf{H}^{\mu}[\Phi(g_i)]. \\
&= \sum_{S\neq\emptyset} \varphi(\widetilde{F}(S)) + \widetilde{F}(S)^2 \sum_{i\in S} \left(\frac{\mathbf{H}^{\mu}[g_i - \mathbf{E}_{\mu}[g_i]]}{\mathbf{Var}_{\mu}[g_i]} + \varphi\left(\frac{1}{\mathbf{Var}_{\mu}[g_i]^{1/2}}\right)\mathop{\mathbf{Var}}_{\mu}[g_i]\right) \\
&= \mathbf{H}^{\eta}[F^{\geq 1}] + \sum_{S\neq\emptyset} \widetilde{F}(S)^2 \sum_{i\in S} \frac{\mathbf{H}^{\mu}[g_i^{\geq 1}]}{\mathbf{Var}_{\mu}[g_i]},
\end{aligned}$$

where the second and third equalities are two applications of Lemma 6.2.1 (for the second equality we view $\widetilde{F}(S)$ as a constant function with $\mathbf{H}^{\mu}[\widetilde{F}(S)] = \varphi(\widetilde{F}(S))$). By the same reasoning, we also have

$$\begin{aligned}
\mathbf{Inf}^{\mu}[f] = \sum_{S\neq\emptyset} \mathbf{Inf}^{\mu}\Big[\widetilde{F}(S)\prod_{i\in S}\Phi(g_i(x^i))\Big] &= \sum_{S\neq\emptyset} \widetilde{F}(S)^2 \sum_{i\in S} \mathbf{Inf}^{\mu}[\Phi(g_i)] \\
&= \sum_{S\neq\emptyset} \widetilde{F}(S)^2 \sum_{i\in S} \frac{\mathbf{Inf}^{\mu}[g_i]}{\mathbf{Var}_{\mu}[g_i]}.
\end{aligned}$$

Here the second equality is by Lemma 6.2.1, again viewing $\widetilde{F}(S)$ as a constant function with $\mathbf{Inf}^{\mu}[\widetilde{F}(S)] = 0$, and the third equality uses the fact that $\mathbf{Inf}^{\mu}[\alpha f] = \alpha^2 \cdot \mathbf{Inf}^{\mu}[f]$ and $\mathbf{Inf}^{\mu}[g_i - \mathbf{E}_{\mu}[g_i]] = \mathbf{Inf}^{\mu}[g_i]$. Finally we see that

$$\mathop{\mathbf{Var}}_{\mu}[f] = \sum_{S\neq\emptyset} \mathbf{Var}_{\mu}\left[\widetilde{F}(S)\prod_{i\in S}\Phi(g_i)\right] = \sum_{S\neq\emptyset} \widetilde{F}(S)^2 \prod_{i\in S} \mathop{\mathbf{Var}}_{\mu}[\Phi(g_i)] = \sum_{S\neq\emptyset} \widetilde{F}(S)^2,$$

where the last quantity is $\mathbf{Var}_{\eta}[F]$. Here the second equality uses the fact that the functions $\Phi(g_i)$ are on disjoint sets of variables (and therefore statistically independent when viewed as random variables), and the third equality holds since $\mathbf{Var}_{\mu}[\Phi(g_i)] = \mathbf{E}[\Phi(g_i)^2] - \mathbf{E}[\Phi(g_i)]^2 = 1$. □

We are now ready to prove our main theorem:

Theorem 68. *Let $F : \{-1,1\}^k \to \mathbb{R}$, and $g_1, \ldots, g_k : \{-1,1\}^{k\ell}_{\mu} \to \{-1,1\}$ where each g_i depends only on the ℓ coordinates in $\{(i-1)\ell + 1, \ldots, i\ell\}$. Let $f(x) = F(g_1(x), \ldots, g_k(x))$ and suppose $C > 0$ satisfies*

1. $\mathbf{H}^{\mu}[g_i^{\geq 1}] \leq C \cdot (\mathbf{Inf}^{\mu}[g_i] - \mathbf{Var}_{\mu}[g_i])$ *for all* $i \in [k]$.

2. $\mathbf{H}^{\eta}[F^{\geq 1}] \leq C \cdot (\mathbf{Inf}^{\eta}[F] - \mathbf{Var}_{\eta}[F])$, *where* $\eta = \langle \mathbf{E}_{\mu}[g_1], \ldots, \mathbf{E}_{\mu}[g_k] \rangle$.

Then $\mathbf{H}^{\mu}[f^{\geq 1}] \leq C \cdot (\mathbf{Inf}^{\mu}[f] - \mathbf{Var}_{\mu}[f])$.

Proof. By our first assumption each g_i satisfies $\mathbf{Inf}^{\mu}[g_i] \geq \frac{1}{C}\mathbf{H}^{\mu}[g^{\geq 1}] + \mathbf{Var}_{\mu}[g_i]$, and so combining this with equation (6.2) of Proposition 6.2.2 we have

$$\begin{aligned} \mathbf{Inf}^{\mu}[f] &= \sum_{S \neq \emptyset} \widetilde{F}(S)^2 \sum_{i \in S} \frac{\mathbf{Inf}^{\mu}[g_i]}{\mathbf{Var}_{\mu}[g_i]} \geq \sum_{S \neq \emptyset} \widetilde{F}(S)^2 \sum_{i \in S} \left(\frac{\mathbf{H}^{\mu}[g_i^{\geq 1}]}{C\,\mathbf{Var}_{\mu}[g_i]} + 1 \right) \\ &= \mathbf{Inf}^{\eta}[F] + \frac{1}{C} \sum_{S \neq \emptyset} \widetilde{F}(S)^2 \sum_{i \in S} \frac{\mathbf{H}^{\mu}[g_i^{\geq 1}]}{\mathbf{Var}_{\mu}[g_i]}. \end{aligned} \tag{6.4}$$

