Testing and Learning Boolean Functions

by

Kevin Michael Matulef

Sc.B., Brown University (2002)
C.A.S.M., University of Cambridge (2003)

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2009

Author . . . .
Department of Mathematics
July 17, 2009

Certified by . . . .
Ronitt Rubinfeld
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Michel X. Goemans
Chairman, Applied Mathematics Committee

Accepted by . . . .
David Jerrison
Chairman, Department Committee on Graduate Students

Testing and Learning Boolean Functions

by

Kevin Michael Matulef

Submitted to the Department of Mathematics
on July 17, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Given a function f on n inputs, we consider the problem of testing whether f belongs to a concept class C, or is far from every member of C. An algorithm that achieves this goal for a particular C is called a *property testing* algorithm, and can be viewed as relaxation of a proper learning algorithm, which must also return an approximation to f if it is in C. We give property testing algorithms for many different classes C, with a focus on those that are fundamental to machine learning, such as halfspaces, decision trees, DNF formulas, and sparse polynomials. In almost all cases, the property testing algorithm has query complexity independent of n, better than the best possible learning algorithm.

Thesis Supervisor: Ronitt Rubinfeld
Title: Professor of Electrical Engineering and Computer Science

Acknowledgements

Put simply, this thesis would not exist without the support, guidance, optimism and good humor of my advisor, Ronitt Rubinfeld. Ronitt introduced me to the research problems that follow, and encouraged me in every one of my endeavors, even those that had nothing to do with research. She is the model advisor, and I cannot thank her enough.

One of the luckiest breaks I had in graduate school was the several summers I spent working with Rocco Servedio. Rocco, I think, is some sort of anomaly; he manages to make the entire process of doing research seem effortless and fun. The keenness of his intellect is matched only by the drollness of his wit. And despite being born in Canada, he is a true American.

I am grateful to the other members of my committee, Mike Sipser and Jon Kelner: Mike helped foster my transition to graduate school, and over the years he has always been available for advice on research, life, and whatever else. I had the pleasure of serving as his teaching assistant, and his reputation as an excellent teacher is well deserved; in the future I can only hope to emulate his clarity.

Jon has assisted me in more capacities than I am able to to list. From applicant to graduate student to professor, he has consistently been one step ahead of me, and from his forward vantage point, kind enough to turn around and tell me what he sees.

I would like to thank all of my coauthors (in no particular order): Ryan O'Donnell, Andrew Wan, Homin Lee, Ilias Diakonikolas, Rocco Servedio, Ronitt Rubinfeld, Krzysztof Onak, Ning Xie, Alexandr Andoni, Tali Kaufman, and Noga Alon. All were a pleasure to work with, and it goes without saying that many of the ideas on the following pages are due to them.

I would like to thank all of my friends; whatever sanity I have left is surely due to them.

Finally, I would like to thank my family- my siblings, cousins, and especially my parents, who taught me my first algorithms, without realizing that's what they were doing.

Kevin Matulef

17 July 2009

Bibliographic Note

This thesis is in large part based upon joint work with other researchers, much of which already appears in published form. The results on testing halfspaces in Chapters 5 and 6 first appeared in [44] and [43] respectively; both were joint work with Ryan O'Donnell, Ronitt Rubinfeld, and Rocco Servedio. The results on testing for concise representations in Chapter 3 first appeared in [17], and were joint work with Ilias Diakonikolas, Homin Lee, Krzysztof Onak, Ronitt Rubinfeld, Rocco Servedio, and Andrew Wan. The result on testing sparse polynomials in Chapter 4 first appeared in [18], and was joint work with Ilias Diakonikolas, Homin Lee, Rocco Servedio, and Andrew Wan.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is about *learning* and *testing*, and the relationship between them. Although this seems to be a topic for a different kind of thesis- perhaps one on education, rather than mathematics or computer science- there is a twist. The objects of our study, the ones doing the learning and testing, are not people, but machines.

What resources are required for a machine to learn a new concept? And what resources are required for a machine to test whether its understanding of a concept is accurate? These are the questions that lie at the heart of this thesis. They are difficult questions to formalize, and we will not claim to provide definitive answers to them here, but we will look at two reasonable and established ways that the theoretical computer science community has modeled them, and compare the two models. What we will find is that for a variety of types of concepts, there is a rich relationship between the problem of learning a new concept and the problem of testing whether a concept has a given form. Techniques from one are often applicable to the other, and in most cases, the testing problem requires significantly fewer resources.

1.1 Computational learning theory

We formalize the notion of a concept by representing it as a *classification function.* For example, consider the "concept" of email spam. Surely, we each know spam messages when we see them. In fact for each of us, there is some function f, consisting of a complex set of rules, that correctly classifies our emails as "spam" or "not spam" according to our tastes. Unfortunately, this function f is not known to the computer that handles our email (and often it's not consciously known to us!). We cannot expect the computer to read our minds, but we *can* hope that by observing some emails that have already been classified, the computer will be able to infer a good approximation to f, which it can use to automatically classify emails in the future.

The spam scenario summarizes the goal of a learning algorithm: given access to examples labeled according to an unknown function f, design a hypothesis h that is a good approximation to f. The hypothesis is considered "good," or close to f, if it is likely to classify a random input the same as f. More formally, if Ω is a distribution over the domain of f, then we say that a hypothesis h is *ϵ-close to f* with respect to Ω if $\Pr_{x \sim \Omega}[f(x) \neq h(x)] \leq \epsilon$.

In general, without making any assumptions, it is impossible to recover a good approximation to f without seeing nearly all of f (i.e. seeing labeled examples of the form $(x, f(x))$ for nearly every input x). The reason, of course, is that the unseen part of f could differ arbitrarily from any purported approximation. However, if one assumes that f comes from some *class* of functions $\mathcal{C}$, then the problem of approximating f may become tractable. The question then becomes: for which concept classes $\mathcal{C}$ is this the case?

One of the most well-known ways to formalize this question is via the *Probably Approximately Correct* (PAC) model of Valiant [61]:

Definition 1 (PAC Learning). *Let $\mathcal{C}$ and $\mathcal{H}$ be classes of functions. We say that $\mathcal{C}$ is* PAC Learnable *by $\mathcal{H}$ using algorithm $\mathcal{A}$ if when given*

1. *parameters $0 < \epsilon, \delta < 1$, and*

2. *access to points distributed according to a fixed distribution Ω and labeled by an unknown function $f \in C$,*

with probability at least $1 - \delta$, $\mathcal{A}$ outputs a hypothesis $h \in H$ that is ϵ-close to f.

The model has many parameters, but the origin of the "PAC" name should be clear. The "probably" comes from the fact that the learning algorithm succeeds with probability at least $1 - \delta$, while the "approximately" comes from the fact that the learning algorithm outputs an ϵ-approximation.

The PAC model has many variants, a few of which will be important to us here. If the algorithm $\mathcal{A}$ requires the ability to query f, we refer to it as *learning with queries*. If the hypothesis class $\mathcal{H}$ is the same as $\mathcal{C}$, we say that $\mathcal{C}$ is *properly* learnable. Finally, if the distribution Ω is the uniform distribution then we say that $\mathcal{C}$ is PAC learnable with *respect to the uniform distribution.*

1.2 Property testing

The learning problem, as formalized by computational learning theory, is in some sense a problem of emulation; the goal of a learning algorithm is to find a way to emulate any function f, assuming it comes from a pre-specified class $\mathcal{C}$. But instead of full-blown emulation, we might simply want to determine something about f, without making any prior assumptions. Specifically we might ask: given access to a unknown function f, can we simply determine if it belongs to the class $\mathcal{C}$? Although providing an exact answer to this question requires seeing all of f, if we ask it in the right way, we might hope to get an approximate answer.

Property testing, initiated by Rubinfeld and Sudan [54] and Goldreich, Golwasser, and Ron [29], provides a framework for doing this. The goal of a property testing algorithm is to accept an object if it has a certain property, and reject the object if it *far* from having the property. In our case, the objects are functions, the property is membership in the class $\mathcal{C}$, and being far from having the property means being ϵ-far from every function in $\mathcal{C}$.

Definition 2 (Property Testing). *A* property tester *for a class $\mathcal{C}$ is an algorithm that when given query access to a function f, outputs*

- *YES with probability $\geq 2/3$ if $f \in \mathcal{C}$*
- *NO with probability $\geq 2/3$ if f is ϵ-far from all $f' \in \mathcal{C}$.*

The complexity of a testing algorithm is measured both in terms of the number of black-box queries it makes to f (*query complexity*) as well as the time it takes to process the results of those queries (*time complexity*). Since queries are often assumed to be the limiting resource, we will primarily concern ourselves with query complexity, but both measures are important.

1.3 Learning versus testing

In [29], Goldreich *et al.* observed that the testing problem is essentially a relaxation of the proper learning problem. This is because an algorithm for properly learning a class $\mathcal{C}$ (more precisely, for properly learning $\mathcal{C}$ under the uniform distribution with queries) can be converted into an algorithm for testing membership in $\mathcal{C}$. To test whether a function f is in $\mathcal{C}$, we can simply run the proper learning algorithm with parameter $\epsilon/2$ to obtain a hypothesis f', and then evaluate f and f' over $O(1/\epsilon)$ random examples to verify that they are close. If f belongs to $\mathcal{C}$, then the proper learning algorithm will find an $\epsilon/2$-accurate hypothesis f', so a multiplicative Chernoff bound can be used to show that the verification step passes; if f is ϵ-far from $\mathcal{C}$, then it is clearly impossible to find such an f', so the verification step fails. (Note here that it is crucial the learning algorithm be proper. Otherwise, even if the algorithm verifies that f' is close to f, there is no guarantee that f is close to $\mathcal{C}$.)

The disadvantage to using learning algorithms for testing is that proper learning algorithms for virtually every interesting class of n-variable functions must make a number of queries that depends on n. The reason is information-theoretic. Suppose a class $\mathcal{C}$ contains k different functions that are all a constant distance c away from each other. In order for the learning algorithm to output a good hypothesis when $\epsilon < c/2$, it must have the capacity to output any of the k different functions, and thus must make at least $\lceil \log k \rceil$ queries merely to specify which one. This means that learning algorithms even for such simple classes as dictator functions (i.e. single boolean literals, a class that contains $2n$ functions, all at least distance $1/2$ from each other) must make $\Omega(\log n)$ queries. Thus, this raises the natural question: when does testing require strictly fewer queries than learning? Is it possible to test membership in a class $\mathcal{C}$ with a number of queries that is less than $O(\log n)$, or perhaps even *independent* of n?

1.4 Our results

The main result of this thesis is that a large number of classes are testable with query complexity independent of the dimension n, and polynomial in the other relevant parameters. This asympototically beats the query complexity of the best possible learning algorithms for these classes. Our primary focus is on classes of Boolean functions, but our techniques are often not limited to the Boolean case. For some classes, such as s-sparse polynomials over $GF(2)$ and ± 1-weight halfspaces, we prove more fine-grained results, showing lower bounds as well as upper bounds. Table 1.1 summarizes our testing results, as well as a selection of previously known results.

Here in more detail is a description of what we prove, along with an outline of the rest of the thesis:

- In Chapter 3, we describe a general method for testing whether a function on n input variables has a concise representation. The method, called "testing by implicit learning," combines ideas from the junta tests of [26] and [6] with ideas from learning theory, and yields property testers that make poly(s/ϵ) queries (independent of n) for Boolean function classes such as s-term DNF formulas (answering the question posed by [51]), size-s decision trees, size-s Boolean formulas, and size-s Boolean circuits, as well as non-Boolean valued function classes such as size-s algebraic circuits and s-sparse polynomials over finite fields.

  We also prove an $\tilde{\Omega}(\sqrt{s})$ query lower bound for nonadaptively testing s-sparse polynomials over finite fields of constant size. This shows that in some instances, our general method yields a property tester with query complexity that is optimal (for nonadaptive algorithms) up to a polynomial factor.

- In Chapter 4 we focus specifically on the problem of testing s-sparse $GF(2)$ polynomials. In contrast to the algorithm from Chapter 3, which makes poly$(s, 1/\epsilon)$ queries but has running time exponential in s and super-polynomial in $1/\epsilon$, in Chapter 4 we give an algorithm that makes $\text{poly}(s, 1/\epsilon)$ queries and also runs in time $\text{poly}(s, 1/\epsilon)$. We achieve this result by extending the "testing by implicit learning" methodology

Class of functions	Number of Queries	Reference
Boolean functions $f : \{0,1\}^n \to \{0,1\}$		
linear functions (parities)	$O(1/\epsilon)$	[9]
Boolean literals (dictators), conjunctions	$O(1/\epsilon)$	[51]
s-term monotone DNFs	$\tilde{O}(s^2/\epsilon)$	[51]
J-juntas	$\tilde{O}(J^2/\epsilon)$ $\tilde{O}(J/\epsilon)$ $\Omega(J)$ (*adaptive*)	[26] [6] [13]
decision lists	$\tilde{O}(1/\epsilon^2)$	Chapter 3
size-s decision trees, size-s branching programs, size-s Boolean formulas, s-term DNFs	$\tilde{O}(s^4/\epsilon^2)$ $\Omega(\log s/\log\log s)$ (*adaptive*)	Chapter 3
s-sparse polynomials over $\mathbb{F}_2$	$\tilde{O}(s^4/\epsilon^2)$, $\tilde{\Omega}(\sqrt{s})$ $O(\text{poly}(s, 1/\epsilon))$ (*time*)	Chapter 3 Chapter 4
size-s Boolean circuits	$\tilde{O}(s^6/\epsilon^2)$	Chapter 3
functions with Fourier degree $\leq d$	$\tilde{O}(2^{6d}/\epsilon^2)$, $\tilde{\Omega}(\sqrt{d})$	Chapter 3
halfspaces	$O(\text{poly}(1/\epsilon))$	Chapter 5
±1-weight halfpsaces	$\tilde{O}(\sqrt{n}/\epsilon^6)$ $\Omega(\log n)$	Chapter 6
Functions on Finite Fields $f : \mathbb{F}^n \to \mathcal{Y}$		
s-sparse polynomials over field of size $\lvert\mathbb{F}\rvert$	$\tilde{O}((s\lvert\mathbb{F}\rvert)^4/\epsilon^2)$, $\tilde{\Omega}(\sqrt{s})$ for $\lvert\mathbb{F}\rvert = O(1)$	Chapter 3
size-s algebraic circuits, size-s algebraic computation trees over $\mathbb{F}$	$\tilde{O}(s^4 \log^3 \lvert\mathbb{F}\rvert/\epsilon^2)$	Chapter 3

Table 1.1: **Selected previous and new testing results.** All the bounds pertain to query complexity, except where indicated by (*time*). The lower bounds are for non-adaptive algorithms, except where indicated by (*adaptive*). Finally, the upper bounds in Chapter 3 are for adaptive algorithms, though in all cases very similar bounds for non-adaptive algorithms can be achieved (see Section 3.4).

from Chapter 3. While the learning component from Chapter 3 involves a brute-force exhaustive search over a concept class, here the learning component is a sophisticated exact learning algorithm for sparse $GF(2)$ polynomials due to Schapire and Sellie [55]. Applying the algorithm of [55] is nontrivial; it requires us to prove new theorems about how sparse $GF(2)$ polynomials simplify under certain restrictions of "low-influence" sets of variables.

- In Chapter 5 we consider the problem of testing whether a function f is a halfspace, i.e. a function of the form $f(x) = \mathrm{sgn}(w \cdot x - \theta)$. We consider halfspaces over the continuous domain $\mathbb{R}^n$ (endowed with the standard multivariate Gaussian distribution) as well as halfspaces over the Boolean cube $\{-1,1\}^n$ (endowed with the uniform distribution). In both cases we give an algorithm for testing halfspaces using only $\mathrm{poly}(\frac{1}{\epsilon})$ queries, again independent of the dimension n.

  In turns out that halfspaces are not amenable to the "implicit learning" approach from the previous chapters. Thus, to achieve our testing algorithms, we prove new structural results about halfspaces. Two simple structural results about halfspaces are at the heart of our approach for the Gaussian distribution: the first gives an exact relationship between the expected value of a halfspace f and the sum of the squares of f's degree-1 Hermite coefficients, and the second shows that any function that approximately satisfies this relationship is close to a halfspace. We prove analogous results for the Boolean cube $\{-1,1\}^n$ (with Fourier coefficients in place of Hermite coefficients) for balanced halfspaces in which all degree-1 Fourier coefficients are small. Dealing with general halfspaces over $\{-1,1\}^n$ poses significant additional complications and requires other ingredients, including, again, utilization of work on testing juntas.

- Finally, in Chapter 6, we consider the problem of testing whether a Boolean function f is a ± 1-*weight halfspace*, i.e. a function of the form $f(x) = \mathrm{sgn}(w_1x_1 + w_2x_2 + \cdots + w_nx_n)$ where the weights w_i take values in $\{-1,1\}$. While one may be tempted to conclude from the previous chapters that all natural classes of functions are testable with query complexity independent of n, in Chapter 6 we show that this is not the case. In particular, to test whether f is a ± 1-weight halfspace versus ϵ-far from all such halfspaces we prove that nonadaptive algorithms must make $\Omega(\log n)$ queries. We complement this lower bound with a sublinear upper bound showing that $O(\sqrt{n} \cdot \text{poly}(\frac{1}{\epsilon}))$ queries suffice.

1.5 Previous work

Both computational learning theory and property testing are rich fields that have inspired an enormous amount of prior work. Here we will review just a small selection of this work, specifically results on testing classes of functions that are of interest to the learning theory community.

One important class of functions is the class of J-juntas, or functions that depend on at most J variables. Fischer *et al.* [26] gave an algorithm to test whether a function $f : \mathcal{X}^n \to \{0,1\}$ is a J-junta with query complexity polynomial in J and $1/\epsilon$. Diakonikolas *et al.* [17] later generalized their work to function with non-Boolean ranges, and Blais [6] improved upon both of these results, giving an (adaptive) algorithm for testing J-juntas with only $\tilde{O}(J/\epsilon)$ queries. This nearly matches an $\Omega(J)$ lower bound proved by Chockler and Gutfeund [13].

One of the motivations for the work in this thesis is the work of Parnas *et al.* [51], who gave algorithms for testing whether Boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$ have certain very simple representations as Boolean formulae. They gave an $O(1/\epsilon)$-query algorithm for testing whether f is a single Boolean literal or a Boolean conjunction, and an $\tilde{O}(s^2/\epsilon)$-query algorithm for testing whether f is an s-term monotone DNF. They also posed as an open question whether a similar testing result can be obtained for the broader class of general (non-monotone) s-term DNF formulas (we resolve this question in Chapter 3).

A variety of papers have looked at the problem of testing whether a function has a special algebraic structure. Blum *et al.* [9] gave an $O(1/\epsilon)$-query algorithm for testing whether a function can be represented as a linear form over a finite field. Their algorithm was subsequently generalized in several works to test whether f can be represented as a low-degree polynomial. In particular, [1, 33, 37] consider the case when f is defined over a small finite field.

Other research in the area includes the work of Kearns and Ron [38], who gave testing algorithms for the classes of interval functions over the continuous interval $[0, 1]$ and for neural networks and decision trees over the continuous cube $[0, 1]^n$. However their results differ from the "standard" property testing results in several ways; for one thing, they view the dimension n as a constant and their algorithms have query complexity that depends (exponentially) on n.

Some of the techniques in this thesis have already been used in subsequent work in property testing and learning. In [30], Gopalan *et al.* show how to test whether a function f has a sparse representation in the Fourier basis, and whether f is a function of a small number of parities. Their techniques are similar in spirit to the junta testing techniques of [26], and the ones we employ in Chapter 3. In [50], O'Donnell and Servedio consider the problem of learning halfspaces from their degree-1 Fourier coefficients. They solve this problem in part using the techniques from Chapter 5.

Chapter 2

Notation And Preliminaries

Functions come in many shapes and forms. In this thesis our primary interest is *classification* functions, or function that map to a finite range. Some classification functions of particular interest to us are:

- **Functions on the Boolean cube.** Boolean functions are a cornerstone of theoretical computer science, and the titular subject of this thesis. Throughout this document, when we say "Boolean functions," we mean binary classification functions with Boolean inputs. We typically represent these functions as $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$.
- **Functions on $\mathbb{R}^n$.** Occasionally (see Chapter 5), we will also consider binary classification functions with real-valued inputs. We will typically represent these functions as $f : \mathbb{R}^n \rightarrow \{-1, 1\}$.
- **Functions on arbitrary product domains.** Generalizing the previous two notions, we sometimes (see Chapter 3) consider classification functions from an arbitrary product domain to an arbitrary finite range, i.e. functions of the form $f : \mathcal{X} \rightarrow \mathcal{Y}$ where $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$ and $\mathcal{Y}$ is a finite set.

