UNIVERSITY OF CALIFORNIA SAN DIEGO

# Discrete Fourier Analysis and Its Applications

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Computer Science

by

Jiapeng Zhang

Committee in charge:

Professor Shachar Lovett, Chair
Professor Samuel Buss
Professor Russell Impagliazzo
Professor James McKernan
Professor Daniele Micciancio

2019

The Dissertation of Jiapeng Zhang is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2019

# DEDICATION

To my family, I could not have done it without you.

# EPIGRAPH

And ye shall know the truth, and the truth shall make you free.

*The Book of John 8:32*

## TABLE OF CONTENTS

# ACKNOWLEDGEMENTS

I am very fortunate to have Shachar Lovett as my advisor, whom continiusly encourages and supports me. Shachar guided me to work on interesting problems, and shared me countless good ideas. Besides advising in research, Shachar also advised me a lot of things in my daily life, including how to build networks in our research community, how to find a good job, how to make good barbecue chicken, how to make my wife happy, and so on.

I thank all of my thesis committe members, Samuel Buss, Russell Impagliazza, James McKernan and Daniele Micciancio.

I am grateful to have collaboration with a lot of outstanding researchers, Yi-Hsiu Chen, Mika Göös, Kun He, Daniel Kane, Qian Li, Xin Li, Shachar Lovett, Shay Moran, Noam Solomon, Xiaoming Sun, Avishay Tal, Bo Tang, and Salil Vadhan. I learned a lot of knowledge from my coauthors.

I thank to students in UCSD theory group, including (but not limited to) Marco Carmosino, Anant Dhayal, Jiawei Gao, Nick Genise, Max Hopkins, Kawa Hosseini, Baiyu Li, Sankeerth Rao. We had a lot of fun during my study time at UCSD.

I also thank to Benny Applebaum, Alexander Belove, Eric Blais, Xue Chen, Siyao Guo, Iftach Haitner, Pan Peng and Avi Wigderson.

My most important acknowledge is to my family and friends. Especially thank to my wife Yingcong Li (her name appeared frequently in the acknowledgement of my research papers), and my parents. They encouraged me to chase my dreams.

Chapter 3 contains a reprint of material as it appears in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms 2018. Shachar Lovett, Avishay Tal and Jiapeng Zhang. *The robust sensitivity of boolean functions.* The dissertation author was a primary investigator and author of this paper.

Chapter 4 contains a reprint of material as it appears in Proceedings of the fifty-first Annual ACM symposium on Theory of computing 2019. Shachar Lovett and Jiapeng Zhang. *DNF sparsification beyond sunflowers.* The dissertation author was a primary investigator and

author of this paper.

Chapter 5 contains a reprint of material as it appears in Proceedings of the forty-seventh Annual ACM symposium on Theory of computing 2015. Shachar Lovett and Jiapeng Zhang. *Improved noisy population recovery, reverse Bonami-Beckner inequality for sparse functions.* The dissertation author was a primary investigator and author of this paper.

# VITA

| | |
|---|---|
| 2011 | Bachelor of Science, Shanghai Jiao Tong University, Shanghai |
| 2016 | Master of Science, University of California San Diego |
| 2019 | Doctor of Philosophy, University of California San Diego |

# PUBLICATIONS

- Shachar Lovett, Noam Solomon and Jiapeng Zhang. *From DNF compression to sunflower theorems via regularity.* 34th Computational Complexity Conference (CCC 2019) [39]
- Shachar Lovett and Jiapeng Zhang. *DNF sparsification beyond sunflowers.* Proceedings of the 51st annual ACM symposium on Theory of computing [44]
- Kun He, Qian Li, Xiaoming Sun and Jiapeng Zhang. *Quantum Lovász Local Lemma: Shearer's Bound is Tight.* Proceedings of the 51st annual ACM symposium on Theory of computing [29]
- Xin Li, Shachar Lovett and Jiapeng Zhang. *Sunflowers and Quasi-Sunflowers from Randomness Extractors.* Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (RANDOM 2018) [35]
- Yi-Hsiu Chen, Mika Göös, Salil Vadhan and Jiapeng Zhang. *A Tight Lower Bound for Entropy Flattening.* 33rd Computational Complexity Conference (CCC 2018) [12]
- Shachar Lovett, Avishay Tal and Jiapeng Zhang. *The robust sensitivity of boolean functions.* Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms Pages 1822-1833 [40]
- Shachar Lovett and Jiapeng Zhang. *On the impossibility of entropy reversal, and its application to zero-knowledge proofs.* Theory of Cryptography Conference Pages 31-55 [43]
- Bo Tang and Jiapeng Zhang. *Barriers to Black-box Constructions of Traitor Tracing Systems.* Theory of Cryptography Conference Pages 3-30 [56]
- Daniel Kane, Shachar Lovett, Shay Moran and Jiapeng Zhang. *Active classification with comparison queries.* IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS) 355-366 [30]
- Shachar Lovett and Jiapeng Zhang. *Noisy population recovery from unknown noise.* Conference on Learning Theory Pages 1417-1431 [42]

- Shachar Lovett and Jiapeng Zhang. *Improved noisy population recovery, reverse Bonami-Beckner inequality for sparse functions.* Proceedings of the forty-seventh annual ACM symposium on Theory of computing, Pages 137-142 [41]

## ABSTRACT OF THE DISSERTATION

Discrete Fourier Analysis and Its Applications

by

Jiapeng Zhang

Doctor of Philosophy in Computer Science

University of California San Diego, 2019

Professor Shachar Lovett, Chair

The topic of discrete Fourier analysis has been extensively studied in recent decades. It plays an important role in theoretical computer science and discrete mathematics. One hand it is interesting to study the structure of boolean functions via discrete Fourier analysis. On the other hand, these structural results also provide a huge number of applications in theoretical computer science, including computational complexity, pseudorandomness, cryptography, learning theory. In this dissertation, we extend some more connections between discrete Fourier analysis and theoretical computer science. In particular, we study the following questions.

- Robust sensitivity of boolean function. In this part, we study the connection between the Fourier tail bound and the sensitivity tail bound of boolean functions, which is an analogue

of the sensitivity conjecture, which was proposed by Nisan [48].

- DNF sparsification. The disjunctive normal form (or DNF) is a widely used representation of boolean functions. It is very interesting to study the structure of DNFs. There are two natural ways to measure the complexity of DNFs, the width and the size. In this thesis, we study a connection between these two measures. We propose a new approach by combing the swithing lemma (a combinatoric tool) and the hypercontrativity inequality (an analytic inequality). This framework does also suggest a new approach to the famous sunflower conjecture.

- Applications in learning theory. In 1989, the first Fourier-based learning algorithms was introduced by a seminar paper of Linial, Mansour and Nisan [37]. Followed by a series of subsequent works, people found that discrete Fourier analysis is powerful to design learning algorithms. One hand sparse Fourier functions are strong enough to approximate a lot of functions, on the other hand sparse Fourier functions are relatively easy to learn. Build on this framework, we give a more efficient algorithm to solve the *population recovery* problem. That is how to recover a unknown distribution from noisy samples.

# Chapter 1

# Introduction

The topic of boolean function analysis studies functions with discrete domain. In the mathematical aspect, it is interesting to study the structure of discrete functions. On the other hand, it has also been found that discrete Fourier analysis has a lot of significant applications in theoretical computer science, such like cryptography [1, 46], complexity [22, 13], combinatorics [35], coding theory [9, 33], learning theory [19, 38, 45, 42, 14].

A major topic in discrete Fourier analysis is to study the Fourier structure of certain boolean functions. Once we have a simple Fourier representation for a function $f$, then we can easily deal with $f$ in algorithmic aspect, such like design a learning algorithm, or construct a pseudorandom generator. For example, Mansour [45] conjectured that any DNF can be approximated by a sparse Fourier function. It looks like a mathematical problem, however once this conjecture has been confirmed, it automatically leads an efficient learning (even agnostic learning) algorithm for DNFs. The research of discrete Fourier analysis builds a bridge between the profound mathematical structures and applications of computer science.

In this thesis, we study three applications of discrete Fourier analysis. 1) we study the connection between Fourier coefficients and the robust sensitivity of boolean functions; 2) we show how Fourier analysis helps to sparsify DNFs; 3) we solve the problem of population recovery by using discrete Fourier analysis.

## 1.1 Our results

### 1.1.1 The robust sensitivity of boolean functions

The *sensitivity* is a common used complexity measure of boolean functions. Intuitively, the sensitivity of a Boolean function $f$ at a given point is the number of coordinates $i$ such that if we flip the $i$'th coordinate, the value of the function changes. The average value of this quantity is exactly the total influence. The sensitivity connects to a lot of complexity measures, such like decision tree complexity, block sensitivity, certificate complexity. Specifically, the sensitivity is upper bounded by all of the measures above. The sensitivity conjecture speculates the other side, i.e., other measures are also upper bounded by the sensitivity. This conjecture was proposed by Nisan [48] in nearly 20 years ago. Despite much research [55, 48, 49, 10, 17, 32, 11, 4, 27, 2, 18, 3, 5, 24, 23, 28, 36, 54, 8] the conjecture remains wide open, where the best upper bounds on the degree are exponential in the sensitivity, and the best separations are quadratic.

As the original sensitivity conjecture seems untractable at the moment, it makes sense to try and relax it. Gopalan, Servedio, Tal and Wigderson [24] showed that functions of bounded sensitivity have most of their Fourier mass supported on low levels of the hypercube. In the same paper, they also speculate that if most of the points have low sensitivity (instead of every point has low sensitivity), then most of the Fourier mass is on small sets, they call this robust sensitivity conjecture.

In a joint work with Lovett and Tal [40], we confirm this conjecture, and our bound is almost tight (up to some constant). Our proof studies a structure of boolean functions with maximum Fourier degree. We show that any boolean function with maximum Fourier degree contains a sensitivity walk that generates every dimension. This structure might have more applications in boolean function analysis, including the original sensitivity conjecture.

### 1.1.2 DNF sparsification

The disjunctive normal form (DNF) is a well-used representation of logical formulae. A DNF is a disjunction of conjunctive clauses; it can also be described as an OR of ANDs. As DNF is a class of well-used formulae, it is natural to study the power of DNFs. In other words, it is important to study the structure of DNFs.

It is well known that Fourier analysis builds a lot of applications to study DNFs. It has been used to design learning algorithms for DNFs [45, 20], and it is also helpful to construct pseudorandom generators for DNFs [13]. In this thesis, we focus on the problem of DNF sparsification. Let $f$ be a DNF. There are two natural complexity measures associates with it: the numbers of clauses, called *size* and denoted $s(f)$; and the maximal number of variables in a clause, called *width* and denoted $w(f)$. It is a folklore result that DNFs of small size can be approximated by DNFs of small width; however the other side is not very clear. Gopalan, Meka and Reingold [21] studied the reverse problem of *DNF sparsification*: can small width DNFs be approximated by DNFs of small size? Their motivation, other than being a natural problem on the structure of DNFs, came from the goal of designing faster deterministic algorithms to approximately count the number of satisfying assignements of a DNF.

Built on the sunflower structure, Gopalan, Meka and Reingold [21] proved that any width-$w$ DNF can be $\varepsilon$-approximated by a DNF of size $(w+\log(1/\varepsilon))^{O(w)}$. They also speculated this bound is not tight, and they conjectured a bound of $(\log(1/\varepsilon))^{O(w)}$. In a joint work with Lovett [44], we prove a new bound of $(1/\varepsilon)^{O(w)}$. We get a worse dependency on $\varepsilon$, however we have a much better dependency on $w$. Instead of using sunflower conjecture, we use a powerful tool from boolean function analysis, that is the hypercontrativity inequality. Our framework is able to break the $w^w$ barrier, which also appears in some related problems, such like Mansour's conjecture and the sunflower conjecture. In a joint work with Lovett and Solomon [39], we prove that a better upper bound DNF sparsification implies a better sunflower upper bound. Thus our framework gives a hint to solve the sunflower conjecture, which has been opened for 60 years.

### 1.1.3 Population recovery

The topic of population recovery was first motivated by learning theory. Consider a database of patients in a hospital, where for each patient the database lists a large number of traits. Researchers are interested in obtaining this database to perform various statistical studies, but due to privacy concerns the database cannot be released. A possible solution (other than deleting identifying parameters of patients, such as their name) is to delete information at random from the database, or even better, add randomness to the information, with the goal that this will maintain the privacy of the original database, but would still provide researchers with useful information. The question is: does this process ensure privacy, or can the original database be recovered (up to its row order) from a lossy or noisy version of it?

The problem of recovery of data from lossy or noisy samples was studied extensively in statistics in the context of continuous distributions, and was introduced to computer science by Kearns et al. [31] who focused on discrete distributions. The problem regained attention recently in a work by Dvir et al. [15], who related it to the problem of learning DNFs from partial information.

Let $k$ be the size of the original database. Kearns et al. [31] gave an algorithm which is exponential in $k$. Wigderson and Yehudayoff [57] developed a framework called "partial identification", and gave an algorithm which runs in time polynomial in $(k^{\log k}, n, 1/\varepsilon)$ for any $\mu > 0$. Moreover, they showed that their framework cannot obtain algorithms running in time better than polynomial in $k^{\log \log k}$. In a joint work with Lovett [41], we develop an alternative framework (built on discrete Fourier analysis), which gives an algorithm running in time polynomial in $k^{\log \log k}$. In chapter 5, we will discuss more details.

# Chapter 2

# The Function Space

In this chapter, we introduce some standard definitions and notations.

## 2.1 Functions on the boolean cube

**Boolean cube.** Let $n \geq 1$ be an integer, denote $[n] = \{1, 2, \ldots, n\}$. We use $\{0,1\}^n$ to denote the set of binary strings with length $n$.

**Definition 1** (Function space)**.** *Let $\mathscr{F} = \{f : \{0,1\}^n \to \mathbb{R}\}$ denote the space of real functions on the boolean cube, with inner product given by*

$$\langle f, g \rangle = 2^{-n} \sum_{x \in \{0,1\}^n} f(x) \cdot g(x).$$

*We shortly denote $\mathbb{E}_x[f(x) \cdot g(x)] := 2^{-n} \sum_{x \in \{0,1\}^n} f(x) \cdot g(x)$.*

**Definition 2.** *We define the $L_2$-norm on $\mathscr{F} = \{f : \{0,1\}^n \to \mathbb{R}\}$ as*

$$\|f\|_2 := \sqrt{\langle f, f \rangle} = \sqrt{\mathop{\mathbb{E}}_x [f(x)^2]}$$

## 2.2 Discrete Fourier transformation

**Definition 3** (Inner product on the boolean cube)**.** *Let $n \geq 1$ be an integer. We also represent every string $x \in \{0,1\}^n$ as an element in $\mathbb{F}_2^n$. We denote $x \oplus y$ as $x + y$ in $\mathbb{F}_2^n$. We define the inner product of $x, y \in \{0,1\}^n$ as*

$$\langle x, y \rangle = \sum_{i \in [n]} x_i \cdot y_i.$$

Let $n \geq 1$ be an integer. For any $S \subseteq [n]$ and $x \in \{0,1\}^n$, we denote by $e_I \in \{0,1\}^n$ the indicator vector for $I$. For $i \in [n]$ we shorthand $e_i = e_{\{i\}}$. We shorthand $\langle x, S \rangle = \langle x, e_S \rangle$ .

**Definition 4** (Fourier coefficient)**.** *Let $f : \{0,1\}^n \to \mathbb{R}$ be a function. For each $S \subseteq [n]$, its associated Fourier coefficient is*

$$\widehat{f}(S) := \mathop{\mathbb{E}}_{x} \left[ f(x) \cdot (-1)^{\langle x, S \rangle} \right]$$

The Fourier degree of $f$ is defined as

$$\deg(f) := \max\{k : \exists S \subseteq [n], (|S| = k) \wedge (\widehat{f}(S) \neq 0)\}$$

**Theorem 5** (Fourier inversion theorem)**.** *Let $f : \{0,1\}^n \to \mathbb{R}$ be a function, then for every $x \in \{0,1\}^n$*

$$f(x) := \sum_{S \subseteq [n]} \widehat{f}(S) \cdot (-1)^{\langle x, S \rangle}$$

**Lemma 5.1** (Parseval's identity)**.** *Let $f : \{0,1\}^n \to \mathbb{R}$ be a function, then*

$$\sum_{S} \widehat{f}(S)^2 = \mathop{E}_{x}[f(x)^2] = \|f\|_2^2$$

## 2.3 The noise operator

In this section, we introduce the noise operator, which is a very useful tool discrete Fourier analysis.

**Definition 6** (Noisy distribution)**.** *Given $x \in \{0,1\}^n$ and a noise parameter $\mu \in [0,1]$, we denote by $\mathbb{N}_\mu(x)$ the distribution over $y \in \{0,1\}^n$, where $\Pr[y_i = x_i] = \frac{1+\mu}{2}$ and $\Pr[y_i \neq x_i] = \frac{1-\mu}{2}$ independently for all $i \in [n]$.*

**Definition 7** (Noise operator)**.** *The noise operator $T_\mu : \mathscr{F} \to \mathscr{F}$ is defined as*

$$(T_\mu f)(x) = \mathop{\mathbb{E}}_{y \sim \mathbb{N}_\mu}[f(y)].$$

**Definition 8** (Stability)**.** *Let $f : \{0,1\}^n \to \{-1,1\}$ be a boolean function. The $\rho$-stability of $f$ is*

$$\mathrm{Stab}_\rho(f) := \mathop{\mathbb{E}}_{x \in \{0,1\}^n, y \sim \mathbb{N}_\rho(x)} [f(x) \cdot f(y)].$$

*It is clear that* $\mathrm{Stab}_\rho(f) = \langle f, T_\rho f \rangle$

**Boolean function approximation.** Let $f, g : \{0,1\}^n \to \{-1,1\}$ be a boolean functions, we say $g$ that $\varepsilon$-approximates $f$ if

$$\Pr_{x \sim \{0,1\}^n}[g(x) \neq f(x)] \leq \varepsilon$$

here $x \sim \{0,1\}^n$ means that $x$ is a uniformly random string on $\{0,1\}^n$

**Influence.** Let $f : \{0,1\}^n \to \{-1,1\}$ be a boolean function. For each $i \in [n]$, we define its *influence* as

$$\mathbb{I}_f(i) := \Pr_{x \sim \{0,1\}^n} [f(x) \neq f(x \oplus e_i)]$$

# Chapter 3

# The Sensitivity of Boolean Functions

In this chapter, we study the notion of *sensitivity*, a well-known parameter to measuare the complexity of boolean functions. Intuitively, the *sensitivity* of $f$ at $x \in \{0,1\}^n$, denoted $s(f,x)$, is the number of neighbours of $x$ in the boolean hypercube where $f$ takes the opposite value. It has been shown there are a lot of connections between the sensitivity and other complexity measures, such like decision tree complexity, real polynomial degree, block sensitivitiy, certificate complexity. All of these measures were motivated from different applications, it is surprisingly to find connections between all of them. In this chapter, we focus on the connection between the sensitivity and the discrete Fourier spectrum. Our study was motived by the *sensitivity conjecture,* which is an important open problem in the area of boolean function analysis.