This along with equations (6.1) and (6.3) of Proposition 6.2.2 completes the proof:

$$\begin{aligned} \mathbf{H}^{\mu}[f^{\geq 1}] &= \mathbf{H}^{\eta}[F^{\geq 1}] + \sum_{S \neq \emptyset} \widetilde{F}(S)^2 \sum_{i \in S} \frac{\mathbf{H}^{\mu}[g_i^{\geq 1}]}{\mathbf{Var}_{\mu}[g_i]} \\ &\leq C \cdot (\mathbf{Inf}^{\eta}[F] - \mathop{\mathbf{Var}}_{\eta}[F]) + \sum_{S \neq \emptyset} \widetilde{F}(S)^2 \sum_{i \in S} \frac{\mathbf{H}^{\mu}[g_i^{\geq 1}]}{\mathbf{Var}_{\mu}[g_i]} \\ &\leq C \cdot (\mathbf{Inf}^{\mu}[f] - \mathop{\mathbf{Var}}_{\eta}[F]) = C \cdot (\mathbf{Inf}^{\mu}[f] - \mathop{\mathbf{Var}}_{\mu}[f]). \end{aligned}$$

Here the first equality is by (6.1), the first inequality by our second assumption, the second inequality by (6.4), and finally the last identity by (6.3). □

6.3 Distribution-independent bound for FEI$^+$

In this section we prove that μ-biased FEI$^+$ holds for all Boolean functions $F : \{-1,1\}_{\mu}^k \to \{-1,1\}$ with factor C independent of the biases $\mu_1, \ldots, \mu_k$ of μ. When $\mu = \langle 0, \ldots 0 \rangle$ is the uniform distribution it is well-known that the FEI conjecture holds with factor $C = O(\log k)$, and a bound of $C \leq 2^k$ is trivial since $\mathbf{Inf}[F]$ is always an integer multiple of 2^{-k} and $\mathbf{H}[F] \leq 1$; neither proofs carry through to the setting of product distributions. We remark that even verifying the seemingly simple claim "there exists a universal constant C such that $\mathbf{H}^{\mu}[\mathsf{MAJ}_3] \leq C \cdot (\mathbf{Inf}^{\mu}[\mathsf{MAJ}_3] - \mathbf{Var}_{\mu}[\mathsf{MAJ}_3])$ for all product distributions $\mu \in [0,1]^3$", where MAJ_3 the majority function over 3 variables, turns out to be technically cumbersome.

The high-level strategy is to bound each of the $2^k - 1$ terms of $\mathbf{H}^{\mu}[F^{\geq 1}]$ individually via the following lemma:

Lemma 6.3.1. *Let $F : \{-1,1\}^k_{\mu} \to \{-1,1\}$. Let $S \subseteq [k]$, $S \neq \emptyset$, and suppose $\widetilde{F}(S) \neq 0$. For any $j \in S$ we have*

$$\widetilde{F}(S)^2 \log\left(\frac{\prod_{i\in S}\sigma_i^2}{\widetilde{F}(S)^2}\right) \leq \frac{2^{2k+1}}{e\ln 2}\cdot \mathop{\mathbf{Var}}_{\mu}[D_{\phi_j^{\mu}}F].$$

Proof. Recall that $\widetilde{F}(S) = \mathbf{E}_{\mu}[\circ_{i\in S} D_{\phi_i^{\mu}} f] = \prod_{i\in S}\sigma_i \mathbf{E}_{\mu}[D_{x^S} f]$, and so

$$\begin{aligned}
\widetilde{F}(S)^2 \log\left(\frac{\prod_{i\in S}\sigma_i^2}{\widetilde{F}(S)^2}\right) &= \prod_{i\in S}\sigma_i^2 \cdot \mathop{\mathbf{E}}_{\mu}[D_{x^S}F]^2 \log\left(\frac{1}{\mathbf{E}[D_{x^S}F]^2}\right) \\
&\leq \frac{2}{e\ln 2}\prod_{i\in S}\sigma_i^2 \cdot \left|\mathop{\mathbf{E}}_{\mu}[D_{x^S}F]\right| \\
&\leq \frac{2}{e\ln 2}\prod_{i\in S}\sigma_i^2 \mathop{\mathbf{Pr}}_{\mu}[D_{x^S}F \neq 0].
\end{aligned}$$

Here the first inequality holds since $t^2\log(1/t^2) \leq 2t/(e\ln 2)$ for all $t \in \mathbb{R}^+$, and the second uses the fact that $D_{x^S}F$ is bounded within $[-1,1]$. Therefore it suffices to argue that

$$\begin{aligned}
\prod_{i\in S}\sigma_i^2 \mathop{\mathbf{Pr}}_{\mu}[D_{x^S}F \neq 0] &\leq 2^{2k}\cdot \mathop{\mathbf{Var}}_{\mu}[D_{\phi_j^{\mu}}F] \\
&= 2^{2k}\sigma_j^2 \cdot \mathop{\mathbf{Var}}_{\mu}[D_j F] \\
&= 2^{2k}\sigma_j^2 \mathop{\mathbf{E}}_{y\in\{-1,1\}^{[n]\setminus S}}\left[\mathop{\mathbf{E}}_{z\in\{-1,1\}^{S\setminus\{j\}}}\left[((D_jF)|_y(z)-\mu)^2\right]\right],
\end{aligned}$$

where $\mu = \mathbf{E}[D_jF]$ and $(D_jF)|_y$ denotes the restriction of D_jF where the coordinates in $[n]\setminus S$ are set according to y. We first rewrite the desired inequality above as

$$\left(2^{-2k}\prod_{i\in S\setminus\{j\}}\sigma_i^2\right)\mathop{\mathbf{E}}_{y\in\{-1,1\}^{[n]\setminus S}}[\mathbf{1}_{D_{x^S}F(y)\neq 0}] \leq \mathop{\mathbf{E}}_{y\in\{-1,1\}^{[n]\setminus S}}\left[\mathop{\mathbf{E}}_{z\in\{-1,1\}^{S\setminus\{j\}}}\left[((D_jF)|_y(z)-\mu)^2\right]\right]$$

and argue that this holds point-wise: for every $y \in [n]\setminus S$ such that $D_{x^S}F(y) \neq 0$,

$$\mathbf{E}\left[((D_jF)|_y(z)-\mu)^2\right] \geq 2^{-2k}\prod_{i\in S\setminus\{j\}}\sigma_i^2.$$