2.1 Decomposition of Functions

One of the most powerful ideas in theoretical computer science, and indeed all of mathematics, is that of decomposing a function into parts. A functional decomposition is particularly useful if the parts are *orthogonal*, or have limited interaction with each other. Such a decomposition allows one to analyze the parts independently, and can often shed new light on the structure of the function at hand. We now look at three different ways of decomposing functions, one for each type of function discussed above.

2.1.1 The Fourier Decomposition

We can think of Boolean functions represented as $f : \{-1,1\}^n \rightarrow \{-1,1\}$ as a subset of the set of all functions of the form $f : \{-1,1\}^n \rightarrow \mathbb{R}$. This set forms a 2^n-dimensional inner product space with inner product

$$\langle f, g \rangle = \mathop{\mathbf{E}}_{x \sim \Omega}[f(x)g(x)]$$

where Ω is the uniform distribution on $\{-1,1\}^n$. This inner product defines a natural norm on functions given by $||f|| = \sqrt{\langle f, f \rangle}$.

An orthonormal basis for this inner product space is provided by the set of 2^n *parity functions* or *characters*, i.e. functions of the form $\chi_S(x) = \prod_{i \in S} x_i$ where $S \subseteq [n]$. It is trivial to verify that this basis is indeed orthonormal, in other words $||\chi_S|| = 1$ for all S, and $\langle \chi_S, \chi_{S'} \rangle = 0$ for all $S' \neq S$. Thus, every function in the space can be expressed as a linear combination of parity functions:

Fact 3 (Fourier Decomposition). *Every function $f : \{-1,1\}^n \rightarrow \mathbb{R}$ has a unique decomposition of the form*

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

where the $\hat{f}(S)$'s are Fourier coefficients *given by $\hat{f}(S) = \langle f(x), \chi_S(x) \rangle$.*

We will often be interested in f's *degree*-1 coefficients, i.e., $\hat{f}(S)$ where $|S| = 1$. For notational ease, we will sometimes write these as $\hat{f}(i)$ rather than $\hat{f}(\{i\})$.

A simple consequence of the orthonormality of the characters is that the inner product of two functions is given by the inner product of their Fourier coefficients. This gives us the following:

Fact 4 (Plancherel and Parseval's Inequalities). *For any functions* $f, g : \{-1,1\}^n \rightarrow \mathbb{R}$ *we have* Plancherel's Equality:

$$\langle f, g \rangle = \mathbf{E}[f(x)g(x)] = \sum_{S \subseteq [n]} \hat{f}(S)\hat{g}(S)$$

and in the special case when $f = g$ *we have* Parseval's Equality*:*

$$||f||^2 = \sum_{S \subseteq [n]} \hat{f}(S)^2$$

In particular when f *is Boolean (with range* $\{-1,1\}$*) this implies* $\sum \hat{f}(S)^2 = 1$.

2.1.2 The Hermite Decomposition

Now we turn our attention to classification functions with real-valued inputs. We represent these as $f : \mathbb{R}^n \rightarrow \{-1,1\}$ and think of them as a subset of all functions of the form $f : \mathbb{R}^n \rightarrow \mathbb{R}$. We treat the set of square-integrable such functions as an inner product space. As before, we use the inner product

$$\langle f, g \rangle = \mathop{\mathbf{E}}_{x \sim \Omega}[f(x)g(x)]$$

but here Ω is the standard n-dimensional Gaussian distribution (that is, each component of x drawn from Ω is distributed according to the standard normal random variable $N(0,1)$). Again, this inner product defines the natural norm given by $||f|| = \sqrt{\langle f, f \rangle}$.

In contrast to the last section, where the parity functions formed an orthonormal basis for function from $\{-1,1\}^n \rightarrow \mathbb{R}$, here it is not as easy to find an orthonormal basis for functions from $\mathbb{R}^n \rightarrow \mathbb{R}$. In fact, even in the case when $n = 1$, the associated inner product space is infinite dimensional, and finding a complete orthonormal basis is non-trivial. Suffice it to say, for $n = 1$ there is a sequence of polynomials called *Hermite polynomials* that form a complete orthonormal basis for the space. The first few of them are $p_0 \equiv 1, p_1(x) = x, p_2(x) = (x^2 - 1)/\sqrt{2}, \ldots$, and in general they can be defined via $\exp(\lambda x - \lambda^2/2) = \sum_{d=0}^{\infty}(\lambda^d/\sqrt{d!})p_d(x)$ where λ is a formal variable. In the case of general n, given $S \in \mathbb{N}^n$, we have that the collection of n-variate polynomials $H_S(x) := \prod_{i=1}^{n} p_{S_i}(x_i)$ forms a complete orthonormal basis for the space. Thus we have the following analogue of Fact 3:

Fact 5 (Hermite Decomposition). *Every square-integrable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ has a unique decomposition as*

$$f(x) = \sum_{S \in \mathbb{N}^n} \hat{f}(S) H_S(x)$$

where the $\hat{f}(S)$'s are Hermite coefficients *given by $\hat{f}(S) = \langle f, H_S \rangle$.*

We note that it follows immediately from the orthonormality of the H_S's that Plancherel and Parsevel's identities (the analogue of Fact 4, where the sums are taken over all $S \in \mathbb{N}^n$) also holds in this setting.

While Hermite coefficients seem more difficult to analyze than Fourier coefficients, we remark that we will almost always be interested in only "degree-0" and "degree-1" Hermite coefficients, and these are the same as the corresponding Fourier coefficients. To be precise, there is a single "degree-0" coefficient $\hat{f}(\mathbf{0})$, which is just equal to $\mathbf{E}[f(x)]$, and there are n "degree-1" coefficients $\hat{f}(\mathrm{e}_i)$, which are just equal to $\mathbf{E}[f(x)x_i]$. The only difference is that the expectations here are taken over the n-dimensional Gaussian distribution rather than the uniform distribution over the Boolean cube.

2.1.3 The Efron-Stein Decomposition

Now we turn our attention to more general functions of the form $f : \mathcal{X} \rightarrow \mathcal{Y}$ where $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$ and $\mathcal{Y}$ is a finite set. One way to represent such functions is by thinking of the range $\mathcal{Y}$ as a subset of $\mathbb{R}^{|\mathcal{Y}|}$, and identifying element y_i in $\mathcal{Y}$ with the standard unit basis vector $\mathrm{e_i}$ in $\mathbb{R}^{|\mathcal{Y}|}$. We call these functions of the form $f : \mathcal{X} \rightarrow \mathcal{Y}$ *pure-valued* functions and think of them as a subset of $f : \mathcal{X} \rightarrow \mathbb{R}^{\mathcal{Y}}$ (henceforth, for notational convenience, we will implicitly assume $\mathcal{Y} = \{\mathrm{e}_1, ...\mathrm{e}_{|\mathcal{Y}|}\}$ unless otherwise stated, and we will abbreviate $\mathbb{R}^{|\mathcal{Y}|}$ by simply writing $\mathbb{R}^{\mathcal{Y}}$).

Let $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$ be a product probability space over $\mathcal{X}$. Then the set of functions of the form $f : \mathcal{X} \rightarrow \mathbb{R}^{\mathcal{Y}}$ forms the inner product space $L^2(\Omega, \mathbb{R}^{\mathcal{Y}})$ under the inner product

$$\langle f, g \rangle = \mathop{\mathbf{E}}_{x \sim \Omega}[\langle f(x), g(x) \rangle_{\mathbb{R}^{\mathcal{Y}}}]$$

wher $\langle \cdot, \cdot \rangle_{\mathbb{R}^{\mathcal{Y}}}$ denotes the standard inner product over $\mathbb{R}^{\mathcal{Y}}$. Note that this is essentially the same as the previous inner products we've defined, however inside the expectation we take $\langle f(x), g(x) \rangle_{\mathbb{R}^{\mathcal{Y}}}$ instead of $f(x) \cdot g(x)$, since the range of the functions is a vector rather than a scalar. As before, the inner product gives way to the norm $||f|| = \sqrt{\langle f, f \rangle}$, and it is easy to see that pure-valued functions have norm 1.

It turns out that even in this more general setting, the functions in $L^2(\Omega, \mathbb{R}^{\mathcal{Y}})$ can be decomposed quite elegantly due to the following theorem:

Theorem 6 (Efron-Stein Decomposition [19]). *Every function in $L^2(\Omega, \mathbb{R}^{\mathcal{Y}})$ has a unique decomposition of the form*

$$f(x) = \sum_{S \subseteq [n]} f^S(x)$$

such that the following properties hold:

- *Each f^S depends only on the coordinates in S.*

- *For every $S' \not\supseteq S$ and any $y \in \mathcal{X}$, we have $\mathbf{E}_{x \sim \Omega}[f^S(x) |\ x_{S'} = y_{S'}] = 0$, where $x_{S'}$ and $y_{S'}$ denote the inputs x and y restricted to the coordinates in S'.*

The Efron-Stein decomposition is an *orthogonal* decomposition. That is, for any function $f(x)$, and any two components of its Efron-Stein decomposition $f^S(x)$ and $f^T(x)$ where $S \neq T$, we have $\langle f^S, f^T \rangle = 0$. Note however that this is not the same as having an orthonormal basis. The components of the decomposition vary for each f. They do not form a basis for $L^2(\Omega, \mathbb{R}^{\mathcal{Y}})$, and they do not necessarily have norm 1. Fortunately, the orthogonality of the decomposition is enough to guarantee us the following generalized version of Parseval's equality:

Fact 7 (Generalized Parseval's Identity). *For every function $f \in L^2(\Omega, \mathbb{R}^{\mathcal{Y}})$, we have*

$$||f||^2 = \sum_{S \subseteq [n]} ||f^S||^2$$

In particular when f is a pure-valued function, $\sum_{S\subseteq[n]} ||f^S||^2 = 1$

Remark The careful reader will note that so far we have actually introduced *two* ways of representing Boolean functions. Earlier we used the representation $f : \{-1,1\}^n \rightarrow \{-1,1\}$ while in this section we've suggested the representation $f : \{-1,1\}^n \rightarrow \{\mathbf{e_1}, \mathbf{e_2}\}$. Both representations are useful in different contexts. With either representation the Efron-Stein decomposition exists, however the decomposition depends on the representation. When f is represented as $f : \{-1,1\}^n \rightarrow \{-1,1\}$, the Efron-Stein decomposition is given by the Fourier decomposition (that is, $f^S = \hat{f}(S)\chi_S$ for each $S \subseteq [n]$). In this sense the Efron-Stein decomposition is a generalization of the Fourier Decomposition to functions which are not necessarily Boolean.

2.2 More on Functions: Distance, Influence, and Juntas

We now introduce some more notation regarding functions. First, we need a way of measuring the distance between functions:

Definition 8 (Distance). *For functions $f, g : \mathcal{X} \rightarrow \mathcal{Y}$ and a probability measure Ω, we say that f and g are ϵ-*far *if*

$$\Pr_{x\sim\Omega}[f(x) \neq g(x)] \geq \epsilon$$

*Otherwise we say f and g are ϵ-*close.

We will often omit the Ω in the subscript when the context is clear. In particular, when f is a Boolean function we will implicitly assume that the probability is evaluated with respect to the uniform distribution over the Boolean cube. Similarly, when f is over R^n we will implicitly assume the probability is with respect to the standard n-dimensional Gaussian distribution.

As we have already seen in the definition of the Efron-Stein decomposition, for inputs $x = (x_1, ..., x_n) \in \mathcal{X}$ we will often be required to refer to subsets of coordinates. Here we develop some convenient shorthand for notating this.

For the elements $x = (x_1, ..., x_n) \in \mathcal{X}$ and the set $S \subseteq [n]$, we let x_S represent the ordered list $(x_i : i \in S)$, and for $x, y \in \mathcal{X}$ we use the notation $x_S y_{\bar{S}}$ to represent the element $z \in \mathcal{X}$ where $z_S = x_S$ and $z_{\bar{S}} = y_{\bar{S}}$.

For a subset $S \subseteq [n]$ and a setting w of the coordinates in S, the restricted function $f_w(x)$ refers to the function on $n - |S|$ coordinates defined by fixing the coordinates in S according to w, and evaluating f on the rest (in other words, $f_w(x) = f(w_S x_{\bar{S}})$).

Finally, for a product probability measure $\Omega = \Omega_1 \times \cdots \times \Omega_n$ we define $\Omega(S)$ to be the product probability measure over just the coordinates in S (i.e. $\Omega(S) = \prod_{i \in S} \Omega_i$). Correspondingly, we use the notation $x \sim \Omega(S)$ to indicate that x is an $|S|$-coordinate input drawn randomly from $\Omega(S)$.

2.2.1 Influence

We now come to one of the most important definitions in this thesis, that of the *influence* of a set of variables of a function f. The influence is a measure of the set's ability to control the output of the function.

In order to define influence, we must first define variance:

Definition 9 (Variance). *The* variance *of a function $f : \mathcal{X} \rightarrow \mathbb{R}^{\mathcal{Y}}$ with respect to a probability measure Ω is*

$$\mathop{\mathbf{V}}_{x \sim \Omega}[f(x)] = \mathop{\mathbf{E}}_{x \sim \Omega}[||f(x)||^2] - \left|\left| \mathop{\mathbf{E}}_{x \sim \Omega}[f(x)] \right|\right|^2$$

This is a generalization of the standard notion of the variance. Usually the variance of a random variable X is given by $\mathbf{E}[X^2] - \mathbf{E}[X]^2$, but here the $|| \cdot ||$ has been added since the range of f is potentially a vector rather than a scalar. Notice that functions which are pure-valued or which map to $\{-1, 1\}$ have norm 1, so the variance of such functions is given by $1 - ||\mathbf{E}_{x \sim \Omega}[f(x)]||^2$ and is therefore bounded between 0 and 1.

We are now ready to define influence:

Definition 10 (Influence). *For a function $f \in L^2(\Omega, \mathbb{R}^{\mathcal{Y}})$, the* influence *of the set $S \subseteq [n]$ of coordinates under the probability measure Ω is defined as*

$$\text{Inf}_f(S) = \mathop{\mathbf{E}}_{z \sim \Omega(\bar{S})}[\mathop{\mathbf{V}}_{x \sim \Omega(S)}[f_z(x)]]$$

Let us take a moment to reflect on this definition. The closer a function is to constant, the closer its variance to is to zero. Thus, the influence of S is a measure of how "non-constant" you expect the function to be when you randomly restrict the coordinates outside of S. If the function does not depend on the coordinates in S at all, then $\mathrm{Inf}_f(S)$ will be zero; otherwise it will be something larger.[1]

This definition of influence is intuitive, but rather hard to work with on its own. For the functions we care about, influence can be nicely expressed as a probability:

Proposition 11. *Let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a pure-valued function. Then for any set $S \subseteq [n]$ we have*

$$\mathrm{Inf}_f(S) = \Pr_{\substack{z \sim \Omega(\bar{S}) \\ x,y \sim \Omega(S)}} [f_z(x) \neq f_z(y)]$$

Proof. The proof follows easily from the definitions and the fact that $||f|| = 1$ for pure-valued functions.

$$\begin{aligned} \mathrm{Inf}_f(S) &= \mathop{\mathbf{E}}_{z \sim \Omega(\bar{S})} [\mathop{\mathbf{V}}_{x \sim \Omega(S)} [f_z(x)]] \\ &= \mathop{\mathbf{E}}_{z \sim \Omega(\bar{S})} [1 - || \mathop{\mathbf{E}}_{x \sim \Omega(S)} [f_z(x)]||^2] \\ &= \mathop{\mathbf{E}}_{z \sim \Omega(\bar{S})} [1 - \mathop{\mathbf{E}}_{x,y \sim \Omega(S)} [\langle f_z(x), f_z(y) \rangle_{\mathbb{R}^{\mathcal{Y}}}]] \\ &= 1 - \Pr_{\substack{z \sim \Omega(\bar{S}) \\ x,y \sim \Omega(S)}} [f_z(x) = f_z(y)] \end{aligned}$$

□

Proposition 11 is important enough that it motivates the definition of a small subroutine

[1] There are multiple different definitions of "influence" in the literature, so the one we give here merits some discussion. Our definition of influence is a direct generalization of the quantity called *variation* in [25]. It is also, as we show in Proposition 11, consistent with the the definition of influence given in [6] (whereas [6] proposes the equation in Proposition 11 as a definition, here it is a consequence of the definition). Our definition is also consistent with the notion of single-coordinate influence discussed in Proposition 14.

We choose our definition of influence because it unifies the definitions in much of the literature. However, we point out that for sets S of more than one coordinate, our definition is different from the definition of influence given by Kahn, Kalai, and Linial [34]. They define influence to be the probability over the setting of coordintes not in S that the resulting restricted function is not constant (in other words, $\mathbf{E}_{z \sim \Omega(\bar{S})}[\mathbf{1}_{f_z(x) \text{ not constant}}]$). For functions that we care about (with norm 1), this is equivalent to $\mathbf{E}_{z \sim \Omega(\bar{S})}[\lceil \mathbf{V}_{x \sim \Omega(S)}[f_z(x)] \rceil]$, thus the quantity that they call influence is always greater than or equal to our influence. Their definition of influence makes sense for some applications, but our definition makes sense for others, in particular, as we shall see in the next section, for analyzing distance to juntas.

called the **Independence-Test**, shown here in Figure 2.2.1. The independence test was first defined by Fischer et. al. in [25], and we will use it ourselves in Chapters 3 and 4. It is easy to see from Proposition 11 that the probability the test rejects is exactly $\text{Inf}_f(S)$.

Independence Test (inputs are $S \subseteq [n]$, and black-box access to $f : \mathcal{X} \rightarrow \mathcal{Y}$)

1. Choose $z \sim \Omega(\bar{S})$ and $x, y \sim \Omega(S)$.
2. If $f_z(x) = f_z(y)$, accept. Otherwise, reject.

Figure 2-1: The subroutine **Independence-Test**.

The independence test gives us an easy algorithmic way to estimate the influence of a set of variables, but from an analytic standpoint it does not tell us much about how to analyze the influence. For this, it turns out that we can relate the influence of a function to its Efron-Stein (or Fourier) decomposition using the following remarkable formula:

Proposition 12 (Decomposition of Influence). *For any $f \in L^2(\Omega, \mathbb{R}^{\mathcal{Y}})$ and any $S \subseteq [n]$*

$$\text{Inf}_f(S) = \sum_{T:\, S \cap T \neq \emptyset} ||f^T||^2$$

A proof of Proposition 12, or its equivalent with Fourier coefficients in place of Efron-Stein magnitudes, is elementary and appears in several places in the literature (see, for instance, [6] for a proof in this setting).

An immediately corollary of Proposition 12 is the monotonicity and subadditivitiy of influence:

Corollary 13 (Monotonicity and Subadditivity). *For $f \in L^2(\Omega, \mathbb{R}^{\mathcal{Y}})$ and any $S, T \subseteq [n]$*

$$\text{Inf}_f(S) \leq \text{Inf}_f(S \cup T) \leq \text{Inf}_f(S) + \text{Inf}_f(T)$$

Single-Coordinate Influence for Boolean Functions

We will often be interested in the special case of the influence when f is a Boolean function and the set S consists of a single coordinate i. In this case we will use the shorthand $\text{Inf}_f(i)$ instead of $\text{Inf}_f(\{i\})$ to denote the influence of variable i. The influence of a single coordinate is a well-studied quantity in computer science, dating back to the seminal works of Ben-Or and Linial [3], and Kahn, Kalai, and Linial [34], who showed that every balanced Boolean function has a coordinate with influence at least $\Omega(\log n/n)$.

Typically, for a Boolean function the influence of coordinate i is defined as the probability that on a random input, flipping bit i changes the function's value. In fact, if the function is represented appropriately, this definition is just a special case of our definition. This is made precise by the following:

Proposition 14 (Single-Coordinate Influence). *Let $f : \{-1,1\}^n \rightarrow \{-1,1\}$ and $i \in [n]$. Then the influence of variable i is given by*

$$\mathrm{Inf}_f(i) = \Pr_x[f(x^{i-}) \neq f(x^{i+})]$$

where x^{i-} and x^{i+} denote x with the i'th bit set to -1 or 1 respectively.