## 3.1 Sensitivity

**Definition 9** (Sensitivity)**.** *Given a boolean function $f : \{0,1\}^n \to \{-1,1\}$ and $x \in \{0,1\}^n$. The sensitivity of $f$ at $x$ is defined as*

$$s(f,x) = \{i \in [n] : f(x) \neq f(x \oplus e_i)\}$$

*The* sensitivity *of the function $f$ is defined as the maximum sensitivity of a vertex, i.e.,*

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

Intuitively, there should be some connections between the sensitivity and the influence. Formally, we have the following lemma,

**Lemma 9.1.** *Let $f : \{0,1\}^n \to \{-1,1\}$ be a boolean function. Then*

$$\mathop{\mathbb{E}}_{x}[s(x,f)] = \sum_{i} \mathbb{I}_f(i)$$

*Proof.* The proof of the lemma is straightforward. By definition,

$$\mathop{\mathbb{E}}_{x}[s(x,f)] = 2^{-n} \cdot \sum_{x} s(f,x) = 2^{-n} \cdot \sum_{x}\sum_{i} \mathbb{1}_{f(x) \neq f(x \oplus e_i)} = \sum_{i} \mathop{\mathbb{E}}_{x}\left[\mathbb{1}_{f(x) \neq f(x \oplus e_i)}\right]$$

The claim then follows by the definition of influence. □

It has been showed that the sensitivity is upper bounded by the Fourier degree. In specific, we have the following theorem.

**Theorem 10** ([48])**.** *There is a constant $c > 1$ such that for any boolean function $f$,*

$$s(f) \leq deg(f)^c$$

The other side, which called the sensitivity conjecture, was originally proposed by Nisan [48]. The sensitivity conjecture is a central open problem in boolean complexity theory. It speculates that functions of low sensitivity must be "simple". This can be phrased in several equivalent formulations. For our purposes, we focus on the Fourier degree of $f$ (see also [23] for other notions in which low sensitivity functions are simple).

**Conjecture 10.1** (Sensitivity conjecture)**.** *There is a constant $c > 1$ such that for any boolean function $f : \{0,1\}^n \to \{-1,1\}$,*

$$deg(f) \leq s(f)^c$$

Despite much research [55, 48, 49, 10, 17, 32, 11, 4, 27, 2, 18, 3, 5, 24, 23, 28, 36, 54, 8] the sensitivity conjecture remains wide open, where the best upper bounds on the degree are exponential in the sensitivity, and the best separations are quadratic. The survey [27] provides a good account of the conjecture, many of its equivalent formulations and consequences, and the progress so far.

## 3.2 Robust sensitivity

As the original sensitivity conjecture seems untractable at the moment, it makes sense to try and relax it. As usual, we denote by $\widehat{f}(S)$ for $S \subseteq [n]$ the Fourier coefficients of a boolean function $f : \{0,1\}^n \to \{-1,1\}$. Parseval's identity implies that $\sum \widehat{f}(S)^2 = 1$. Gopalan, Servedio, Tal and Wigderson [24] showed that functions of bounded sensitivity have most of their Fourier mass supported on low levels of the hypercube.

**Theorem 11** (Theorem 1.2 in [24]). *Let $f : \{0,1\}^n \to \{-1,1\}$ be a Boolean function. If the sensitivity of $f$ is $s = s(f)$ then for every $\varepsilon > 0$,*

$$\sum_{|S| \geq O(s \log(1/\varepsilon))} \widehat{f}(S)^2 \leq \varepsilon.$$

Observe that there is a disconnect between the assumption and conclusion of Theorem 11 in the following sense. The assumption (bounded sensitivity) is very sensitive to changes in the function $f$. Indeed, changing even one value can increase the sensitivity from 0 to $n$. However, the conclusion (bounded Fourier tail) is not sensitive to small changes in the function. Thus, it makes sense to relax the assumption: instead of assuming that $s(f,x)$ is bounded for *all* $x$, we will assume that it is the case for *most* $x$, and attempt to reach a similar conclusion.

This question was explicitly phrased as an open problem by Gopalan et al. [24]. Given a boolean function $f : \{0,1\}^n \to \{-1,1\}$, consider two distributions over integers $0,\dots,n$:

1. The Fourier distribution of $f$, where one chooses a set $S \subseteq [n]$ with probability $\widehat{f}(S)^2$ and computes its size $|S|$.

2. The sensitivity distribution of $f$, where one chooses a random point $x \in \{0,1\}^n$ and computes its sensitivity $s(f,x)$.

They speculate the robust sensitivity conjecture that if most of the points have low sensitivity, then most of the Fourier mass is on small sets. Formally, this is expressed via corresponding moments of the two distributions.

**Conjecture 11.1** (Conjecture 1.3 in [23]). *For every $d \geq 1$ there exists a constant $a_d$ such that the following holds. For any $n \geq 1$ and any Boolean function $f : \{0,1\}^n \to \{-1,1\}$ it holds that*

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 |S|^d \leq a_d \cdot \mathop{\mathbb{E}}_{x \in \{0,1\}^n} [s(f,x)^d + 1].$$

It is easy to verify that Conjecture 11.1 with a good enough constant $a_d$ (concretely, $a_d = d^d \cdot 2^{O(d)}$) implies Theorem 11, even if we replace the assumption that the maximum sensitivity of $f$ is at most $s$, with the weaker assumption that the $d$-th moment of the sensitivity is at most $s^d$.

In a joint work with Lovett and Tal [40], we proved this conjecture, with near optimal bounds. As we demonstrate below, achieving near optimal bounds is crucial for certain applications, as it allows to tightly relate the Fourier distribution and the sensitivity distribution of boolean functions. The following is our main theorem, which is a slight re-formulation of Conjecture 11.1.

**Theorem 12** ([40]). *Let $f : \{0,1\}^n \to \{-1,1\}$. For any $d \geq 1$ it holds that*

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 |S|^d \leq a_d \cdot \mathop{\mathbb{E}}_{x \in \{0,1\}^n} \left[ s(f,x)^d + 1 \right],$$

*for $a_d = d^{13d} \cdot 2^{O(d)}$.*

We first give below two corollaries of Theorem 12. We note that both are possible only because of the near optimality of our bound on $a_d$ (namely $a_d \leq d^{O(d)}$).

**Exponential sensitivity tails implies exponential Fourier tails.** As a first corollary, we show that if the sensitivity distribution has an exponentially decaying tail, then the same holds for the Fourier distribution. This shows that indeed we can replace the condition of bounded maximal sensitivity with a fast enough decay of the sensitivities.

**Corollary 12.1.** *Let $f : \{0,1\}^n \to \{-1,1\}$. Assume that for some $s \geq 1$, for any $\lambda \geq 1$ the number of nodes $x \in \{0,1\}^n$ for which $s(f,x) \geq \lambda s$ is at most $2^{n-\lambda}$. Then for any $\varepsilon > 0$ it holds that*

$$\sum_{|S| \geq O(s \log^{14}(1/\varepsilon))} \widehat{f}(S)^2 \leq \varepsilon \qquad \forall \varepsilon > 0.$$

**High degree nodes in induced subgraphs of the hypercube.** The second corollary is in graph theory, and involves induced subgraphs of the hypercube. For $A \subset \{0,1\}^n$ let $G[A]$ denote the induced sub-graph of the hypercube $\{0,1\}^n$ on the vertices in $A$. Let $A^c = \{0,1\}^n \setminus A$ denote the complement of $A$.

Gotsman and Linial [25] proved that the sensitivity conjecture is equivalent to the following conjecture: if $|A| \neq 2^{n-1}$ then either in $G[A]$ or in $G[A^c]$, there is a node whose degree is at least $n^c$ for some absolute constant $c > 0$. Below we note that Theorem 12 immediately implies this if we assume that $|A| \geq (1+\varepsilon)2^{n-1}$ for exponentially small $\varepsilon$. In fact, there are many such nodes. Observe that this beats the naive averaging argument, which requires that $\varepsilon \geq 1/n$.

**Corollary 12.2.** *Let $A \subset \{0,1\}^n$ of size $|A| \geq (1+\varepsilon)2^{n-1}$. Then either in $G[A]$ or in $G[A^c]$, there exist $\varepsilon^{O(1)}2^n$ vertices whose degree is at least $\Omega(n/\log^{13}(1/\varepsilon))$.*

The proofs of Corollaries 12.1 and 12.2 is given in Section 3.5.

**Conjectured optimal parameters.** We conjecture that the bound on $a_d$ in Theorem 12 can be improved to $a_d \leq d^d \cdot 2^{O(d)}$. If so, this will imply the strongest quantitative form of Conjecture 11.1, as an example in [24] shows that necessarily $a_d \geq d^{d(1-o(1))}$. We note that under this conjecture, Corollaries 12.1 and 12.2 improve. Concretely:

- Corollary 12.1, under the same assumption, would give that $\sum_{|S|\geq O(s\log^2(1/\varepsilon))}\widehat{f}(S)^2 \leq \varepsilon$.
- Corollary 12.2, under the same assumption, would give that either in $G[A]$ or in $G[A^c]$, there exist $\varepsilon^{O(1)}2^n$ vertices whose degree is at least $\Omega(n/\log(1/\varepsilon))$.

We denote by $\mathscr{H}_n$ the $n$-dimensional hypercube, whose vertices are $V(\mathscr{H}_n) = \{0,1\}^n$ and edges are $E(\mathscr{H}_n) = \{(x, x\oplus e_i) : x \in \{0,1\}^n, i \in [n]\}$. Given two vectors $x,y \in \{0,1\}^n$, we shorthand $x+y$ for $x\oplus y$ whenever the context is clear. In particular, edges of the hypercube are written as $(x, x+e_i)$. We say that an edge $(x, x+e_i)$ has direction $i$.

**Chapter organization.** In Section 3.3 we prove a weak form of Theorem 12, where the bound on $a_d$ is $a_d = 2^{O(d^2)}$ instead of $a_d = d^{O(d)}$. Although this bound is insufficient for the applications described in Corollary 12.1 and Corollary 12.2, it is useful to build intuition, as the proof is significantly simpler than the proof of Theorem 13. The actual proof of Theorem 13 appears in Section 3.4. We prove Corollaries 12.1 and 12.2 in Section 3.5. We discuss open problems in Section 3.6.

## 3.3 A weak form of the main theorem

First, we rephrase Theorem 13 in terms of $\binom{|S|}{d}$ instead of $|S|^d$. Using the fact that for any $d$ (even $d > |S|$), we have $|S|^d \leq d^d \cdot \left(\binom{|S|}{d} + 1\right)$, it is enough to prove the following.

**Theorem 13.** *Let $f : \{0,1\}^n \to \{-1,1\}$. For any $d \geq 1$ it holds that*

$$\sum_{S\subseteq[n]} \widehat{f}(S)^2 \binom{|S|}{d} \leq a_d \cdot \mathop{\mathbb{E}}_{x\in\{0,1\}^n}\left[s(f,x)^d\right],$$

*where $a_d \leq d^{12d} \cdot 2^{O(d)}$.*

We prove Theorem 13 in Section 3.4. In this section, we prove a weak form of Theorem 13 with $a_d = d^{O(d^2)}$ instead of $a_d = d^{O(d)}$. While these bounds are insufficient for the applications in Corollaries 12.1 and 12.2, it would be instructive in order to build intuition.

**Theorem 14** (Weak form of Theorem 13)**.** *Let $f : \{0,1\}^n \to \{-1,1\}$. For any $d \geq 1$ it holds that*

$$\sum_{S \subseteq [n]} \widehat{f}(S)^2 \binom{|S|}{d} \leq a_d \cdot \mathop{\mathbb{E}}_{x \in \{0,1\}^n} \left[ s(f,x)^d \right]$$

*where $a_d \leq 2^{\binom{d}{2}+d}$.*

We prove Theorem 14 in the reminder of this section. The first step is to replace the Fourier moments with a more combinatorial expression.

### 3.3.1 Fourier moments and max degree cubes

**Definition 15** (Sub-cubes)**.** *For $v \in \{0,1\}^n$ and $I \subset [n]$ let*

$$C(v,I) := \{x \in \{0,1\}^n : x_i = v_i \ \forall i \notin I\}$$

*denote a sub-cube. The dimension of the sub-cube is $|I|$. Note that $C(v,I) = C(v',I)$ for all $v' \in C(v,I)$. We denote by $\mathscr{C}(n,d)$ the set of all $d$-dimensional cubes in $\{0,1\}^n$.*

Given $C = C(v,I) \in \mathscr{C}(n,d)$, the restriction of $f : \{0,1\}^n \to \{-1,1\}$ to $C$ is $f|_C : \{0,1\}^I \to \{-1,1\}$ given by $f|_C(x) = f(y)$ where $y_i = x_i$ for $i \in I$ and $y_i = v_i$ for $i \notin I$. We say that $f|_C$ has *max degree* if its degree as a multilinear real polynomial over $\{x_i : i \in I\}$ is maximal, namely $d$. This is equivalent to $\widehat{f|_C}(I) \neq 0$.

The following lemma connects the Fourier moments of $f$ and the number of maximal degree cubes in $f$. It appears in a slightly different formulation as Theorem 3.2 in [24].

**Lemma 15.1.** *Let $f : \{0,1\}^n \to \{-1,1\}$. Fix $d \geq 1$. Define*

$$A := 2^n \sum_{S \subseteq [n]} \widehat{f}(S)^2 \binom{|S|}{d}$$

*and*

$$B := |\{C \in \mathscr{C}(n,d) : f|_C \text{ has max degree}\}|.$$

*Then*

$$2^{-d}B \leq A \leq 2^{d}B.$$

*Proof.* For a function $g : \{0,1\}^n \to \mathbb{R}$ define its directional derivative in direction $i \in [n]$ as $\Delta_i g : \{0,1\}^n \to \mathbb{R}$ given by $\Delta_i g(x) = g(x+e_i) - g(x)$. For a set of directions $I = \{i_1,\dots,i_d\}$ the iterated derivative is defined as

$$\Delta_I f(x) = (\Delta_{i_1} \dots \Delta_{i_d} f)(x) = \sum_{J \subseteq I} (-1)^{|I|-|J|} f(x+e_J).$$

In particular, the iterative derivative does not depend on the order of $i_1,\dots,i_d$, making $\Delta_I f$ well defined. Define

$$T := \{(x,I) : x \in \{0,1\}^n, I \subset [n], |I| = d, \Delta_I f(x) \neq 0\}.$$

We will see that $|T|$ is directly related to $B$, while $A$ is related to the expression

$$\sum_{(x,I) \in T} (\Delta_I f(x))^2.$$

We first show that $B = 2^{-d}|T|$. To see that, fix a $d$-dimensional cube $C = C(v,I)$ and consider $f|_C$. Note that $\Delta_I f(v)$ is the sum with alternating signs of the points of $C$. In particular, if we let $f|_C(x) = \sum_{J \subseteq I} \widehat{f|_C}(J)(-1)^{\langle x,e_J \rangle}$ be the Fourier decomposition of $f|_C$, then

$$\Delta_I f(v) = \pm 2^d \cdot \widehat{f|_C}(I).$$

(the sign can be computed explicitly as $(-1)^{\langle v,e_I \rangle}$, but we don't need it). In particular, $f|_C$ has max degree iff $\Delta_I f(v) \neq 0$; namely exactly when $(v,I) \in T$. As this holds for any $v' \in C$ we have that

$$2^d B = |T|.$$

Next we relate $T$ to $A$. To that end, we explore the effect of derivatives on the Fourier decomposition. It is easy to see that the Fourier decomposition of $\Delta_i f$ is

$$\Delta_i f(x) = 2 \sum_{S \subseteq [n]: i \in S} \widehat{f}(S)(-1)^{\langle x, e_S \rangle}.$$

Applying this iteratively for $I \subset [n]$ of size $|I| = d$ gives

$$\Delta_I f(x) = 2^d \sum_{S \subseteq [n]: I \subseteq S} \widehat{f}(S)(-1)^{\langle x, e_S \rangle}.$$

Thus we have

$$\sum_{x \in \{0,1\}^n} (\Delta_I f(x))^2 = 2^n \cdot 2^{2d} \sum_{S \subseteq [n]: I \subseteq S} \widehat{f}(S)^2.$$

Summing over all sets $I$ with $|I| = d$, and restricting to $(x, I) \in T$ (otherwise by definition $\Delta_I f(x) = 0$ contributes nothing to the sum) gives

$$\sum_{(x,I) \in T} (\Delta_I f(x))^2 = 2^n \cdot 2^{2d} \sum_S \widehat{f}(S)^2 \binom{|S|}{d} = 2^{2d} A.$$

To conclude, note that whenever $(x, I) \in T$ then $1 \leq (\Delta_I f(x))^2 \leq 2^{2d}$, where the lower bound follows from $\Delta_I f(x)$ being a nonzero integer, and the upper bound from the fact that $\Delta_I f(x)$ is the sum with alternating signs of $2^d$ evaluations of a Boolean function $f$. Thus

$$2^d B = |T| \leq \sum_{(x,I) \in T} (\Delta_I f(x))^2 \leq 2^{2d} |T| = 2^{3d} B$$

and hence

$$2^{-d} B \leq A \leq 2^d B.$$

□

### 3.3.2 Sensitivity graph and related notions

Let $f:\{0,1\}^n \to \{-1,1\}$. Given Lemma 15.1, we focus on bounding the number of $d$-dimensional cubes $C$ such that $f|_C$ has max degree. The following definitions are from [24]. We define two notions of "sensitive edges" for edges of the hypercube. The first is sensitivity with respect to a boolean function defined on the hypercube; the second is sensitivity with respect to a path in the hypercube. We would mainly be interested in situations when the two coincide.

**Definition 16.** *Let $f:\{0,1\}^n \to \{-1,1\}$. The sensitivity graph $G_f$ of $f$ is the sub-graph of $\mathscr{H}_n$ whose edges are*

$$E(G_f) := \{(x, x+e_i) : x \in \{0,1\}^n, i \in [n], f(x) \neq f(x+e_i)\}.$$

*Edges of $G_f$ are called "sensitive edges" of $\mathscr{H}_n$ with respect to $f$.*

**Definition 17.** *Let $P$ be a walk (i.e. a path) in $\mathscr{H}_n$, whose vertices are $v_0, v_1, \ldots, v_m \in \{0,1\}^n$. Let $i_1, \ldots, i_m \in [n]$ be the directions of the edges of $P$, namely $v_i = v_{i-1} + e_i$. An edge $(v_j, v_{j+1})$ is said to be a* leading edge *of the walk if there is no $j' < j$ for which $i_{j'} = i_j$. Namely, the edge $(v_j, v_{j+1})$ is the first edge in the walk in direction $i_j$. In such a case, we also say that $v_j$ is a* sensitive node. *We further define:*

- *Sensitive nodes of $P$: $V(P) = (v_{j_1}, \ldots, v_{j_d})$.*
- *Sensitive directions of $P$: $I(P) = (i_{j_1}, \ldots, i_{j_d})$.*
- *Dimension of $P$: $\dim(P) = |V(P)| = |I(P)|$.*

**Definition 18** (Walk sensitive for a function)**.** *Let $f:\{0,1\}^n \to \{-1,1\}$. A walk $P$ in $\mathscr{H}_n$ is sensitive for $f$ if the sensitive edges of $P$ are also sensitive edges for $f$.*

**Definition 19** (Proper walk)**.** *Let $f:\{0,1\}^n \to \{-1,1\}$ and $1 \leq d \leq n$. A proper walk $P$ with respect to $f$, of dimension $d$, is given by:*

- *Its sensitive nodes $V(P) = (v_1, \ldots, v_d)$, where $v_1, \ldots, v_d \in \{0,1\}^n$.*

- *Its sensitive directions $I(P) = (i_1, \ldots, i_d)$, where $i_1, \ldots, i_d \in [n]$ are distinct.*

*Such that they satisfy:*

- *$f(v_j) \neq f(v_j + e_{i_j})$ for $j = 1, \ldots, d$.*

- *$v_j \in C(v_1, \{i_1, \ldots, i_{j-1}\})$ for $j = 2, \ldots, d$.*

*A proper walk can be extended to a walk in $\mathscr{H}_n$ with sensitive nodes $V(P)$ and sensitive directions $I(P)$, by connecting each $v_j$ to $v_{j+1}$ using some shortest walk. By definition, this part of the walk will only use edges with directions in $\{i_1, \ldots, i_j\}$. The resulting walk is sensitive for $f$.*

Given a proper walk $P$ with $V(P) = (v_1, \ldots, v_d)$ and $I(P) = (i_1, \ldots, i_d)$, we say that it *realizes* the sub-cube $C(P) := C(v_1, I(P))$. Equivalently, $C(P)$ is the minimal sub-cube which contains all the edges $(v_j, v_j + e_{i_j})$.