To see this, fix $y \in \{-1,1\}^{[n]\setminus S}$ such that $(D_{x^S}F)(y) \neq 0$. Viewing $(D_{x^S}F)$ as $(D_{x^{S\setminus\{j\}}}D_jF)$, it follows that $(D_jF)|_y$ is non-constant. Since $(D_jF)|_y$ takes values in $\{-1,0,1\}$, there must

exist some $z^* \in \{-1,1\}^{S\setminus\{j\}}$ such that $|(D_jF)|_y(z^*) - \mu| \geq \frac{1}{2}$ and so indeed

$$\begin{aligned}\mathbf{E}\left[((D_jF)|_y(z)-\mu)^2\right] &\geq \left(\frac{1}{2}\right)^2 \mathbf{Pr}[z=z^*] \\ &= \frac{1}{4}\prod_{i\in S\setminus\{j\}}\frac{1\pm\mu_i}{2} \geq \frac{1}{4}\prod_{i\in S\setminus\{j\}}\frac{\sigma_i^2}{4} \geq 2^{-2k}\prod_{i\in S\setminus\{j\}}\sigma_i^2.\end{aligned}$$

Here the penultimate inequality uses the fact that $1\pm\mu_i \geq (1-\mu_i^2)/2 = \sigma_i^2/2$, which holds since $|\mu_i| \leq 1$. □

Theorem 69. *Let $F : \{-1,1\}_\mu^k \to \{-1,1\}$. Then $\mathbf{H}^\mu[F^{\geq 1}] \leq 2^{O(k)} \cdot (\mathbf{Inf}^\mu[F] - \mathbf{Var}_\mu[F])$.*

Proof. The claim can be equivalently stated as $\mathbf{H}^\mu[F^{\geq 1}] \leq 2^{O(k)} \sum_{i=1}^n \mathbf{Var}_\mu[D_{\phi_i^\mu}F]$, since

$$\sum_{i=1}^n \mathbf{Var}[D_{\phi_i^\mu}F] = \sum_{|S|\geq 2}|S|\cdot\widetilde{F}(S)^2 \leq 2\sum_{|S|\geq 2}(|S|-1)\cdot\widetilde{F}(S)^2 = 2\cdot(\mathbf{Inf}^\mu[F]-\mathbf{Var}_\mu[F]).$$

By Lemma 6.3.1, the contribution of each $S \neq \emptyset$ to $\mathbf{H}^\mu[F^{\geq 1}]$ is $2^{O(k)}\,\mathbf{Var}_\mu[D_{\phi_j^\mu}F]$, where j is any element of S. Summing over all $2^k - 1$ non-empty subsets S of $[k]$ completes the proof. □

6.3.1 FEI$^+$ for read-once formulas

Finally, we combine our two main results so far, the composition theorem (Theorem 68) and the distribution-independent universal bound (Theorem 69), to prove Conjecture 3 for read-once formulas with arbitrary gates of bounded arity.

Definition 77. Let $\mathcal{B}$ be a set of Boolean functions. We say that a Boolean function f is a formula over the basis $\mathcal{B}$ if f is computable by a formula with gates belonging to $\mathcal{B}$. We say that f is a read-once formula over $\mathcal{B}$ if every variable appears at most once in the formula for f.

Corollary 6.3.2. *Let $C > 0$ and $\mathcal{B}$ be a set of Boolean functions, and suppose $\mathbf{H}^\mu[F] \leq C\cdot(\mathbf{Inf}^\mu[F]-\mathbf{Var}_\mu[F])$ for all $F \in \mathcal{B}$ and product distributions μ. Let $\mathcal{C}$ be the class of read-once formulas over the basis $\mathcal{B}$. Then $\mathbf{H}^\mu[f] \leq C\cdot(\mathbf{Inf}^\mu[f]-\mathbf{Var}_\mu[f])$ for all $f \in \mathcal{C}$ and product distributions μ.*

Proof. We proceed by structural induction on the formula computing f. The base case holds since the μ-biased Fourier expansion of the dictator x_1 and anti-dictator $-x_1$ is $\pm(\mu_1 + \sigma_1\phi_1^\mu(x))$ and so $\mathbf{H}^\mu[f^{\geq 1}] = \widetilde{f}(\{1\})^2 \log(\sigma_1^2/\widetilde{f}(\{1\})^2) = \sigma_1^2 \log(\sigma_1^2/\sigma_1^2) = 0$.

For the inductive step, suppose $f = F(g_1, \ldots, g_k)$, where $F \in \mathcal{B}$ and $g_1, \ldots, g_k$ are read-once formulas over $\mathcal{B}$ over disjoint sets of variables. Let μ be any product distribution over the domain of f. By our induction hypothesis we have $\mathbf{H}^\mu[g_i^{\geq 1}] \leq C \cdot (\mathbf{Inf}^\mu[g_i] - \mathbf{Var}_\mu[g_i])$ for all $i \in [k]$, satisfying the first requirement of Theorem 68. Next, by our assumption on $F \in \mathcal{B}$, we have $\mathbf{H}^\eta[F^{\geq 1}] \leq C \cdot (\mathbf{Inf}^\eta[F] - \mathbf{Var}_\eta[F])$ for all product distributions η, and in particular, $\eta = \langle \mathbf{E}_\mu[g_1], \ldots, \mathbf{E}_\mu[g_k] \rangle$, satisfying the second requirement of Theorem 68. Therefore, by Theorem 68 we conclude that $\mathbf{H}^\mu[f] \leq C \cdot (\mathbf{Inf}^\mu[f] - \mathbf{Var}_\mu[f])$. □