Proof. According to our definition of influence, we have $\mathrm{Inf}_f(i) = \mathop{\mathbf{E}}_{z \sim \Omega([n]\setminus i)}[\mathop{\mathbf{V}}_{x \sim \Omega(i)}[f_z(x)]]$. Here the restricted function f_z is just a function of the single coordinate i. It is easy to see that f_z has variance 1 if $f_z(x^{i-}) \neq f_z(x^{i+})$ and variance 0 otherwise. Hence the expectation of the variance is just equal to $\Pr_x[f(x^{i-}) \neq f(x^{i+})]$ and the proof is complete. □

In Chapters 5 and 6 we will be particularly interested in functions which are *unate*:

Definition 15 (Unate functions). *A Boolean function $f : \{-1,1\}^n \rightarrow \{-1,1\}$ is* unate *if it is monotone increasing or monotone decreasing as a function of variable x_i for each i.*

It turns out that for unate functions, single-coordinate influences have a particularly nice relationship to Fourier coefficients:

Fact 16. *If $f : \{-1,1\}^n \rightarrow \{-1,1\}$ is unate then* $\mathrm{Inf}_f(i) = |\hat{f}(i)|$.

We will prove this fact explictly in Lemma 152 in Chapter 6.

2.2.2 Juntas

An important special type of function that we will refer to over and over again in this thesis is the junta:

Definition 17 (Juntas). *A function f is a* junta on $J \subseteq [n]$ *if f only depends on the coordinates in J.*

It follows immediately from the definition of juntas and influence that the property of being a junta on J is characterized by the variables outside of J having no influence:

Proposition 18. *A function f is a junta on J if and only if* $\mathrm{Inf}_f([n] \backslash J) = 0$.

In fact this characterization is somewhat robust, as the following proposition shows:

Proposition 19. *Let $f : \mathcal{X} \to \mathcal{Y}$ and suppose that for some set J, we have* $\mathrm{Inf}_f([n] \backslash J) \leq \epsilon$. *Then f is ϵ-close to a junta on J. In fact it is ϵ-close to the junta given by*

$$h(x) = \mathop{\mathbf{Plur}}_{z}[f(x_J z_{\bar{J}})]$$

where $\mathbf{Plur}$ *denotes the "plurality" or "most-common-output" operator (with ties broken arbitrarily).*

The following proof is taken from [6], generalizing an argument from [25]. It is included here only for completeness:

Proof. ([6])

$$\begin{aligned}
\Pr[f(x) \neq h(x)] &= 1 - \mathop{\mathbf{E}}_{x}[\langle f(x), h(x) \rangle] \\
&= 1 - \mathop{\mathbf{E}}_{x}\left[\langle \mathop{\mathbf{E}}_{z}[f(x_J z_{\bar{J}})], h(x) \rangle_{\mathbb{R}^{\mathcal{Y}}}\right] \\
&= 1 - \mathop{\mathbf{E}}_{x}\left[\left\| \mathop{\mathbf{E}}_{z}[f(x_J z_{\bar{J}})] \right\|_{\infty} \left\| \mathop{\mathbf{E}}_{z}[f(x_J z_{\bar{J}})] \right\|_{1}\right] \\
&\leq 1 - \mathop{\mathbf{E}}_{x}\left[\left\| \mathop{\mathbf{E}}_{z}[f(x_J z_{\bar{J}})] \right\|_{2}^{2}\right] \\
&= 1 - \sum_{S \subseteq J} \|f^S\|_2^2 \\
&= \sum_{S : S \cap ([n] \backslash J) \neq \emptyset} \|f^S\|_2^2 \\
&= \mathrm{Inf}_f([n] \backslash J)
\end{aligned}$$

The first equality follows from the fact that f and h are pure-valued. The second follows from the fact that h only depends on the coordinates in J and linearity of expectation. The third equality from the fact that $\langle \mathbf{E}_z[f(x_J z_{\bar{J}})], h(x)\rangle_{\mathbb{R}^{\mathcal{Y}}} = \|\mathbf{E}_z[f(x_J z_{\bar{J}})]\|_\infty$ and $\|\mathbf{E}_z[f(x_J z_{\bar{J}})]\|_1 = 1$ for pure-valued functions. The inequality is a special case of Holder's inequality. The following equality follows from the fact that $\mathbf{E}_z[f(x_J z_{\bar{J}})] = \sum_{S \subseteq J} f^S(x)$. The next equality uses Parseval, and finally the last equality is Proposition 12. □

2.3 Probability Bounds

In addition to the standard Markov and Chernoff bounds, we will often make use of the following claims:

Proposition 20. *If X is a random variable taking values in the range $[-1, 1]$, its expectation can be estimated to within an additive $\pm\epsilon$, with confidence $1-\delta$, using $O(\log(1/\delta)/\epsilon^2)$ queries.*

Proof. This follows from a standard additive Chernoff bound. We shall sometimes refer to this as "empirically estimating" the value of $\mathbf{E}[X]$. □

Proposition 21 (Fischer *et al.* [26]). *Let $X = \sum_{i=1}^{l} X_i$ be a sum of non-negative independent random variables X_i, and denote expectation of X by α. If every X_i is bounded above by t, then*

$$\Pr[X < \eta\alpha] < \exp\left(\frac{\alpha}{et}(\eta e - 1)\right)$$

for every $\eta > 0$.

Chapter 3

Testing for Concise Representations

3.1 Introduction

In this chapter we study the problem of testing whether a function has a concise representation. Our main result is a general algorithm that can be used to test whether an unknown function belongs to one of many different representation classes, as long as the representation class satisfies certain conditions. We show that this algorithm yields property testers for many classes that were not previously known to be testable. These include Boolean function classes such as decision lists, size-s decision trees, size-s branching programs, s-term DNF (resolving an open question of Parnas *et al.* [51]), size-s Boolean formulas, size-s Boolean circuits, and s-sparse polynomials over $\mathbb{F}_2$, as well as non-Boolean classes such as size-s algebraic circuits, size-s algebraic computation trees, and s-sparse polynomials over finite fields. For each of these classes the testing algorithm uses poly$(s, 1/\epsilon)$ many queries, independent of the number n of inputs to the function (the running time is exponential in s, though linear in n). These testing results are included in the top part of Table 1.1. We note that our general algorithm can also be easily shown to yield property testers for all of the classes tested in [51]; the query complexities would be slightly larger than in [51], but would not require a specialized algorithm for each problem.

We also prove a lower bound; we show that any non-adaptive algorithm to test s-sparse polynomials over finite fields of constant size must make $\tilde{\Omega}(\sqrt{s})$ queries. Since this is within a polynomial factor of our upper bound, this result shows that in at least one instance our general algorithm yields a tester that is nearly optimal. (For testing other representation classes, there is a larger gap between our upper and lower bounds. We give some simple but fairly weak lower bounds for other representation classes in Section 3.7.)

Our techniques. Our approach combines ideas from the junta test of Fischer *et al.* [26] with ideas from learning theory. As mentioned in the introduction to this thesis, the basic idea of using a learning algorithm to do property testing goes back to Goldreich *et al.* [29], who observed that any proper learning algorithm for a class $\mathcal{C}$ can immediately be used as a testing algorithm for $\mathcal{C}$. However, it is well known that proper learning algorithms for virtually every interesting class of n-variable functions must make at least $\Omega(\log n)$ queries. Thus this testing-by-learning approach did not previously yield any strong results for testing interesting function classes.

We get around this impediment by making the key observation that many interesting classes $\mathcal{C}$ of functions are "well-approximated" by juntas in the following sense: every function in $\mathcal{C}$ is close to some function in $\mathcal{C}_J$, where $\mathcal{C}_J \subseteq \mathcal{C}$ and every function in $\mathcal{C}_J$ is a J-junta. For example, every s-term DNF over $\{0,1\}^n$ is τ-close to an s-term DNF that depends on only $s \log s/\tau$ variables, since each term with more than $\log s/\tau$ variables can be removed from the DNF at the cost of at most τ/s error. Roughly speaking, our algorithm for testing whether f belongs to $\mathcal{C}$ works by attempting to learn the "structure" of the junta in $\mathcal{C}_J$ that f is close to *without actually identifying the relevant variables on which the junta depends*. If the algorithm finds such a junta function, it accepts, and if it does not, it rejects. Our approach can be characterized as *testing by implicit learning* (as opposed to the explicit proper learning in the approach of Goldreich *et al.* [29]), since we are "learning" the structure of the junta to which f is close without explicitly identifying its relevant variables. Indeed, avoiding identifying the relevant variables is what makes it possible to have query complexity independent of n.

We find the structure of the junta f' in $\mathcal{C}_J$ that f is close to by using the techniques of [26]. As in [26], we begin by randomly partitioning the variables of f into subsets and identifying which subsets contain an influential variable (the random partitioning ensures that with high probability, each subset contains at most one such variable if f is indeed in $\mathcal{C}$). Next, we create a sample of random labeled examples $(x^1, y^1), (x^2, y^2), ..., (x^m, y^m)$, where each x^i is a string of length J (not length n; this is crucial to the query complexity of the algorithm) whose bits correspond to the influential variables of f, and where y^i corresponds with high probability to the value of junta f' on x^i. Finally, we exhaustively check whether any function in $\mathcal{C}_J$ – over J input variables – is consistent with this labeled sample. This step takes at least $|\mathcal{C}_J|$ time steps, which is exponential in s for most of the classes we test; but since $|\mathcal{C}_J|$ is independent of n, we are able to get away with an overall query complexity that is independent of n. (The overall time complexity is linear as a function of n; note that such a runtime dependence on n is inevitable since it takes n time steps simply to prepare a length-n query string to the black-box function.) We explain our testing algorithm in more detail in Section 3.2 and prove correctness of the algorithm in Section 3.3. We apply the theorem to obtain new testing results for different classes of Boolean and non-Boolean functions in Section 3.5.

Finally, we prove our lower bound for testing s-sparse polynomials over finite fields in two stages. We first show that any non-adaptive algorithm that can successfully distinguish a linear form $x_{i_1} + \cdots + x_{i_s}$ (over s randomly selected variables from $x_1, \ldots, x_n$) from a linear form $x_{i_1} + \cdots + x_{i_{s+p}}$ (over $s + p$ randomly selected variables, where p is the characteristic of the finite field) must make $\tilde{\Omega}(\sqrt{s})$ queries. This is a technical generalization of a similar result for $\mathbb{F}_2$ in [26]; the heart of our proof is an extension of a convergence type result about random walks over $\mathbb{Z}_2^q$ with arbitrary step distribution to random walks over $\mathbb{Z}_p^q$. (As an interesting side product, the latter also partially answers a question posed in [26] as to what groups possess a similar convergence type property.) We then prove that every s-sparse polynomial g over finite field $\mathbb{F}$ is "far" from every affine function with at least $s + 1$ non-zero coefficients. This result does not have an analogue in [26] (that paper establishes a lower bound on distinguishing size-s parities from size-$(s + 2)$ parities, and it is trivially true that every size-s parity is far from every size-$(s + 2)$ parity) and its proof requires several ideas; our argument uses random restrictions chosen according to a distribution that depends on the structure of the polynomial g. We present these results in Section 3.6.

Notational Note: The techniques in this chapter are applicable to functions with non-Boolean domain and range. We will denote functions in this chapter as $f : \mathcal{X}^n \rightarrow \mathcal{Y}$, where

$\mathcal{X}$ and $\mathcal{Y}$ are finite sets, and we will let Ω denote the uniform distribution over $\mathcal{X}^n$. (Note that this is a slight abuse of notation from the preliminaries section, where we represented the domain as a product $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ of n potentitally different sets, and we placed no restrictions on the probability measure Ω).

Bibliographic Note: The results in this chapter originally appeared in [17], however the presentation here has been simplified. While [17] contained a technique for generalizing the junta test of Fischer et. al. [25] to functions with non-Boolean range, here we use the Efron-Stein decomposition instead, obviating the need for the previous technique. This was inspired by Eric Blais's junta test in [6], and we are grateful to him for the improvement.

3.2 The test and an overview of its analysis

In this section we present our testing algorithm and give an intuitive explanation of how it works. We close this section with a detailed statement of our main theorem, Theorem 23, describing the correctness and query complexity of the algorithm.

3.2.1 Subclass approximators

Let $\mathcal{C}$ denote a class of functions from $\mathcal{X}^n$ to $\mathcal{Y}$. We will be interested in classes of functions that can be closely approximated by juntas in the class. We have the following:

Definition 22. *For $\tau > 0$, we say that a subclass $\mathcal{C}(\tau) \subseteq \mathcal{C}$ is a $(\tau, J(\tau))$-*approximator *for $\mathcal{C}$ if*

- *$\mathcal{C}(\tau)$ is closed under permutation of variables, i.e. if $f(x_1, \ldots, x_n) \in \mathcal{C}(\tau)$ then $f(x_{\sigma_1}, \ldots, x_{\sigma_n})$ is also in $\mathcal{C}(\tau)$ for every permutation σ of $[n]$; and*
- *for every function $f \in \mathcal{C}$, there is a function $f' \in \mathcal{C}(\tau)$ such that f' is τ-close to f and f' is a $J(\tau)$-junta.*

Typically for us $\mathcal{C}$ will be a class of functions with size bound s in some particular representation, and $J(\tau)$ will depend on s and τ. (A good running example to keep in mind is $\mathcal{X} = \{0, 1\}$, $\mathcal{Y} = \{-1, 1\}$, and $\mathcal{C}$ is the class of all functions that have s-term DNF

Identify-Critical-Subsets (input is black-box access to $f : \mathcal{X}^n \to \mathcal{Y}$ and $\epsilon > 0$)

1. Partition the variables $x_1, \dots, x_n$ into r random subsets by assigning each of $x_1, \dots, x_n$ equiprobably to one of $I_1, \dots, I_r$.
2. Choose s random subsets $\Lambda_1, \dots, \Lambda_s \subseteq [r]$ of size $J(\tau^\star)$ by uniformly choosing without repetitions $J(\tau^\star)$ members of $[r]$. Each set Λ_i determines a block $B_i \overset{\text{def}}{=} \bigcup_{j \in \Lambda_i} I_j$. (Note that we do not guarantee that the blocks are disjoint.)
3. Apply h iterations of the *independence test* (see Figure 2.2.1) to each block B_i. If all of the independence test iterations applied to block B_i accept, then B_i is declared to be *influence-free*, and all the subsets I_j with $j \in \Lambda_i$ are declared to be influence-free on its behalf.
4. If:

   (a) at least half of the blocks $B_1, \dots, B_s$ are influence-free; and

   (b) except for at most $J(\tau^\star)$ subsets, every subset in the partition $I_1, \dots, I_r$ is declared influence-free on behalf of some block,

   then output the list $I_{i_1}, \dots, I_{i_j}$ of those subsets that are *not* declared to be influence-free. (We call these the *critical* subsets.) Otherwise, halt and output "Not in $\mathcal{C}$."

Figure 3-1: The subroutine **Identify-Critical-Subsets.**

representations. In this case we may take $\mathcal{C}(\tau)$ to be the class of all s-term $\log(s/\tau)$-DNFs, and we have $J(\tau) = s\log(s/\tau)$.) Our techniques will work on function classes $\mathcal{C}$ for which $J(\tau)$ is a slowly growing function of $1/\tau$ such as $\log(1/\tau)$. In Section 3.5 we will consider many different specific instantiations of $\mathcal{C}$ and corresponding choices of $\mathcal{C}(\tau)$.

We write $\mathcal{C}(\tau)_k$ to denote the subclass of $\mathcal{C}(\tau)$ consisting of those functions that depend only on variables in $\{x_1, \dots, x_k\}$. We may (and will) view functions in $\mathcal{C}(\tau)_k$ as taking k instead of n arguments from $\mathcal{X}$.

3.2.2 Explanation of our testing algorithm

Our algorithm for testing whether a function $f : \mathcal{X}^n \to \mathcal{Y}$ belongs to $\mathcal{C}$ or is ϵ-far from $\mathcal{C}$ is given in Figures 3-1 through 3-3. Given $\epsilon > 0$ and black-box access to f, the algorithm performs three main steps:

```
**Construct-Sample** (input is the list $I_{i_1}, \ldots, I_{i_j}$ output by **Identify-Critical-Subsets** and black-box access to $f$)

1. Repeat the following $m$ times to construct a set $S$ of $m$ labeled examples $(x, y) \in \mathcal{X}^{J(\tau^\star)} \times \mathcal{Y}$, where $\mathcal{X} = \{\omega_0, \omega_1, \ldots, \omega_{|\mathcal{X}|-1}\}$:
    (a) Draw $z$ uniformly from $\mathcal{X}^n$. Let $X_q \overset{\text{def}}{=} \{i : z_i = \omega_q\}$, for each $0 \leq q \leq |\mathcal{X}| - 1$.
    (b) For $\ell = 1, \ldots, j$
        i. $w \overset{\text{def}}{=} 0$
        ii. For $k = 1, \ldots, \lceil \lg |\mathcal{X}| \rceil$
            A. $\mathcal{X}_0 \overset{\text{def}}{=}$ union of $(X_q \cap I_{i_\ell})$ taken over all $0 \leq q \leq |\mathcal{X}| - 1$ such that the $k$-th bit of $q$ is zero
            B. $\mathcal{X}_1 \overset{\text{def}}{=}$ union of $(X_q \cap I_{i_\ell})$ taken over all $0 \leq q \leq |\mathcal{X}| - 1$ such that the $k$-th bit of $q$ is one
            C. Apply $g$ iterations of the *independence test* to $\mathcal{X}_0$. If any of the $g$ iterations reject, mark $\mathcal{X}_0$. Similarly, apply $g$ iterations of the *independence test* to $\mathcal{X}_1$; if any of the $g$ iterations reject, mark $\mathcal{X}_1$.
            D. If exactly one of $\mathcal{X}_0$, $\mathcal{X}_1$ (say $\mathcal{X}_b$) is marked, set the $k$-th bit of $w$ to $b$.
            E. If neither of $\mathcal{X}_0$, $\mathcal{X}_1$ is marked, set the $k$-th bit of $w$ to unspecified.
            F. If both $\mathcal{X}_0$, $\mathcal{X}_1$ are marked, halt and output "no".
        iii. If any bit of $w$ is unspecified, choose $w$ at random from $\{0, 1, \ldots, |\mathcal{X}| - 1\}$.
        iv. If $w \notin [0, |\mathcal{X}| - 1]$, halt and output "no."
        v. Set $x_\ell = \omega_w$.
    (c) Evaluate $f$ on $z$, assign the remaining $J(\tau^\star) - j$ coordinates of $x$ randomly, and add the pair $(x, f(z))$ to the sample of labeled examples being constructed.
```

Figure 3-2: The subroutine **Construct-Sample.**

1. **Identify critical subsets.** In Step 1, we first randomly partition the variables $x_1, \ldots,$ x_n into r disjoint subsets $I_1, \ldots, I_r$. We then attempt to identify a set of $j \leq J(\tau^\star)$ of these r subsets, which we refer to as *critical* subsets because they each contain a "highly relevant" variable. (For now the value $\tau^\star$ should be thought of as a small quantity; we discuss how this value is selected below.) This step is essentially the same as the 2-sided test for J-juntas from Section 4.2 of Fischer *et al.* [26]. We will show that if f is close to a $J(\tau^\star)$-junta then this step will succeed w.h.p., and if f is far from every $J(\tau^\star)$-junta then this step will fail w.h.p.

2. **Construct a sample.** Let $I_{i_1}, \ldots, I_{i_j}$ be the critical subsets identified in the previous step. In Step 2 we construct a set S of m labeled examples $\{(x^1, y^1), \ldots, (x^m, y^m)\}$, where each x^i is independent and uniformly distributed over $\mathcal{X}^{J(\tau^\star)}$. We will show that if f belongs to $\mathcal{C}$, then with high probability there is a fixed $f'' \in \mathcal{C}(\tau^\star)_{J(\tau^\star)}$ such that each y^i is equal to $f''(x^i)$. On the other hand, if f is far from $\mathcal{C}$, then we will show that w.h.p. no such $f'' \in \mathcal{C}(\tau^\star)_{J(\tau^\star)}$ exists.

   To construct each labeled example, we again borrow a technique outlined in [26]. We start with a uniformly random $z \in \mathcal{X}^n$. We then attempt to determine how the j highly relevant coordinates of z are set. Although we don't know which of the coordinates of z are highly relevant, we do know that, assuming the previous step was successful, there should be one highly relevant coordinate in each of the critical subsets. We use the independence test repeatedly to determine the setting of the highly relevant coordinate in each critical subset.