### 3.3.3 Proper walks in maximal degree cubes

Let $f : \{0,1\}^n \to \{-1,1\}$. Gopalan et al. [24] proved that if $f|_C$ has maximal degree, then $C$ is realized by some proper walk (in fact, they prove that there exists such a proper walk with a succinct description, which allows for better quantitative bounds; for now, we ignore this aspect, and re-inspect it in Section 3.4). We will ask for a proper walk where the first node has maximal sensitivity.

**Definition 20** (First-maximal proper walk)**.** *Let $P$ be a proper walk with respect to $f$, with sensitive nodes $V(P) = (v_1, \ldots, v_d)$. We say that $P$ is* first-maximal *if $s(f, v_1) \geq s(f, v_i)$ for all $i = 2, \ldots, d$.*

**Lemma 20.1.** *Let $f : \{0,1\}^n \to \{-1,1\}$, $C \in \mathscr{C}(n,d)$ such that $f|_C$ has maximal degree $d$. Then $C$ is realized by a first-maximal proper walk with respect to $f$.*

*Proof.* Let $g = f|_C$. For a sensitive edge $(x,x')$ for $g$, define its weight as

$$w(x,x') = \max(s(f,x), s(f,x')).$$

We will prove that there exists a $d$-dimensional proper walk $P$ with respect to $g$, with sensitive nodes $V(P) = (v_1, \dots, v_d)$ and sensitive directions $I(P) = (i_1, \dots, i_d)$, such that

$$w(v_1, v_1 + e_{i_1}) \geq w(v_2, v_2 + e_{i_2}) \geq \dots \geq w(v_d, v_d + e_{i_d}).$$

We first observe that this suffices for the lemma. We may assume that $s(f,v_1) \geq s(f, v_1 + e_{i_1})$, as otherwise we can set the starting point to be $v_1 + e_{i_1}$ without changing any of the properties of the proper walk. Then by design for every $j = 2, \dots, d$ we have

$$s(f,v_1) = w(v_1, v_1 + e_{i_1}) \geq w(v_j, v_j + e_{i_j}) \geq s(f, v_j).$$

Next, we prove the existence of such a walk by induction on $d$. For $d = 1$ this is obvious, so assume $d \geq 2$. Let $(y,y')$ be a sensitive edge in $G_g$ with minimal weight $w(y,y')$. Assume that $y' = y + e_\ell$. If $g$ has maximal degree $d$, then at least one of the restrictions $g|_{x_\ell=0}$ or $g|_{x_\ell=1}$ must have maximal degree $d-1$ in their respective sub-cube. Assume without loss of generality that this holds for $g|_{x_\ell=0}$ and that $y_\ell = 0$. By induction there is a proper walk with the required conditions, realizing the sub-cube $\{x : x_\ell = 0\}$ of dimension $d-1$, given by sensitive nodes $v_1, \dots, v_{d-1}$ and sensitive directions $i_1, \dots, i_{d-1}$. To complete the walk we set $v_d = y$ and $i_d = \ell$. □

### 3.3.4 Putting it together

Let $f : \{0,1\}^n \to \{-1,1\}$. By Lemma 20.1, in any $d$-dimensional sub-cube $C$ where $f|_C$ has maximal degree, we can find a first-maximal proper walk realizing it. Thus, instead of counting maximal degree sub-cubes, we will count first-maximal proper walks.

**Claim 20.1.** *The number of $d$-dimensional first-maximal proper walks in $G_f$, which start at a given node $x$, is at most*

$$2^{\binom{d}{2}} s(f,x)^d.$$

*Proof.* We wish to count $d$-dimensional proper walks $P$ with respect to $f$. Let $V(P) = (v_1,\dots,v_d)$ and $I(P) = (i_1,\dots,i_d)$. We assume $v_1 = x$, hence there are $s(f,x)$ possible values for $i_1$. Given that we already defined $v_1,\dots,v_{j-1}$ and $i_1,\dots,i_{j-1}$, we have by assumption that $v_j \in C(v_1,\{i_1,\dots,i_{j-1}\})$, and hence it has at most $2^{j-1}$ different possibilities. Given a choice of $v_j$, the number of choices for $i_j$ is at most $s(f,v_j) \le s(f,x)$. Thus we can bound the number of such walks by

$$2^{1+2+\ldots+d-1} \cdot s(f,x)^d = 2^{\binom{d}{2}} s(f,x)^d.$$

□

We complete the proof of Theorem 14 below.

*Proof of Theorem 14.* Let $A = 2^n \sum_{S\subseteq[n]} \widehat{f}(S)^2 \binom{|S|}{d}$, $B = |\{C \in \mathscr{C}(n,d) : f|_C \text{ has max degree}\}|$ and $D = \sum_{x\in\{0,1\}^n} s(f,x)^d$. By Lemma 15.1 we have $A \le 2^d B$. By Lemma 20.1 we can bound $B$ by the number of $d$-dimensional first-maximal proper walks with respect to $f$, and by Claim 20.1 this number is bounded by $2^{\binom{d}{2}} D$. Thus

$$2^n \sum_{S\subseteq[n]} \widehat{f}(S)^2 \binom{|S|}{d} \le 2^d B \le 2^{\binom{d}{2}+d} \sum_{x\in\{0,1\}^n} s(f,x)^d.$$

The theorem follows by dividing both sides by $2^n$. □

## 3.4 Proof of the main theorem

We prove Theorem 13 in this section. We will follow the proof of Theorem 14 and make careful modifications and optimizations, that would allow us to improve the bound from the weak bound of $a_d = d^{O(d^2)}$ to the near optimal bound of $a_d = d^{O(d)}$.

A keen reader can see that the main reason for the loss of parameters in Theorem 14 is the number of potential first-maximal proper walks in a max degree function, which we naively bounded by $2^{\binom{d}{2}}$. In order to obtain a better bound, we need to define more carefully what do we mean by a "description" of a proper walk. This notion was studied implicitly in [24] (see Lemma 5.5), and we define it here explicitly.

**Definition 21** (Signature of a walk)**.** *Let $P$ be a $d$-dimensional walk in $\{0,1\}^n$. Let $V(P) = (v_1,\dots,v_d)$ and $I(P) = (i_1,\dots,i_d)$. By construction, we have $v_{j+1} \in C(v_1,\{i_1,\dots,i_j\})$ for all $j = 1,\dots,d-1$. This means that there exists $r_{i,j} \in \{0,1\}$ such that*

$$v_{j+1} = v_1 + r_{j,1} \cdot e_{i_1} + \ldots + r_{j,j} \cdot e_{i_j}.$$

*The* signature *of $P$ is*

$$R(P) = (r_{i,j} : 1 \le i \le j \le d-1) \in \{0,1\}^{\binom{d}{2}}.$$

We next define when a family of walks has a succinct description.

**Definition 22** (Signature of a family of walks)**.** *Let $\mathscr{P}$ be a family of walks in $\{0,1\}^d$. The signatures of $\mathscr{P}$ are*

$$R(\mathscr{P}) = \{R(P) : P \in \mathscr{P}\} \subset \{0,1\}^{\binom{d}{2}}.$$

*If $|R(\mathscr{P})| \le 2^b$ then we say that $\mathscr{P}$ can be described using $b$ bits.*

We also need to extend the notion of first-maximal proper walks, in a way that breaks the relation between the sub-cube and the global sensitivity of the function on $\mathscr{H}_n$.

**Definition 23.** *Let $P$ be a $d$-dimensional walk whose sensitive nodes are $V(P) = (v_1,\dots,v_d)$. Let $w : \{0,1\}^d \to \mathbb{R}$ be some weight function on the nodes of the hypercube. We say that $P$ is* first-maximal with respect to *$w$ if $w(v_1) \ge w(v_i)$ for all $i = 2,\dots,d$.*

In the applications we will use $g = f|_C$ with weight function $w(x) = s(f,x)$. However, making the general definition allows to focus on the restricted function $f|_C$ and forget about the function $f$. The following definition isolates our notion of "efficient description" of a first-maximal proper walks.

**Definition 24.** *Fix $d \geq 1$. We say that first-maximal proper walks in $d$ dimensions can be described using $b$ bits if the following holds. For any function $g : \{0,1\}^d \to \{-1,1\}$ of maximal degree $d$, and any weight function $w : \{0,1\}^d \to \mathbb{R}$, there exist a $d$-dimensional walk $P_{g,w}$ which is proper with respect to $g$, and first-maximal with respect to $w$, such that the family*

$$\mathscr{P}_{proper,\, first-maximal} := \{P_{g,w}\}$$

*can be described using $b$ bits (recall Definition 22).*

One can verify that Lemma 20.1 can be extended to an arbitrary weight function. Thus, it establishes that first-maximal proper walks in $d$ dimensions can be described using $\binom{d}{2}$ bits. This motivates the question of looking for the minimal such description length. This is further motivated by the following lemma.

**Lemma 24.1.** *Assume that first-maximal proper walks in dimension $d$ can be described using $b$ bits. Then Theorem 13 holds with the bound $a_d = 2^{d+b}$.*

*Proof.* Let $\mathscr{R} \subset \{0,1\}^{\binom{d}{2}}$ be a set of size $|\mathscr{R}| \leq 2^b$, such that for any function $g : \{0,1\}^d \to \{-1,1\}$, and any weight function $w : \{0,1\}^d \to \mathbb{R}$, there exists a $d$-dimensional walk $P_{g,w}$ which is proper with respect to $g$, and first-maximal with respect to $w$, such that $R(P_{g,w}) \in \mathscr{R}$.

The only change needed in the proof of Theorem 13 is in Claim 20.1, where instead of allowing for an arbitrary first-maximal proper walk, we only allow for walks $P$ for which $R(P) \in \mathscr{R}$. Thus the number of first-maximal proper walks starting at node $x$ can be bounded by $s(f,x)^d|\mathscr{R}|$ and the rest of the proof remains as is. $\square$

Gopalan et al. [24] proved (Lemma 5.5) that if we remove the requirement that the walk is first-maximal, then proper walks can be described using $4d$ bits. However, their proof does not give the first-maximal condition, which is why their proof only works assuming a bound on the maximal sensitivity of $f$. We conjecture that such a bound can be obtained also with the first-maximal condition.

**Conjecture 24.1.** *For any $d \geq 1$, first-maximal proper walks in dimension $d$ can be described using $O(d)$ bits.*

Conjecture 24.1 would give optimal bounds in Theorem 13. Below, we give a nearly tight bound.

**Theorem 25.** *For any $d \geq 1$, first-maximal proper walks in dimension $d$ can be described using $12d\log d$ bits.*

Theorem 13 follows immediately from Theorem 25 and Lemma 24.1. Below, we give the details necessary to prove Theorem 25. We start with some more definitions from [24].

### 3.4.1 Sensitive trees

Let $g : \{0,1\}^d \to \{-1,1\}$. Its corresponding sensitivity graph is $G_g$. We will generally assume that $g$ has max degree, although the following statements also follow from a weaker assumption that $g$ has maximal decision tree depth $d$.

**Definition 26** (Sensitive tree)**.** *Let $g : \{0,1\}^d \to \{-1,1\}$. A sensitive tree for $g$ is a sub-tree $T$ of $G_g$ such that all edges of $T$ have distinct directions. We denote by $V(T)$ the nodes of $T$, by $I(T)$ the directions of the edges of $T$, and by $C(T)$ the minimal sub-cube that contains $T$.*

The following claim is Lemma 5.3 in [24], which shows how to get a proper walk from a sensitive tree. It also shows that such walks can be succinctly described.

**Claim 26.1** (Proper walk from a sensitive tree)**.** *Let $g : \{0,1\}^d \to \{-1,1\}$, and let $T$ be a sensitive tree for $g$. Then for every $v \in V(T)$ there exists a proper walk $P = P_{tree}(v;T)$ with respect to $g$, such that $v$ is the first node in $P$, $V(P) \subseteq V(T)$ and $I(P) = I(T)$. Furthermore, let*

$$\begin{aligned}\mathscr{P}_{tree} := \{P_{tree}(v;T) :& g : \{0,1\}^d \to \{-1,1\},\\ & T \text{ sensitive tree for } g,\ v \in V(T)\}.\end{aligned}$$

*Then $\mathscr{P}_{tree}$ can be described using $2d$ bits.*

*Proof.* Given a sensitive tree $T$ with respect to $g$, consider the walk obtained by performing a depth first search on $T$ starting at $v$. This gives the required proper walk. To analyze the signatures of $\mathscr{P}_{tree}$, note that if $T$ is a tree with $k$ edges, then a depth first search in $T$ is a walk of length $2k$ which can be described as a sequence of length $2k$ with two types of operations: “follow next sensitive edge” or “backtrack”. Moreover, there are exactly $k$ of each type. This determines the signature of the walk. Thus the total number of different signatures in $\mathscr{P}_{tree}$ is at most

$$|R(\mathscr{P}_{tree})| \le \sum_{k=1}^{d} \binom{2k}{k} \le 2^{2d}.$$

□

**Definition 27** (Shifting a sensitive tree)**.** *Let $g : \{0,1\}^d \to \{-1,1\}$ and let $T$ be a sensitive tree for $g$. We say that $T$ can be shifted in direction $J \subseteq [d]$, where $J \cap I(T) = \emptyset$, if $f(x) = f(x+e_J)$ for all nodes $x$ of $T$. In such a case, we denote by $T+e_J$ the tree obtained by shifting all nodes and edges of $T$ by $e_J$. Observe that $T+e_J$ is also a sensitive tree for $g$.*

**Definition 28.** *Let $g : \{0,1\}^d \to \{-1,1\}$ and let $T$ be a sensitive tree for $g$. Let $I \subset [d]$ disjoint from $I(T)$. We say that $T$ is invariant to shifts supported on directions $I$, if for any $J \subseteq I$ we can shift $T$ in direction $J$. Equivalently, if $f(x) = f(x+e_J)$ for all $x \in V(T)$ and all $J \subseteq I$. In the case that $I = [d] \setminus I(T)$ we say that $T$ is* maximally invariant to shifts.

The following claim is essentially Lemma 4.6 in [24].

**Claim 28.1.** *Let $g : \{0,1\}^d \to \{-1,1\}$. Let $T$ be a sensitive tree with respect to $g$. Let $I \subset [d]$ disjoint from $I(T)$. Then there exists $I' \subseteq I$, and a sensitive tree $T'$ with respect to $g$, such that the following holds:*

- *$I(T') = I(T) \cup I'$.*
- *$T'$ is invariant to shifts supported on directions $I \setminus I'$.*
- *There exists $J \subseteq I'$ such that $T + e_J$ is a sub-tree of $T'$.*

*Proof.* We build $T'$ greedily. Set initially $T' = T$ and $I' = \emptyset$. If $T'$ is invariant to shifts supported on $I \setminus I'$, we are done. Otherwise, let $J$ be minimal such that $g(v+e_J) \neq g(v)$ for some $v \in V(T')$. Choose some arbitrary $j \in J$. By assumption $T'$ can be shifted in direction $J \setminus \{j\}$, so set $T' = T' + e_{J\setminus\{j\}}$. Now, there exists some $v \in V(T')$ for which $g(v) \neq g(v+e_j)$. Thus, we can add a new sensitive edge $(v, v+e_j)$ to $T'$, and add $j$ to $I'$. Repeat this process until it terminates. □

Let $v \in \{0,1\}^d$. We say that a sensitive tree $T$ agrees with $v$ on coordinates $I \subset [n]$, where $I \cap I(T) = \emptyset$, if $v_i = x_i$ for all $x \in C(T)$ and all $i \in I$. Note that if a sensitive tree $T$ is invariant to shifts supported on directions $I$, then for any $v$ there exists some shift $T' = T + e_J$ for $J \subseteq I$ such that $T'$ agrees with $v$ on $I$.

### 3.4.2 Sensitive tree chains

**Definition 29** (Sensitive tree chain)**.** *Let $g : \{0,1\}^d \to \{-1,1\}$. A sequence of sensitive trees $T_1, \dots, T_m$ with respect to $g$ is called a* sensitive tree chain *if for each $i = 2, \dots, m$, $V(T_i) \cap C(T_{i-1})$ is nonempty. We define $V(T_1, \dots, T_m) := V(T_1) \cup \dots \cup V(T_m)$ and $I(T_1, \dots, T_m) := I(T_1) \cup \dots \cup I(T_m)$.*

Note that if $T_1, \dots, T_m$ is a sensitive tree chain with respect to $g$, then so is any subsequence. Namely, for any $i \leq j$ we have that $T_i, \dots, T_j$ is also a sensitive tree chain with respect to $g$.

**Claim 29.1.** *Let $g : \{0,1\}^d \to \{-1,1\}$. Let $T_1,\ldots,T_m$ be a sensitive tree chain for g. For every $v \in V(T_1)$ there exists a proper walk $P = P_{chain}(v;T_1,\ldots,T_m)$ with respect to g, such that v is the first node in P, $V(P) \subseteq V(T_1,\ldots,T_m)$ and $I(P) = I(T_1,\ldots,T_m)$.*

*Proof.* Let $v_1 = v$, and for $i > 1$ fix some $v_i \in V(T_i) \cap C(T_{i-1})$. Consider the following walk: start with a tree walk $P_{tree}(v_1;T_1)$, which traverses $T_1$ and starts and ends with $v_1$. Then choose a shortest path from $v_1$ to $v_2$, which by assumption only uses directions in $I(T_1)$. Proceed with a tree walk $P_{tree}(v_2;T_2)$, which traverses $T_2$ and starts and ends with $v_2$. Then choose a shortest path from $v_2$ to $v_3$, which by assumption only uses directions in $I(T_2)$. Iterate this procedure until we cover all trees. □

**Definition 30** (Disjoint sensitive tree chain)**.** *Let $g : \{0,1\}^d \to \{-1,1\}$. Let $T_1,\ldots,T_m$ be a sensitive tree chain with respect to g. It is said to be* disjoint *if $I(T_1),\ldots,I(T_m)$ are pairwise disjoint.*

Gopalan et al. [24] proved that for any function of maximal degree, there exists a disjoint sensitive tree chain which cover all directions.

**Lemma 30.1** (Lemma 5.2 in [24])**.** *Let $g : \{0,1\}^d \to \{-1,1\}$ of maximal degree. There exists a disjoint sensitive tree chain $T_1,\ldots,T_m$ with respect to g, such that $I(T_1,\ldots,T_m) = [d]$.*

Gopalan et al. [24] also showed that for these disjoint sensitive tree chains, their corresponding proper walks can be descried using $4d$ bits.

**Lemma 30.2** (Lemma 5.5 in [24])**.** *Define*

$$\begin{aligned}\mathscr{P}_{disjoint} := \{P(v;T_1,\ldots,T_m) \mid g : \{0,1\}^d \to \{-1,1\},& \\ v \in V(T_1),\ T_1,\ldots,T_m& \\ \textit{disjoint sensitive tree}& \\ \textit{chain for } g\}.&\end{aligned}$$

*Then $\mathscr{P}_{disjoint}$ can be described using $4d$ bits.*

*Proof sketch.* We show that $O(d)$ bits are enough, where with some optimizations this can be made $4d$. Let $d_i = \dim(T_i)$ where by the disjointness assumption $\sum d_i \leq d$. Fix $v_1 \in V(T_1)$ and $v_i \in V(T_i) \cap C(T_{i-1})$. Each walk $P_{tree}(v_i; T_i)$ can be encoded using $2d_i$ bits, as we saw in Claim 26.1. The shift from $v_i \in V(T_i)$ to $v_{i+1} \in V(T_{i+1})$ can be encoded using additional $d_i$ bits. In addition, we need symbols to denote when a description of a tree starts and ends, and when the description of a shift starts and ends. Each of these is repeated at most $d$ times. □

The main problem with tree chains $T_1, \ldots, T_m$ is that they allow to "move" only in one direction, that is following the sequence $T_1, T_2, \ldots, T_m$, but not in the reverse direction. In the next section, we introduce reversible tree chains, which allow to move in both directions. These will turn out to be crucial for the purpose of designing first-maximal proper walks.