By Theorem 69, for any set $\mathcal{B}$ of Boolean functions with maximum arity k and product distribution μ, every $F \in \mathcal{B}$ satisfies $\mathbf{H}^\mu[F] \leq 2^{O(k)} \cdot (\mathbf{Inf}^\mu[F] - \mathbf{Var}_\mu[F])$. Combining this with Corollary 6.3.2 yields the following:

Theorem 70. *Let $\mathcal{B}$ be a set of Boolean functions with maximum arity k, and $\mathcal{C}$ be the class of read-once formulas over the basis $\mathcal{B}$. Then $\mathbf{H}^\mu[f] \leq 2^{O(k)} \cdot (\mathbf{Inf}^\mu[f] - \mathbf{Var}_\mu[f])$ for all $f \in \mathcal{C}$ and product distributions μ.*

6.4 Lower bound on the constant of the FEI conjecture

The tools we develop in this work also yield an explicit function f achieving the best-known ratio between $\mathbf{H}[f]$ and $\mathbf{Inf}[f]$ (*i.e.* a lower bound on the constant C in the FEI conjecture). We will use the following special case of Proposition 6.2.2 on the behavior of spectral entropy and total influence under composition:

Lemma 6.4.1 (Amplification lemma)**.** *Let $F : \{-1,1\}^k \to \{-1,1\}$ be a balanced Boolean function. Let $f_0 = F$, and define $f_m = F(f_{m-1}(x^1), \ldots, f_{m-1}(x^k))$ for all $m \geq 1$. Then*

$$\frac{\mathbf{H}[f_m]}{\mathbf{Inf}[f_m]} = \frac{\mathbf{H}[F]}{\mathbf{Inf}[F]} + \frac{\mathbf{H}[F]}{\mathbf{Inf}[F](\mathbf{Inf}[F]-1)} - \frac{\mathbf{H}[F]}{\mathbf{Inf}[F]^{m+1}(\mathbf{Inf}[F]-1)}.$$

Theorem 71. *There exists an infinite family of functions $f_m : \{-1,1\}^{6^m} \to \{-1,1\}$ such that $\lim_{m\to\infty} \mathbf{H}[f_m]/\mathbf{Inf}[f_m] \geq 6.278944$.*

Proof. Let

$$g = (\overline{x}_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge x_4) \vee (x_1 \wedge \overline{x}_2 \wedge x_5 \wedge x_6) \vee (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_4 \wedge x_5).$$

It can be checked that g is a balanced function with $\mathbf{H}[F] \geq 3.92434$ and $\mathbf{Inf}[F] = 1.625$. Applying Lemma 6.4.1 with $F = g$, we get

$$\lim_{m \to \infty} \frac{\mathbf{H}[f_m]}{\mathbf{Inf}[f_m]} \geq \frac{3.92434}{1.625} + \frac{3.92434}{1.625 \times 0.625} = 6.278944.$$

□

Chapter 7

Conclusions

Our results in this thesis shed new light on the mathematical structure of Boolean functions, and we have seen how this improved structural understanding can be leveraged to obtain new computational results, both algorithmic upper bounds and complexity-theoretic lower bounds, in property testing, learning theory, and circuit complexity. Many interesting questions remain to be answered; in this final chapter we highlight a few that we find particularly intriguing.

Testing Monotonicity. In Chapter 2 we gave an $\tilde{O}(n^{5/6})$-query non-adaptive algorithm for testing the monotonicity of Boolean functions, and showed that any non-adaptive algorithm must make $\tilde{\Omega}(n^{1/5})$ many queries. While our results bring the gap between the upper and lower bounds for non-adaptive algorithms down to a polynomial factor, an exponential gap remains in the best-known bounds for *adaptive* algorithms. Is there a polylog(n)-query adaptive algorithm for testing the monotonicity of Boolean functions?

Agnostic Learning. In Chapter 3 we gave a characterization of the statistical query complexity of agnostically learning Boolean functions under product distributions. With this characterization in hand a next natural step is to investigate the complexity of agnostic learning in other variants of the PAC model; a particularly well-studied one is that of agnostic learning from *membership queries*. Is there a poly(n)-time membership query algorithm for agnostically learning the class of n-variable DNF formulas to any constant

excess error $\varepsilon > 0$? Is there a $\mathrm{poly}(n, 1/\varepsilon)$-time membership query for agnostically learning the class of n-variable linear threshold functions to any excess error $\varepsilon > 0$?

Approximation by Small-Depth Circuits. In our concluding remarks to Chapters 4 and 5 we listed a few of the many open problems concerning the complexity of exact versus approximate computation in small-depth circuits. Here we mention one more, related to our results in Chapter 5 on the role of negations in the approximation of monotone functions by DNF formulas. Beyond DNF formulas, one may ask quantitatively just how powerful negations can be in the approximation of monotone functions for many other classes of circuits. The following question concerning the possibility of strengthening the Okol'nishnikova–Ajtai–Gurevich theorem is due to Gil Kalai [Kalai, 2010]: Is there a monotone function in AC^0 that cannot be approximated by monotone AC^0?

The Fourier Entropy–Influence Conjecture. Given its implications for learning theory, the theory of pseudorandomness, and random graph theory, resolving the Fourier Entropy–Influence Conjecture is evidently of significant importance. The following consequence of the FEI conjecture, sufficient to imply Mansour's conjecture and hence the existence of an efficient membership query algorithm for agnostically learning DNF formulas, is open: Can every Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ be ε-approximated in ℓ_2 by a $2^{O_\varepsilon(\mathbf{Inf}[f])}$-sparse polynomial $p : \{-1,1\}^n \to \mathbb{R}$?