   For example, suppose that $\mathcal{X} = \{0, 1\}$ and I_1 is a critical subset. To determine the setting of the highly relevant coordinate of z in critical subset I_1, we subdivide I_1 into two sets: the subset $\mathcal{X}_0 \subseteq I_1$ of indices where z is set to 0, and the subset $\mathcal{X}_1 = I_1 \backslash \mathcal{X}_0$ of indices where z is set to 1. We can then use the independence test on both $\mathcal{X}_0$ and $\mathcal{X}_1$ to find out which one contains the highly relevant variable. This tells us whether the highly relevant coordinate of z in subset I_1 is set to 0 or 1. We repeat this process for each critical subset in order to find the settings of the j highly relevant coordinates of z; these form the string x. (The other $J(\tau^\star) - j$ coordinates of x are set to random

Check-Consistency (input is the sample S output by **Identify-Critical-Subsets**)

(a) Check every function in $\mathcal{C}(\tau^\star)_{J(\tau^\star)}$ to see if any of them are consistent with sample S. If so output "yes" and otherwise output "no."

Figure 3-3: The subroutine **Check-Consistency.**

values; intuitively, this is okay since they are essentially irrelevant.) We then output $(x, f(z))$ as the labeled example.

3. **Check consistency.**

   Finally, in Step 3 we search through $\mathcal{C}(\tau^\star)_{J(\tau^\star)}$ looking for a function f'' over $\mathcal{X}^{J(\tau^\star)}$ that is consistent with all m examples in S. (Note that this step takes $\Omega(|\mathcal{C}(\tau^\star)_{J(\tau^\star)}|)$ time but uses no queries.) If we find such a function then we accept f, otherwise we reject.

3.2.3 Sketch of the analysis

We now give an intuitive explanation of the analysis of the test.

Completeness. Suppose f is in $\mathcal{C}$. Then there is some $f' \in \mathcal{C}(\tau^\star)$ that is $\tau^\star$-close to f. Intuitively, $\tau^\star$-close is so close that for the entire execution of the testing algorithm, the black-box function f might as well actually be f' (the algorithm only performs $\ll 1/\tau^\star$ many queries in total, each on a uniform random string, so w.h.p. the view of the algorithm will be the same whether the target is f or f'). Thus, for the rest of this intuitive explanation of completeness, we pretend that the black-box function is f'.

Recall that the function f' is a $J(\tau^\star)$-junta. Let us refer to the variables, x_i, that have $\mathrm{Inf}_f(x_i) > \theta$ (recall that $\mathrm{Inf}_f(x_i)$ is a measure of the influence of variable x_i, and θ is some threshold to be defined later) as the *highly relevant* variables of f'. Since f' is a junta, in Step 1 we will be able to identify a collection of $j \leq J(\tau^\star)$ "critical subsets" with high probability. Intuitively, these subsets have the property that:

- each highly relevant variable occurs in one of the critical subsets, and each critical subset contains at most one highly relevant variable (in fact at most one relevant variable for f');

- the variables outside the critical subsets are so "irrelevant" that w.h.p. in all the queries the algorithm makes, it doesn't matter how those variables are set (randomly flipping the values of these variables would not change the value of f' w.h.p.).

Given critical subsets from Step 1 that satisfy the above properties, in Step 2 we construct a sample of labeled examples $S = \{(x^1, y^1), \dots, (x^m, y^m)\}$ where each x^i is independent and uniform over $\mathcal{X}^{J(\tau^\star)}$. We show that w.h.p. there is a $J(\tau^\star)$-junta $f'' \in C(\tau^\star)_{J(\tau^\star)}$ with the following properties:

- there is a permutation $\sigma : [n] \to [n]$ for which $f''(x_{\sigma(1)}, \dots, x_{\sigma(J(\tau))})$ is close to $f'(x_1, \dots, x_n)$;
- The sample S is labeled according to f''.

Finally, in Step 3 we do a brute-force search over all of $\mathcal{C}(\tau^\star)_{J(\tau^\star)}$ to see if there is a function consistent with S. Since f'' is such a function, the search will succeed and we output "yes" with high probability overall.

Soundness. Suppose now that f is ϵ-far from $\mathcal{C}$.

One possibility is that f is ϵ-far from every $J(\tau^\star)$-junta; if this is the case then w.h.p. the test will output "no" in Step 1.

The other possibility is that f is ϵ-close to a $J(\tau^\star)$-junta f' (or is itself such a junta). Suppose that this is the case and that the testing algorithm reaches Step 2. In Step 2, the algorithm tries to construct a set of labeled examples that is consistent with f'. The algorithm may fail to construct a sample at all; if this happens then it outputs "no." If the algorithm succeeds in constructing a sample S, then w.h.p. this sample is indeed consistent with f'; but in this case, w.h.p. in Step 3 the algorithm will not find any function $g \in \mathcal{C}(\tau^\star)_{J(\tau^\star)}$ that is consistent with all the examples. (If there were such a function g, then standard arguments in learning theory show that w.h.p. any such function $g \in \mathcal{C}(\tau^\star)_{J(\tau^\star)}$ that is consistent with S is in fact close to f'. Since f' is in turn close to f, this would mean that g is close to f. But g belongs to $\mathcal{C}(\tau^\star)_{J(\tau^\star)}$ and hence to $\mathcal{C}$, so this violates the assumption that f is ϵ-far from $\mathcal{C}$.)

3.2.4 The main theorem (Theorem 23)

We now state our main theorem, which is proved in detail in Section 3.3. The algorithm $\mathcal{A}$ is adaptive, but in Section 3.4 we discuss how to make it non-adaptive with only a slight increase in query complexity.

Theorem 23. *There is an algorithm $\mathcal{A}$ with the following properties:*

Let $\mathcal{C}$ be a class of functions from $\mathcal{X}^n$ to $\mathcal{Y}$. Suppose that for every $\tau > 0$, $\mathcal{C}(\tau) \subseteq \mathcal{C}$ is a $(\tau, J(\tau))$-approximator for $\mathcal{C}$. Suppose moreover that for every $\epsilon > 0$, there is a τ satisfying

$$\tau \leq \kappa \cdot \frac{\epsilon^2}{\ln(|\mathcal{X}|) \cdot J(\tau)^2 \cdot \ln^2(J(\tau)) \cdot \ln\ln(J(\tau)) \cdot \ln^2(|\mathcal{C}(\tau)_{J(\tau)}|) \cdot \ln(\frac{\ln(|\mathcal{X}|)}{\epsilon} \ln|\mathcal{C}(\tau)_{J(\tau)}|)}, \tag{3.1}$$

where $\kappa > 0$ is a fixed absolute constant. Let $\tau^\star$ be the largest value τ satisfying (3.1) above. Then algorithm $\mathcal{A}$ makes:

$$\begin{aligned}
& 2sh + (2gJ(\tau^\star)\lceil \lg|\mathcal{X}|\rceil + 1)m \\
= \;& \Theta\left(\frac{1}{\epsilon} J(\tau^\star)^2 \ln^2(J(\tau^\star)) \log\log J(\tau^\star) \ln(|\mathcal{C}(\tau^\star)_{J(\tau^\star)}|)\right) \\
& + \Theta\left(\frac{\lg|\mathcal{X}|}{\epsilon^2} J(\tau^\star)^2 \ln^2(|\mathcal{C}(\tau^\star)_{J(\tau^\star)}|) \ln(\frac{1}{\epsilon} \ln(|\mathcal{C}(\tau^\star)_{J(\tau^\star)}|))\right) \\
= \;& \tilde{O}\left(\frac{\ln|\mathcal{X}|}{\epsilon^2} J(\tau^\star)^2 \ln^2(|\mathcal{C}(\tau^\star)_{J(\tau^\star)}|)\right)
\end{aligned}$$

many black-box queries to f, and satisfies the following:

- *If $f \in \mathcal{C}$ then $\mathcal{A}$ outputs "yes" with probability at least $2/3$;*
- *If f is ϵ-far from $\mathcal{C}$ then $\mathcal{A}$ outputs "no" with probability at least $2/3$.*

Here are some observations to help interpret the bound (3.1). Note that if $J(\tau)$ grows too rapidly as a function of $1/\tau$, e.g. $J(\tau) = \Omega(1/\sqrt{\tau})$, then there will be no $\tau > 0$ satisfying inequality (3.1). On the other hand, if $J(\tau)$ grows slowly as a function of $1/\tau$, e.g. $\log(1/\tau)$, then it is may be possible to satisfy (3.1).

In all of our applications $J(\tau)$ will grow as $O(\log(1/\tau))$, and $\ln |C(\tau)_{J(\tau)}|$ will always be at most $\text{poly}(J(\tau))$, so (3.1) will always be satisfiable. The most typical case for us will be that $J(\tau) \leq \text{poly}(s)\log(1/\tau)$ (where s is a size parameter for the class of functions in question) and $\ln |C(\tau)_{J(\tau)}| \leq \text{poly}(s) \cdot \text{poly}\log(1/\tau)$, which yields $\tau^\star = \tilde{O}(\epsilon^2)/\text{poly}(s)$ and an overall query bound of $\text{poly}(s)/\tilde{O}(\epsilon^2)$.

3.3 Proof of Theorem 23

Let us describe how the parameters s, h, g and m mentioned above (and others) are set. (The table below should perhaps be glossed over on a first reading, but will be useful for subsequent reference.) Given $\epsilon > 0$, let $\tau^\star$ be as described in the theorem statement. We set:

$r \overset{\text{def}}{=} 25J(\tau^\star)^2$	$\Theta(J(\tau^\star)^2)$
$s \overset{\text{def}}{=} 25J(\tau^\star)(7 + \ln r)$	$\Theta(J(\tau^\star)\ln J(\tau^\star))$
$\epsilon_2 \overset{\text{def}}{=} \frac{\epsilon}{2}$	$\Theta(\epsilon)$
$m \overset{\text{def}}{=} \frac{\ln 6\lvert\mathcal{C}(\tau^\star)_{J(\tau^\star)}\rvert}{\epsilon_2}$	$\Theta(\frac{1}{\epsilon}\ln(\lvert\mathcal{C}(\tau^\star)_{J(\tau^\star)}\rvert))$
$\epsilon_1 \overset{\text{def}}{=} \frac{1}{200m}$	$\Theta(\epsilon/\ln(\lvert\mathcal{C}(\tau^\star)_{J(\tau^\star)}\rvert))$
$\theta \overset{\text{def}}{=} \frac{\epsilon_1 J(\tau^\star)}{6er}$	$\Theta(\epsilon/(\ln(\lvert\mathcal{C}(\tau^\star)_{J(\tau^\star)}\rvert)J(\tau^\star)))$
$g \overset{\text{def}}{=} \frac{\ln\left(100mJ(\tau^\star)\lceil\lg\lvert\mathcal{X}\rvert\rceil\right)}{\theta}$	$\Theta\left(\frac{1}{\epsilon}J(\tau^\star)\ln(\lvert\mathcal{C}(\tau^\star)_{J(\tau^\star)}\rvert)\ln\left(\frac{\ln\lvert\mathcal{X}\rvert}{\epsilon}J(\tau^\star)\ln(\lvert\mathcal{C}(\tau^\star)_{J(\tau^\star)}\rvert)\right)\right)$
$h \overset{\text{def}}{=} \frac{(3+2\ln s)}{\theta}$	$\Theta(\frac{1}{\epsilon}\ln(\lvert\mathcal{C}(\tau^\star)_{J(\tau^\star)}\rvert)J(\tau^\star)\ln J(\tau^\star)\ln\ln J(\tau^\star))$

where e is the base of the natural logarithm. Note that $\epsilon_1 + \epsilon_2 < \epsilon$.

Observe that for some suitable (small) absolute constant $\kappa > 0$, our setting of parameters and choice of $\tau^\star$ yields the following bounds that we will use later:

- $2mgJ(\tau^\star)\lceil\lg|\mathcal{X}|\rceil \cdot \tau^\star \leq 1/100$ (used in Lemma 31)

- $2sh \cdot \tau^\star \leq 1/100$ (used in Corollary 30),

- $m(\epsilon_1 + \tau^\star) < 1/100$ (used in Lemma 31).

3.3.1 Step 1: Identifying critical subsets.

Step 1 of the algorithm consists of running the procedure **Identify-Critical-Subsets**, reproduced for convenience in Figure 3-1. This procedure performs $2sh$ queries to f. The procedure is nearly identical to the "two-sided" junta test of Section 4.2 of Fischer *et al.* with two small differences. The first is that we have adjusted various constant factors slightly (we need a smaller failure probability because we are using this in the context of a larger test). The second is that **Identify-Critical-Subsets** outputs the list of subsets that are declared to be not influence-free (whereas the Fischer *et al.* test simply accepts or rejects f), since we will need these subsets for the rest of our test.

We now prove two quick lemmata that will be useful in establishing the soundness and completeness of the algorithm.

Lemma 24. *Let f be a function with at most $J(\tau^\star)$ variables x_i that have $\mathrm{Inf}_f(\{i\}) \geq \theta$. Then with probability at least $1 - 1/400$, each of the variables x_i that have $\mathrm{Inf}_f(\{i\}) \geq \theta$ occurs in some subset I_ℓ that is not declared influence-free by* **Identify-Critical-Subsets**.

Proof. Fix a variable x_i such that $\mathrm{Inf}_f(\{i\}) \geq \theta$. Let I_ℓ denote the subset to which x_i belongs. By the monotonicity and subadditivity of influence (Lemma 13) we have that

$$\theta \leq \mathrm{Inf}_f(\{i\}) \leq \mathrm{Inf}_f(I_\ell) \leq \mathrm{Inf}_f(B_k)$$

where B_k is any block such that $\ell \in \Lambda_k$. This implies that for any such block B_k, the probability that all h iterations of the independence test accept is at most $(1-\theta)^h < \frac{1}{20s^2} < \frac{1}{400sJ(\tau^\star)}$. So the probability that any block that contains x_i is declared influence-free is at most $\frac{1}{400J(\tau^\star)}$. By a union bound over all at most $J(\tau^\star)$ variables x_i that have $\mathrm{Inf}_f(\{i\}) \geq \theta$, the probability that any block that contains such a variable causes any subset I_ℓ containing the variable to be declared influence-free is at most $1/400$. □

Lemma 25. *Let V be any set of $\leq J(\tau^\star)$ variables from $x_1, \ldots, x_n$. Then with probability at least $1 - 1/25$, every subset I_ℓ, $1 \leq \ell \leq r$, contains at most one variable from V.*

Proof. For any fixed pair of variables in V, the probability that they end up in the same subset I_i is $1/r$. Thus by a union bound the probability that any pair of variables from V

end up in the same subset is at most

$$\frac{1}{r}\binom{|V|}{2} < \frac{|V|}{2} \leq \frac{J(\tau^\star)^2}{r} = \frac{1}{25}$$

□

Let $\mathcal{K} \subseteq [n]$ denote a set of coordinates satisfying $\mathrm{Inf}_f(\overline{\mathcal{K}}) < \epsilon_1$. Lemma 19 states that the following function:

$$h(x) \stackrel{\mathrm{def}}{=} \mathop{\mathbf{Plur}}_{z}[f(x_{\mathcal{K}} z_{\overline{\mathcal{K}}})] \tag{3.2}$$

is ϵ_1-close to f.

Let $\mathcal{J}$ denote the set of those coordinates on which f has binary influence at least θ. To prove the soundness of **Identify-Critical-Subsets**, we must prove that if f passes **Identify-Critical-Subsets** with probability greater than 1/3, then it is ϵ_1-close to a $J(\tau^\star)$-junta. This is accomplished by showing that $|\mathcal{J}| \leq J(\tau^\star)$, and that $\mathcal{J}$ can be used in place of $\mathcal{K}$ above, *i.e.*, $\mathrm{Inf}_f(\overline{\mathcal{J}}) < \epsilon_1$. Then we can invoke Lemma 19 to finish the proof. In addition, we will also prove some properties about the subsets $I_{i_1}, \ldots, I_{i_j}$ output by the algorithm.

Lemma 26. *If f passes* **Identify-Critical-Subsets** *with probability higher than 1/3, then:*

(i) $|\mathcal{J}| \leq J(\tau^\star)$*;*

(ii) $\mathrm{Inf}_f(\overline{\mathcal{J}}) < \epsilon_1$*,*

and f is thus ϵ_1-close to a $J(\tau^\star)$-junta by Lemma 19.

Let h be defined as in Equation (3.2) using $\mathcal{J}$ as the set $\mathcal{K}$. Suppose that f passes **Identify-Critical-Subsets** *with probability greater than 1/3. Then given that f passes, the sets output by the algorithm, $I_{i_1}, \ldots, I_{i_j}$, have the following properties with probability at least 6/7:*

(iii) Every $x_i \in \mathcal{J}$ occurs in some subset I_{i_ℓ} that is output;

(iv) Every subset I_{i_ℓ}, $1 \leq \ell \leq j$, contains at most one variable from $\mathcal{J}$.

Proof. **Condition (i):** (paraphrasing Prop. 3.1 and Lemma 4.3 of [26]) Suppose $|\mathcal{J}| > J(\tau^\star)$. Then with probability at least $3/4$ (using the same argument as in the proof of

Lemma 25), the number of subsets I_{i_ℓ} containing an element from $\mathcal{J}$ is at least $J(\tau^\star)+1$. For any fixed subset I_{i_ℓ} that contains an element from $\mathcal{J}$ and any fixed block B containing I_{i_ℓ}, the probability of B being declared influence-free is bounded by:

$$(1-\theta)^h = (1-\theta)^{(3+2\ln s)/\theta} < \frac{1}{20s(J(\tau^\star)+1)}.$$

Union bounding over the at most s blocks to which the subset I_{i_ℓ} can belong, and union bounding over $J(\tau^\star)+1$ subsets that contain an element from $\mathcal{J}$, we have that with probability at least $\frac{3}{4}\cdot\frac{19}{20} > \frac{2}{3}$, at least $J(\tau^\star)+1$ subsets are not declared influence-free and consequently f does not pass **Identify-Critical-Subsets**. Thus, if f passes **Identify-Critical-Subsets** with probability at least $1/3$, it must be the case that $|\mathcal{J}| \le J(\tau^\star)$.

Condition (ii): (paraphrasing Prop. 3.1 and Lemma 4.3 of [26]) Suppose $\mathrm{Inf}_f(\overline{\mathcal{J}}) \ge \epsilon_1$. We will show that each block B_ℓ has high influence with high probability. This will imply that the number of blocks not declared influence-free is larger than $s/2$ with high probability, so the test will reject with probability at least $2/3$.

In order to show that each block has reasonably high influence, we will make use of the following technical tool which was defined by Fischer *et al.* [26].

Definition 27. Let f be a function that maps $\mathcal{X}^n$ to $\{-1,1\}$, and let $\mathcal{J} \subseteq [n]$ be a set of coordinates. For each coordinate $i \in [n]$, we define the *unique variation of i with respect to $\mathcal{J}$* as

$$\mathrm{Ur}_f(i) \stackrel{\mathrm{def}}{=} \mathrm{Inf}_f([i]\backslash\mathcal{J}) - \mathrm{Inf}_f([i-1]\backslash\mathcal{J}),$$

and for $I \subseteq [n]$ we define the unique variation of I as

$$\mathrm{Ur}_f(I) \stackrel{\mathrm{def}}{=} \sum_{i\in I} \mathrm{Ur}_f(i).$$

The most important property of the unique variation that distinguishes it from the other notions of influence is that for any set of coordinates, its unique variation simply equals the sum of the unique variation of each of its coordinates. This makes it easy to compute the expected value of the unique variation on a random subset of coordinates. Furthermore, the following properties hold.

Lemma 28 (Fischer *et al.* [26]).

- *For any coordinate* $i \in [n]$, $\mathrm{Ur}_f(\{i\}) \leq \mathrm{Inf}_f(\{i\})$.
- *For every set* $I \subseteq [n]$ *of coordinates,* $\mathrm{Ur}_f(I) \leq \mathrm{Inf}_f(I \backslash \mathcal{J})$.
- $\mathrm{Ur}_f([n]) = \mathrm{Ur}_f([n]\backslash \mathcal{J}) = \mathrm{Inf}_f([n]\backslash \mathcal{J})$.