### 3.4.3 Reversible tree chains

Up until now, we relied on the definition of [24]. In this section, we give several new definitions for combinatorial sub-structures of the hypercubes, which our improved bound hinges upon.

Given trees $T_1, \ldots, T_m$, we define by $C(T_1, \ldots, T_m)$ the smallest sub-cube that contains all their edges. The following definition is a weak form of a sensitive tree chain, that will be important for us.

**Definition 31** (Weak sensitive tree chain). *Let $g : \{0,1\}^d \to \{-1,1\}$. A sequence of sensitive trees $T_1, \ldots, T_m$ with respect to $g$ is called a* weak sensitive tree chain *if for each $i = 2, \ldots, m$, $V(T_i) \cap C(T_1, \ldots, T_{i-1})$ is nonempty (as opposed to $V(T_i) \cap C(T_{i-1}) \neq \emptyset$ in Definition 29).*

**Claim 31.1.** *Let $g : \{0,1\}^d \to \{-1,1\}$. Let $T_1, \ldots, T_m$ be a weak sensitive tree chain for $g$. For every $v \in V(T_1)$ there exists a proper walk $P = P_{weak-chain}(v; T_1, \ldots, T_m)$ with respect to $g$, such that $v$ is the first node in $P$, $V(P) \subseteq V(T_1, \ldots, T_m)$ and $I(P) = I(T_1, \ldots, T_m)$.*

*Proof.* The proof is identical to that of Claim 29.1, except that after the traversal on $T_i$ we may change coordinates in $I(T_1) \cup \ldots \cup I(T_i)$ to get to $T_{i+1}$ (as opposed to just changing the coordinates in $I(T_i)$, as done in Claim 29.1). □

**Definition 32** (Reversible sensitive tree chain)**.** *Let $g : \{0,1\}^d \to \{-1,1\}$. A reversible sensitive tree chain for $g$ is comprised of:*

- *A disjoint sensitive tree chain $T_1, \ldots, T_m$ where $I(T_1, \ldots, T_m) = [d]$.*
- *A weak sensitive tree chain $T'_m, \ldots, T'_1$ (in this order!) where $I(T'_m, \ldots, T'_1) = [d]$.*

*Such that*

- *Each $T_i$ is a sub-tree of $T'_i$*
- *The sets $I(T'_i) \setminus I(T_i)$ for $i = 1, \ldots, m$ are pairwise disjoint.*

Reversible sensitive tree chains allow us to construct first-maximal walks, as they support proper walks which start at any node of $T'_1, \ldots, T'_m$.

**Claim 32.1.** *Let $g : \{0,1\}^d \to \{-1,1\}$. Assume that there exists a reversible sensitive tree chain $(T_1, \ldots, T_m; T'_m, \ldots, T'_1)$ for $g$. Then, for any weight function $w : \{0,1\}^d \to \mathbb{R}$, there exists a walk $P = P_{g,w}$ which is proper with respect to $g$, and first-maximal with respect to $w$. In addition:*

- *$V(P) \subset V(T'_1, \ldots, T'_m)$.*
- *$I(P) = [d]$.*
- *The length of $P$ is at most $12d$.*

*Proof.* Let $V = V(T'_1, \ldots, T'_m)$. Let $v \in V$ for which $w(v)$ is maximal. We will construct a walk $P$ as above starting at $v$. Assume that $v \in V(T'_j)$. The walk $P$ is composed of:

- The shortest path in the tree $T'_j$ from $v$ to some $v' \in V(T_j)$.

- The walk $P_{chain}(v';T_j,\dots,T_m)$, ending at some $v'' \in V(T_m)$.
- The walk $P_{weak-chain}(v'';T'_m,\dots,T'_1)$.

The first two claims clearly hold. We next bound the length of $P$.

In order to bound the length of walk, the first part has length at most $\dim(T'_i) \leq d$. The second walk has length bounded by $\sum_{i=j}^{m} 3\dim(T_i) \leq 3d$, which follows as we assume that $I(T_1),\dots,I(T_m)$ are disjoint, and that $C(T_i) \cap V(T_{i+1}) \neq \emptyset$ for $i = 1,\dots,m-1$. The length of the third part can be bounded as follows.

The walk in the third part $P_{weak-chain}(v'';T'_m,\dots,T'_1)$ selects vertices $v'_{m-1},\dots,v'_1$ in $V(T'_{m-1}),\dots,V(T'_1)$, respectively, such that for all $i = m-1,\dots,1$ we have $v'_i \in C(T'_{i+1},\dots,T'_m)$. The walk starts at $v'_m := v'' \in V(T'_m)$, and explores $T'_m$ using $P_{tree}(v'_m;T'_m)$ that starts and ends at $v'_m$. We then take the shortest walk in $\mathscr{H}_d$ from $v'_m$ to $v'_{m-1}$(we explain why the walk is proper below). From $v'_{m-1}$ explore $T'_{m-1}$ using $P_{tree}(v'_{m-1};T'_{m-1})$, and then take the shortest walk in $\mathscr{H}_d$ from $v'_{m-1}$ to $v'_{m-2}$. We continue this way until we reach $v'_1$, where we explore $T'_1$ using $P_{tree}(v'_1;T'_1)$.

First, we argue that the walk is proper. Recall that when moving from $v'_{i+1}$ to $v'_i$, for $i = m-1,\dots,1$, we take the shortest path in $\mathscr{H}_d$ between the two vertices. Since $v'_i \in C(v'_{i+1}, I(T'_{i+1},\dots,T'_m))$, we only change coordinates in $I(T'_{i+1},\dots,T'_m)$ which means that the walk is indeed proper.

Next, we wish to bound the length of the shortest path from $v'_{i+1}$ to $v'_i$, i.e., the distance between $v'_i$ and $v'_{i+1}$ in $\mathscr{H}_d$. Denote by $d_H(u,v)$ the distance between two nodes $u$ and $v$ in $\mathscr{H}_d$ (i.e., their Hamming distance). To bound $d_H(v'_i, v'_{i+1})$, we use the fact that $T_1,\dots,T_m$ is a disjoint sensitive tree chain (that is, we are using the forward chain to bound the length of the backward walk!). Since $T_1,\dots,T_m$ is a disjoint sensitive tree chain, there exist $v_i \in V(T_i)$ and $v_{i+1} \in V(T_{i+1})$ with distance at most $\dim(T_i)$ between them (simply take $v_{i+1} \in V(T_{i+1}) \cap C(T_i)$

and any $v_i \in V(T_i)$). By the triangle inequality,

$$\begin{aligned} & d_H(v_i', v_{i+1}') \\ \le & d_H(v_i', v_i) + d_H(v_i, v_{i+1}) + d_H(v_{i+1}, v_{i+1}') \\ \le & \dim(T_i') + \dim(T_i) + \dim(T_{i+1}'), \end{aligned}$$

where we used the fact that $v_i', v_i \in V(T_i')$ to bound the first summand and that $v_{i+1}, v_{i+1}' \in V(T_{i+1}')$ to bound the third. Thus, the total length of the third part is at most

$$\sum_{j=1}^{m} \dim(T_j) + \sum_{j=1}^{m} 4\dim(T_j') \le 9d,$$

where $\sum_{j=1}^{m} \dim(T_j) = d$ and where $\sum_{j=1}^{m} \dim(T_j') \le 2d$ by our assumption that $I(T_1') \setminus I(T_1), \dots, I(T_m') \setminus I(T_m)$ are disjoint. Thus we can bound the length of the total walk by $12d$. □

The following lemma shows how, starting from a sensitive tree chain, we can construct a reversible sensitive tree chain.

**Lemma 32.1.** *Let $g : \{0,1\}^d \to \{-1,1\}$. Assume that there exists a disjoint sensitive tree chain $T_1, \dots, T_m$ for $g$ such that $I(T_1, \dots, T_m) = [d]$. Then there exists a reversible sensitive tree chain for $g$.*

Before proving Lemma 32.1 we need an extension of Claim 28.1 to a weak sensitive tree chain.

**Claim 32.2.** *Let $g : \{0,1\}^d \to \{-1,1\}$. Let $T_1, \dots, T_m$ be a weak sensitive tree chain with respect to $g$. Let $I \subset [d]$ disjoint from $I(T_1, \dots, T_m)$. Then there exists $I' \subseteq I$, and a weak sensitive tree chain $T_1', \dots, T_m'$ with respect to $g$, such that the following hold:*

- *$I(T_i') = I(T_i) \cup I_i'$, where $I_1', \dots, I_m'$ is a partition of $I'$.*
- *For all $i = 1, \dots, m$, $T_i'$ is invariant to shifts supported on directions $I \setminus I'$.*

- *There exists $J \subseteq I'$ such that for all i, $T_i + e_J$ is a sub-tree of $T_i'$.*

*Proof.* The proof is nearly identical to that of Claim 28.1. Let initially $T_i' = T_i, I' = \emptyset$. If all of $T_1', \dots, T_m'$ are invariant to shifts supported on directions $I \setminus I'$, we are done. Otherwise, pick minimal $J \subset I \setminus I'$ for which some $T_i'$ cannot be shifted in direction $J$, and pick $j \in J$. Replace each $T_i'$ with $T_i' + e_{J \setminus \{j\}}$, and observe that $T_1', \dots, T_m'$ is still a weak sensitive tree chain with respect to $g$. Choose $v \in V(T_i')$ such that $g(v) \neq g(v + e_j)$, add the edge $(v, v + e_j)$ to $T_i'$, and add $j$ to $I'$. Repeat this process until it terminates. □

*Proof of Lemma 32.1.* Let $T_1, \dots, T_m$ be the initial disjoint sensitive tree chain. Throughout the proof, we will modify $T_1, \dots, T_m$ by the following operations: for some $i \in [m]$ we will choose $J \subseteq I(T_i)$, and replace $T_{i+1}, \dots, T_m$ with $T_{i+1} + e_J, \dots, T_m + e_J$, while assuring that the latter are also sensitive trees for $g$. Observe that such operations maintain the property that $T_1, \dots, T_m$ is a disjoint sensitive tree chain, and that they do not change $I(T_j)$ for any $j$.

We construct $T_m', \dots, T_1'$ in this order. In the $i$-th iteration (where $i = m, \dots, 1$), we will construct $T_i'$, and along the way also change $T_{i+1}', \dots, T_m'$ and $T_{i+1}, \dots, T_m$. We will obtain the following invariant at the end of the $i$-th iteration (and the beginning of the $i-1$ iteration):

- $T_j$ is a sub-tree of $T_j'$ for all $j = i, \dots, m$.

- $T_1, \dots, T_m$ is a disjoint sensitive tree chain.

- $T_m', \dots, T_i'$ is a weak sensitive tree chain.

- $C(T_m', \dots, T_i') = C(T_i, \dots, T_m)$.

The first iteration, for $i = m$, is very simple: take $T_m' = T_m$. At the beginning of the $i$-th iteration, for $i < m$, we have already constructed $T_m', \dots, T_{i+1}'$ that satisfy the requirements above. Apply Claim 32.2 to the weak sensitive tree chain $T_m', \dots, T_{i+1}'$ with $I = I(T_i)$ (which by induction is disjoint from $I(T_m, \dots, T_{i-1}) = I(T_m', \dots, T_{i-1}'))$. This results in a weak sensitive tree chain $T_m'', \dots, T_{i+1}''$ and a set $I' = I(T_i) \cap I(T_{i+1}'', \dots, T_m'')$ such that

- There exists $J' \subseteq I'$ such $T'_j + e_{J'}$ is a sub-tree of $T''_j$ for all $j = i+1, \ldots, m$.
- The directions $I(T''_{i+1}) \cap I(T_i), \ldots, I(T''_m) \cap I(T_i)$ are disjoint and partition $I'$.
- For all $j = i+1, \ldots, m$, $T''_j$ is invariant to shifts supported on directions $I \setminus I'$.

Next, choose some $v_i \in V(T_i)$. Let $I'' = I \setminus I'$. Let $J'' \subseteq I''$ be a shift so that $T''_{i+1} + e_{J''}$ will agree with $v_i$ on the coordinates $I''$. Define $T'''_j = T''_j + e_{J''}$ for $j = i+1, \ldots, m$. Note that for $J = J' \cup J''$ we have that $T_j + e_J$ is a sub-tree of $T'''_j$. Perform the following operations:

- Set $T'_j = T'''_j$ for $j = i+1, \ldots, m$.
- Set $T_j = T_j + e_J$ for $j = i+1, \ldots, m$.
- Set $T'_i = T_i$.

We claim that this satisfies the required conditions for the end of the $i$-iteration.

First, we have that $T_j$ is a sub-tree of $T'_j$ for $j = i, \ldots, m$. Second, $T_1, \ldots, T_m$ is still a disjoint sensitive tree chain, as we shifted $T_{i+1}, \ldots, T_m$ by some $J \subseteq I(T_i)$. Next, we need to show that $T'_m, \ldots, T'_i$ is a weak sensitive tree chain.

Recall that by definition that means that $C(T'_m, \ldots, T'_{j+1}) \cap V(T'_j) \neq \emptyset$ for all $j = m-1, \ldots, i$. First, we claim that this holds for $j = m-1, \ldots, i+1$. This is true since it held at the beginning of the $i$-th iteration, and the only change is that we shifted all trees $T'_{i+1}, \ldots, T'_m$ by the same shift $e_J$, and potentially replaced them by larger sensitive trees containing them. So, it also holds at the end of the $i$-th iteration. Next, we show that for $j = i$.

Recall that we chose the shift $J$ so that for some $v_i \in V(T_i)$, $C(T'_{i+1})$ agrees with $v_i$ on $I'' = I(T_i) \setminus I(T'_m, \ldots, T'_{i+1})$. By the assumption that $C(T'_m, \ldots, T'_{i+1}) = C(T_{i+1}, \ldots, T_m)$ which held at the beginning of the $i$-th iteration, and since we only shifted and extended $T'_{i+1}, \ldots, T'_m$ by some directions in $I(T_i)$, we have that $T'_{i+1}, \ldots, T'_m \subset C(T_i, \ldots, T_m)$. As each $T_j$ is a sub-tree of $T'_j$, and as $T'_i = T_i$, this implies that $C(T'_m, \ldots, T'_i) = C(T_i, \ldots, T_m)$. But then $C(T'_{i+1})$ also agrees with $v_i$ on $I(T_1, \ldots, T_{i-1})$. This then implies that $v_i \in C(T'_m, \ldots, T'_{i+1})$. Thus $V(T_i) \cap C(T'_m, \ldots, T'_{i+1})$ is nonempty, as claimed. □

### 3.4.4 Completing the proof

We conclude the proof of Theorem 25. Let $g : \{0,1\}^d \to \{-1,1\}$ of maximal degree. By Lemma 30.1 there exists a disjoint sensitive tree chain $T_1,\dots,T_m$ for $g$, such that $I(T_1,\dots,T_m) = [d]$. We may thus apply Lemma 32.1, which shows the existence of a reversible sensitive tree chain for $g$. Claim 32.1 then shows that there exists a proper walk for $g$ of length at most $12d$. To conclude, observe that for any length $\ell \geq 1$, if we define a family of walks in $\mathscr{H}_d$ of length $\ell$,

$$\mathscr{P}_{length\,\ell} := \{P \text{ walk of length } \ell \text{ in } \mathscr{H}_d\}$$

then $\mathscr{P}$ can be described using $\ell \cdot \log d$ many bits, simply by giving the edges in the walk. This shows that first-maximal proper walks in dimension $d$ can be described using $12d\log d$ bits. [1]

## 3.5 Proofs of Corollaries

We prove in this section the two corollaries of Theorem 12 given in the introduction.

*Proof of Corollary 12.1.* For any $d \geq 1$ it is easy to verify that the assumption implies

$$\mathop{\mathbb{E}}_{x\in\{0,1\}^n}\left[s(f,x)^d + 1\right] \leq s^d \cdot d^d \cdot 2^{O(d)}.$$

Fix $\varepsilon > 0$ and let $\lambda$ to be determined later. Then by Theorem 12 we have

$$\begin{aligned}\sum_{|S|\geq\lambda s} \widehat{f}(S)^2 &\leq \frac{1}{(\lambda s)^d}\sum \widehat{f}(S)^2|S|^d \\ \leq & \frac{d^{13d}2^{O(d)}}{(\lambda s)^d} \mathop{\mathbb{E}}_{x\in\{0,1\}^n}\left[s(f,x)^d + 1\right] \\ \leq & \left(O\left(\frac{d^{14}}{\lambda}\right)\right)^d.\end{aligned}$$

[1] The constant 12 is not optimal. We chose to compromise optimizing the constant, in order to make the presentation simpler.

The claim follows by setting $d = \log(1/\varepsilon)$ and $\lambda = O(d^{14})$. □

*Proof of Corollary 12.2.* Let $g : \{0,1\}^n \to \{-1,1\}$ be the indicator function of $A$, namely $g(x) = 1$ if $x \in A$ and $g(x) = -1$ if $x \notin A$. Let $\text{Parity}(x) = (-1)^{x_1+\dots+x_n}$ be the parity function, and define $f(x) = g(x)\text{Parity}(x)$. Note that if $x \in A$ then the degree of $x$ in $G[A]$ equals $s(f,x)$, and if $x \notin A$ then the degree of $x$ in $G[A^c]$ equals $s(f,x)$. Thus, we can rephrase the claim as

$$\Pr_{x \in \{0,1\}^n}\left[s(f,x) \geq \frac{n}{c\log^{13}(1/\varepsilon)}\right] \geq \varepsilon^c,$$

for some absolute constant $c > 0$.

Next, observe that $\widehat{f}([n]) = \widehat{g}(\emptyset) = \mathbb{E}[g] = 2\varepsilon$. By Theorem 12 for any $d \geq 1$ we have

$$\begin{aligned}&\mathop{\mathbb{E}}_{x\in\{0,1\}^n}[s(f,x)^d]\\ \geq& -1 + \frac{1}{d^{13d}2^{O(d)}}\sum \widehat{f}(S)^2|S|^d\\ \geq& -1 + \frac{\varepsilon^2 n^d}{d^{13d}2^{O(d)}}.\end{aligned}$$

On the other hand, for any $\lambda > 0$ we have

$$\mathop{\mathbb{E}}_{x\in\{0,1\}^n}[s(f,x)^d] \leq \Pr[s(f,x) \geq \lambda n] \cdot n^d + (\lambda n)^d.$$

Setting $\lambda = 1/(c\log^{13}(1/\varepsilon))$ gives

$$\begin{aligned}&\Pr\left[s(f,x) \geq \frac{n}{c\log^{13}(1/\varepsilon)}\right]\\ \geq& \frac{\varepsilon^2}{d^{13d}2^{O(d)}} - \frac{1}{c^d\log^{13d}(1/\varepsilon)} - \frac{1}{n^d}.\end{aligned}$$

The claim follows by setting $d = \log(1/\varepsilon)$ and choosing $c > 0$ large enough. □

## 3.6 Open problems

Our main result is Theorem 13, which proves Conjecture 11.1. As we discussed in the introduction, we suspect that our quantitative bounds are sub-optimal. The conjectured bound below will allow us to match the results of [24], which assumed a bound on the maximal sensitivity.

**Conjecture 32.1.** *Theorem 12 holds with a bound of* $a_d \leq d^d \cdot 2^{O(d)}$.

Lets assume this conjecture for now. Corollary 12.2 finds many nodes of large degree in either $G[A]$ or $G[A^c]$. However, it makes sense that it suffices to consider the larger of either $A$ or $A^c$.