Bibliography

[Ailon *et al.*, 2007] Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Estimating the distance to a monotone function. *Random Struct. Algorithms*, 31(3):371–383, 2007. 2.1

[Ajtai and Grevich, 1987] Miklós Ajtai and Yuri Grevich. Monotone versus positive. *Journal of the ACM*, 34(4):1004–1015, 1987. 5.1, 5.1

[Ajtai, 1983] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983. 4.1

[Alon and Boppana, 1987] N. Alon and R. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7:1–22, 1987. 5.1

[Amano, 2011] Kazuyuki Amano. Tight bounds on the average sensitivity of k-CNF. *Theory of Computing*, 7(1):45–48, 2011. 4.1.1, 4.5.1, 5.4.3

[Austrin and Håstad, 2011] Per Austrin and Johan Håstad. Randomly supported independence and resistance. *SIAM Journal on Computing*, 40(1):1–27, 2011. 3.1.1

[Austrin and Mossel, 2009] Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. *Computational Complexity*, 18(2):249–271, 2009. 3.1.1

[Batu *et al.*, 2004] Tugkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *ACM Symposium on Theory of Computing*, pages 381–390, 2004. 2.1

[Beame *et al.*, 2012] Paul Beame, Russell Impagliazzo, and Srikanth Srinivasan. Approximating AC^0 by small height decision trees and a deterministic algorithm for $\#\mathsf{AC}^0\mathsf{SAT}$. In *IEEE Conference on Computational Complexity*, pages 117–125, 2012. 4.1

[Beckner, 1975] William Beckner. Inequalities in Fourier analysis. *Annals of Mathematics*, 102:159–182, 1975. 3.3.2

[Bellare *et al.*, 1996] Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos Kiwi, and Madhu Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory*, 42(6):1781–1795, 1996. 1.3

[Bernasconi *et al.*, 1997] A. Bernasconi, B. Codenotti, and J. Simon. On the Fourier analysis of Boolean functions. Technical report, Istituto di Matematica Computazionale, 1997. 4.5.1

[Blais and O'Donnell, 2010] Eric Blais and Ryan O'Donnell. Lower bounds for testing function isomorphism. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, pages 235–246, 2010. 2.1.1, 2.2.2

[Blais and Tan, 2013] Eric Blais and Li-Yang Tan. Approximating Boolean functions with depth-2 circuits. In *Proceedings of the 28th Annual IEEE Conference on Computational Complexity*, pages 74–85, 2013. 4.1.1, 5.1, 5.1, 5.1, 5.1.1, 5.1.2

[Blais *et al.*, 2012] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012. 2.1

[Blais *et al.*, 2013a] Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions on hypergrid domains. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:36, 2013. 2.1, 2.1

[Blais *et al.*, 2013b] Eric Blais, Li-Yang Tan, and Andrew Wan. Edge-isoperimetric inequalities via the entropy method. Manuscript, 2013. 4.5.1, 4.7

[Blum *et al.*, 1993] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47:549–595, 1993. 1.3

[Blum *et al.*, 1994] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262. ACM, 1994. 3.1.3, 3.2

[Blum *et al.*, 1998] A. Blum, C. Burch, and J. Langford. On learning monotone boolean functions. In *Proceedings of FOCS*, pages 408–415, 1998. 3.1.2, 3.1.2, 3.1.2, 3.3.1, 25

[Bonami, 1970] Aline Bonami. Étude des coefficients Fourier des fonctions de $L^p(G)$. *Annales de l'Institute Fourier*, 20(2):335–402, 1970. 3.3.2

[Boppana, 1997] Ravi Boppana. The average sensitivity of bounded-depth circuits. *Information Processing Letters*, 63(5):257–261, 1997. 4.1.1, 4.5.1, 4.5.1

[Briët *et al.*, 2012] Jop Briët, Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012. 2.1, 2.1, 2.1

[Bshouty and Tamon, 1996] Nader Bshouty and Christino Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996. 1.3, 3.1.2, 3.1.2, 3.1.2, 5.1.2

[Cai, 1989] Jin-Yi Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. *J. Comput. Syst. Sci.*, 38(1):68–85, 1989. 4.1

[Chakrabarty and Seshadhri, 2013a] Deeparnab Chakrabarty and C. Seshadhri. A $o(n)$ monotonicity tester for boolean functions over the hypercube. In *ACM Symposium on Theory of Computing*, pages 411–418, 2013. 1.4, 2.1, 2.1, 2.1.2, 3, 2.5.1, 2.5.1, 2.5.4, 10

[Chakrabarty and Seshadhri, 2013b] Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and lipschitz testing over hypercubes and hypergrids. In *ACM Symposium on Theory of Computing*, pages 419–428, 2013. 2.1, 2.1

[Chakrabarty and Seshadhri, 2013c] Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. In *APPROX-RANDOM*, pages 425–435, 2013. 2.1, 2.1

[Chan and Potechin, 2012] Siu Man Chan and Aaron Potechin. Tight bounds for monotone switching networks via fourier analysis. In *Symposium on Theory of Computing (STOC)*, pages 495–504, 2012. 5.1

[Chor *et al.*, 1985] Benny Chor, Oded Goldreich, Johan Hasted, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *Foundations of Computer Science, 1985., 26th Annual Symposium on*, pages 396–407. IEEE, 1985. 3.1.1

[Chow, 1961] Chao-Kong Chow. On the characterization of threshold functions. In *Proceedings of the 2nd Annual Symposium on Switching Circuit Theory and Logical Design (FOCS)*, pages 34–38, 1961. 4.4

[Cohen *et al.*, 2005] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes*. North-Holland Mathematical Library. Elsevier Science, 2005. 4.3.2

[Crama and Hammer, 2011] Yves Crama and Peter L. Hammer. *Boolean Functions - Theory, Algorithms, and Applications*, volume 142 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2011. 4.1.1

[De *et al.*, 2010] Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. In *Proceedings of the 14th Annual International Workshop on Randomized Techniques in Computation*, pages 504–517, 2010. 6.1, 6.1.1

[Dodis *et al.*, 1999] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonocity. In *Proceedings of RANDOM*, pages 97–108, 1999. 2.1

[Ergün *et al.*, 2000] Funda Ergün, Sampath Kannan, S. Ravi Kumar, Ronitt Rubinfeld, and Mahesh Vishwanthan. Spot-checkers. *Journal of Computer and System Sciences*, 60:717–751, 2000. Earlier version in STOC'96. 2.1