Proof. The proof is straightforward from the definition and Proposition 12. □

Now fix any value $\ell \in [s]$. The block B_ℓ is a random set of variables independently containing each variable x_i coordinate with probability $J(\tau^\star)/r$. Let $\mathrm{Ur}_f(I)$ be the unique variation of a set I with respect to $\mathcal{J}$ (see Definition 27). Then the expected value of the unique influence of B_ℓ is

$$\mathbf{E}[\mathrm{Ur}_f(B_\ell)] = \frac{J(\tau^\star)}{r}\mathrm{Ur}_f(\overline{\mathcal{J}}) = \frac{J(\tau^\star)}{r}\mathrm{Inf}_f(\overline{\mathcal{J}}) \geq \frac{\epsilon_1 J(\tau^\star)}{r}.$$

By Lemma 28 and Lemma 21 (taking $\eta = 1/2e$, $t = \theta$ and $\alpha = \frac{\epsilon_1 J(\tau^\star)}{r}$ in Lemma 21), we have

$$\Pr\left[\mathrm{Inf}_f(B_\ell) < \frac{\epsilon_1 J(\tau^\star)}{2er}\right] \leq \Pr\left[\mathrm{Ur}_f(B_\ell) < \frac{\epsilon_1 J(\tau^\star)}{2er}\right] < \exp\left(-\frac{\epsilon_1 J(\tau^\star)}{2er\theta}\right) = e^{-3} < \frac{1}{12}$$

Hence the probability that the influence of B_ℓ is less than $\epsilon_1 J(\tau^\star)/2er = 3\theta$ is less than $1/12$. This implies that the expected number of blocks with influence less than 3θ is smaller than $s/12$. From Markov's inequality we get that with probability at least $1 - \frac{1}{6}$, there are less than $s/2$ blocks with influence smaller than 3θ.

The probability of a block with influence greater than 3θ being declared influence free is at most:

$$(1-3\theta)^h = (1-3\theta)^{(3+2\ln s)/\theta} < e^{-(9+6\ln s)} < \frac{1}{1000s},$$

and therefore with probability at least $1 - \frac{1}{1000}$ none of these blocks are declared influence free. So with overall probability at least $1 - (\frac{1}{6} + \frac{1}{1000}) > \frac{2}{3}$, more than $s/2$ blocks are declared influence-free and the test rejects.

Condition (iii): We may suppose that f passes **Identify-Critical-Subsets** with probability greater than 1/3. Then we know that $|\mathcal{J}| \leq J(\tau^\star)$ by Condition (i). By Lemma 24, given that f passes **Identify-Critical-Subsets**, the probability that some $x_i \in \mathcal{J}$ does not occur in some subset I_{i_ℓ} output by the algorithm is at most $3/400$. (The bound is $3/400$ rather than $1/400$ because we are conditioning on f passing **Identify-Critical-Subsets**, which takes place with probability at least $1/3$.)

Condition (iv): As above we may suppose that f passes **Identify-Critical-Subsets** with probability greater than 1/3. By Condition (i) we know that $|\mathcal{J}| \leq J(\tau^\star)$, so we may apply Lemma 25. Hence conditioned on f passing **Identify-Critical-Subsets** (an event which has probability at least $1/3$), the probability that any subset I_{i_ℓ} output by the algorithm includes more than one relevant variable of h is at most $3/25$.

Summing the probabilities, we get that conditions (iii) and (iv) are true with probability at least $1 - (\frac{3}{400} + \frac{3}{25}) > \frac{6}{7}$. □

Fischer *et al.* establish completeness by showing that if f is a junta then with probability at least $2/3$ conditions (a) and (b) are both satisfied in Step 4. However we need more than this, since we are going to use the subsets $I_{i_1}, \ldots, I_{i_j}$ later in the test. We will prove:

Lemma 29. *Suppose that f is a $J(\tau^\star)$-junta. Let $\mathcal{K}$ be the set of variables satisfying $\mathrm{Inf}_f(\{i\}) \geq \theta$. Then with probability at least $6/7$, algorithm* **Identify-Critical-Subsets** *outputs a list of $j \leq J(\tau^\star)$ subsets $I_{i_1}, \ldots, I_{i_j}$ with the property that:*

(i) each variable $x_i \in \mathcal{K}$ occurs in some subset I_ℓ that is output;

(ii) $\mathrm{Inf}_f(\overline{\mathcal{K}}) < \epsilon_1$;

(iii) Every subset I_{i_ℓ}, $1 \leq \ell \leq j$, contains at most one relevant variable for f.

Proof. **Condition (a):** Fix any partition $I_1, \ldots, I_r$. If f is a $J(\tau^\star)$-junta, then it is independent of all but at most $J(\tau^\star)$ subsets in the partition. Hence for any fixed ℓ, the probability over the selection of the blocks that f is independent of B_ℓ is at least:

$$\begin{aligned}\binom{r-J(\tau^\star)}{J(\tau^\star)}\Big/\binom{r}{J(\tau^\star)} &> \left(\frac{r-2J(\tau^\star)}{r-J(\tau^\star)}\right)^{J(\tau^\star)}\\ &= \left(1-\frac{J(\tau^\star)}{r-J(\tau^\star)}\right)^{J(\tau^\star)}\\ &> 1-\frac{J(\tau^\star)^2}{r-J(\tau^\star)}\\ &\geq \frac{23}{24}.\end{aligned}$$

The probability that f depends on more than half of the blocks is therefore smaller than $\frac{2}{24}$ using the Markov inequality. (See [26], Lemma 4.2).

Condition (b) fails with probability at most:

$$r\left(1-\frac{1}{25J(\tau^\star)}\right)^s = r\left(1-\frac{1}{25J(\tau^\star)}\right)^{25J(\tau^\star)(7+\ln r)} < r\cdot S\frac{1}{1000r} = \frac{1}{1000},$$

(see [26], Lemma 4.2, which uses $s = 20J(3+\ln r)$ instead).

Condition (i): Since we assume that f is a $J(\tau^\star)$-junta we may apply Lemma 24, and thus the probability that any variable x_i that has $\mathrm{Inf}_f(\{i\}) \geq \theta$ occurs in a subset I_ℓ that is declared influence-free by **Identify-Critical-Subsets** is at most $1/400$.

Condition (ii): Let $\mathcal{L}$ denote the relevant variables for f that are not in $\mathcal{K}$, and let $\mathcal{T}$ denote $[n]\setminus(\mathcal{K}\cup\mathcal{L})$. By Lemma 13 we have

$$\mathrm{Inf}_f(\mathcal{L}) \leq \sum_{i\in\mathcal{L}}\mathrm{Inf}_f(\{i\}) \leq J(\tau^\star)\theta = J(\tau^\star)\frac{\epsilon_1 J(\tau^\star)}{6e\cdot 25J(\tau^\star)^2} < \epsilon_1.$$

We have that $\overline{\mathcal{K}} = \mathcal{L}\cup\mathcal{T}$, so by Lemma 13 we get

$$\mathrm{Inf}_f(\overline{\mathcal{K}}) = \mathrm{Inf}_f(\mathcal{L}\cup\mathcal{T}) \leq \mathrm{Inf}_f(\mathcal{L}) + \mathrm{Inf}_f(\mathcal{T}) = \mathrm{Inf}_f(\mathcal{L}) \leq \epsilon_1.$$

Condition (iii): Suppose there are precisely $j' \leq J(\tau^\star)$ many relevant variables. Then by Lemma 25, the probability that any subset $I_1,\ldots,I_r$ ends up with two or more relevant variables is at most $1/25$.

Summing failure probabilities, we find that all the required conditions are fulfilled with probability at least $1 - (1/12 + 1/1000 + 1/400 + 1/25)$ which is greater than $6/7$. □

We are ultimately interested in what happens when **Identify-Critical-Subsets** is run on a function from $\mathcal{C}$. Using the above, we have:

Corollary 30. *Suppose f is $\tau^\star$-close to some $J(\tau^\star)$-junta f'. Then with probability at least $5/6$, algorithm* **Identify-Critical-Subsets** *outputs a list of $j \leq J(\tau^\star)$ subsets $I_{i_1}, \dots, I_{i_j}$ with the property that*

(i') each variable x_i which has $\mathrm{Inf}_{f'}(\{i\}) \geq \theta$ occurs in some subset I_ℓ that is output;

(ii') $\mathrm{Inf}_{f'}(\overline{\mathcal{K}}) < \epsilon_1$;

(iii') Every subset I_{i_ℓ}, $1 \leq \ell \leq j$, contains at most one relevant variable for f'.

Proof. Observe that each of the $2sh$ queries that **Identify-Critical-Subsets** performs is on an input that is selected *uniformly at random* from $\mathcal{X}^n$ (note that the query points are not all independent of each other, but each one considered individually is uniformly distributed). Since f and f' disagree on at most a $\tau^\star$ fraction of all inputs, the probability that **Identify-Critical-Subsets** queries any point on which f and f' disagree is at most $2sh \cdot \tau^\star < 1/100$. Since by Lemma 29 we know that conditions (i'), (ii') and (iii') would hold with probability at least $6/7$ if the black-box function were f', we have that conditions (i), (ii) and (iii) hold with probability at least $6/7 - 1/100 > 5/6$ with f as the black-box function. □

3.3.2 Step 2: Constructing a sample.

Step 2 of the algorithm consists of running the procedure **Construct-Sample**. The algorithm makes $(2gj\lceil \lg |\mathcal{X}| \rceil + 1)m$ many queries to f, and either outputs "no" or else outputs a sample of m labeled examples (x, y) where each x belongs to $\mathcal{X}^{J(\tau^\star)}$.

We introduce some notation. Given functions $f : \mathcal{X}^n \rightarrow \mathcal{Y}$ and $f' : \mathcal{X}^j \rightarrow \mathcal{Y}$ with $j \leq n$ and a permutation $\sigma : [n] \rightarrow [n]$, we write $f \overset{\sigma}{\sim} f'$ to indicate that $\forall x \in \mathcal{X}^n$: $f'(x_{\sigma(1)}, \dots, x_{\sigma(j)}) = f(x_1, \dots, x_n)$. If $f : \mathcal{X}^n \rightarrow \mathcal{Y}$ is a function with j relevant variables, we use f_j^σ to mean the function over j variables that results by mapping the i-th relevant variable under f to the i-th character of a j-character string over $\mathcal{X}$; i.e. if σ is a permutation which induces such a mapping, then f_j^σ is the function satisfying $f \overset{\sigma}{\sim} f_j^\sigma$. Given a function $f : \mathcal{X}^j \rightarrow \mathcal{Y}$ and permutation $\sigma : [n] \rightarrow [n]$, we write $f_\uparrow^\sigma$ to denote the j-junta satisfying $f_\uparrow^\sigma \overset{\sigma}{\sim} f$.

Lemma 31. *Given $f : \mathcal{X}^n \rightarrow \mathcal{Y}$ and some $J(\tau^\star)$-junta f' that is $\tau^\star$-close to f, let $\mathcal{K}$ be the set of variables satisfying $\mathrm{Inf}_{f'}(\{i\}) \geq \theta$. Suppose* **Construct-Sample** *is given oracle access to f and inputs $I_{i_1}, \ldots, I_{i_j}$, with $j \leq J(\tau^\star)$, where:*

1. *Each variable $x_i \in \mathcal{K}$ is contained in one of $I_{i_1}, \ldots, I_{i_j}$;*

2. *$\mathrm{Inf}_{f'}(\overline{\mathcal{K}}) < \epsilon_1$;*

3. *Every subset I_{i_ℓ}, $1 \leq \ell \leq j$, contains at most one relevant variable for f'.*

Let h be the function defined as in Equation 3.2 using the set $\mathcal{K}$. Let $\mathcal{H} \subseteq \mathcal{K}$ be the set of relevant variables for h, and let $\sigma : [n] \rightarrow [n]$ be some permutation which maps the variable from $\mathcal{H}$ in bin I_{i_ℓ} to bit ℓ. Then with probability at least $1 - 3/100$, **Construct-Sample** *outputs a set of m uniform, random examples labeled according to a $J(\tau^\star)$-junta g which depends on no variables outside of $\mathcal{K}$ and satisfies $\Pr_{z \in \mathcal{X}^n}[g_\uparrow^\sigma(z) \neq f'(z)] \leq \epsilon_1$.*

Proof. By Lemma 19 we have that $\Pr_{z \in \mathcal{X}^n}[h(z) \neq f'(z)] \leq \epsilon_1$. We now show that except with probability less than $3/100$, **Construct-Sample** produces a set S of m examples that are uniform, random, and labeled according to $g \overset{\mathrm{def}}{=} h_{J(\tau^\star)}^\sigma$ (note that $g_\uparrow^\sigma \equiv h$).

Consider a particular iteration of Step 1 of **Construct-Sample**. The iteration generates an example x that is uniform random and labeled according to g if

(a) for every bin I_{i_ℓ} which contains a variable from $\mathcal{H}$, Step 1(b)ii constructs the index w such that X_w contains that variable;

(b) for every bin I_{i_ℓ} that contains no variable from $\mathcal{H}$, in every iteration of Step 1(b)ii(C) at most one of $\mathcal{X}_0, \mathcal{X}_1$ is marked, and the value w that is considered in Step 1(b)iv lies in $[0, |\mathcal{X}| - 1]$; and

(c) $h(z) = f(z)$.

Item (a) ensures that if I_{i_ℓ} contains a variable from $\mathcal{H}$, then x_ℓ takes the value of that variable under the assignment z (and, since z is a uniform random value, so is x_ℓ). Item (b) ensures that if I_{i_ℓ} contains no variable from $\mathcal{H}$, **Construct-Sample** does not output "no" and assigns x_ℓ a uniform random value, because x_ℓ either gets a fresh uniform random value in Step 1(b)iii or gets the value of z (which is uniform random). Together, these ensure that $g(x) = g(z_{\sigma(1)}, \ldots, z_{\sigma(J(\tau^\star))})$, and item (c) ensures that the label for the example x will be $h(z) = g(x)$.

It remains to bound the probability that any of (a), (b), or (c) fail to hold. Suppose first that every query of every iteration of the independence test is answered according to f'. Then item (3) implies that (a) can only fail to hold if we do not manage to figure out some bit of w in Step 1(b)ii for some ℓ for which I_{i_ℓ} contains a variable from $\mathcal{H}$ (which means that all g executions of the independence test pass for that bit failed), and it also implies that condition (b) holds (it is possible for a bit of w to be unspecified, but not for both $\mathcal{X}_0, \mathcal{X}_1$ to be marked or for w to be set to an out-of-range value). Thus the probability that either (a) or (b) fails to hold is at most

$$j\lceil \lg |\mathcal{X}|\rceil (1-\theta)^g + 2jg\lceil \lg |\mathcal{X}|\rceil \cdot \tau^\star,$$

where the first term bounds the probability that all $g\lceil \lg |\mathcal{X}|\rceil$ executions of the independence test pass for some ℓ and the second term bounds the probability that any execution of the independence test queries a point z such that $f(z) \neq f'(z)$. Finally, the probability that (c) fails to hold is at most $\epsilon_1 + \tau^\star$.

Now considering all m iterations, we have that the overall probability of either outputting "no" or obtaining a bad example in the m-element sample is at most $mj\lceil \lg |\mathcal{X}|\rceil (1-\theta)^g + 2jgm\lceil \lg |\mathcal{X}|\rceil \cdot \tau^\star + (\epsilon_1 + \tau^\star)m \leq 1/100 + 1/100 + 1/100$, and the lemma is proved. □

3.3.3 Step 3: Checking consistency.

The final step of the algorithm, Step 3, is to run **Check-Consistency**. This step makes no queries to f.

The following two lemmata establish completeness and soundness of the overall test and conclude the proof of Theorem 23.

Lemma 32. *Suppose that $f \in \mathcal{C}$. Then with probability at least 2/3, algorithm $\mathcal{A}$ outputs yes.*

Proof. Let f' be some $J(\tau^\star)$-junta in $\mathcal{C}(\tau^\star)$ that is $\tau^\star$-close to f. By Corollary 30, we have that except with probability at most $1/6$, f passes **Identify-Critical-Subsets** and the inputs $I_{i_1}, \dots, I_{i_j}$ given to **Construct-Sample** will satisfy conditions (i')-(iii'). Let $\mathcal{K}$ be the set consisting of those variables that have binary influence at least θ under f'. We use Lemma 31 to conclude that with probability at least $1 - 3/100$, **Construct-Sample** outputs m uniform, random examples labeled according to some $J(\tau^\star)$-junta g satisfying $\Pr_z[g^\sigma_\uparrow(z) \neq f'(z)] \leq \epsilon_1$. Let σ' map the variables in $\mathcal{K}$ to the same values as σ, but also map the remaining, possibly relevant variables of f' to the remaining $J(\tau^\star) - j$ bits. Clearly $\Pr_z[g^{\sigma'}_\uparrow(z) \neq f'(z)] \leq \epsilon_1$, and since the relevant variables of $g^{\sigma'}_\uparrow$ (which are contained in $\mathcal{K}$) are a subset of the relevant variables of f', we have that $\Pr_x[g(x) \neq (f')^{\sigma'}_{J(\tau^\star)}(x)] \leq \epsilon_1$.

Assuming that **Construct-Sample** outputs m uniform random examples labeled according to g, they are also labeled according to $f'^{\sigma'}_{J(\tau^\star)} \in \mathcal{C}(\tau^\star)_{J(\tau^\star)}$ except with probability at most $\epsilon_1 m$. Summing all the failure probabilities, we have that **Check-Consistency** does not output "yes" with probability at most $1/6 + 3/100\epsilon_1 m < 1/3$, and the lemma is proved. □

Lemma 33. *Suppose that f is ϵ-far from $\mathcal{C}$. Then the probability that algorithm $\mathcal{A}$ outputs "yes" is less than 1/3.*

Proof. We assume that f passes **Identify-Critical-Subsets** with probability greater than 1/3 (otherwise we are done), and show that if f passes **Identify-Critical-Subsets**, it will be rejected by **Construct-Sample** or **Check-Consistency** with probability at least 2/3.

Assume f passes **Identify-Critical-Subsets** and outputs $I_{i_1}, \dots, I_{i_j}$. Using Lemma 26, we know that except with probability at most 1/7, $\mathcal{J}$, the set of variables with binary influence at least θ under f, satisfies:

- $\mathrm{Inf}_f(\overline{\mathcal{J}}) < \epsilon_1$;

- each variable in $\mathcal{J}$ is contained in some bin I_{i_ℓ} that is output;

- each bin I_{i_ℓ} contains at most one variable from $\mathcal{J}$.

As in Lemma 31, we construct a function h using the variables in $\mathcal{J}$ according to Equation 3.2 in Section 3.3.1. Let $\mathcal{H} \subseteq \mathcal{J}$ be the set of relevant variables for h, and let $\sigma : [n] \rightarrow [n]$ be as in Lemma 31. We have that $\Pr_{z \in \mathcal{X}^n}[h(z) \neq f(z)] \leq \epsilon_1$. We show that with probability greater than $1 - 2/100$, **Construct-Sample** either outputs "no" or a set of m uniform, random examples labeled according to $g \overset{\text{def}}{=} h^{\sigma}_{J(\tau^\star)}$.

Consider a particular random draw of $z \in \mathcal{X}^n$ As in Lemma 31, this draw will yield a uniform, random example $x \in \mathcal{X}^{J(\tau^\star)}$ for g as long as

(a) for every bin I_{i_ℓ} which contains a variable from $\mathcal{H}$, Step 1(b)ii constructs the index w such that X_w contains that variable;

(b) for every bin I_{i_ℓ} that contains no variable from $\mathcal{H}$, in every iteration of Step 1(b)ii(C) at most one of $\mathcal{X}_0, \mathcal{X}_1$ is marked, and the value w that is considered in Step 1(b)iv lies in $[0, |\mathcal{X}| - 1]$; and

(c) $h(z) = f(z)$.

The probability of (c) failing is bounded by ϵ_1. The probability of (a) failing is at most $j\lceil \lg |\mathcal{X}| \rceil (1 - \theta/2)^g < \frac{1}{100m}$. If neither (a) nor (c) occurs, then the example satisfies (a), (b) and (c) unless it fails to satisfy (b), but if it fails to satisfy (b) **Construct-Sample** outputs "no" in Step 1(b).ii.F or Step 1(b).iv. a Thus if f passes **Identify-Critical-Subsets**, we have that with probability at least

$$1 - 1/7 - 1/100 - \epsilon_1 m \geq 1 - 1/7 - 2/100 > 1 - 1/6$$

Construct-Sample either outputs "no" or it outputs a set of m uniform random examples for g.