**Conjecture 32.2.** *Let* $A \subset \{0,1\}^n$ *of size* $|A| \geq (1+\varepsilon)2^{n-1}$. *Then* $G[A]$ *contains* $\varepsilon^{O(1)}2^n$ *vertices whose degree is at least* $\Omega(n/\log(1/\varepsilon))$.

The main technical component of our proof is the structure of the sensitivity graph for functions of maximal degree. Our techniques, however, apply equally well under the weaker assumption that the decision tree complexity of the function is maximal. In fact, any complexity measure where if $f$ has maximal complexity then, for any bit $x_i$, one of the restrictions $f|_{x_i=0}$ or $f|_{x_i=1}$ has maximal complexity would do.

**Acknowledgement** Chapter 3 contains a reprint of material as it appears in Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms 2018. Shachar Lovett, Avishay Tal and Jiapeng Zhang. *The robust sensitivity of boolean functions.* The dissertation author was a primary investigator and author of this paper.

# Chapter 4

# DNF Sparsification

The disjunctive normal form (DNF) is a well-used representation of logical formulae. A DNF is a disjunction of conjunctive clauses; it can also be described as an OR of ANDs. Functions which can represented as small CNFs or DNFs are central in computational complexity theory, and have been widely studied. We focus on DNFs in this chapter.

Given a DNF $f$, there are two natural ways to measure its complexity: the number of clauses, called the size of $f$; and the maximal number of variables in a clause, called the width of $f$. It is a folklore result that DNFs of small size can be approximated by DNFs of small width; concretely, we have the following theorem.

**Theorem 33.** *Let $f = \varphi_1 \vee \cdots \vee \varphi_s$ be a DNF of size s, where each $\varphi_i$ is a conjunctive clause. Then it can be $\varepsilon$-approximated by a DNF $f'$ of width* $\log(s/\varepsilon)$.

The proof of this theorem is pretty straightforward. We can easily obtain $f'$ by removing the large clauses.

*Proof.* Define the set as $B := \{i \in [t] : \text{ the width of } \varphi_i \text{ is smaller than } \log(s/\varepsilon)\}$. Define the DNF $f'$ as

$$f' = \bigvee_{i \in B} \varphi_i.$$

Then

$$\Pr_x[f(x) \neq f(x')] \leq \Pr_x[\exists i \notin B, \varphi_i(x) = 1] \leq \sum_{i \notin B} \Pr_x[\varphi_i(x) = 1] \leq s \cdot 2^{-\log(s/\varepsilon)} \leq \varepsilon.$$

Then claim the follow. □

Gopalan, Meka and Reingold [21] studied the reverse problem of *DNF sparsification*: can small width DNFs be approximated by DNFs of small size? Their motivation, other than being a natural problem on the structure of DNFs, came from the goal of designing faster deterministic algorithms to approximately count the number of satisfying assignements of a DNF. In specific, their main structural result on DNFs is the following.

**Theorem 34** ([21]). *Let $f$ be a boolean function which can be expressed as a width-$w$ DNF. Then for every $\varepsilon > 0$, $f$ can be $\varepsilon$-approximated by a DNF of width $w$ and size $(w\log(1/\varepsilon))^{O(w)}$.*

It was conjectured in [21] that the term $(w\log(1/\varepsilon))^w$ is not tight. In particular, Conjecture 6.1 in their paper speculates that the bound can be improved to $c(\varepsilon)^w$, and moreover that possibly one can take $c(\varepsilon) = (\log 1/\varepsilon)^{O(1)}$. Joint with Lovett [44], we resolve the weaker conjecture.

**Theorem 35** ([44]). *Let $f$ be a boolean function which can be expressed as a width-$w$ DNF. Then for every $\varepsilon > 0$, $f$ can be $\varepsilon$-approximated by a DNF of width $w$ and size $(1/\varepsilon)^{O(w)}$.*

While the dependence we obtain on the error $\varepsilon > 0$ is probably sub-optimal, our main goal was to sharpen the dependence on the width $w$, from $w^{O(w)}$ to $2^{O(w)}$ (for a fixed error $\varepsilon$).

## 4.1 Proof overview

Let $f = \varphi_1 \vee \ldots \vee \varphi_t$ be a DNF, where each $\varphi_i$ is a clause (conjunction of literals). The main object which underlies our work is the function which maps an input to the *first* clause which satisfies it. We call this the *DNF index function*. Observe that this depends on the specific structure of the DNF, and not just the boolean function it computes. Moreover, it depends on the order of the clauses.

**Definition 36** (DNF index function)**.** *Let $f = \varphi_1 \vee \ldots \vee \varphi_t$ be a DNF. The index function of $f$ is a function $Ind_f : \{0,1\}^n \to \{0,\ldots,t\}$ defined as follows:*

$$Ind_f(x) = \begin{cases} 0 & \text{if } f(x) = 0 \\ \min\{i \in [t] : \varphi_i(x) = 1\} & \text{if } f(x) = 1 \end{cases}$$

Let $p_i = \Pr_x[\mathrm{Ind}_f(x) = i]$ denote the fraction of inputs such that the $i$-th clause is the first clause that satisfies them. The following is a natural approach for DNF sparsification: only keep clauses $\varphi_i$ for which $p_i$ is noticeable.

However, it is not clear how many noticeable clauses could there be. For example, a bad scenario would be if $p_i = 1/t$ for all $i$; in this case there would be no way to significantly sparsify the DNF. However, this cannot be the case, as if $\varphi_1$ as width $w$ then $p_1 = 2^{-w}$. So, the main challenge is to show that there is a small set of indices $I \subset [t]$, such that $\sum_{i \notin I} p_i \leq \varepsilon$. Our main theorem shows that this holds for $|I| = (1/\varepsilon)^{O(w)}$.

Next, we highlight how we show that. At its core, our argument has two parts: a combinatorial part, where we prove switching lemma for the DNF index function; and an analytic part, where we analyze the noise sensitivity of the index function and connect it to the problem of DNF sparsification.

**Combinatorial part: Switching lemma.** The behaviour of DNFs under random restrictions have been well studied. Razborov [51], refining previous work of Håstad [26], showed that DNFs simplify under random restrictions. See also [7] for an exposition.

We need a few standard definitions. A *restriction* is $\rho \in \{0,1,*\}^n$. An $(n,k)$*-random restriction* is a uniform restriction $\rho \in \{0,1,*\}^n$ with exactly $k$ stars. Given a boolean function $f : \{0,1\}^n \to \{0,1\}$, its restriction under $\rho$ is denoted $f|_\rho : \{0,1\}^{\rho^{-1}(*)} \to \{0,1\}$. Given a function $f : \{0,1\}^n \to X$ for some finite set $X$, we denote its *decision tree complexity* by $\mathrm{dt}(f)$.

The well-known switching lemma for DNFs[26, 51, 7] is the following result. Let $f$ be an $n$-variate boolean function computed by a width-$w$ DNF. Let $k = \alpha n$. Let $\rho \in \{0,1,*\}^n$ be an $(n,k)$-random restriction. Then for any $d \geq 1$,

$$\Pr_{\rho}\left[\mathrm{dt}(f|_{\rho}) \geq d\right] \leq (7\alpha w)^d.$$

We extend this result to the DNF index function. Assume that $f = \varphi_1 \vee \ldots \vee \varphi_t$ and let $\mathrm{Ind}_f : \{0,1\}^n \to \{0,\ldots,t\}$ be its associated DNF index function. We prove (lemma 36.1) that for any $d \geq 1$,

$$\Pr_{\rho}\left[\mathrm{dt}(\mathrm{Ind}_f|_{\rho}) \geq d\right] \leq (32\alpha w)^d.$$

We note that Rossman [53] introduced the index function under the name “first witness function”, and proved a similar switching lemma.

**Analytic part: Noise sensitivity.** Let $I$ denote the set of noticeable clauses $i \in [t]$. Our goal is to upper bound $\sum_{i \notin I} p_i$. To that end, we study the behaviour of the index function under noise. Given $x \in \{0,1\}^n$ let $\mathbb{N}_\rho(x)$ denote the noise distribution around $x$, where $y \sim \mathbb{N}_\rho(x)$ is sampled by taking $\Pr[x_i = y_i] = \rho$ independently for $i \in [n]$. The main observation is that that if all (or most) of the $p_i$ are negligible, then $\mathrm{Ind}_f$ cannot be stable under noise. This is since if $\mathrm{Ind}_f(x) = i$, and $p_i$ is tiny, then if we sample $y \sim \mathbb{N}_\rho(x)$ then with high probability $\mathrm{Ind}_f(y) \neq i$. This follows from the well known fact (whose proof is based on the hypercontractive inequality) that small sets in the hypercube are not noise stable.

So, our goal is to show that $\mathrm{Ind}_f$ is noise stable. Concretely, we say that an input $x$ which satisfies $f$ is $(\rho,\gamma)$-stable for $f$ if

$$\Pr_{y \sim \mathbb{N}_\rho(x)}\left[\mathrm{Ind}_f(x) = \mathrm{Ind}_f(y)\right] \geq \gamma.$$

Thus, if most inputs are stable, then if we first sample $x \in \{0,1\}^n$ uniformly, and then take $i = \mathrm{Ind}_f(x)$, then with high probability $p_i$ is noticeable. This then implies that $\sum p_i$ is concentrated on a small set $I$. To conclude, we need to show that indeed most inputs $x \in f^{-1}(1)$ are noise stable.

This in turn follows from our switching lemma for $\mathrm{Ind}_f$. Consider an equivalent way to jointly sample $x, y$, where first we sample $\rho \in \{0,1,*\}^n$ where $\Pr[\rho_i = *] = 1 - \rho$, and then sample $x, y$ conditioned on $x_i = y_i = \rho_i$ whenever $\rho_i \neq *$. If $\mathrm{Ind}_f|_\rho$ has a small depth decision tree, then there is a noticeable probability that it evaluates to the same leaf on both $x, y$. That is, that $\mathrm{Ind}_f(x) = \mathrm{Ind}_f(y)$. Thus, the switching lemma allows us to prove that most inputs are noise stable, completing the proof.

## 4.2 Switching lemma for DNF index function

Let $f$ be a width-$w$ DNF. Recall that the index function of $f$ maps an input to the first clause that is satisfies, or to 0 if no clause is satisfied. The main goal of this section is to prove a switching lemma for the DNF index function. We start with some preliminary definitions.

**Decision tree.** Let $g : \{0,1\}^n \to X$ be a function where $X$ is some finite set. A decision tree for $g$ is a binary tree whose nodes are labeled by variables and whose leaves are labeled by elements of $X$. The decision tree complexity of $g$, denoted $\mathrm{dt}(g)$, is the minimal depth of a decision tree computing $g$.

**Restrictions.** A restriction is $\rho \in \{0,1,*\}^n$. Given a function $g : \{0,1\}^n \to X$, its restriction $g|_\rho$ is the sub-function obtained by restricting to inputs which agree with $\rho$. That is, let $S = \{i : \rho_i = *\}$ be the "alive" variables. Then $g|_\rho : \{0,1\}^S \to X$ by mapping $z \in \{0,1\}^S$ to $g(x)$, where $x_i = z_i$ if $i \in S$ and $x_i = \rho_i$ otherwise.

**Random restrictions.** An $(n,k)$-random restriction is the the uniform distribution over restrictions $\rho \in \{0,1,*\}^n$ with exactly $k$ stars.

The following is the main result of this section.

**Lemma 36.1** (Switching lemma for the DNF index function)**.** *Let $f$ be a width-$w$ DNF on $n$ variables, and let $Ind_f$ be its DNF index function. Let $k = \alpha n$ and let $\rho$ be an $(n,k)$-random restriction.. Then for every $d \geq 1$,*

$$\Pr_{\rho}[dt(Ind_f|_{\rho}) \geq d] \leq (32\alpha w)^d.$$

*Proof.* We assume $\alpha \leq 1/32w$ otherwise the claim is trivial. Let $\rho \in \{0,1,*\}^n$. We say that $\rho$ is "bad" if $\mathrm{dt}(\mathrm{Ind}_f|_{\rho}) \geq d$. We use a compression argument, similar to the one used by Razborov [51] to prove the switching lemma for DNFs.

The DNF $f = \varphi_1 \vee \ldots \vee \varphi_t$ is fixed throughout. Let $V_j$ denote the variables that appear in $\varphi_j$. We use the following notations. Given two strings $a, a'$ their concatenation is $a \circ a'$. Given a known set $W$ of size $w$, and a set $V \subset W$ of a known size, we can uniquely describe $V$ by a string in $[w]^{|V|}$. We denote this representation $\mathrm{SetIndex}(W,V)$. We define three operations on restrictions:

- **Append:** given a restriction $\rho \in \{0,1,*\}^n$ and a partial input $u \in \{0,1\}^S$ where $S \subset \rho^{-1}(*)$, we denote by $\mathrm{append}(\rho,u)$ the restriction obtained by appending $u$ to $\rho$:

$$\mathrm{append}(\rho,u) = \begin{cases} u_i & \text{if } i \in S \\ \rho_i & \text{otherwise} \end{cases}$$

- **Delete:** given a restriction $\rho \in \{0,1,*\}^n$ and a set $S \subset \rho^{-1}(\{0,1\})$, we denote by delete$(\rho, S)$ the restriction obtained by setting the symbols in $S$ to stars:

$$\text{delete}(\rho, S) = \begin{cases} * & \text{if } i \in S \\ \rho_i & \text{otherwise} \end{cases}$$

- **Update:** given a restriction $\rho \in \{0,1,*\}^n$ and a partial input $u \in \{0,1\}^S$ where $S \subset \rho^{-1}(\{0,1\})$, we denote by update$(\rho, u)$ the restriction obtained by updating the elements in $S$ to $u$:

$$\text{update}(\rho, u) = \begin{cases} u_i & \text{if } i \in S \\ \rho_i & \text{otherwise} \end{cases}$$

We next present the encoding and decoding algorithms.

```
                         $Encode(\rho)$

Input: restriction $\rho \in \{0,1,*\}^n$.
Output: restriction $\tau \in \{0,1,*\}^n$, string $a \in \mathbb{N}^*$.

  1. Initialize $\tau = \rho$. Initialize $a$ to be an empty string.
  2. For $j = 1, \ldots, t$ do:
     (a) If $\varphi_j|_\rho \equiv 0$ then skip to the next $j$.
     (b) If $\varphi_j|_\rho \equiv 1$ then abort the loop.
     (c) Otherwise compute:
           i. $A_j = \{i \in V_j : \rho_i = *\}$ the alive variables in $\varphi_j$.
          ii. $u_j \in \{0,1\}^{A_j}$ an assignment under which $\text{dt}(\text{Ind}_f|_{\text{append}(\rho,u_j)})$ is maximized.
         iii. $v_j \in \{0,1\}^{A_j}$ an assignment under which $\varphi_j|_{\text{append}(\rho,v_j)} \equiv 1$.
```

```
        (d) Update:
              i. $\rho = \text{append}(\rho, u_j)$.
              ii. $\tau = \text{append}(\tau, v_j)$.
              iii. $a = a \circ |A_j| \circ \text{SetIndex}(V_j, A_j) \circ u_j$.
    3. Return $\tau, a$.
```

```
                              $Decode(\tau, a)$

Input: restriction $\tau \in \{0,1,*\}^n$, string $a \in \mathbb{N}^*$.
Output: restriction $\rho \in \{0,1,*\}^n$.

    1. Initialize $A$ to be an empty set.
    2. For $j = 1, \ldots, t$ do:
        (a) If $\varphi_j|_\tau \equiv 0$ then skip to the next $j$.
        (b) Otherwise read from $a$: $A_j \subset V_j$ and $u_j \in \{0,1\}^{A_j}$.
        (c) Update:
              i. $A = A \cup A_j$.
              ii. $\tau = \text{update}(\tau, u_j)$.
    3. Return $\rho = \text{delete}(\tau, A)$.
```

We first argue that the encoding and decoding are correct.

**Lemma 36.2.** *For any $\rho \in \{0,1,*\}^n$ it holds that*

$$DECODE(ENCODE(\rho)) = \rho.$$

*Proof.* Let $\tau, a = ENCODE(\rho)$. Note that if $\rho_i \neq *$ then $\tau_i = \rho_i$. So, we just need to verify that the decoding procedure deletes exactly the elements that were appended in the encoding procedure, namely $\cup A_j$. Say that an index $j \in [t]$ is active if in the encoding procedure, we have that $\varphi_j|_\rho$ is non-constant when it is considered. Let $J = \{j_1, \ldots, j_r\}$ denote the set of active indices. The main observation is that these are also the indices in which in the decoding procedure we have $\varphi_j|_\rho \not\equiv 0$. In fact, one can further verify that $\varphi_j|_\rho \equiv 1$ in these cases. To conclude note that the auxiliary string $a$ allows to precisely recover the sets $A_j$. □

To conclude the proof we need to bound the probability that $\rho$ is bad. To do so, we bound the size of the set $\{ENCODE(\rho) : \rho \text{ is bad}\}$. Assume that $\tau, a = ENCODE(\rho)$. As $\rho$ is bad, we have $\mathrm{dt}(\mathrm{Ind}_f|_\rho) \geq d$. This means that $m = \sum |A_j| \geq d$ by the choice of the $u_j$. Given a fixed $m$ we bound the number of choices for $\tau, a$.

The restriction $\tau$ has exactly $k-m$ stars, and so has $\binom{n}{k-m} 2^{n-k+m}$ options. Assume there are $r$ sets $A_j$ with $|A_j| > 0$. The number of choices of $|A_1|, \ldots, |A_r|$ is equal to the number of ways we can decompose $m = a_1 + \ldots + a_r$ with $a_i \geq 1$, which equals $\binom{m-1}{r-1}$. The sum of these over all $r$ is $2^{m-1}$. Given that $|A_j| > 0$ for some $j$, the number of options for $\mathrm{SetIndex}(V_j, A_j)$ is $w^{|A_j|}$ and the number of choices for $u_j$ is $2^{|A_j|}$. So we obtain

$$\{ENCODE(\rho) : \rho \text{ is bad}\} \leq \sum_{m \geq d} \binom{n}{k-m} 2^{n-k+m} (4w)^m.$$

On the other hand, the total number of restrictions $\rho$ with exactly $k$ stars equals $\binom{n}{k} 2^{n-k}$. So we obtain that

$$\begin{aligned}\Pr[\rho \text{ is bad}] &\le \sum_{m\ge d} \frac{\binom{n}{k-m}}{\binom{n}{k}}(8w)^m \\ &\le \sum_{m\ge d}\left(\frac{\alpha}{1-\alpha}\right)^m (8w)^m \\ &\le \sum_{m\ge d}(16\alpha w)^m \le (32\alpha w)^d,\end{aligned}$$

where the last inequality follows from the assumption $\alpha \le 1/32w$. □

We would need the following simple corollary. Let $\mathbb{R}_{n,\alpha}$ be the distribution over restrictions $\{0,1,*\}^n$ where $\Pr[\rho_i = *] = \alpha$ and $\Pr[\rho_i = 0] = \Pr[\rho_i = 1] = \frac{1-\alpha}{2}$.