[Feldman and Kothari, 2013] V. Feldman and P. Kothari. Learning coverage functions and private release of marginals. *arXiv, CoRR*, abs/1304.2079, 2013. 3.1.3

[Feldman *et al.*, 2011] Vitaly Feldman, Homin K. Lee, and Rocco A. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. In *COLT*, pages 273–292, 2011. 3.1.3

[Feldman *et al.*, 2012] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM J. Comput.*, 41(6):1558–1590, 2012. 3.1

[Feldman, 2012] V. Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012. 3.1.3, 3.2

[Feller, 1968] W. Feller. *An introduction to probability theory and its applications.* John Wiley & Sons, 1968. 3.4

[Filmus *et al.*, 2013] Yuval Filmus, Toniann Pitassi, Robert Robere, and Stephen A. Cook. Average case lower bounds for monotone switching networks. In *Symposium on Foundations of Computer Science (FOCS)*, 2013. 5.1

[Fischer *et al.*, 2002] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 474–483, 2002. 1.4, 2.1, 8

[Fischer, 2004] Eldar Fischer. On the strength of comparisons in property testing. *Inf. Comput.*, 189(1):107–116, 2004. 2.1

[Friedgut and Kalai, 1996] Ehud Friedgut and Gil Kalai. Every monotone graph property has a sharp threshold. *Proceedings of the American Mathematical Society*, 124(10):2993–3002, 1996. 6.1

[Friedgut, 1998] Ehud Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):27–36, 1998. 5.4.3, 6.1

[Furst *et al.*, 1984] Merrick Furst, James Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. 4.1

[Glagolev, 1967] V. V. Glagolev. Nekotorye otsenki dizyunktivnykh normalnykh form funktsiĭ algebry logiki. *Problemy Kibernetiki*, 19:75–95, 1967. 1

[Goldmann and Håstad, 1998] Mikael Goldmann and Johan Håstad. Monotone circuits for connectivity have depth $(\log n)^{2\text{-}o(1)}$. *SIAM J. Comput.*, 27(5):1283–1294, 1998. 5.1

[Goldreich *et al.*, 1998] Oded Goldreich, Shafi Goldwasser, Eric Lehman, and Dana Ron. Testing monotonicity. In *IEEE Symposium on Foundations of Computer Science*, pages 426–435, 1998. 2.1, 2.1.2

[Goldreich *et al.*, 2000] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samordinsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000. 2.1

[Gopalan *et al.*, 2008a] Parikshit Gopalan, Adam Kalai, and Adam Klivans. Agnostically learning decision trees. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 527–536, 2008. 6.1

[Gopalan *et al.*, 2008b] Parikshit Gopalan, Adam Kalai, and Adam Klivans. A query algorithm for agnostically learning DNF? In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 515–516, 2008. 6.1

[Gopalan *et al.*, 2010] Parikshit Gopalan, Ryan O'Donnell, Yi Wu, and David Zuckerman. Fooling functions of halfspaces under product distributions. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity*, pages 223–234, 2010. 2.1.1

[Gopalan *et al.*, 2013] Parikshit Gopalan, Raghu Meka, and Omer Reingold. Dnf sparsification and a faster deterministic counting algorithm. *Computational Complexity*, 22(2):275–310, 2013. 5.4.3

[Grigni and Sipser, 1995] Michelangelo Grigni and Michael Sipser. Monotone separation of logarithmic space from logarithmic depth. *J. Comput. Syst. Sci.*, 50(3):433–437, 1995. 5.1

[Halevy and Kushilevitz, 2008] Shirley Halevy and Eyal Kushilevitz. Testing monotonicity over graph products. *Random Struct. Algorithms*, 33(1):44–67, 2008. 2.1

[Håstad, 1986] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *STOC*, pages 6–20, 1986. 4.1

[Håstad, 2012] Johan Håstad. On the correlation of parity and small-depth circuits. Technical Report TR12-137, Electronic Colloquium on Computational Complexity, 2012. 4.1, 4.1

[Haussler, 1992] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992. 3.1

[Heiman *et al.*, 1993] Rafi Heiman, Ilan Newman, and Avi Wigderson. On read-once threshold formulae and their randomized decision in tree complexity. *Theor. Comput. Sci.*, 107(1):63–76, 1993. 6.1.1

[Impagliazzo *et al.*, 2012] Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithm for AC^0. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 961–972, 2012. 4.1

[Ioffe and Tikhomirov, 1968] Aleksandr Ioffe and Vladimir Tikhomirov. Duality of convex functions and extremum problems. *Russ. Math. Surv.*, 23, 1968. 3.1.1, 3.2

[Jackson, 1997] Jeffrey Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997. 5.1.2

[Kabatyanski and Panchenko, 1988] G. A. Kabatyanski and V.I. Panchenko. Packings and coverings of the hamming space by unit balls. *Dokl. Akad. Nauk SSSR*, 303(3):550–552, 1988. 4.1.1, 4.3.2

[Kahn *et al.*, 1988] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on Boolean functions. In *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–80, 1988. 1.3, 3.3.4.1, 3.4.2, 6.1

[Kalai *et al.*, 2008] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008. 1.4, 3.1, 3.1.1, 13, 3.1.2, 3.1.2, 3.1.3, 3.2.1, 22

[Kalai, 2007] Gil Kalai. The entropy/influence conjecture. Posted on Terence Tao's *What's new* blog, http://terrytao.wordpress.com/2007/08/16/gil-kalai-the-entropyinfluence-conjecture/, 2007. 6.1

[Kalai, 2010] Gil Kalai. Noise stability and threshold circuits. Gil Kalai's *Combinatorics and more* blog, http://gilkalai.wordpress.com/2010/02/10/noise-stability-and-threshold-circuits/, 2010. 7

[Karchmer and Wigderson, 1988] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 539–550, New York, NY, USA, 1988. ACM. 5.1

[Karchmer *et al.*, 1991] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via direct sum in communication coplexity. In *Structure in Complexity Theory Conference*, pages 299–304, 1991. 5.1