Suppose **Construct-Sample** outputs such a set of examples. We claim that with probability at least $1 - 1/6$ over the choice of random examples for g, **Check Consistency** will output "no". Suppose that **Check Consistency** finds some $g' \in \mathcal{C}(\tau^\star)_{J(\tau^\star)}$ consistent with all m examples. Then g' cannot be ϵ_2-close to g. (Otherwise, we have that

$\Pr_z[g'^{\sigma}_{\uparrow}(z) \neq g^{\sigma}_{\uparrow}(z)] \leq \epsilon_2$, from which it follows that $\Pr_z[g'^{\sigma}_{\uparrow}(z) \neq f(z)] \leq \epsilon_2 + \epsilon_1 < \epsilon$ since $g^{\sigma}_{\uparrow}(z)$ is ϵ_1-close to f. But $g' \in \mathcal{C}(\tau^\star)_{J(\tau^\star)}$, so $g'^{\sigma}_{\uparrow} \in \mathcal{C}(\tau^\star) \subseteq \mathcal{C}$ which contradicts our assumption that f is ϵ-far from $\mathcal{C}$.) By choice of m, the probability there exists a $g' \in \mathcal{C}(\tau^\star)_{J(\tau^\star)}$ consistent with all m examples that is not ϵ_2-close to g is at most $|\mathcal{C}(\tau^\star)_{J(\tau^\star)}|(1-\epsilon_2)^m = 1/6$. Thus, if f passes **Identify-Critical-Subsets**, then **Construct-Sample** and **Check-Consistency** output "yes" with probability less than $1/6 + 1/6 < 1/3$. This proves the lemma. □

3.4 Making the algorithm non-adaptive

The algorithm $\mathcal{A}$ presented in the previous section is adaptive. In this section, we show that $\mathcal{A}$ can be made non-adaptive without considerably increasing its query complexity.

The only part of our current algorithm that fails to be non-adaptive is Step 2, the **Construct-Sample** subroutine, which relies on knowledge of the critical subsets identified in Step 1. To remove this reliance, one approach is to modify the **Construct-Sample** subroutine (in particular the `for`-loop in step 1(b)) so that it iterates over every subset rather than just the critical ones. This modified subroutine can be run before the critical subsets are even identified, and the queries it makes can be stored for future use. Later, when the critical subsets are identified, the queries made during the iterations over non-critical subsets can be ignored. Since there are $\Theta(J(\tau^\star)^2)$ total subsets compared to the $\Theta(J(\tau^\star))$ critical ones, the cost of this modified algorithm is an additional factor of $\Theta(J(\tau^\star))$ in the query complexity given in Theorem 23. For all of our applications, this translates to only a small polynomial increase in query complexity (in most cases, merely an additional factor of $\Theta(s)$).

We briefly sketch a more efficient approach to nonadaptivity; this is done essentially by combining Steps 1 and 2. Specifically, each of the m examples that we currently generate in Step 2 can be generated using the techniques from Step 1. To generate a single example, we take a random assignment to all of the variables, and we split each set I_i of variables into $|\mathcal{X}|$ sets $I_{i,\omega}$, where $I_{i,\omega}$ consists of those variables in I_i that were assigned ω. We get $\Theta(|\mathcal{X}| J(\tau^\star)^2)$ sets of variables. Now, as in the **Identify-Critical-Subsets** subroutine, we create $k = O(J(\tau^\star)\log(|\mathcal{X}|J(\tau^\star)))$ blocks, each consisting of exactly $|\mathcal{X}|J(\tau^\star)$ sets $I_{i,\omega}$ chosen at random. We run the independence test $\Theta(\frac{1}{\theta}\log(km))$ times on each of these blocks, and declare influence free those not rejected even once. If for each critical subset I_i, at least $|\mathcal{X}| - 1$ sets $I_{i,\omega}$ are declared influence free on behalf of some block, the remaining $I_{i,\omega}$ which are not declared influence free give us the values of the influential variables. One can show that this happens with probability $1 - O(1/m)$. Therefore when the procedure is repeated to generate all m examples, the probability of overall success is constant. Without going into a detailed analysis, the query complexity of this modified algorithm is essentially the same as that given in Theorem 23, namely $\tilde{O}\left(\frac{\ln|\mathcal{X}|}{\epsilon^2} J(\tau^\star)^2 \ln^2(|\mathcal{C}(\tau^\star)_{J(\tau^\star)}|)\right)$. Thus, for all of our applications, we can achieve non-adaptive testers with the same complexity bounds stated in Theorems 34 and 38.

3.5 Applications to Testing Classes of Functions

The algorithm $\mathcal{A}$ in Theorem 23 can be applied to many different classes of functions that were not previously known to be testable. The following two subsections state and prove our results for Boolean and non-Boolean functions, respectively. These testing results are collected in Table 1.1.

3.5.1 Boolean Functions

Theorem 34. *For any s and any $\epsilon > 0$, Algorithm $\mathcal{A}$ yields a testing algorithm for*

1. *decision lists using $\tilde{O}(1/\epsilon^2)$ queries;*

2. *size-s decision trees using $\tilde{O}(s^4/\epsilon^2)$ queries;*

3. *size-s branching programs using $\tilde{O}(s^4/\epsilon^2)$ queries;*

4. *s-term DNF using $\tilde{O}(s^4/\epsilon^2)$ queries;*

5. *size-s Boolean formulas using $\tilde{O}(s^4/\epsilon^2)$ queries;*

6. *size-s Boolean circuits using $\tilde{O}(s^6/\epsilon^2)$ queries;*

7. *functions with Fourier degree at most* d *using* $\tilde{O}(2^{6d}/\epsilon^2)$ *queries.*

Proof. We describe each class of functions and apply Theorem 23 to prove each part of the theorem.

Decision Lists. A *decision list* L of length m is described by a list $(\ell_1, b_1), \ldots, (\ell_m, b_m)$, b_{m+1} where each ℓ_i is a Boolean literal and each b_i is an output bit. Given an input $x \in \{0,1\}^n$ the value of L on x is b_j, where $j \geq 1$ is the first value such that ℓ_j is satisfied by x. If ℓ_j is not satisfied by x for all $j = 1, \ldots, m$ then the value of $L(x)$ is b_{m+1}.

Let $\mathcal{C}$ denote the class of all Boolean functions computed by decision lists. Since only a $1/2^j$ fraction of inputs x cause the $(j+1)$-st literal ℓ_j in a decision list to be evaluated, we have that the class $\mathcal{C}(\tau) \stackrel{\text{def}}{=} \{$all functions computed by decision lists of length $\log(1/\tau)\}$ is a $(\tau, J(\tau))$-approximator for $\mathcal{C}$, where $J(\tau) \stackrel{\text{def}}{=} \log(1/\tau)$. We have $|\mathcal{C}(\tau)_{J(\tau)}| \leq 2 \cdot 4^{\log(1/\tau)}(\log(1/\tau))!$. This yields $\tau^\star = \tilde{O}(\epsilon^2)$, so Theorem 23 thus yields part (1) of Theorem 34.

Decision Trees. A *decision tree* is a rooted binary tree in which each internal node is labeled with a variable x_i and has precisely two children and each leaf is labeled with an output bit. A decision tree computes a Boolean function in the obvious way: given an input x, the value of the function on x is the output bit reached by starting at the root and going left or right at each internal node according to whether the variable's value in x is 0 or 1. The *size* of a decision tree is simply the number of leaves of the tree (which is one more than the number of internal nodes).

Let $\mathcal{C}$ denote the class of all Boolean functions computed by decision trees of size at most s. It is obvious that any size-s decision tree depends on at most s variables. We may thus take $\mathcal{C}(\tau) \stackrel{\text{def}}{=} \mathcal{C}$ and we trivially have that $\mathcal{C}(\tau)$ is a $(\tau, J(\tau))$-approximator for $\mathcal{C}$ with $J(\tau) \stackrel{\text{def}}{=} s$.

Now we bound $|\mathcal{C}(\tau)_{J(\tau)}|$ by $(8s)^s$. It is well known that the number of s-leaf rooted binary trees in which each internal node has precisely two children is the Catalan number $C_{s-1} = \frac{1}{s}\binom{2s-2}{s-1}$, which is at most 4^s. For each of these possible tree topologies there are at most s^{s-1} ways to label the $s-1$ internal nodes with variables from $x_1, \ldots, x_s$. Finally, there are precisely 2^s ways to choose the leaf labels. So the total number of decision trees of size s over variables $x_1, \ldots, x_s$ is at most $4^s \cdot s^{s-1} \cdot 2^s < (8s)^s$.

We thus have $\tau^\star = \tilde{O}(\epsilon^2/s^4)$ in Theorem 23, and we obtain part (2) of Theorem 34.

Branching Programs. Similar results can be obtained for *branching programs.* A branching program of size s is a rooted s-node directed acyclic graph with two sink nodes labeled 0 and 1. Each internal node has fanout two (and arbitrary fan-in) and is labeled with a variable from $x_1, \ldots, x_n$. Given an input x, the value of the branching program on x is the output bit reached as described above.

Let $\mathcal{C}$ denote the class of all s-node branching programs over $\{0,1\}^n$. As with decision trees we may take $\mathcal{C}(\tau) \stackrel{\text{def}}{=} \mathcal{C}$ and $J(\tau) \stackrel{\text{def}}{=} s$. We show that $|\mathcal{C}(\tau)_{J(\tau)}| \leq s^s(s+1)^{2s}$.

The graph structure of the DAG is completely determined by specifying the endpoints of each of the two outgoing edges from each of the s internal vertices. There are at most $s+1$ possibilities for each endpoint (at most $s-1$ other internal vertices plus the two sink nodes), so there are at most $(s+1)^{2s}$ possible graph structures. There are at most s^s ways to label the s nodes with variables from $\{x_1, \ldots, x_s\}$. Thus the total number of possibilities for a size-s branching program over $x_1, \ldots, x_s$ is at most $s^s(s+1)^{2s}$.

Again we have $\tau^\star = \tilde{O}(\epsilon^2/s^4)$, so Theorem 23 yields part (3) of Theorem 34.

DNF Formulas. An s-term DNF formula is an s-way OR of ANDs of Boolean literals. A *k-DNF* is a DNF in which each term is of length at most k.

It is well known that any s-term DNF formula over $\{0,1\}^n$ is τ-close to a $\log(s/\tau)$-DNF with at most s terms (see e.g. [62] or Lemma 35 below). Thus if $\mathcal{C}$ is the class of all s-term DNF formulas over $\{0,1\}^n$, we may take $\mathcal{C}(\tau)$ to be the class of all s-term $\log(s/\tau)$-DNF, and we have that $\mathcal{C}(\tau)$ is a $(\tau, J(\tau))$-approximator for $\mathcal{C}$ with $J(\tau) \stackrel{\text{def}}{=} s\log(s/\tau)$. An easy counting argument shows that $|\mathcal{C}(\tau)_{J(\tau)}| \leq (2s\log(s/\tau))^{s\log(s/\tau)}$. We get $\tau^\star = \tilde{O}(\epsilon^2/s^4)$, so Theorem 23 yields part (4) of Theorem 34.

Boolean Formulas. We define a *Boolean formula* to be a rooted tree in which each internal node has arbitrarily many children and is labeled with either AND or OR and each leaf is labeled with a Boolean variable x_i or its negation $\overline{x}_i$. The size of a Boolean formula is the number of AND/OR gates it contains.

Let $\mathcal{C}$ denote the class of all Boolean formulas of size at most s. Similar to the case of

DNF, we have the following easy lemma:

Lemma 35. *Any size-s Boolean formula (or size-s circuit) over $\{0,1\}^n$ is τ-close to a size-s formula (or size-s circuit) in which each gate has at most $\log(s/\tau)$ inputs that are literals.*

Proof. If a gate g has more than $\log(s/\tau)$ many inputs that are distinct literals, the gate is τ/s-approximated by a constant function (1 for OR gates, 0 for AND gates). Performing such a replacement for each of the s gates in the circuit yields a τ-approximator for the overall formula (or circuit). □

We may thus take $\mathcal{C}(\tau)$ to be the class of all size-s Boolean formulas in which each gate has at most $\log(s/\tau)$ distinct literals among its inputs, and we have that $\mathcal{C}(\tau)$ is a $(\tau, J(\tau))$-approximator for $\mathcal{C}$ with $J(\tau) \overset{\text{def}}{=} s\log(s/\tau)$. An easy counting argument shows that $|\mathcal{C}(\tau)_{J(\tau)}| \leq (2s\log(s/\tau))^{s\log(s/\tau)+s}$; for each of the s gates there is a two-way choice for its type (AND or OR) and an at most s-way choice for the gate that it feeds into. There are also at most $\log(s/\tau)$ literals from $x_1, \ldots, x_{s\log(s/\tau)}, \overline{x}_1, \ldots, \overline{x}_{s\log(s/\tau)}$ that feed into the gate. Thus there are at most $(2s\log(s/\tau))^{\log(s/\tau)+1}$ possibilities for each of the s gates, and consequently at most $(2s\log(s/\tau))^{s\log(s/\tau)+s}$ possibilities overall. Again we get $\tau^\star = \tilde{O}(\epsilon^2/s^4)$, which gives part (5) of Theorem 34.

Boolean Circuits. An even broader representation scheme is that of *Boolean circuits.* A Boolean circuit of size s is a rooted DAG with s internal nodes, each of which is labeled with an AND, OR or NOT gate. (We consider circuits with arbitrary fan-in, so each AND/OR node is allowed to have arbitrarily many descendants.) Each directed path from the root ends in one of the $n+2$ sink nodes $x_1, \ldots, x_n, 0, 1$.

For $\mathcal{C}$ the class of all size-s Boolean circuits, using Lemma 35 we may take $\mathcal{C}(\tau)$ to be the class of all size-s Boolean circuits in which each gate has at most $\log(s/\tau)$ distinct literals among its inputs, and we have that $\mathcal{C}(\tau)$ is a $(\tau, J(\tau))$-approximator for $\mathcal{C}$ with $J(\tau) \overset{\text{def}}{=} s\log(s/\tau)$. It is easy to see that $|\mathcal{C}(\tau)_{J(\tau)}| \leq 2^{2s^2+4s}$. To completely specify a size-s Boolean circuit, it suffices to specify the following for each of the s gates: its label (three possibilities, AND/OR/NOT) and the set of nodes to which it has outgoing edges (at most 2^{2s+2} possibilities, since this set is a subset of the $s+2$ sink nodes and the s internal nodes).

This results in $\tau^\star = \tilde{O}(\epsilon^2/s^6)$, and consequently Theorem 23 yields part (6) of Theorem 34.

Functions with bounded Fourier degree. For convenience here we take $\mathcal{X} = \{-1,1\}$. Recall that every Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ has a unique Fourier representation, i.e. a representation as a multilinear polynomial with real coefficients: $f(x) = \sum_{S\subseteq[n]} \hat{f}(S) \prod_{i\in S} x_i$. The coefficients $\hat{f}(S)$ are the *Fourier coefficients* of f. The *Fourier degree* of f is the degree of the above polynomial, i.e. the largest value d for which there is a subset $|S| = d$ with $\hat{f}(S) \neq 0$.

Let $\mathcal{C}$ denote the class of all Boolean functions over $\{-1,1\}^n$ with Fourier degree at most d. Nisan and Szegedy [48] have shown that any Boolean function with Fourier degree at most d must have at most $d2^d$ relevant variables. We thus may take $\mathcal{C}(\tau) \overset{\text{def}}{=} \mathcal{C}$ and $J(\tau) \overset{\text{def}}{=} d2^d$. The following lemma gives a bound on $|\mathcal{C}(\tau)_{J(\tau)}|$:

Lemma 36. *For any $d > 0$ we have $|\mathcal{C}(\tau)_{J(\tau)}| < 2^{d^2 \cdot 2^{2d}}$.*

Proof. We first establish the following simple claim:

Claim 37. *Suppose the Fourier degree of $f : \{-1,1\}^n \to \{-1,1\}$ is at most d. Then every nonzero Fourier coefficient of f is an integer multiple of $1/2^{d-1}$.*

Proof. Let us view $f : \{-1,1\}^n \to \{-1,1\}$ as a polynomial with real coefficients. Define the polynomial $p(x_1,\ldots,x_n)$ as

$$p(x_1,\ldots,x_n) = \frac{f(2x_1-1,\ldots,2x_n-1)+1}{2}.$$

The polynomial p maps $\{0,1\}^n$ to $\{0,1\}$. Since f is a multilinear polynomial of degree at most d, so is p. Now it is well known that there is a unique multilinear polynomial that computes any given mapping from $\{0,1\}^n$ to $\{0,1\}$, and it is easy to see that this polynomial has all integer coefficients. Since

$$f(x_1,\ldots,x_n) = 2p\left(\frac{1+x_1}{2},\ldots,\frac{1+x_n}{2}\right) - 1,$$

it follows that every coefficient of f is an integer multiple of $\frac{1}{2^{d-1}}$, and the claim is proved. □

To prove Lemma 36 we must bound the number of distinct Boolean functions with Fourier degree at most d over variables $x_1, \ldots, x_{d2^d}$. First observe that there are at most $D = \sum_{i=0}^{d} \binom{d2^d}{i} \leq (d2^d)^d$ monomials of degree at most d over these variables.

If $f : \{-1,1\}^{d2^d} \to \{-1,1\}$ has Fourier degree at most d, then by Claim 37 every Fourier coefficient is an integer multiple of $1/2^{d-1}$. Since the sum of squares of all Fourier coefficients of any Boolean function is 1, at most 2^{2d-2} of the D monomials can have nonzero Fourier coefficients, and each such coefficient takes one of at most 2^d values. Thus there can be at most

$$\binom{D}{2^{2d-2}} \cdot (2^d)^{2^{2d-2}} \leq (D2^d)^{2^{2d-2}} < 2^{d^2 \cdot 2^{2d}}$$

many Boolean functions over $x_1, \ldots, x_{d2^d}$ that have Fourier degree at most d. □

We thus get that $\tau^\star = \tilde{O}(\epsilon^2/2^{6d})$, and Theorem 23 yields part (7) of Theorem 34. □

3.5.2 Non-Boolean Functions

Theorem 38. *For any s and any $\epsilon > 0$, Algorithm $\mathcal{A}$ yields a testing algorithm for*

1. *s-sparse polynomials over finite field $\mathbb{F}$ using $\tilde{O}((s|\mathbb{F}|)^4/\epsilon^2)$ queries;*
2. *size-s algebraic circuits over finite ring or field $\mathbb{F}$ using $\tilde{O}(s^4 \log^3 |\mathbb{F}|/\epsilon^2)$ queries;*
3. *size-s algebraic computation trees over finite ring or field $\mathbb{F}$ using $\tilde{O}(s^4 \log^3 |\mathbb{F}|/\epsilon^2)$ queries.*

Proof. We describe each class of functions and apply Theorem 23 to prove each part of the theorem.

Sparse Polynomials over Finite Fields. Let $\mathbb{F}$ denote any finite field and let $\mathcal{Y} = \mathbb{F}$. An *s-sparse polynomial over* $\mathbb{F}$ is a multivariate polynomial in variables $x_1, \ldots, x_n$ with at most s nonzero coefficients.

Let us say that the *length* of a monomial is the number of distinct variables that occur in it (so for example the monomial $3x_1^2x_2^4$ has length two). We have the following:

Lemma 39. *Any s-sparse polynomial over $\mathbb{F}$ is τ-close to an s-sparse polynomial over $\mathbb{F}$ in which each monomial has length at most $|\mathbb{F}|\ln(s/\tau)$.*

Proof. If a monomial has length ℓ greater than $|\mathbb{F}|\ln(s/\tau)$, then it can be τ/s-approximated by 0 (for a uniform random $x \in \mathbb{F}^n$, the probability that the monomial is not 0 under x is $(1 - 1/|\mathbb{F}|)^\ell$). Performing this approximation for all s terms yields a τ-approximator for the polynomial. □

For $\mathcal{C}$ = the class of all s-sparse polynomials in n variables over finite field $\mathbb{F}$, we have that the class $\mathcal{C}(\tau)$ of all s-sparse polynomials over finite field $\mathbb{F}$ with all monomials of length at most $|\mathbb{F}|\ln(s/\tau)$ is a $(\tau, J(\tau))$-approximator with $J(\tau) = s|\mathbb{F}|\ln(s/\tau)$. The following counting argument shows that

$$|\mathcal{C}(\tau)_{J(\tau)}| \leq (s|\mathbb{F}|^3\ln(s/\tau))^{s|\mathbb{F}|\ln(s/\tau)}.$$

Consider a single monomial M. To specify M we must specify a coefficient in $\mathbb{F}$, a subset of at most ℓ of the $J(\tau)$ possible variables that have nonzero degree (at most $J(\tau)^\ell$ possibilities), and for each of these variables we must specify its degree, which we may assume is at most $|\mathbb{F}| - 1$ since $\alpha^{|\mathbb{F}|} = \alpha$ for every α in finite field $\mathbb{F}$. Thus there are at most $|\mathbb{F}|(J(\tau)|\mathbb{F}|)^\ell$ possibilities for each monomial, and consequently at most $|\mathbb{F}|^s(J(\tau)|\mathbb{F}|)^{s\ell} = |\mathbb{F}|^s(s|\mathbb{F}|^2\ln(s/\tau))^{s|\mathbb{F}|\ln(s/\tau)} \leq (s|\mathbb{F}|^3\ln(s/\tau))^{s|\mathbb{F}|\ln(s/\tau)}$ possible polynomials overall.