**Corollary 36.1.** *Let $f$ be a width-$w$ DNF on $n$ variables, and let $Ind_f$ be its DNF index function. Let $\rho \sim \mathbb{R}_{n,\alpha}$. Then for every $d \ge 1$,*

$$\Pr_{\rho\sim\mathbb{R}_{n,\alpha}}[dt(Ind_f|_\rho) \ge d] \le (64\alpha w)^d + 2^{-\Omega(\alpha n)}.$$

*Proof.* Let $\rho \sim \mathbb{R}_{n,\alpha}$ and let $k = |\rho^{-1}(*)|$. Conditioned on $|\rho^{-1}(*)| = k$, the distribution of $\rho$ is an $(n,k)$-random restriction. Namely, it is uniform in $U_{n,k}$, the set of restrictions in $\{0,1,*\}^n$ with exactly $k$ stars. Then

$$\begin{aligned}&\Pr_{\rho\sim\mathbb{R}_{n,\alpha}}[\mathrm{dt}(\mathrm{Ind}_f|_\rho) \ge d] \\ =&\sum_k \Pr_{\rho\sim\mathbb{R}_{n,\alpha}}[|\rho^{-1}(*) = k|] \cdot \Pr_{\rho\in U_{n,k}}[\mathrm{dt}(\mathrm{Ind}_f|_\rho) \ge d].\end{aligned}$$

The probability that $k \ge 2\alpha n$ is exponentially small in $\alpha n$. Whenever $k \le 2\alpha n$ we use lemma 36.1 to deduce the bound. □

## 4.3 DNF sparsification

The goal of this section is to prove the following theorem.

**Theorem 37.** *Let $f = \varphi_1 \vee \ldots \vee \varphi_t$ be a width-w DNF. Then for every $\varepsilon > 0$, there exists a subset $I \subset [t]$ of size $|I| \le (1/\varepsilon)^{O(w)}$ such that the following holds. Let $f' = \bigvee_{i \in I} \varphi_i$. Then*

$$\Pr[f(x) \neq f'(x)] \le \varepsilon.$$

### 4.3.1 Noticeable indices

Let $f = \varphi_1 \vee \ldots \vee \varphi_t$ be a width-$w$ DNF. To recall, $\mathrm{Ind}_f$ is the index function of $f$, which maps an input $x$ to the first clause that it satisfies, or to 0 if $f(x) = 0$. The main question we study is: how are the outputs of the DNF index function distributed? for example, can they be uniform in $[t]$? we show that the answer is no if $t$ is too large, which leads us to be able to approximate $f$ as a smaller DNF.

**Definition 38** (Noticeable index)**.** *Let $f = \varphi_1 \vee \ldots \vee \varphi_t$ be a DNF. An index $i \in [t]$ is called $\tau$-noticeable if*

$$\Pr_{x \in \{0,1\}^n} \left[ Ind_f(x) = i \right] \ge \tau.$$

For example, if $f$ is a DNF, and $\varphi_1$ is the first clause with $w$ variables, then 1 is $(2^{-w})$-noticeable since $\Pr[\varphi_1(x) = 1] = 2^{-w}$. We denote the set of all noticeable indices by

$$I(f, \tau) = \{i \in [t] : i \text{ is } \tau\text{-noticeable}\}.$$

The following claim is straightforward.

**Lemma 38.1.** $|I(f, \tau)| \le 1/\tau$.

*Proof.* Ii $i \in I(f, \tau)$ then $\Pr\left[\mathrm{Ind}_f(x) = i\right] \ge \tau$. These events are disjoint for different $i$. □

### 4.3.2 Noise stability

We use the following shorthand: $|g| = \Pr_{x\in\{0,1\}^n}[g(x)=1]$ is the fraction of inputs on which $g$ accepts. We will need the following fact which follows from the hyper-contractive inequality (see for example [50], page 259).

**Lemma 38.2.** *Let $g:\{0,1\}^n \to \{0,1\}$ be a boolean function. Then $\mathrm{Stab}_\rho(g) \le |g|^{\frac{2}{1+\rho}}$.*

The following claim is a simple corollary of Fact 38.2. It studies the noise sensitivity of a decomposition of a boolean function $f$ into disjoint boolean functions $g_1,\ldots,g_t$.

**Lemma 38.3.** *Let $f = g_1+\ldots+g_t$ where $f,g_1,\ldots,g_t:\{0,1\}^n \to \{0,1\}$ are boolean functions. Given a parameter $\tau \in [0,1]$ define*

$$I = \{i \in [t] : |g_i| \ge \tau\}.$$

*Then*

$$\sum_{i\notin I} \mathrm{Stab}_\rho(g_i) \le \tau^{\frac{1-\rho}{1+\rho}}.$$

*Proof.* Lemma 38.2 gives that for $i \notin I$ we have $\mathrm{Stab}_\rho(g_i) \le |g_i|^{\frac{2}{1+\rho}} \le |g_i| \cdot \tau^{\frac{1-\rho}{1+\rho}}$. Thus

$$\sum_{i\notin I} \mathrm{Stab}_\rho(g_i) \le \tau^{\frac{1-\rho}{1+\rho}} \sum_{i\notin I} |g_i| \le \tau^{\frac{1-\rho}{1+\rho}}.$$

□

### 4.3.3 Noise stability of the index function

The noise stability of boolean function is a well-studied topic. Here, we study the noise stability of the DNF index function.

**Definition 39** (Stable and sensitive inputs)**.** *Let $f$ be a DNF, $Ind_f$ be its DNF index function, and* *let $x \in \{0,1\}^n$ be an input which satisfies $f$. The input $x$ is called $(\rho,\gamma)$-stable for $f$ if*

$$\Pr_{y \sim \mathbb{N}_\rho(x)} \left[ \mathit{Ind}_f(x) = \mathit{Ind}_f(y) \right] \geq \gamma.$$

*Otherwise, $x$ is called $(\rho,\gamma)$-sensitive for $f$.*

**Definition 40** (Index sensitivity)**.** *The* $(\rho,\gamma)$-index sensitivity *of $f$ is the fraction of $(\rho,\gamma)$-sensitive inputs for $f$,*

$$\begin{aligned} &\mathit{IndexSensitivity}(f,\rho,\gamma) \\ =& \Pr_{x \in \{0,1\}^n} \left[ f(x) = 1 \wedge x \textit{ is } (\rho,\gamma)\textit{-sensitive for } f \right]. \end{aligned}$$

The following lemma connects the index sensitivity to DNF sparsification.

**Lemma 40.1.** *Let $f = \varphi_1 \vee \ldots \vee \varphi_t$ be a DNF. Fix $\rho,\gamma,\tau \in [0,1]$. Let $I = I(f,\tau)$ be the set of $\tau$-noticeable clauses of $f$, and define $f' = \bigvee_{i \in I} \varphi_i$. Then*

$$\Pr[f(x) \neq f'(x)] \leq \mathit{IndexSensitivity}(f,\rho,\gamma) + \gamma^{-1} \tau^{\frac{1-\rho}{1+\rho}}.$$

*Proof.* Observe that $f'(x) \leq f(x)$ for all $x$. So, if $f(x) \neq f'(x)$ then necessarily $f'(x) = 0, f(x) = 1$ and $\mathrm{Ind}_f(x) \notin I$. Let $I^c = [t] \setminus I$. Then

$$\Pr[f(x) \neq f'(x)] \leq \sum_{i \in I^c} \Pr[\mathrm{Ind}_f(x) = i].$$

To simplify notation, for $x \in \{0,1\}^n$ let $E(x)$ denote the event "$x$ is $(\rho,\gamma)$-stable for $f$". Then we can bound

$$\begin{aligned} &\Pr[f(x) \neq f'(x)] \\ \leq& \Pr[f(x) = 1 \wedge \neg E(x)] + \sum_{i \in I^c} \Pr[\mathrm{Ind}_f(x) = i \wedge E(x)]. \end{aligned}$$

The first term equals IndexSensitivity$(f,\rho,\gamma)$. To bound the second term, Fix $i \in I^c$. Conditioned on $E(x)$ we have $\Pr_{y\sim\mathbb{N}_\rho(x)}[\mathrm{Ind}_f(x) = \mathrm{Ind}_f(y)] \geq \gamma$. Thus

$$\begin{aligned}&\Pr_{x\in\{0,1\}^n, y\sim\mathbb{N}_\rho(x)}[\mathrm{Ind}_f(x) = \mathrm{Ind}_f(y) = i \wedge E(x)]\\ \geq&\gamma\cdot\Pr[\mathrm{Ind}_f(x) = i \wedge E(x)].\end{aligned}$$

Let $g_i : \{0,1\}^n \to \{0,1\}$ be the indicator of the event $\mathrm{Ind}_f(x) = i$, so that $f = g_1 + \ldots + g_t$. Then we have

$$\begin{aligned}&\Pr[\mathrm{Ind}_f(x) = i \wedge E(x)]\\ \leq&\gamma^{-1}\cdot\Pr[\mathrm{Ind}_f(x) = \mathrm{Ind}_f(y) = i]\\ =&\gamma^{-1}\cdot\mathrm{Stab}_\rho(g_i).\end{aligned}$$

To conclude we have

$$\Pr[f(x) \neq f'(x)] \leq \mathrm{IndexSensitivity}(f,\rho,\gamma) + \gamma^{-1}\cdot\sum_{i\in I^c}\mathrm{Stab}_\rho(g_i).$$

The bound now follows from claim 38.3. □

Thus, we reduced the problem of compressing DNFs to that of bounding the index sensitivity of DNFs. The following lemma shows that for width-$w$ DNFs is, most of their inputs are stable at noise level $\rho = 1 - O(1/w)$. Its proof uses the switching lemma for the DNF index function, or more precisely corollary 36.1.

**Lemma 40.2.** *Let $f$ be an n-variate width-w DNF. Set $\rho = 1 - \frac{1}{128w}$ and let $\gamma = 2^{-d}$ for an integer $d \geq 1$. Then*

$$\mathit{IndexSensitivity}(f,\rho,\gamma) \leq 2\gamma + 2^{-\Omega(n/w)}.$$

*Proof.* Let $x \in \{0,1\}^n$ sampled uniformly and let $y \sim \mathbb{N}_\rho(x)$. It will be convenient to sample $x, y$

in an equivalent but different way. Recall that $\mathbb{R}_{n,\alpha}$ is a distribution over restrictions $\rho \in \{0,1,*\}^n$ where $\Pr[\rho_i = *] = \alpha$ and $\Pr[\rho_i = 0] = \Pr[\rho_i = 1] = \frac{1-\alpha}{2}$. Then we can sample $(x,y)$ as follows:

1. Sample $\rho \sim \mathbb{R}_{n,\alpha}$ where $\alpha = 1-\rho$. Let $S = \{i : \rho_i = *\}$.

2. Sample $x|_S \in \{0,1\}^S$ uniformly, and set $x_i = \rho_i$ if $i \notin S$.

3. Sample $y|_S \in \{0,1\}^S$ uniformly, and set $y_i = \rho_i$ if $i \notin S$.

Next, fix $\rho$ and assume that $\mathrm{dt}(\mathrm{Ind}_f|_\rho) = d$. Then in particular, the probability that $x|_S, y|_S$ take the same path in the decision tree is at least $2^{-d}$. So we obtain that

$$\Pr_{x,y}\left[\mathrm{Ind}_f(x) = \mathrm{Ind}_f(y)|\rho\right] \geq 2^{-\mathrm{dt}(\mathrm{Ind}_f|_\rho)}.$$

Let $p(x)$ denote the probability that $\mathrm{dt}(\mathrm{Ind}_f|_\rho) \geq d$ when $\rho$ is sampled conditioned on $x$. Then

$$\Pr_{y}\left[\mathrm{Ind}_f(x) = \mathrm{Ind}_f(y)|x\right] \geq 2^{-(d-1)}(1-p(x)).$$

We would like to show that for most $x$ it holds that $p(x) \leq 1/2$; such $x$ will be $(\rho, 2^{-d})$-stable for $f$. That is, we wish to upper bound

$$p = \Pr_x\left[p(x) \geq 1/2\right] \leq 2 \Pr_\rho\left[\mathrm{dt}(\mathrm{Ind}_f|_\rho) \geq d\right].$$

corollary 36.1 bounds the right hand side, and for $\alpha = 1/128w$ gives

$$p \leq 2^{1-d} + 2^{-\Omega(n/w)}.$$

□

We now prove theorem 37.

*Proof of theorem 37.* Let $\rho, \gamma, \tau \in [0,1]$ to be optimized shortly. Let $I = I(f,\tau)$, where $|I| \le 1/\tau$ by Claim 38.1. Combining lemma 40.1 with lemma 40.2 gives the bound

$$\Pr[f \neq f'] \le O(\gamma) + 2^{-\Omega(n/w)} + \gamma^{-1}\tau^{\frac{1-\rho}{1+\rho}}.$$

First, we note that we may assume that $n = \Omega(w\log(1/\varepsilon))$ as otherwise the theorem holds vacuously, as the total number of possible width $w$ clauses is $2^w\binom{n}{w} \le O(\log 1/\varepsilon)^w$. Let $\rho = 1 - 1/128w$, $\gamma = O(\varepsilon)$ and $\tau = \varepsilon^{O(w)}$. Then

$$\Pr[f(x) \neq f'(x)] \le \varepsilon.$$

□

## 4.4 Open problems and future direction

As we shortly describe, the same challenge appears in two other related problem: the Erdős-Rado sunflower conjecture [16] and Mansour's conjecture [45].

### 4.4.1 Connections to the sunflower conjecture

The sunflower structure was first introduced by Erdős-Rado. It deals with set systems. A $w$-set system is a collection of sets in which all sets contain at most $w$ elements. To see the relation between set systems and DNFs, note that if $\mathscr{F}$ is a set system of sets $S \subset [n]$, then there is a natural associated (monotone) DNF given by $f(x) = \vee_{S\in\mathscr{F}} \wedge_{i\in S} x_i$. The DNF associated to a $w$-set system is a width-$w$ DNF. In the other direction, any width-$w$ DNF contains a large unary DNF (concretely, with at least a $2^{-w}$ fraction of the clauses) [21]. Unary DNFs are similarly equivalent to set systems.

We next introduce sunflowers, which are widely studied in combinatorics.

**Definition 41** (Sunflower [16])**.** *A collection of $r$ sets $S_1, \dots, S_r$ is called an $r$-sunflower if all the pairwise intersections $S_i \cap S_j$ are the same.*

The Erdős-Rado sunflower lemma [16] states that if $\mathscr{F}$ is a $w$-uniform set system, and $|\mathscr{F}| > w!(r-1)^w$, then $F$ must contain an $r$-sunflower. The well-known sunflower conjecture is that the dependence on $w$ can be improved.

**Conjecture 41.1** (Sunflower conjecture)**.** *For any $r \geq 3$ there exists $c = c(r)$ such that the following holds. If $F$ is a $w$-uniform set system, and $|F| > c^w$, then $F$ contains an $r$-sunflower.*

The sunflower conjecture has been open for nearly 60 years. Despite much research, the best bounds, even for $r = 3$, still are of the order of $w^w$.

**Approximate sunflowers.** Rossman [52] defined the notion of an approximate sunflower, motivated by applications in complexity theory.

**Definition 42** (Approximate sunflower)**.** *A set system $\mathscr{F}$ is a $\gamma$-approximate sunflower if the following holds. Let $C = \cap_{S \in \mathscr{F}} S$ be the common core of all sets in $\mathscr{F}$, and define $\mathscr{F}' = \{S \setminus C : S \in \mathscr{F}\}$. Let $f'$ be the monotone DNF associated with $\mathscr{F}'$. Then $\Pr[f'(x) = 1] \geq 1 - e^{-\gamma}$.*

Rossman proved that if $\mathscr{F}$ is a $w$-uniform set system of size $|\mathscr{F}| > (w \log(1/\gamma))^{O(w)}$, then $\mathscr{F}$ must contain a $\gamma$-approximate sunflower. Similarly to the sunflower conjecture, it is reasonable to conjecture that a better bound holds.

**Conjecture 42.1** (Approximate sunflower conjecture)**.** *For any $\gamma > 0$ there exists $c = c(\gamma) > 0$ such that the following holds. If $F$ is a $w$-uniform set system, and $|F| > c^w$, then $F$ contains an $\gamma$-approximate sunflower.*

In fact, one can show [35, 39] that the approximate sunflower conjecture implies the sunflower conjecture.

**Connection to DNF sparsification.** The main tool used in [21] to achieve their result about DNF sparsification is the sunflower and approximate sunflower lemmas stated above. Roughly speaking, they used the approximate sunflower lemma to compress an approximate sunflower to its common core. A barrier towards an improved dependence on the width $w$ in their result, is that the dependence on $w$ in both lemmas is of the order of $w^w$. Thus, one of the main motivations of the current work is to achieve DNF sparsification that breaks the $w^w$ bound. A more ambitious goal is to use the connection between sunflower structure and DNF sparsification as highlighted in [21], together with our improved DNF sparsification result, to obtain improved bounds for the sunflower conjecture.

**Upper approximation DNF implies sunflower structures.** In this paper, we show that every width-$w$ DNF $f$ can be $\varepsilon$-approximated by a width-$w$ DNF $f'$ of size $(1/\varepsilon)^{O(w)}$. In addition, $f'$ *lower bounds* $f$, that is $f'(x) \leq f(x)$ for all $x \in \{0,1\}^n$. In a joint work with Lovett and Solomon [39], we show that if one can get similar bounds where $f'$ *upper bounds* $f$, then this would imply improved bounds for the sunflower conjecture.

### 4.4.2 Connections to Mansour's conjecture

Mansour's conjecture [45] deals with the approximation of DNFs by sparse polynomials. We say that a boolean function $f : \{0,1\}^n \to \{0,1\}$ can be $\varepsilon$-approximated by a polynomial of sparsity $t$ if there exists a polynomial $p : \{0,1\}^n \to \mathbb{R}$ with at most $t$ monomials such that

$$\mathop{\mathbb{E}}_{x \in \{0,1\}^n} \left[ (f(x) - p(x))^2 \right] \leq \varepsilon.$$

**Conjecture 42.2** (Mansour's conjecture for size)**.** *For any $\varepsilon > 0$ there exists $c = c(\varepsilon) > 0$ such that the following holds. Any DNF of size $s$ can be $\varepsilon$-approximated by a polynomial of sparsity $c^{\log s}$.*

One of the motivations behind Mansour's conjecture, other than a better understanding of the structure of DNFs, is that it would give an efficient agnostic learning algorithm for DNFs [20, 19].

As was noted in [21], it makes sense to speculate a similar conjecture for bounded width DNFs. As any DNF of size $s$ can be approximated by a DNF of width $w = O(\log(s/\varepsilon))$, this latter conjecture is stronger.

**Conjecture 42.3** (Mansour's conjecture for width)**.** *For any $\varepsilon > 0$ there exists $c = c(\varepsilon) > 0$ such that the following holds. Any DNF of width $w$ can be $\varepsilon$-approximated by a polynomial of sparsity $c^w$.*

The best known bound for Mansour's conjecture for width [45] is that it holds for sparsity $(w \log(1/\varepsilon))^{O(w)}$ (the bound for size holds by approximating a bounded size DNF with a bounded width DNF). So again, we see the $w^w$ term appearing, where the conjecture asks if it can be improved to $c(\varepsilon)^{O(w)}$ (and moreover that $c(\varepsilon) = O(\log 1/\varepsilon)$). In fact, Mansour shows that his technique would not yield a $w^{o(w)}$-type bound, so other ideas are necessary. A direct corollary of our main theorem is that both versions of Mansour's conjecture are equivalent.

**Corollary 42.1.** *Conjecture 42.2 and conjecture 42.3 are equivalent.*

**Acknowledgement** Chapter 4 contains a reprint of material as it appears in Proceedings of the fifty-first Annual ACM symposium on Theory of computing 2019. Shachar Lovett and Jiapeng Zhang. *DNF sparsification beyond sunflowers.* The dissertation author was a primary investigator and author of this paper.

# Chapter 5

# Population Recovery

A huge number of Fourier-based learning algorithms have been discovered in the past three decades. [19, 20, 34, 37, 45]. One hand, a lot of simple functions can be approximated by sparse real polynomials; On the other hand, the Fourier-based learning algorithm provides a generic framework to learn sparse polynomials. In our work, we focus on learning distributions from noisy samples, this problem is called population recovery.