[Kearns *et al.*, 1994] M. Kearns, R. Schapire, and L. Sellie. Toward Efficient Agnostic Learning. *Machine Learning*, 17(2/3):115–141, 1994. 1.4, 3.1

[Kearns, 1998] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998. 3.1, 3.1.3

[Keller *et al.*, 2012] Nathan Keller, Elchanan Mossel, and Tomer Schlank. A note on the entropy/influence conjecture. *Discrete Mathematics*, 312(22):3364–3372, 2012. 6.1.1

[Klivans and Sherstov, 2007] Adam R Klivans and Alexander A Sherstov. Unconditional lower bounds for learning intersections of halfspaces. *Machine Learning*, 69(2-3):97–114, 2007. 3.1.3

[Klivans and Sherstov, 2010] Adam R. Klivans and Alexander A. Sherstov. Lower bounds for agnostic learning via approximate rank. *Computational Complexity*, 19(4):581–604, 2010. 3.1

[Klivans *et al.*, 2004] Adam Klivans, Ryan O'Donnell, and Rocco Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer and System Sciences*, 68(4):808–840, 2004. 4.1.1, 4.6

[Klivans *et al.*, 2010] Adam Klivans, Homin Lee, and Andrew Wan. Mansour's Conjecture is true for random DNF formulas. In *Proceedings of the 23rd Annual Conference on Learning Theory*, pages 368–380, 2010. 6.1.1, 6.1.1

[Korshunov, 1969] Aleksej Dmitrievich Korshunov. Verkhnyaya otsenka slozhnosti kratchaĭshikh dnf pochti vsekh bulevykh funktsiĭ. *Kibernetika*, 6:1–8, 1969. 1

[Korshunov, 1981] Aleksej Dmitrievich Korshunov. O slozhnosti kratchaĭshikh dizyunktivnykh normalnykh form bulevykh funktsiĭ. *Metody Diskretnogo Anal*, 37:9–41, 1981. 1

[Korshunov, 1983] Aleksej Dmitrievich Korshunov. O slozhnosti kratchaĭshikh dizyunktivnykh normalnykh form sluchanykh bulevykh funktsiĭ. *Metody Diskretnogo Anal*, 40:25–53, 1983. 4.1, 1

[Korshunov, 2003] A. D. Korshunov. Monotone Boolean functions. *Russian Math. Surveys (Uspekhi Mat. Nauk)*, 58(5):929–1001, 2003. 5.1

[Krause and Pudlák, 1997] Matthias Krause and Pavel Pudlák. On the computational power of depth-2 circuits with threshold and modulo gates. *Theoretical Computer Science*, 174(1–2):137–156, 1997. 5.1.2

[Krivelevich *et al.*, 2003] Michael Krivelevich, Benny Sudakov, and Van H. Vu. Covering codes with improved density. *IEEE Transactions on Information Theory*, 49(7):1812–1815, 2003. 4.3.2

[Kuznetsov, 1983] S. E. Kuznetsov. O nizhneĭ otsenke dliny kratchaĭsheĭ dnf pochti vsekh bulevykh funktsiĭ. *Veroyatnoste Metody Kibernetiki*, 19:44–47, 1983. 4.1, 1, 4.1.1

[Linial *et al.*, 1989] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform and Learnability. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 574–579, 1989. 4.1

[Lovász, 1987] László Lovász. *An algorithmic theory of numbers, graphs and convexity*, volume 50. SIAM, 1987. 2

[Luby and Wigderson, 1995] Michael Luby and Avi Wigderson. *Pairwise independence and derandomization*. Citeseer, 1995. 3.1.1

[Lupanov, 1961] Oleg Lupanov. Implementing the algebra of logic functions in terms of constant depth formulas in the basis &, $\vee$, $\neg$. *Dokl. Ak. Nauk. SSSR*, 136:1041–1042, 1961. 4.1

[Mansour, 1994] Yishay Mansour. Learning Boolean functions via the Fourier Transform. In Vwani Roychowdhury, Kai-Yeung Siu, and Alon Orlitsky, editors, *Theoretical Advances in Neural Computation and Learning*, chapter 11, pages 391–424. Kluwer Academic Publishers, 1994. 6.1

[Mansour, 1995] Y. Mansour. An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50:543–550, 1995. 3.3.2

[Matulef *et al.*, 2009] Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco Servedio. Testing ± 1-weight halfspaces. In *Proceedings of the 13th Annual International Workshop on Randomized Techniques in Computation*, pages 646–657, 2009. 2.1.1

[Matulef *et al.*, 2010] Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco Servedio. Testing halfspaces. *SIAM Journal on Computing*, 39(5):2004–2047, 2010. 2.1.1

[Mossel and O'Donnell, 2002] Elchanan Mossel and Ryan O'Donnell. On the noise sensitivity of monotone functions. In *Mathematics and Computer Science II*, pages 481–495. Springer, 2002. 3.1.2

[Mossel *et al.*, 2004] E. Mossel, R. O'Donnell, and R. Servedio. Learning functions of k relevant variables. *Journal of Computer & System Sciences*, 69(3):421–434, 2004. Previously published as "Learning juntas". 3.1.2

[Mossel, 2008] Elchanan Mossel. Gaussian bounds for noise correlation of functions and tight analysis of Long Codes. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 156–165, 2008. 2.1.1

[O'Donnell and Servedio, 2008] Ryan O'Donnell and Rocco Servedio. Learning monotone decision trees in polynomial time. *SIAM Journal on Computing*, 37(3):827–844, 2008. 1.3, 3.1.2, 6.1

[O'Donnell and Wimmer, 2007] Ryan O'Donnell and Karl Wimmer. Approximation by DNF: examples and counterexamples. In *Proceedings of the 34th Annual International Colloquium on Automata, Languages and Programming*, pages 195–206, 2007. 4.1, 5.1, 5.1.2, 5.5