Setting $\tau^\star = \tilde{O}(\epsilon^2/(s|\mathbb{F}|)^4)$ and applying Theorem 23 yields part (1) of Theorem 38.

Algebraic Circuits. Let $\mathbb{F}$ denote any finite ring or field and let $\mathcal{Y} = \mathbb{F}$. A size-s *algebraic circuit* (or *straight line program*) over $\mathbb{F}^n$ is a rooted directed acyclic graph with s internal nodes (each with two inputs and one output) and $n + k$ leaf nodes for some $k \geq 0$ (each with no inputs and arbitrarily many outputs). The first n leaf nodes are labeled with the input variables $x_1, \ldots, x_n$, and the last k leaf nodes are labeled with arbitrary constants α_i from $\mathbb{F}$. Each internal node is labeled with a gate from $\{+, \times, -\}$ and computes the sum, product, or difference of its two input values (if $\mathbb{F}$ is a field we allow division gates as well).

Let $\mathcal{C}$ denote the class of all Boolean functions computed by algebraic circuits of size at most s over variables $x_1, \ldots, x_n$. (Here we analyze the simpler case of circuits with $+$, $\times$, $-$ gates; our analysis can easily be extended to handle division gates as well.) Any size-s algebraic circuit depends on at most $2s$ variables. We may thus take $\mathcal{C}(\tau) \stackrel{\text{def}}{=} \mathcal{C}$ and we trivially have that $\mathcal{C}(\tau)$ is a $(\tau, J(\tau))$-approximator for $\mathcal{C}$ with $J(\tau) \stackrel{\text{def}}{=} 2s$. Now we show that $|\mathcal{C}(\tau)_{J(\tau)}| \leq (75|\mathbb{F}|^2 s^2)^s$.

A size s algebraic circuit can read at most $2s$ leaves as each internal node has two inputs. Thus it can read at most $2s$ constant leaves, and at most $2s$ input leaves. To completely specify a size-s algebraic circuit, it suffices to specify the $2s$ constant leaf nodes and the following for each of the s gates: its label (at most three possibilities) and the two nodes to which it has outgoing edges (at most $(5s)^2$ possibilities, since it can hit two of the at most $4s$ leaves and the s internal nodes). Thus there are at most $|\mathbb{F}|^{2s}(75s^2)^s$ different algebraic circuits.

Equation 3.1 in Theorem 23 is satisfied for small τ's, but we do not care how large the optimum $\tau^\star$ is as $J(\tau)$ does not depend on τ. Eventually, Theorem 23 yields part (2) of Theorem 38.

Algebraic Computation Trees. Let $\mathbb{F}$ denote any finite ring or field and let $\mathcal{Y} = \mathbb{F}$. A size-s *algebraic computation tree* over input variables $x_1, \ldots, x_n$ is a rooted binary tree with the following structure. There are s leaves, each describes an output value which is either a constant, an input variable, or one of the variables computed in the ancestors of the leaf. Each internal node has two children and is labeled with y_v, where $y_v = y_u \circ y_w$ and y_u, y_w are either inputs, the labels of ancestor nodes, or constants, and the operator $\circ$ is one of $\{+, -, \times, \div\}$ (the last one only if $\mathbb{F}$ is a field). An input that reaches such a node branches left if $y_v = 0$ and branches right if $y_v \neq 0$.

Let $\mathcal{C}$ denote the class of all functions computed by algebraic computation trees of size at most s over $x_1, \ldots, x_n$. Any size-s algebraic computation tree depends on at most $3s$ variables. So similar to algebraic circuits, we can take $\mathcal{C}(\tau) \stackrel{\text{def}}{=} \mathcal{C}$ and $J(\tau) \stackrel{\text{def}}{=} 3s$. Now we show that $|\mathcal{C}(\tau)_{J(\tau)}| \leq 16^s(|\mathbb{F}| + 4s)^{3s}$.

As in the boolean case, the number of s-leaf rooted binary trees in which each internal node has precisely two children is at most 4^s. A tree has $s - 1$ internal nodes and s leaves.

For each of these possible tree topologies there are at most $4(|\mathbb{F}| + 4s)^2$ ways to label the $s-1$ internal nodes (with one of 4 operations on two constants, variables or ancestor nodes). Finally, there are at most $(|\mathbb{F}| + 4s)^s$ ways to choose the leaf labels. So the total number of decision trees of size s over variables $x_1, \ldots, x_{3s}$ is at most $4^s \cdot (4(|\mathbb{F}| + 4s)^2)^{s-1} \cdot (|\mathbb{F}| + 4s)^s \leq 16^s(|\mathbb{F}| + 4s)^{3s}$.

As before we do not care what the optimal $\tau^\star$ in Theorem 23 is. Finally, we obtain query complexity $\tilde{O}(s^4 \log^3 |\mathbb{F}|/\epsilon^2)$ by Theorem 23, that is, we obtain part (3) of Theorem 38. □

3.6 Lower bounds for testing sparse polynomials

One consequence of Theorem 23 is a poly(s/ϵ)-query algorithm for testing s-sparse polynomials over finite fields of fixed size (independent of n). In this section we present a polynomial lower bound for non-adaptive algorithms for this testing problem. Our main result in this section is the following theorem:

Theorem 40. *Let $\mathbb{F}$ be any fixed finite field, i.e. $|\mathbb{F}| = O(1)$ independent of n. There exists a fixed constant $\epsilon > 0$ (depending on $|\mathbb{F}|$) such that any* non-adaptive *ϵ-testing algorithm for the class of s-sparse polynomials over $\mathbb{F}^n$ must make $\tilde{\Omega}(\sqrt{s})$ queries.*

To prove Theorem 40 we use Yao's principle [63] in (what has become) a standard way for proving lower bounds in property testing (e.g. see [23]). We present two distributions D_{YES} and D_{NO}, the former on inputs satisfying the property (i.e. s-sparse polynomials from $\mathbb{F}^n$ to $\mathbb{F}$), the latter on inputs that are ϵ-far from satisfying it, and show that any deterministic (non-adaptive) algorithm making "few" queries cannot distinguish between a random draw from D_{YES} versus a random draw from D_{NO}. By standard arguments (see for example Lemma 8.1 in [23]), it suffices to argue that for any query set $Q \subset \mathbb{F}^n$ of cardinality $q = \tilde{O}(\sqrt{s})$ the induced distributions on $\mathbb{F}^q$ (obtained by restricting the randomly chosen functions to these q points) have statistical distance less than $1/3$.

Throughout this section we write $\mathbb{F}$ to denote the finite field with P elements, where $P = p^k$ is a prime power. We consider the following two distributions over functions mapping $\mathbb{F}^n$ to $\mathbb{F}$:

- A draw from D_{YES} is obtained as follows: independently and uniformly (with repetitions) draw s variables $x_{i_1}, \dots, x_{i_s}$ from $x_1, \dots, x_n$, and let $f(x) = x_{i_1} + \dots + x_{i_s}$.
- A draw from D_{NO} is obtained as follows: independently and uniformly (with repetitions) draw $s+p$ variables $x_{i_1}, \dots, x_{i_{s+p}}$ from $x_1, \dots, x_n$, and let $f(x) = x_{i_1} + \dots + x_{i_{s+p}}$.

It is clear that every draw from D_{YES} is an s-sparse polynomial over $\mathbb{F}$, and that for any $n = \omega((s+p)^2)$ almost all the probability mass of D_{NO} is on functions with $s+p$ distinct nonzero coefficients.

Theorem 40 then follows from the following two results:

Theorem 41. *Let A be any non-adaptive algorithm which is given black-box access to a function $f : \mathbb{F}^n \to \mathbb{F}$ and outputs either "yes" or "no." Then we have*

$$\left| \Pr_{f \in D_{\text{YES}}}[A^f \textit{ outputs "yes"}] - \Pr_{f \in D_{\text{NO}}}[A^f \textit{ outputs "yes"}] \right| \leq \frac{1}{3}$$

unless A makes $\tilde{\Omega}(\sqrt{s})$ queries to the black-box function f.

Theorem 42. *Let*

$$\Phi(P) \stackrel{\text{def}}{=} 1/(P^{P^2+P^{10P^2+26}}).$$

Fix any $s \leq n-1$. Let g be an s-sparse polynomial in $\mathbb{F}[x_1, \dots, x_n]$. Then g is $\Phi(P)$-far from every affine function over $\mathbb{F}$ in which $s+1$ or more variables have nonzero coefficients, i.e. every function of the form

$$a_1 x_1 + \dots + a_{s+r} x_{s+r} + b \tag{3.3}$$

where $0 \neq a_i \in \mathbb{F}$, $b \in \mathbb{F}$, and $r \geq 1$.

Theorem 41 shows that any non-adaptive algorithm that can successfully distinguish a random linear form $x_{i_1} + \dots + x_{i_s}$ from a random linear form $x_{i_1} + \dots + x_{i_{s+p}}$ must make $\tilde{\Omega}(\sqrt{s})$ queries; this is a technical generalization of a similar result for $\mathbb{F}_2$ in [26]. Theorem 42 establishes that every function $x_{i_1} + \dots + x_{i_{s+p}}$ is far from every s-sparse polynomial over $\mathbb{F}$. Together these results imply that any testing algorithm for s-sparse $\mathbb{F}$ polynomials must be able to distinguish length-s linear forms from length-$(s+p)$ linear forms, and must make $\tilde{\Omega}(\sqrt{s})$ queries. We prove these theorems in the following subsections.

We note that it is conceivable that a stronger version of Theorem 42 might be true in which $\Phi(P)$ is replaced by an absolute constant such as $1/3$; however Theorem 42 as stated suffices to give our desired lower bound.

3.6.1 s-sparse linear forms are indistinguishable from $(s+p)$-sparse linear forms.

First, let us recall the definition of statistical distance:

Definition 43 (statistical distance). *Let S be a finite set and $\mathbb{P}, \mathbb{Q}$ be probability measures on $(S, 2^S)$. The* statistical distance *between $\mathbb{P}$ and $\mathbb{Q}$ is defined by* $\|\mathbb{P}-\mathbb{Q}\| \stackrel{\text{def}}{=} \max_{A\subseteq S} |\mathbb{P}(A)-\mathbb{Q}(A)|$.

The following fact is an immediate consequence of the definition:

Fact 44. $\|\mathbb{P}-\mathbb{Q}\| \equiv \frac{1}{2}\sum_{x\in S} |\mathbb{P}(x)-\mathbb{Q}(x)| \equiv \sum_{x\in S} \left(\mathbb{P}(x)-\mathbb{Q}(x)\right)^{\pm}$.

We now explain how Theorem 41 can be reduced to a convergence-type result about random walks on the group $\mathbb{Z}_p^q$ (Theorem 45 below). We remark that the argument given here is an immediate generalization of the corresponding argument in Section 6 of [26]. Our main technical contribution is in fact the proof of Theorem 45.

Recall that a non-adaptive testing algorithm queries a fixed subset $\mathcal{Q}$ of the domain $\mathbb{F}^n$, where $|\mathbb{F}| = P = p^k$ is a prime power. To prove Theorem 41, it suffices to argue that for any query set $\mathcal{Q} \subset \mathbb{F}^n$ of cardinality $q = |\mathcal{Q}| = \tilde{O}(\sqrt{s})$ the induced distributions on $\mathbb{F}^q$ (obtained by restricting the randomly chosen functions to these q points) have a statistical distance less than $1/3$.

Let us now describe the distributions induced by D_{YES} and D_{NO} on $\mathbb{F}^q$. Let $r_1, \ldots, r_q \in \mathbb{F}^n$ be the queries, and let M be a $q \times n$ matrix with rows $r_1, \ldots, r_q$. To choose an element $x \in \mathbb{F}^q$ according to the first (induced) distribution, we choose at random (with repetitions)

s columns of M and sum them up. This gives us an element of $\mathbb{F}^q$. The same holds for the second distribution, the only difference being that we choose $s + p$ columns.

For $x \in \mathbb{F}^q \cong \mathbb{Z}_p^{kq}$, let $\mathbb{P}(x)$ be the probability of choosing x when we pick a column of M at random. Consider a random walk on the group $\mathbb{Z}_p^{kq}$, starting at the identity element, in which at every step we choose an element of the group according to $\mathbb{P}$ and add it to the current location. Let $\mathbb{P}_t$ be the distribution of this walk after t steps. Observe that $\mathbb{P}_s$ and $\mathbb{P}_{s+p}$ are exactly the distributions induced by D_{YES} and D_{NO}. We want to show that for s sufficiently large compared to q, the distributions $\mathbb{P}_s$ and $\mathbb{P}_{s+p}$ are close with respect to the statistical distance. To do this, it suffices to prove the following theorem:

Theorem 45. *Let r be a prime, $q \in \mathbb{N}^*$ and $\mathbb{P}$ be a probability measure on $\mathbb{Z}_r^q$. Consider the random walk X on $\mathbb{Z}_r^q$ with step distribution $\mathbb{P}$. Let $\mathbb{P}_t$ be the distribution of X at step t. There exists an absolute constant $C > 0$ such that for every $0 < \delta \leq 1/2$, if $t \geq C\frac{\log 1/\delta}{\delta} \cdot r^4 \log r \cdot q^2 \log^2(q+1)$ then $\|\mathbb{P}_t - \mathbb{P}_{t+r}\| \leq \delta$.*

Indeed, since the underlying additive group of the field $\mathbb{F}$ is $\mathbb{Z}_p^k$, by applying the above theorem for $r = p$ and $q' = kq$ the result follows. We prove Theorem 45 in the following subsection.

3.6.2 Periodicity in random walks

To prove Theorem 45, we start with some basic definitions and facts about random walks on (finite) groups. For a detailed treatment of the subject, see [16] and references therein. For basic facts about Fourier Analysis on finite groups, see [58, 60].

Let $(G, +)$ be a finite group. For any probability measures $\mathbb{P}$, $\mathbb{Q}$ on G, the convolution $(\mathbb{P} * \mathbb{Q})$ of $\mathbb{P}$ and $\mathbb{Q}$ is the probability measure on G defined by:

$$(\mathbb{P} * \mathbb{Q})(y) = \sum_{x \in G} \mathbb{P}(x)\mathbb{Q}(x + y)$$

Let $\mathbb{P}_1, \ldots, \mathbb{P}_n$ be probability measures on G. The *convolution product* of the $\mathbb{P}_i$'s, is defined as follows:

$$\{*\prod\}_{i=j}^{j}\mathbb{P}_i \stackrel{\text{def}}{=} \mathbb{P}_j$$
$$\{*\prod\}_{i=j}^{n}\mathbb{P}_i \stackrel{\text{def}}{=} \mathbb{P}_j * \{*\prod\}_{i=j+1}^{n}\mathbb{P}_i, \quad \text{if } n > j$$

Similarly, $\mathbb{P}^{*n}$, *the n-fold convolution product of $\mathbb{P}$ with itself* is defined by: $\mathbb{P}^{*1} \stackrel{\text{def}}{=} \mathbb{P}$ and $\mathbb{P}^{*n} \stackrel{\text{def}}{=} \mathbb{P}^{*(n-1)} * \mathbb{P}$, if $n > 1$.

A distribution (probability measure) $\mathbb{P}$ on G induces a random walk on G as follows: Denoting by X_n its position at time n, the walk starts at the identity element of G ($n = 0$) and at each step selects an element $\xi_n \in G$ according to $\mathbb{P}$ and goes to $X_{n+1} = \xi_n + X_n$. Denote by $\mathbb{P}_n$ the distribution of X_n. Since X_n is the sum of n independent random variables with distribution $\mathbb{P}$, it follows that $\mathbb{P}_n = \mathbb{P}^{*n}$.

We will be interested in such random walks on *finite abelian groups* and in particular on the group $(\mathbb{Z}_r^q, +)$, where $+$ denotes componentwise addition modulo r. We remark that for abelian groups, the convolution operation is commutative. In fact, commutativity is crucially exploited in the proof of the theorem.

For a function $f : \mathbb{Z}_r^q \to \mathbb{C}$, we define its Fourier transform $\widehat{f} : \mathbb{Z}_r^q \to \mathbb{C}$ by

$$\widehat{f}(x) \stackrel{\text{def}}{=} \frac{1}{r^q} \sum_{y \in \mathbb{Z}_r^q} f(y)(\omega_r)^{\langle x,y \rangle}$$

where $\omega_r \stackrel{\text{def}}{=} e^{2\pi i/r}$ and for $x, y \in \mathbb{Z}_r^q$ we denote $\langle x, y \rangle \stackrel{\text{def}}{=} (\sum_{i=1}^q x_i y_i) \mod r$.

Fact 46. *Let $\mathbb{P}$, $\mathbb{Q}$ be probability measures on $\mathbb{Z}_r^q$. Then, $\widehat{\mathbb{P} * \mathbb{Q}}(y) = r^q \cdot \widehat{\mathbb{P}}(y) \cdot \widehat{\mathbb{Q}}(y)$, $y \in \mathbb{Z}_r^q$.*

For $p \geq 1$ and $f : \mathbb{Z}_r^q \to \mathbb{C}$, the l_p norm of f is defined by $\|f\|_p \stackrel{\text{def}}{=} \{\mathbb{E}_{x \in \mathbb{Z}_r^q}[|f(x)|^p]\}^{1/p}$. The inner product of $f, g : \mathbb{Z}_r^q \to \mathbb{C}$ is defined as: $\langle f, g \rangle \stackrel{\text{def}}{=} \mathbb{E}_{x \in \mathbb{Z}_r^q}[f(x)\overline{g(x)}]$.

Fact 47 (Parseval's identity). *Let $f : \mathbb{Z}_r^q \to \mathbb{C}$. Then, $\|f\|_2^2 \equiv \langle f, f \rangle = \sum_{x \in \mathbb{Z}_r^q} |\hat{f}|^2(x)$.*

Proof of Theorem 45.

The special case of this theorem for $r = 2$ was proved by Fischer *et al.* [26]. Our proof is a technical generalization of their proof. Moreover, our proof has the same overall structure as the one in [26]. However, one needs to overcome several difficulties in order to achieve this generalization.

We first give a high-level overview of the overall strategy. Any given $x \in (\mathbb{Z}_r^q)^*$ partitions the space into r non-empty subspaces $V_i^x = \{y \in \mathbb{Z}_r^q : \langle y, x \rangle = i\}$ for $i = 0, 1, \ldots, r-1$. We say that an $x \in (\mathbb{Z}_r^q)^*$ is *degenerate* if there exists some i whose probability measure $\mathbb{P}(V_i^x)$ is "large". (We note that the definition of degeneracy in the proof of [26] is quite specialized for the case $r = 2$. They define a direction to be degenerate if one of the subspaces V_0^x, V_1^x has "small" probability. Our generalized notion - that essentially reduces to their definition for $r = 2$ - is the conceptually correct notion and makes the overall approach work.)

We consider two cases: If all the Fourier coefficients of $\mathbb{P}$ are not "very large" (in absolute value), then we can show by standard arguments (see e.g. [16]) that the walk is close to stationarity after the desired number of steps. Indeed, in such a case the walk converges rapidly to the uniform distribution (in the "classical" sense, i.e. $||\mathbb{P}_t - \mathcal{U}|| \to 0$ as t approaches infinity).

If, on the other hand, there exists a "very large" Fourier coefficient of $\mathbb{P}$, then we argue that there must also exist a degenerate direction (this is rather non-trivial) and we use induction on the dimension q. It should be noted that in such a case the walk *may not converge at all in the classical sense.* (An extreme such case would be, for example, if $\mathbb{P}$ was concentrated on one element of the group.)