## 5.1 Noisy distribution

A formal description of the population recovery problem is as follows. Suppose there is an unknown distribution $\pi$ over $\{0,1\}^n$, and an error parameter $0 < \mu < 1$. Lossy samples from it are obtained as follows:

- Sample a string $x \in \{0,1\}^n$ according to $\pi$.
- Replace each coordinate of $x$ independently with a ? with probability $1-\mu$.

Noisy samples from it are obtained as follows:

- Sample a string $x \in \{0,1\}^n$ according to $\pi$.
- Flip each coordinate of $x$ independently with probability $(1-\mu)/2$.

In both cases, the goal is to reconstruct $\pi$ up to a small additive error $\varepsilon$. That is, we would like to output a list of strings $S$ and an estimate $\tilde{\pi}(x)$ for $x \in S$, such that $|\tilde{\pi}(x) - \pi(x)| \leq \varepsilon$ for all $x \in S$, and $\pi(x) < \varepsilon$ for $x \notin S$.

It should be clear that the problem is trivial when $\mu = 1$ (as no error is introduced), is harder the smaller $\mu$ is, and is intractable for $\mu = 0$. Moreover, the recovery problem from lossy samples is easier than the recovery problem from noisy samples, since if we replace each ? with a random bit, we obtain the noisy model. Indeed, the known algorithms for the lossy problem are better than those known for the noisy problem. In [15] a polynomial time algorithm (in $n, 1/\varepsilon$) for the lossy recovery problem was given whenever $\mu \gtrapprox 0.365$. This was improved to $\mu > 1 - 1/\sqrt{2} \approx 0.3$ in [6]. Finally, a polynomial time algorithm for any $\mu > 0$ was given in [47].

For the noisy problem, algorithms are known only when the support size of $\pi$, which we denote by $k$, is bounded. Kearns et al. [31] gave an algorithm which is exponential in $k$. Wigderson and Yehudayoff [57] developed a framework called ”partial identification”, and gave an algorithm which runs in time polynomial in $(k^{\log k}, n, 1/\varepsilon)$ for any $\mu > 0$. In this chapter, we show an alternative framework, which gives an algorithm running in time polynomial in $k^{\log \log k}$.

**Theorem 43** ([41])**.** *For any $\mu > 0$ there exists an algorithm for the noisy recovery problem, running in time poly$(k^{O_\mu(\log \log k)}, n, 1/\varepsilon)$.*

A subsequent work of De, Saks and Tang [14] (built on refined discrete Fourier analysis) further improved the upper bound to poly$(k)$.

**Theorem 44** ([14])**.** *For any $\mu > 0$, there exists an algorithm for the noisy population recovery problem, running in time poly$((k/\varepsilon)^{O_\mu(1)}, n)$.*

An interesting property of the noisy recovery problem is that the algorithmic problem reduces to a purely *information theoretic* problem. Recall the noise operator $T_\mu$ on functions

$f:\{0,1\}^n \to \mathbb{R}$ which defined as

$$(T_\mu f)(x) = \mathop{\mathbb{E}}_{e \sim \mathbb{N}_\mu} [f(x \oplus e)],$$

where $\mathbb{N}_\mu$ is the noisy distribution. If $\pi$ is a distribution on $\{0,1\}^n$, then $T_\mu \pi$ is the distribution of its noisy samples. Now, if $\pi_1, \pi_2$ are two distributions on $\{0,1\}^n$, each of support of size $k$ and with a noticeable statistical distance, then any recovery algorithm would need to distinguish the two noisy distributions. In particular, there should be noticeable statistical distance between $T_\mu \pi_1$ and $T_\mu \pi_2$. Surprisingly, it turns out that if this holds for any pair of distributions, then the noisy recovery problem can be solved efficiently, for example by computing the maximum likelihood estimator which is a convex optimization problem. See eg [6, 47] for details. Thus, we can formulate the following information theoretic problem, which is equivalent to the existence of efficient algorithms for noisy population recovery.

Let $f:\{0,1\}^n \to \mathbb{R}$ be a function of bounded support (e.g. $f = \frac{1}{2}(\pi_1 - \pi_2)$). Let $\|f\|_1 = \sum_x |f(x)|$. Define

$$\Delta(k,\mu) := \sup_{\mathrm{supp}(f) \le k} \frac{\|f\|_1}{\|T_\mu f\|_1}.$$

Then $\Delta(k,\mu)$ is a lower bound on any recovery algorithm for noisy population recovery with error $\varepsilon \le 1/k$; and on the other hand, the maximum likelihood estimator converges to the correct solution in time polynomial in $(\Delta(k,\mu), n, \varepsilon^{-1})$. Our main technical contribution is the following theorem, which shows that $\Delta(k,\mu) \le k^{O(\log\log k + \log 1/\mu)}$. Theorem 43 then follows by the above discussion.

**Theorem 45.** *Let* $f:\{0,1\}^n \to \mathbb{R}$ *with* $\mathrm{supp}(f) = k$*. Then*

$$\|T_\mu f\|_1 \ge k^{-O(\log\log k + \log 1/\mu)} \|f\|_1.$$

## 5.2 Proof overview

Let $f : \{0,1\}^n \to \mathbb{R}$ be a function with support of size $k$, where we may assume $\|f\|_1 = 1$. If we could find a noticeable Fourier coefficient $\widehat{f}(S)$ where $S$ has low hamming weight, we could lower bound $\|T_\mu f\|_1$ since

$$\|T_\mu f\|_1 \geq |\widehat{T_\mu f}(S)| = \mu^{|S|}|\widehat{f}(S)|.$$

As a first step, we show (Lemma 45.1) an extension of this lower bound. If we define a function $g(x) = f(x) \cdot \Pr_e[x \oplus e \in E]$, and $E \subset \{0,1\}^n$ is any subset, then

$$\|T_\mu f\|_1 \geq \mu^{|S|}|\widehat{g}(S)|.$$

Next, we choose the subset $E$ to control the properties of $g$. Let $\mathrm{supp}(f) = \{x_1, \ldots, x_k\}$ and assume $|f(x_1)|$ is maximal, and in particular $|f(x_1)| \geq 1/k$. We choose $E$ to contain only points which are closer to $x_1$ than to all the other $x_i$ which are far enough from $x_1$. With this choice, we prove (Lemma 45.2) that $g(x_1) \approx f(x_1)$ while $g(x_i)$ decays exponentially fast in the hamming distance between $x_1$ and $x_i$. This allows us to approximate $g$ by a function $h$ supported on a small hamming ball around $x_1$.

Finally, we restrict our attention to functions supported on small hamming balls. We show that if $h$ has support of size $k$ and is supported in a hamming ball of radius $r$, then there exists $S \subset [n]$ of size $|S| \leq \log k$ such that $|\widehat{h}(S)| \geq k^{-O(\log r)}$. Putting these together, it turns out that one should consider balls of radius $r = O(\log k \log\log k)$, which imply the bound.

## 5.3 Preliminary

For $x \in \{0,1\}^n$ let $|x|$ denote the hamming weight of $x$. For $x, y \in \{0,1\}^n$ let $\mathrm{dist}(x,y)$ denote their hamming distance. Let $B(n,r) = \{x \in \{0,1\}^n : |x| \leq r\}$ denote the hamming ball of radius $r$ in $\{0,1\}^n$. We will mostly be interested in the $L_1$ norm $\|f\|_1 = \sum |f(x)|$. For an operator $T : \mathscr{F} \to \mathscr{F}$ its $L_1$ to $L_1$ norm is defined as $\|T\|_{1\to 1} = \sup \|Tf\|_1/\|f\|_1$, where the supremum is taken over all nonzero functions. The support of a function $f$ is the set of elements with nonzero value, $\mathrm{supp}(f) = \{x : f(x) \neq 0\}$.

For a noise parameter $0 < \mu < 1$, let $\mathbb{N}_\mu$ denote the distribution of $e \in \{0,1\}^n$ given by $\Pr[e_i = 0] = (1+\mu)/2$ and $\Pr[e_i = 1] = (1-\mu)/2$ independently for all $i \in [n]$.

## 5.4 Lower bounding the norm of noisy functions

Let $f : \{0,1\}^n \to \mathbb{R}$ with bounded support. We restate Theorem 45 for the convenience of the reader.

**Theorem 45 (restated).** *Let $f : \{0,1\}^n \to \mathbb{R}$ with* $\mathrm{supp}(f) = k$. *Then*

$$\|T_\mu f\|_1 \geq k^{-O(\log\log k + \log 1/\mu)} \|f\|_1.$$

We may assume without loss of generality that $\|f\|_1 = 1$. A simple lower bound on $\|T_\mu f\|_1$ follows if $f$ has a noticeable Fourier coefficient of low hamming weight. For any $S \subset [n]$,

$$\|T_\mu f\|_1 \geq |\widehat{T_\mu f}(S)| = \mu^{|S|} |\widehat{f}(S)|.$$

As a first step, we show that the same bound holds if one replaces $f$ with any function of the form $g(x) = f(x) \Pr[x + e \in E]$, where $e \sim \mathbb{N}_\mu$ and $E \subset \{0,1\}^n$ is any subset.

**Lemma 45.1.** *Let $f : \{0,1\}^n \to \mathbb{R}$ be a function and let $E \subset \{0,1\}^n$. Define $g : \{0,1\}^n \to \mathbb{R}$ by*

$$g(x) = f(x) \Pr_{e \sim \mathbb{N}_\mu} [x + e \in E].$$

*Then for any $S \subset [n]$ we have*

$$\|T_\mu f\|_1 \geq \mu^{|S|} |\widehat{g}(S)|.$$

We defer the proof of this lemma to section 5.5. Assume that $\mathrm{supp}(f) = \{x_1, \ldots, x_k\}$ with $|f(x_1)| \geq 1/k$. We choose $E$ so that $g(x_1) \approx f(x_1)$ but $g(x_i)$ decays exponentially in $\mathrm{dist}(x_1, x_i)$. This will allow us to approximate $g$ by a function bounded in a hamming ball of low radius. Specifically, we choose

$$\begin{aligned} E = \big\{ y \in \{0,1\}^n : \mathrm{dist}(x_1, y) < \mathrm{dist}(x_i, y) \\ \text{for all } x_i \text{ such that } \mathrm{dist}(x_1, x_i) \geq \log(k)/\mu^2 \big\} \end{aligned} \tag{5.1}$$

**Lemma 45.2.** *For the set $E$ defined in* (5.1) *and $g = f \cdot T_\mu 1_E$ we have*

1. $|g(x_1)| \geq |f(x_1)|/2 \geq 1/2k$.
2. *If* $\mathrm{dist}(x_1, x_i) \geq \log(k)/\mu^2$ *then* $|g(x_i)| \leq |f(x_i)| \cdot \exp(-\mu^2 \cdot \mathrm{dist}(x_1, x_i))$.

We defer the proof of this lemma to section 5.5. As the values in $g$ decay exponentially fast, we can well approximate $g$ with a function supported on a hamming ball of low radius.

**Corollary 45.1.** *Let $f : \{0,1\}^n \to \mathbb{R}$ with $|\mathrm{supp}(f)| = k$, and let $g = f \cdot T_\mu 1_E$ for the set $E$ defined in* (5.1)*. For any $r \geq \log(k)/\mu^2$ there exist a function $h : \{0,1\}^n \to \mathbb{R}$ such that*

1. $\mathrm{supp}(h) \leq k$, $\mathrm{supp}(h) \subseteq B(n,r)$, $\|h\|_1 \geq 1/2k$.
2. $\|g - h\|_1 \leq \exp(-r\mu^2)\|f\|_1$. *In particular,* $|\widehat{g}(S)| \geq |\widehat{h}(S)| - \exp(-r\mu^2)\|f\|_1$ *for any* $S \subset [n]$.

*Proof.* Take $h(x) = g(x)$ if $|x| \leq r$, and $h(x) = 0$ otherwise. The properties follow immediately from Lemma 45.2. □

This motivates the study of functions supported in a hamming ball of low radius. We may assume the hamming ball is centered around 0 by shifting the function. We show that such functions have noticeable Fourier coefficients of low hamming weight.

**Lemma 45.3.** *Let $h : \{0,1\}^n \to \mathbb{R}$ be a function with $|\mathrm{supp}(h)| = k, \mathrm{supp}(h) \subseteq B(n,r)$. Then there exists $S \subset [n]$, $|S| \leq \log k$ such that*

$$|\widehat{h}(S)| \geq k^{-\log(4r)} \|h\|_1.$$

We defer the proof of this lemma to section 5.5. Theorem 45 follows by combining Lemma 45.1, Corollary 45.1 and Lemma 45.3.

*Proof of Theorem 45.* Assume without loss of generality that $\|f\|_1 = 1$. Set $E \subset \{0,1\}^n$ as given in (5.1), and set $g(x) = f(x)\Pr[x+e \in E]$ where $e \sim \mathbb{N}_\mu$. By Lemma 45.1 we have $\|T_\mu f\|_1 \geq |\widehat{g}(S)|\mu^{|S|}$ for all $S \subseteq [n]$. Let $r \geq \log(k)/\mu^2$ to be optimized later, and apply Corollary 45.1 to find a function $h : \{0,1\}^n \to \mathbb{R}$ such that $|\mathrm{supp}(h)| = k, \mathrm{supp}(h) \subset B(n,r)$ and $|\widehat{g}(S)| \geq |\widehat{h}(S)| - \exp(-r\mu^2)$. Applying Lemma 45.3 to $h$, there exists $S \subseteq [n]$, $|S| \geq \log k$ such that $|\widehat{h}(S)| \geq k^{-\log(4r)}\|h\|_1$. We also know that $\|h\|_1 \geq 1/2k$. Putting these together, we obtain the lower bound

$$\|T_\mu f\|_1 \geq \mu^{\log k}\left((1/2k) \cdot k^{-\log(4r)} - \exp(-r\mu^2)\right).$$

Setting $r = O(\log k \cdot \log\log k \cdot \log(1/\mu)/\mu^2)$ we get that $\exp(-r\mu^2) \leq (1/4k)k^{-\log(4r)}$ and hence

$$\|T_\mu f\|_1 \geq k^{-O(\log\log k + \log 1/\mu)}.$$

□

## 5.5 Missing proofs

**Proof of Lemma 45.1** We restate Lemma 45.1 for the convenience of the reader.

**Lemma 45.1 (restated).** *Let $f : \{0,1\}^n \to \mathbb{R}$ be a function and let $E \subset \{0,1\}^n$. Define $g : \{0,1\}^n \to \mathbb{R}$ by*

$$g(x) = f(x) \Pr_{e \sim \mathbb{N}_\mu}[x + e \in E].$$

*Then for any $S \subset [n]$ we have*

$$\|T_\mu f\|_1 \geq \mu^{|S|} |\widehat{g}(S)|.$$

We will need a few auxiliary claims first. For $i \in [n]$ define $T_{\mu,i} : \mathscr{F} \to \mathscr{F}$ to be the operator that adds noise only in coordinate $i$,

$$(T_{\mu,i} f)(x) = \frac{1+\mu}{2} \cdot f(x) + \frac{1-\mu}{2} \cdot f(x^i),$$

where $x^i$ is the element obtain by flipping the $i$-th bit of $x$. The following claim bounds the norm of $T_{\mu,i}$ and its inverse.

**Claim 45.1.** $\|T_{\mu,i}\|_{1\to 1} = 1$ *and* $\|T_{\mu,i}^{-1}\|_{1\to 1} = 1/\mu$.

*Proof.* The bound $\|T_{\mu,i} f\|_1 \leq \|f\|_1$ is immediate, and is tight for $f = 1$. To derive the bound on $T_{\mu,i}^{-1}$, let $x_0, x_1$ be such that $(x_0)_i = 0, (x_1)_i = 1$ and $(x_0)_j = (x_1)_j$ for all $j \neq i$. If $(f(x_0), f(x_1)) = (a,b)$ then $((T_{\mu,1}^{-1} f)(x_0), (T_{\mu,1}^{-1} f)(x_1)) = (1/2\mu) \cdot ((1+\mu)a - (1-\mu)b, -(1-\mu)a + (1+\mu)b)$. Then $|(T_{\mu,i}^{-1} f)(x_0)| + |(T_{\mu,i}^{-1} f)(x_1)| \leq (1/\mu)|a+b| = (1/\mu)(|f(x_0)| + |f(x_1)|)$. The claim follows by summing over all choices for $x_0, x_1$, and noting that the bound is tight for $f(x) = (-1)^{x_i}$. □

For $S \subset [n]$ define the operator $T_{\mu,S} : \mathscr{F} \to \mathscr{F}$ to add noise to the coordinates in $S$. Formally, $T_{\mu,S} = \prod_{i \in S} T_{\mu,i}$. Note that $T_\mu = T_{\mu,[n]}$. Claim 45.1 implies that

$$\|T_{\mu,S}\|_{1\to 1} \leq 1, \qquad \|T_{\mu,S}^{-1}\|_{1\to 1} \leq (1/\mu)^{|S|}. \tag{5.2}$$

*Proof of Lemma 45.1.* Note that for any two functions $f', f'' \in \mathscr{F}$ we have

$$\left\langle f', T_\mu f'' \right\rangle = \mathop{\mathbb{E}}_{e \sim \mathbb{N}_\mu} \sum_{x \in \{0,1\}^n} f'(x) f''(x+e) = \mathop{\mathbb{E}}_{e \sim \mathbb{N}_\mu} \sum_{x \in \{0,1\}^n} f'(x+e) f''(x) = \left\langle T_\mu f', f'' \right\rangle .$$

We have $g(x) = f(x) \cdot (T_\mu 1_E)(x)$. Define an operator $X_S : \mathscr{F} \to \mathscr{F}$ by $(X_S f)(x) = f(x) \chi_S(x)$. Then

$$\widehat{g}(S) = \sum_{x \in \{0,1\}^n} f(x) T_\mu 1_E(x) \chi_S(x) = \left\langle X_S f, T_\mu 1_E \right\rangle = \left\langle T_\mu X_S f, 1_E \right\rangle .$$

In particular, since $\|1_E\|_\infty = 1$ we obtain that

$$|\widehat{g}(S)| \leq \|T_\mu X_S f\|_1. \tag{5.3}$$

Next, let $S^c = [n] \setminus S$ be the complement of $S$, and decompose $T_\mu = T_{\mu,S} T_{\mu,S^c}$. Note that the operators $T_{\mu,S^c}$ and $X_S$ commute. Hence

$$T_\mu X_S f = T_{\mu,S} T_{\mu,S^c} X_S f = T_{\mu,S} X_S T_{\mu,S^c} f = T_{\mu,S} X_S T_{\mu,S}^{-1} T_\mu f.$$

To conclude, we bound

$$\|T_\mu X_S f\|_1 \leq \|T_{\mu,S}\|_{1 \to 1} \|X_S\|_{1 \to 1} \|T_{\mu,S}^{-1}\|_{1 \to 1} \|T_\mu f\|_1 \leq (1/\mu)^{|S|} \|T_\mu f\|_1,$$

where we apply Claim 45.1 and the obvious bound $\|X_S\|_{1 \to 1} = 1$. □

**Proof of Lemma 45.2** We restate Lemma 45.2 for the convenience of the reader.

**Lemma 45.2 (restated).** *For the set $E$ defined in* (5.1) *and $g = f \cdot T_\mu 1_E$ we have*

1. $|g(x_1)| \geq |f(x_1)|/2 \geq 1/2k$.

2. *If* $\mathrm{dist}(x_1, x_i) \geq \log(k)/\mu^2$ *then* $|g(x_i)| \leq |f(x_i)| \cdot \exp(-\mu^2 \cdot \mathrm{dist}(x_1, x_i))$.