[O'Donnell and Wimmer, 2013] Ryan O'Donnell and Karl Wimmer. KKL, Kruskal-Katona, and Monotone nets. *SIAM J. Comput.*, 42(6):2375–2399, 2013. 3.1.2, 3.4

[O'Donnell *et al.*, 2011] Ryan O'Donnell, John Wright, and Yuan Zhou. The Fourier Entropy-Influence Conjecture for certain classes of boolean functions. In *Proceedings of the 38th Annual International Colloquium on Automata, Languages and Programming*, pages 330–341, 2011. 1.4, 6.1.1

[O'Donnell, 2003] Ryan O'Donnell. *Computational applications of noise sensitivity*. PhD thesis, Massachusetts Institute of Technology, 2003. 3.1.2, 3.3.2

[O'Donnell, 2004] Ryan O'Donnell. Hardness amplification within NP. *Journal of Computer and System Sciences*, 69(1):68–94, 2004. 1.3

[O'Donnell, 2014] Ryan O'Donnell. *The Analysis of Boolean Functions*. Cambridge Univ. Press, 2014. Preliminary version at analysisofbooleanfunctions.org. 2.1.3, 3.3.2

[Okol'nishnikova, 1982] E. Okol'nishnikova. On the influence of negations on the complexity of a realization of monotone Boolean functions by formulas of bounded depth (in Russian). *Metody Diskret. Analiz.*, 38:74–80, 1982. 5.1

[Pippenger, 2003] Nicholas Pippenger. The shortest disjunctive normal form of a random boolean function. *Random Structures & Algorithms*, 22(2):161–186, 2003. 1, 4.1.1

[Potechin, 2010] Aaron Potechin. Bounds on monotone switching networks for directed connectivity. In *Symposium on Foundations of Computer Science (FOCS)*, pages 553–562, 2010. 5.1

[Quine, 1954] Willard Van Orman Quine. Two theorems about truth functions. *Bol. Soc. Math. Mexicana*, 10:64–70, 1954. 4.1, 5.1

[Raz and McKenzie, 1999] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. 5.1

[Raz and Wigderson, 1990] R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. In *Proceedings of the 22nd ACM Symposium on Theory of Computing*, pages 287–292, 1990. 5.1

[Razborov and Rudich, 1997] A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997. 5.1

[Razborov, 1985] Aleksandr A. Razborov. Lower bounds for the monotone complexity of some boolean functions. *Soviet Mathematics Doklady*, 31:354–357, 1985. 5.1, 5.1

[Ron and Servedio, 2013] Dana Ron and Rocco A. Servedio. Exponentially improved algorithms and lower bounds for testing signed majorities. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1319–1336, 2013. 2.1.1

[Ron *et al.*, 2012] Dana Ron, Ronitt Rubinfeld, Muli Safra, Alex Samorodnitsky, and Omri Weinstein. Approximating the influence of monotone Boolean functions in $O(\sqrt{n})$ query complexity. *ACM Transactions on Computation Theory*, 4(4):11, 2012. 2.1

[Rubinfeld and Servedio, 2009] Ronitt Rubinfeld and Rocco A. Servedio. Testing monotone high-dimensional distributions. *Random Struct. Algorithms*, 34(1):24–44, 2009. 2.1

[Rubinfeld and Sudan, 1996] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. 1.2, 1.4

[Sapozhenko, 1972] A. A. Sapozhenko. O slozhnosti dizyunktivnykh normalnykh form, poluchaemykh s pomoshchyu gradientnogo algoritma. *Diskretnyĭ Anal*, 21:62–71, 1972. 1

[Schilling, 1990] Mark F Schilling. The longest run of heads. *College Math. J*, 21(3):196–207, 1990. 3.4.1

[Servedio, 2004a] R. Servedio. On learning monotone DNF under product distributions. *Information and Computation*, 193(1):57–74, 2004. 3.1.2, 6.1

[Servedio, 2004b] Rocco Servedio. Monotone Boolean formulas can approximate monotone linear threshold functions. *Discrete Applied Mathematics*, 142(1-3):181–187, 2004. 1.3

[Sherstov, 2011] Alexander A. Sherstov. The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011. 3.1.1, 3.1.3

[Simon, 2007] H. Simon. A characterization of strong learnability in the statistical query model. In *Proceedings of Symposium on Theoretical Aspects of Computer Science*, pages 393–404, 2007. 3.1.3

[Szörényi, 2009] Balázs Szörényi. Characterizing statistical query learning: simplified notions and proofs. In *Algorithmic Learning Theory*, pages 186–200. Springer, 2009. 3.1.3, 3.2

[Talagrand, 1993] M. Talagrand. Isoperimetry, logarithmic Sobolev inequalities on the discrete cube and Margulis' graph connectivity theorem. *GAFA*, 3(3):298–314, 1993. 3.4.2

[Talagrand, 1996] M. Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996. 3.1.2

[Tardos, 1988] Éva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988. 5.1, 5.1

[Traxler, 2009] Patrick Traxler. Variable influences in conjunctive normal forms. In *SAT*, pages 101–113, 2009. 4.5.1

[Valiant and Valiant, 2011] Gregory Valiant and Paul Valiant. Estimating the unseen: an $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. In *ACM Symposium on Theory of Computing*, pages 685–694, 2011. 2.1.1, 2.3, 2.3

[Valiant, 1984] Leslie Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. 1.2, 1.4

[Valiant, 2012] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 11–20. IEEE, 2012. 3.1.2, 3.1.3

[Wieder, 2002] Udi Wieder. Tennis for the people II. *http://windowsontheory.org/2012/11/16/tennis-for-the-people-ii/*, 2002. 3.1.2

[Yang, 2005] Ke Yang. New lower bounds for statistical query learning. *Journal of Computer and System Sciences*, 70(4):485–509, 2005. 3.2

[Yao, 1977] Andrew Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing*, pages 222–227, 1977. 2.1.1, 2.2

[Yao, 1979] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979. 1.2

[Yao, 1985] Andrew Yao. Separating the polynomial time hierarchy by oracles. In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, pages 1–10, 1985. 4.1