*Proof.* We first lower bound $|g(x_1)|$. Let $s = \log(k)/\mu^2$. By definition $g(x_1) = f(x_1)\Pr[x_1 + e \in E]$, where $e \sim \mathbb{N}_\mu$. If we let $y = x_1 + e$ then we can upper bound the probability that $x_1 + e \notin E$ by the union bound

$$\Pr[x_1 + e \notin E] \leq \sum_{i:\mathrm{dist}(x_1,x_i)\geq s} \Pr[\mathrm{dist}(x_1,y) \geq \mathrm{dist}(x_i,y)].$$

Let $S_i$ denote the coordinates in which $x_1, x_i$ differ, where $|S_i| \geq s$. Then $dist(x_1,y) \geq \mathrm{dist}(x_i,y)$ iff the hamming weight of $e$ restricted to $S$ is at least $|S|/2$. As each bit of $e$ is 1 with probability $(1-\mu)/2$ independently, we apply the Chernoff bound and obtain

$$\Pr[\mathrm{dist}(x_1,y) \geq \mathrm{dist}(x_i,y)] = \Pr\left[\sum_{j\in S_i} e_j \geq |S_i|/2\right] \leq \exp(-2|S_i|\mu^2) \leq 1/2k.$$

Hence $\Pr[x_1 + e \in E] \geq 1/2$ and $|g(x_1)| \geq |f(x_1)|/2$. To upper bound $|g(x_i)|$, we upper bound $\Pr[x_i + e \in E]$. Now, if $x_i + e \in E$ then in particular $\mathrm{dist}(x_1, x_i + e) < \mathrm{dist}(x_i, x_i + e)$, or equivalently the hamming weight of $e$ restricted to $S_i$ exceeds $|S_i|/2$. Applying again the Chernoff bound,

$$\Pr[x_i + e \in E] \leq \Pr\left[\sum_{j\in S_i} e_j \geq |S_i|/2\right] \leq \exp(-2|S_i|\mu^2).$$

□

**Proof of Lemma 45.3** We restate Lemma 45.3 for the convenience of the reader. For convenience, we denote the function studied by $f$.

**Lemma 45.3 (restated).** *Let $f : \{0,1\}^n \to \mathbb{R}$ be a function with $|\mathrm{supp}(f)| = k, \mathrm{supp}(f) \subseteq B(n,r)$. Then there exists $S \subset [n]$, $|S| \leq \log k$ such that*

$$|\widehat{f}(S)| \geq k^{-\log(4r)} \|f\|_1.$$

In order to prove Lemma 45.3, we find a low degree polynomial $p$ which computes $f$ on its support. A function $p : \{0,1\}^n \to \mathbb{R}$ is a degree $d$ polynomial if $p(x) = \sum_{|S| \le d} p_S \cdot \chi_S(x)$. We note that in our normalization, $\widehat{p}(S) = 2^n p_S$. To simplify notation define $|p| = \sum_S |p_S|$.

**Proposition 45.1.** *Let $f : \{0,1\}^n \to \mathbb{R}$ be a function with $|\mathrm{supp}(f)| = k, \mathrm{supp}(f) \subseteq B(n,r)$. Then there exists a polynomial $p$ of degree at most $\log k$ such that*

*(i)* $p(x) = f(x)$ *for all* $x \in \mathrm{supp}(f)$.

*(ii)* $|p| \le k \cdot r^{\log k} \cdot \|f\|_1$.

We first show that Lemma 45.3 follows immediately from Proposition 45.1.

*Proof of Lemma 45.3 from Proposition 45.1.* Consider $\langle f, p \rangle$. On the one hand,

$$\langle f, p \rangle = \sum_x f(x) p(x) = \sum_x f(x)^2 \ge 1/k.$$

On the other hand, by Parseval's identity,

$$\langle f, p \rangle = \sum_S \widehat{f}(S) p_S \le \max\{|\widehat{f}(S)| : |S| \le \log k\} \cdot |p|.$$

Hence there exists $S \subset [n]$, $|S| \le \log k$ such that $\widehat{f}(S) \ge 1/(k|p|)$. □

We now move to prove Proposition 45.1 by induction. We first define $F(k,r)$ to be the minimal bound on $|p|$ for which Proposition 45.1 holds. For technical reasons, we will require $p(x) = f(x)$ also for some $x$ outside the support of $f$. Formally, we define $f : X \to \mathbb{R}$ where we implicitly assume that $f(x) = 0$ for all $x \notin X$, but it could be that $f(x) = 0$ for some $x \in X$. We require that $p(x) = f(x)$ for all $x \in X$.

**Definition 46** ($F(k,r)$ function)**.** *For $k, r \ge 1$ define $F(k,r) \ge 0$ to be the minimal quantity such that the following holds. For any $n \ge 1$, any set $X \subset B(n,r)$ of size $|X| \le k$ and any function $f : X \to \mathbb{R}$, there exists a polynomial $p$ of degree at most* $\log k$ *such that* *(i)* $p(x) = f(x)$ *for all* $x \in X$.

*(ii)* $|p| \leq F(k,r)\|f\|_1$.

*If no such polynomial exists, set* $F(k,r) = \infty$.

We will also need a refinement based on the sum of hamming weights in $X$. Define $W(X) = \sum_{x \in X} |x|$ to be the sum of hamming weights in $X$.

**Definition 47** ($F(k,r;w)$ function)**.** *For* $k,r,w \geq 1$ *define* $F(k,r;w) \geq 0$ *to be the minimal quantity such that the following holds. For any* $n \geq 1$*, any set* $X \subset B(n,r)$ *of size* $|X| \leq k$ *and* $W(X) \leq w$ *and any function* $f : X \to \mathbb{R}$*, there exists a polynomial* $p$ *of degree at most* $\log k$ *such that*

*(i)* $p(x) = f(x)$ *for all* $x \in X$.

*(ii)* $|p| \leq F(k,r;w)\|f\|_1$.

*If no such polynomial exists, set* $F(k,r;w) = \infty$.

Note that $F(k,r;kr) = F(k,r)$. We now prove a recursive formula on $F(k,r;w)$;

**Proposition 47.1.** $F(k,r;w) \leq \max_{1 \leq a \leq k/2}\{F(k,r;w-a) + F(a,r)\}$.

*Proof.* We prove the proposition by induction on $n$. Let $X \subset B(n,r)$ with $|X| \leq k$, $W(x) \leq w$ and let $f : X \to \mathbb{R}$. We assume without loss of generality that $\|f\|_1 = 1$. Define $X_0, X_1, X_* \subseteq \{0,1\}^{n-1}$ as

$$X_0 = \{x \in \{0,1\}^{n-1} : x0 \in X, x1 \notin X\},$$

$$X_1 = \{x \in \{0,1\}^{n-1} : x1 \in X, x0 \notin X\},$$

$$X_* = \{x \in \{0,1\}^{n-1} : x0, x1 \in X\}.$$

Note that $X_0, X_1, X_*$ are disjoint and that $|X_0| + |X_1| + 2|X_*| = |X| \leq k$. Let $\{i, j\} = \{0, 1\}$ be such that $|X_i| \leq |X_j|$. Define $Y, Z \subseteq \{0,1\}^{n-1}$ by

$$\begin{aligned} Y &= X_0 \cup X_1 \cup X_* \\ Z &= X_i \cup X_* \end{aligned}$$

Note that by our assumption, $|Z| \leq k/2$. If $|Z| = 0$ then the last bit in all elements of $X$ is always $j$, hence we can reduce to dimension $n-1$ and continue by induction. Thus, we assume that $|Z| \geq 1$. Define a function $g : Y \to \mathbb{R}$ by

$$g(x) = \begin{cases} f(xj) & \text{if } x \in X_j \cup X_* \\ f(xi) & \text{if } x \in X_i \end{cases}$$

and a function $h : X \to \mathbb{R}$ by

$$h(x) = \begin{cases} 0 & \text{if } x \in X_i \\ f(xi) - f(xj) & \text{if } x \in X_* \end{cases}.$$

Let $x = x'x_n$ with $x' \in \{0,1\}^{n-1}, x_n \in \{0,1\}$ and observe that for all $x \in X$,

$$f(x) = g(x') + h(x') \cdot 1_{x_n = i}. \tag{5.4}$$

We now apply the proposition inductively to $g, h$. For $g$, we have $\|g\|_1 \leq 1$, $Y \subset B(n-1, r)$, $|Y| \leq k$ and $W(Y) = W(X) - |X_1| - |X_*| - W(X_*) \leq w - |Z|$. Hence there exists a polynomial $p_g$ of degree $\log k$ such that $p_g(x') = g(x')$ for all $x' \in Y$ and $|p_g| \leq F(k, r; w - |Z|)$. For $h$, we have $\|h\|_1 \leq 1$, $Z \subset B(n-1, r)$ and $|Z| \leq k/2$. Hence there exists a polynomial $p_h$ of degree

$\log|Z| \leq \log k - 1$ such that $p_h(x') = h(x')$ for all $x' \in Z$ and $|p_h| \leq F(|Z|, r)$. Define

$$p(x) = p_g(x') + p_h(x') 1_{x_n = i}$$

so that $p(x) = f(x)$ for all $x \in X$. Note that since $\deg(p_g) \leq \log k, \deg(p_h) \leq \log k - 1$ then $\deg(p) \leq \log k$. Finally, we bound $|p|$ by

$$|p| \leq |p_g| + |p_h 1_{x_n=i}| = |p_g| + |p_h||1_{x_n=i}| = |p_g| + |p_h| \leq F(k, r; w - |Z|) + F(|Z|, r).$$

□

**Proposition 47.2.** $F(k, r) \leq k \cdot r^{\log k}$.

*Proof.* As $r$ never changes throughout the induction, set $G(k) = F(k, r)$ and $G(k; w) = F(k, r; w)$. We prove the proposition by induction on $k$. By Proposition 47.1 we have

$$G(k; w) \leq \max_{1 \leq a \leq k/2} \left\{ G(k; w - a) + G(a) \right\}.$$

Expanding $G(k; w - a)$ recursively, we obtain the bound

$$G(k) = G(k; kr) \leq \max_{a_1 + \ldots + a_t \leq kr, 1 \leq a_1, \ldots, a_t \leq k/2} \left\{ \sum_{i=1}^{t} G(a_i) \right\}. \tag{5.5}$$

Let $a_1, \ldots, a_t$ be the parameters that maximize (5.5). By induction, $G(a_i) \leq a_i r^{\log a_i} \leq a_i r^{\log k - 1}$, where we used the fact that $a_i \leq k/2$. Hence

$$G(k) \leq \left( \sum_{i=1}^{t} a_i \right) r^{\log k - 1} \leq kr \cdot r^{\log k - 1} = k \cdot r^{\log k}.$$

□

## 5.6 Open problems

In this paper, we proved an improved bound $k^{O(\log\log k)}$ for population recovery. In a subsequent work of De, Saks and Tang [14] further improved it to $k^{O(1)}$. In both of our studies, we assumed the noisy parameter $\mu$ is known. It is more interesting to consider the case of unknown noise. In a joint work with Lovett [42], we have a quasi-polynomial time algorithm to solve this question, and our algorithm was built on framework of [57]. It is interesting to ask whether we can get unknown noise population recovery based on discrete Fourier analysis.

**Acknowledgement** Chapter 5 contains a reprint of material as it appears in Proceedings of the forty-seventh Annual ACM symposium on Theory of computing 2015. Shachar Lovett and Jiapeng Zhang. *Improved noisy population recovery, reverse Bonami-Beckner inequality for sparse functions.* The dissertation author was a primary investigator and author of this paper.

# Bibliography

[1] Adi Akavia, Shafi Goldwasser, and Samuel Safra. Proving hard-core predicates using list decoding. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 146. IEEE Computer Society, 2003.

[2] Andris Ambainis, Mohammad Bavarian, Yihan Gao, Jieming Mao, Xiaoming Sun, and Song Zuo. Tighter relations between sensitivity and other complexity measures. In *Automata, Languages, and Programming: 41st International Colloquium, ICALP 2014, Proceedings, Part I*, pages 101–113. Springer, 2014.

[3] Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Sensitivity versus certificate complexity of boolean functions. In *International Computer Science Symposium in Russia*, pages 16–28. Springer, 2016.

[4] Andris Ambainis and Xiaoming Sun. New separation between s(f) and bs(f). *arXiv preprint arXiv:1108.3494*, 2011.

[5] Nikhil Balaji, Samir Datta, Raghav Kulkarni, and Supartha Podder. Graph properties in node-query setting: Effect of breaking symmetry. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, pages 17:1–17:14, 2016.

[6] Lucia Batman, Russell Impagliazzo, Cody Murray, and Ramamohan Paturi. Finding heavy hitters from lossy or noisy data. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 347–362. Springer, 2013.

[7] Paul Beame. A switching lemma primer. Technical report, Technical Report UW-CSE-95-07-01, Department of Computer Science and Engineering, University of Washington, 1994.

[8] Shalev Ben-David, Pooya Hatami, and Avishay Tal. Low-sensitivity functions from unambiguous certificates. In *Conference on Innovations in Theoretical Computer Science (ITCS 2017)*, 2017.

[9] Richard E. Blahut. Transform techniques for error control codes. *IBM Journal of Research and development*, 23(3):299–315, 1979.

[10] Harry Buhrman and Ronald De Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[11] Sourav Chakraborty. *Sensitivity, block sensitivity and certificate complexity of Boolean functions*. PhD thesis, Masters thesis, University of Chicago, USA, 2005. 23, 2005.

[12] Yi-Hsiu Chen, Mika Göös, Salil P Vadhan, and Jiapeng Zhang. A tight lower bound for entropy flattening. In *33rd Computational Complexity Conference (CCC 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[13] Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 504–517. Springer, 2010.

[14] Anindya De, Michael Saks, and Sijian Tang. Noisy population recovery in polynomial time. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 675–684. IEEE, 2016.

[15] Zeev Dvir, Anup Rao, Avi Wigderson, and Amir Yehudayoff. Restriction access. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 19–33. ACM, 2012.

[16] Paul Erdős and R Rado. Intersection theorems for systems of sets. *Journal of the London Mathematical Society*, 35(1):85–90, 1960.

[17] Anna Gál and Adi Rosén. A theorem on sensitivity and applications in private computation. *SIAM Journal on Computing*, 31(5):1424–1437, 2002.

[18] Justin Gilmer, Michal Kouckỳ, and Michael E Saks. A new approach to the sensitivity conjecture. In *Conference on Innovations in Theoretical Computer Science (ITCS 2015)*, pages 247–254. ACM, 2015.

[19] Parikshit Gopalan, Adam Kalai, and Adam R Klivans. A query algorithm for agnostically learning DNF?. In *COLT*, pages 515–516, 2008.

[20] Parikshit Gopalan, Adam Tauman Kalai, and Adam R Klivans. Agnostically learning decision trees. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 527–536. ACM, 2008.

[21] Parikshit Gopalan, Raghu Meka, and Omer Reingold. DNF sparsification and a faster deterministic counting algorithm. *Computational Complexity*, 22(2):275–310, 2013.

[22] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 120–129. IEEE, 2012.

[23] Parikshit Gopalan, Noam Nisan, Rocco A Servedio, Kunal Talwar, and Avi Wigderson. Smooth boolean functions are easy: efficient algorithms for low-sensitivity functions. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 59–70. ACM, 2016.

[24] Parikshit Gopalan, Rocco Servedio, Avishay Tal, and Avi Wigderson. Degree and sensitivity: tails of two distributions. *arXiv preprint arXiv:1604.07432*, 2016.

[25] Craig Gotsman and Nathan Linial. The equivalence of two problems on the cube. *Journal of Combinatorial Theory, Series A*, 61(1):142–146, 1992.

[26] Johan Håstad. Computational limitations of small-depth circuits. 1987.

[27] Pooya Hatami, Raghav Kulkarni, and Denis Pankratov. Variations on the sensitivity conjecture. *Theory of Computing, Graduate Surveys*, 4:1–27, 2011.

[28] Kun He, Qian Li, and Xiaoming Sun. A tighter relation between sensitivity and certificate complexity. *arXiv preprint arXiv:1609.04342*, 2016.

[29] Kun He, Qian Li, Xiaoming Sun, and Jiapeng Zhang. Quantum lovász local lemma: Shearer's bound is tight. In *Proceedings of the 51st annual ACM symposium on Theory of computing*, 2019.

[30] Daniel M Kane, Shachar Lovett, Shay Moran, and Jiapeng Zhang. Active classification with comparison queries. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 355–366. IEEE, 2017.

[31] Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 273–282. ACM, 1994.

[32] Claire Kenyon and Samuel Kutin. Sensitivity, block sensitivity, and -block sensitivity of boolean functions. *Information and Computation*, 189(1):43–53, 2004.

[33] Shrinivas Kudekar, Santhosh Kumar, Marco Mondelli, Henry D Pfister, Eren Şaşolu, and Rüdiger L Urbanke. Reed–muller codes achieve capacity on erasure channels. *IEEE Transactions on information theory*, 63(7):4298–4316, 2017.

[34] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[35] Xin Li, Shachar Lovett, and Jiapeng Zhang. Sunflowers and quasi-sunflowers from randomness extractors. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[36] Chengyu Lin and Shengyu Zhang. Sensitivity conjecture and log-rank conjecture for functions with small alternating numbers. *arXiv preprint arXiv:1602.06627*, 2016.

[37] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. In *30th Annual Symposium on Foundations of Computer Science*, pages 574–579. IEEE, 1989.

[38] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, 40(3):607–620, 1993.

[39] Shachar Lovett, Noam Solomon, and Jiapeng Zhang. From dnf compression to sunflower theorems via regularity. *arXiv preprint arXiv:1903.00580*, 2019.

[40] Shachar Lovett, Avishay Tal, and Jiapeng Zhang. The robust sensitivity of boolean functions. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1822–1833. SIAM, 2018.

[41] Shachar Lovett and Jiapeng Zhang. Improved noisy population recovery, and reverse bonami-beckner inequality for sparse functions. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 137–142. ACM, 2015.

[42] Shachar Lovett and Jiapeng Zhang. Noisy population recovery from unknown noise. In *Conference on Learning Theory*, pages 1417–1431, 2017.

[43] Shachar Lovett and Jiapeng Zhang. On the impossibility of entropy reversal, and its application to zero-knowledge proofs. In *Theory of Cryptography Conference*, pages 31–55. Springer, 2017.

[44] Shachar Lovett and Jiapeng Zhang. Dnf sparsification beyond sunflowers. In *Proceedings of the 51st annual ACM symposium on Theory of computing*, 2019.

[45] Yishay Mansour. An $n^{O(\log\log n)}$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50(3):543–550, 1995.

[46] James L Massey. The discrete fourier transform in coding and cryptography. 1998.

[47] Ankur Moitra and Michael Saks. A polynomial time algorithm for lossy population recovery. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 110–116. IEEE, 2013.

[48] Noam Nisan. Crew prams and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991.

[49] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational complexity*, 4(4):301–313, 1994.

[50] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[51] Alexander A Razborov. Bounded arithmetic and lower bounds in boolean complexity. In *Feasible Mathematics II*, pages 344–386. Springer, 1995.

[52] Benjamin Rossman. The monotone complexity of k-clique on random graphs. *SIAM Journal on Computing*, 43(1):256–279, 2014.

[53] Benjamin Rossman. An entropy proof of the switching lemma and tight bounds on the decision-tree size of ac0. 2017.

[54] Karthik C. S. and Sébastien Tavenas. On the sensitivity conjecture for disjunctive normal forms. In *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2016, December 13-15, 2016, Chennai, India*, pages 15:1–15:15, 2016.

[55] Hans-Ulrich Simon. A tight $\Omega(\log\log n)$-bound on the time for parallel ram's to compute nondegenerated boolean functions. In *International Conference on Fundamentals of Computation Theory*, pages 439–444. Springer, 1983.

[56] Bo Tang and Jiapeng Zhang. Barriers to black-box constructions of traitor tracing systems. In *Theory of Cryptography Conference*, pages 3–30. Springer, 2017.

[57] Avi Wigderson and Amir Yehudayoff. Population recovery and partial identification. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 390–399. IEEE, 2012.