Remark: It seems that our proof can be easily modified to hold for any *finite abelian group*. (We remind the reader that any such group can be uniquely expressed as the direct sum of cyclic groups.) Perhaps, such a result would be of independent interest. We have not attempted to do so here, since it is beyond the scope of our lower bound. Note that, with the exception of the inductive argument, all the other components of our proof work (in this generalized setting) without any changes. It is very likely that a more complicated induction would do the trick.

Now let us proceed with the actual proof. We make essential use of two lemmata. The first one is a simple combinatorial fact that is used several times in the course of the proof:

Lemma 48. *Let n be a positive integer greater than 1 and $\epsilon \in (0, 1/2]$ be a constant. Consider a complex number $\mathrm{v} \in \mathbb{C}$ expressible as a (non-trivial) convex combination of the n-th roots of unity all of whose coefficients are at most $1-\epsilon$. Then, we have $|\mathrm{v}| \leq 1-\epsilon/2n^2$.*

Proof. We can write $\mathrm{v} = \sum_{j=0}^{n-1} v_j \omega_n^j$, with $\omega_n = e^{2\pi i/n}$, $v_j \geq 0$, $\sum_{j=0}^{n-1} v_j = 1$ and $\max_j v_j \leq 1-\epsilon$. For the proof it will be helpful to view the ω_n^j's as unit vectors in the complex plane (the angle between two "adjacent" such vectors being $\theta_n = 2\pi/n$).

By assumption, it is clear that at least two distinct coefficients must be non-zero. We claim that the length of the vector v is maximized (over all possible "legal" choices of the v_j's) when exactly two of the coefficients are non-zero, namely two coefficients corresponding to consecutive n-th roots of unity.

This is quite obvious, but we give an intuitive argument. We can assume that $n \geq 5$; otherwise the claim is straightforward. Consider the unit vector e (this vector corresponds to one of the ω_n^j's) whose coefficient v_{e} in v is maximum. We want to "distribute" the remaining "mass" $1 - v_{\mathrm{e}}$ to the other coordinates (n-th roots) so as to maximize the length $|\mathrm{v}|$. First, observe that vectors whose angle with e is at least $\pi/2$ do not help; so we can assume the corresponding coefficients are zero. Now consider the set of vectors "above" e (whose angle with e is less than $\pi/2$). We can assume that their "mass" (i.e. sum of coefficients) is concentrated on the unit vector e_a adjacent to e (whose angle with e is minimum); this maximizes their total contribution to the length of the sum. By a symmetric argument, the same holds for the set of vectors "below" e (denote by e_b the corresponding adjacent vector). Finally, it is easy to see that in order to maximize the total contribution of e_a and e_b to the length of the sum, one of them must have zero weight (given that their total mass is "fixed").

Now let us proceed with the proof of the upper bound. By symmetry, it is no loss of generality to assume that $v_0, v_1 > 0$ with $v_0 \geq v_1$. The claim now follows from the following sequence of elementary calculations:

$$\begin{aligned}
|\mathbf{v}|^2 = v_0^2 + v_1^2 + 2v_0v_1\cos\theta_n &= 1 - 2v_0v_1\big(1-\cos\theta_n\big)\\
&= 1 - 2v_0\big(1-v_0\big)\big(1-\cos(2\pi/n)\big)\\
&\le 1 - 2\epsilon(1-\epsilon)\big(1-\cos(2\pi/n)\big)\\
&\le 1 - \epsilon\big(1-\cos(2\pi/n)\big)\\
&\le 1 - \epsilon/n^2
\end{aligned}$$

The last inequality above follows by observing that $\cos(2\pi/n) \le 1 - 1/n^2$, $n \ge 2$. The elementary inequality $\sqrt{1-x} \le 1 - x/2$ completes the argument. □

Our second lemma is an analytical tool giving a (relatively sharp) upper bound on the statistical distance between two distributions. It should be noted that this result is a variant of the "upper bound lemma" [16], which has been used in numerous other random walk problems.

Lemma 49 (upper bound lemma, [16]). *In the context of Theorem 45, for any* $t \ge 0$, *we have:*

$$\|\mathbb{P}_t - \mathbb{P}_{t+r}\|^2 \le r^q \cdot \sum_{x\in(\mathbb{Z}_r^q)^*} |\alpha(x)|^{2t}.$$

Proof. We have:

$$\begin{aligned}
\|\mathbb{P}_t - \mathbb{P}_{t+r}\|^2 &= (r^{2q}/4)\cdot\|\mathbb{P}_t - \mathbb{P}_{t+r}\|_1^2 && (3.4)\\
&\le (r^{3q}/4)\cdot\|\mathbb{P}_t - \mathbb{P}_{t+r}\|_2^2 && (3.5)\\
&= (r^{3q}/4)\cdot\sum_{x\in\mathbb{Z}_r^q}\big|\widehat{\mathbb{P}}_t(x) - \widehat{\mathbb{P}_{t+r}}(x)\big|^2 && (3.6)\\
&= (r^{3q}/4)\cdot\sum_{x\in(\mathbb{Z}_r^q)^*}\big|r^{q(t-1)}\big(\widehat{\mathbb{P}}(x)\big)^t - r^{q(t+r-1)}\big(\widehat{\mathbb{P}}(x)\big)^{t+r}\big|^2 && (3.7)\\
&= (r^q/4)\sum_{x\in(\mathbb{Z}_r^q)^*}\big|\alpha^t(x) - \alpha^{t+r}(x)\big|^2 && (3.8)\\
&\le r^q\sum_{x\in(\mathbb{Z}_r^q)^*}|\alpha(x)|^{2t} && (3.9)
\end{aligned}$$

Step (3.4) follows directly from the definitions of the statistical distance and the l_1 norm. Step (3.5) easily follows from the Cauchy-Schwarz inequality and step (3.6) from the Parseval identity. For Step (3.7) notice that $\widehat{\mathbb{P}}_t(y) = r^{q(t-1)}\left(\widehat{\mathbb{P}}(y)\right)^t$ and $\widehat{\mathbb{P}}(\mathbf{0}) = 1/r^q$. Step (3.8) is immediate by the definition of α and Step (3.9) follows from the triangle inequality. □

Let $X_t \in \mathbb{Z}_r^q$ be the position of the random walk at time t and $\mathbb{P}_t$ its distribution. By assumption $X_0 = 0$. As previously mentioned, $\mathbb{P}_t = \mathbb{P}^{*t}$. It is easy to show that the statistical distance $||\mathbb{P}_t - \mathbb{P}_{t+r}||$ is monotone non-increasing in t; we are interested in the first time $t = t(r, q)$ for which $\mathbb{P}_t$ and $\mathbb{P}_{t+r}$ are δ-close.

Notation. For $q \in \mathbb{N}$, define $b(q) \stackrel{\text{def}}{=} q^2 \log^2(q+1)$, $d(r) \stackrel{\text{def}}{=} r^4 \log r$, $S_q \stackrel{\text{def}}{=} \sum_{j=1}^{q} j/b(j)$, $S \stackrel{\text{def}}{=} \lim_{j\to\infty} S_j$ and $t_q \stackrel{\text{def}}{=} C\frac{\log(1/\delta)}{\delta} d(r)b(q)$.

Throughout the proof, we assume for simplicity that t_q is an integer. If $\mathbb{P}$ is a probability measure on $\mathbb{Z}_r^q$ and $\widehat{\mathbb{P}}$ is its Fourier transform, we denote $\alpha(x) \stackrel{\text{def}}{=} r^q\widehat{\mathbb{P}}(x)$. A word concerning absolute constants. The letter C will always denote an absolute constant, but as is customary the value of C need not be the same in all its occurrences. Also note that S is an absolute constant, so C can depend on S.

Theorem 45 follows from the following claim:

Claim 50. *There exists a universal constant $C > 0$ such that for any $0 < \delta \leq 1/2$, any $t \geq t_q$ and any probability measure $\mathbb{P}$ on $\mathbb{Z}_r^q$ it holds $||\mathbb{P}_t - \mathbb{P}_{t+r}|| \leq \frac{\delta}{S} \cdot S_q < \delta$.*

We will prove the claim by induction on q.

Base case ($q = 1$). Given an arbitrary probability measure $\mathbb{P}$ on the discrete circle $\mathbb{Z}_n$, $n \in \mathbb{N}^*$, we will show that, for all $t \geq t_1 \equiv C\frac{\log 1/\delta}{\delta} \cdot n^4 \log n$, it holds $||\mathbb{P}_t - \mathbb{P}_{t+n}|| \leq \frac{\delta}{S}$.

Set $\epsilon_0 := \frac{\delta}{Sn}$ and consider the following two cases below:

Case I (There exists a $k \in \mathbb{Z}_n$ such that $\mathbb{P}(k) \geq 1 - \epsilon_0$.) In this case, we claim that for all $t \in \mathbb{N}^*$ it holds $||\mathbb{P}_t - \mathbb{P}_{t+n}|| \leq n\epsilon_0 = \delta/S$. (In fact, this holds independently of the value of the time t.) This should be intuitively obvious, but we give an argument.

Recall that the statistical distance $\|\mathbb{P}_t - \mathbb{P}_{t+c}\|$ is a monotone non-increasing function of t for any constant c. Hence, $\|\mathbb{P}_t - \mathbb{P}_{t+n}\| \leq \|\mathbb{P} - \mathbb{P}_{n+1}\|$ and it suffices to argue that $\|\mathbb{P} - \mathbb{P}_{n+1}\| \leq n\epsilon_0$. The crucial fact is that for all $i \in \mathbb{Z}_n$ we have $\mathbb{P}_{n+1}(i) \geq (1-n\epsilon_0)\cdot\mathbb{P}(i)$. This directly implies that $\|\mathbb{P}-\mathbb{P}_{n+1}\| \equiv \sum_{i\in\mathbb{Z}_n}(\mathbb{P}(i)-\mathbb{P}_{n+1}(i))^+ \leq n\epsilon_0\cdot\sum_{\{i:\mathbb{P}(i)>\mathbb{P}_{n+1}(i)\}}\mathbb{P}(i) \leq n\epsilon_0 \cdot \sum_{i\in\mathbb{Z}_n}\mathbb{P}(i) = n\epsilon_0$.

To see that the aforementioned fact is true, observe that for any $i \in \mathbb{Z}_n$, conditioned on the walk being at position i at time $t = 1$, with probability at least $(1-\epsilon)^n$ each of the next n steps is k, so with probability at least $(1-\epsilon_0)^n \geq 1 - n\epsilon_0$ the walk is at position i again at time $t = n+1$.

Case II (For all $k \in \mathbb{Z}_n$ it holds $\mathbb{P}(k) \leq 1 - \epsilon_0$.) Note that, for $k \in \mathbb{Z}_n$, we can write $\alpha(k) = \sum_{l=0}^{n-1}\mathbb{P}(l)\cdot\omega_n^{k\cdot l}$, where $\omega_n = e^{2\pi i/n}$. Since $\mathbb{P}$ is a probability measure, it follows that $\alpha(0) = 1$. Now observe that for $k \in \mathbb{Z}_n^*$, $\alpha(k)$ is a convex combination of n-th roots of unity with coefficients at most $1 - \epsilon_0$. Hence, an application of Lemma 48 gives the following corollary:

Corollary 51. *For all $k \in \mathbb{Z}_n^*$, it holds $|\alpha(k)| \leq 1 - \frac{\delta}{2Sn^3}$.*

We have now set ourselves up for an application of Lemma 49. For any $t \in \mathbb{N}$ with $t \geq t_1$, we thus get:

$$\begin{aligned}
\|\mathbb{P}_t - \mathbb{P}_{t+n}\|^2 &\leq n\sum_{i\in\mathbb{Z}_n^*}|\alpha(i)|^{2t} \\
&\leq n^2\left(1-\frac{\delta}{2Sn^3}\right)^{2t} \leq n^2\left(1-\frac{\delta}{2Sn^3}\right)^{2t_1} \\
&\leq n^2\left(e^{-\frac{\delta}{2Sn^3}}\right)^{2t_1} = n^2 e^{-Cn\log n\log(1/\delta)/S}
\end{aligned}$$

where we used the elementary inequality $1 - x \leq e^{-x}$, for $x \in [0,1]$. For large enough C, we have $\|\mathbb{P}_t - \mathbb{P}_{t+n}\|^2 \leq (\delta/S)^2$ and the base case is proved.

Induction Step: Assume that the claim holds for $q-1$, i.e. that for any $t \geq t_{q-1}$ and any probability measure $\mathbb{P}$ on $\mathbb{Z}_r^{q-1}$ it holds $\|\mathbb{P}_t - \mathbb{P}_{t+r}\| \leq \frac{\delta}{S}\cdot S_{q-1}$. We will prove that the claim also holds for q.

For $x \in (\mathbb{Z}_r^q)^*$ and $i = 0, 1, \ldots, r-1$ define $V_i^x \overset{\text{def}}{=} \{y \in \mathbb{Z}_r^q : \langle y, x \rangle = i\}$. At this point we are ready to formally define the notion of degenerate direction:

Definition 52. *We say that* $x \in (\mathbb{Z}_r^q)^*$ *is a* degenerate direction *if there exists an* $i \in \{0, 1, \ldots, r-1\}$ *such that* $\mathbb{P}(V_i^x) \geq 1 - \frac{2\delta q}{\sqrt{C} r^2 b(q)}$.

We distinguish the following two cases below:

Case I (For all $x \in (\mathbb{Z}_r^q)^*$ it holds $|\alpha(x)| < 1 - \frac{\delta q}{\sqrt{C} r^4 b(q)}$.) Note that, since $\mathbb{P}$ is a probability distribution, we have $\alpha(\mathbf{0}) = 1$. Now, for $t \geq t_q$ Lemma 49 yields:

$$\begin{aligned}
\|\mathbb{P}_t - \mathbb{P}_{t+r}\|^2 &\leq r^q \sum_{x \in (\mathbb{Z}_r^q)^*} |\alpha(x)|^{2t} \\
&\leq r^{2q}\Big(1 - \frac{\delta q}{\sqrt{C} r^4 b(q)}\Big)^{2t} \leq r^{2q}\Big(1 - \frac{\delta q}{\sqrt{C} r^4 b(q)}\Big)^{2t_q} \\
&\leq r^{2q}\big(e^{-\frac{\delta q}{\sqrt{C} r^4 b(q)}}\big)^{2t_q} = r^{2q} e^{-2q \log r \sqrt{C} \log 1/\delta}
\end{aligned}$$

Similarly, if C is large enough, we have $\|\mathbb{P}_t - \mathbb{P}_{t+r}\| \leq \delta/S \leq \frac{\delta}{S} \cdot S_q$.

Case II (There exists some $x_0 \in (\mathbb{Z}_r^q)^*$ such that $|\alpha(x_0)| \geq 1 - \frac{\delta q}{\sqrt{C} r^4 b(q)}$.)

Since r is a prime, we may assume without loss of generality that $x_0 = \varepsilon_1 = (1 \mathbf{0}_{q-1})$. Then, for $i = 0, 1, \ldots, r-1$, we have $V_i \equiv V_i^{x_0} = \{y \equiv (y_1, y_2, \ldots, y_q) \in \mathbb{Z}_r^q : y_1 = i\}$; note that each V_i is isomorphic to $\mathbb{Z}_r^{q-1}$.

Now observe that we can write $\alpha(x_0) = \sum_{j=0}^{r-1} \mathbb{P}(V_j) \omega_r^j$ with $\sum_j \mathbb{P}(V_j) = 1, \mathbb{P}(V_j) \geq 0$. That is, $\alpha(x_0)$ is a convex combination of r-th roots of unity whose absolute value is at least $1 - \epsilon'/2r^2$, where $\epsilon' := \frac{2\delta q}{\sqrt{C} r^2 b(q)}$. Thus, (the contrapositive of) Lemma 48 implies that there must exist some $j \in \{0, 1, \ldots, r-1\}$ with $\mathbb{P}(V_j) \geq 1 - \frac{2\delta q}{\sqrt{C} r^2 b(q)}$ (i.e. x_0 is degenerate). Clearly, it is no loss of generality to assume that $j = 0$, i.e. $\mathbb{P}(V_0) \geq 1 - \frac{2\delta q}{\sqrt{C} r^2 b(q)}$.

For $i = 0, 1, \ldots, r-1$ and $j = t_q, t_q + r$, consider the conditional probability measures $\mathbb{P}_j^i = (\mathbb{P}_j | V_i)$. All the $2r$ distributions obtained in this manner can be viewed as distributions on $\mathbb{Z}_r^{q-1}$. By the law of total probability, we can write: $\mathbb{P}_j = \sum_{i=0}^{r-1} \mathbb{P}_j(V_i) \cdot \mathbb{P}_j^i$.

Since $\mathbb{P}(V_0) \geq 1 - \frac{2\delta q}{\sqrt{C r^2 b(q)}}$, it follows that $|\mathbb{P}_t(V_i) - \mathbb{P}_{t+r}(V_i)| \leq \frac{2\delta q}{\sqrt{C r b(q)}}$, for all $i \in \{0, 1, \ldots, r-1\}$. (In fact, this holds independently of the value of the time t). This can be shown by an argument similar to that in Case I of the induction basis.

We will show using the induction hypothesis that for $i = 0, 1, \ldots, r-1$ and $t \geq t_q$ it holds:

$$||\mathbb{P}^i_t - \mathbb{P}^i_{t+r}|| \leq \frac{\delta}{S} \cdot \left(S_{q-1} + \frac{q}{2b(q)}\right)$$

We claim that this will conclude the proof. This follows from the following chain of inequalities:

$$\begin{aligned} ||\mathbb{P}_t - \mathbb{P}_{t+r}|| &\leq \sum_{i=0}^{r-1} |\mathbb{P}_t(V_i) - \mathbb{P}_{t+r}(V_i)| + \Big|\Big|\sum_{i=0}^{r-1} \mathbb{P}_t(V_i) \cdot (\mathbb{P}^i_t - \mathbb{P}^i_{t+r})\Big|\Big| && (3.10) \\ &\leq \frac{2\delta q}{\sqrt{C b(q)}} + \frac{\delta}{S}\left(S_{q-1} + \frac{q}{2b(q)}\right) && (3.11) \\ &\leq \frac{\delta}{S} S_q && (3.12) \end{aligned}$$

Step (3.10) follows easily from the triangle inequality (recall that the statistical distance is a norm) and by using the fact that the $\mathbb{P}^i_j$'s are distributions. For Step (3.11) observe that the second summand in (3.10) is a convex combination and Step (3.12) assumes that C is large enough.

To finish the proof we show that $||\mathbb{P}^0_t - \mathbb{P}^0_{t+r}|| \leq \frac{\delta}{S} \cdot \left(S_{q-1} + \frac{q}{2b(q)}\right)$. The proofs for the $r-1$ remaining cases are very similar.

For $i = 0, 1, \ldots, r-1$ denote $\mathbb{P}^i = (\mathbb{P}|V_i)$. Let $\mathbf{N}_j = (N^1_j, \ldots, N^{r-1}_j)$ be a random vector such that the random variable N^l_j ($l = 1, 2, \ldots, r-1$) counts the number of times the walk makes a step $x \in \mathbb{Z}^q_r$ with $x_1 = l$ during the first j steps. Consider a vector $\mathbf{s} = (s_1, s_2, \ldots, s_{r-1})$ such that $|\mathbf{s}| \overset{\text{def}}{=} \sum_{i=1}^{r-1} s_i \leq j$ and $\sum_{k=1}^{r-1} k s_k \equiv 0 \mod r$. Then, we have:

$$(\mathbb{P}^0_j|\mathbf{N}_j = \mathbf{s}) = \left(\{*\prod\}_{i=1}^{r-1}(\mathbb{P}^i)^{*s_i})\right) * (\mathbb{P}^0)^{*(j-|\mathbf{s}|)}$$

where by $\{*\prod\}$ we denote the convolution product. The above equality holds for the following reason: The distribution on the left hand side is the distribution on $V_0 \cong \mathbb{Z}_r^{q-1}$ given that the walk makes s_l steps x with $x_1 = l$ $(l = 1, 2, \ldots, r-1)$ (and $j - |\mathbf{s}|$ steps with $x_1 = 0$). The equality follows by commutativity.

Therefore, by the law of total probability, we can write $\mathbb{P}_j^0$ as the following convex combination of conditional distributions:

$$\mathbb{P}_j^0 = \sum_{(\sum_{k=1}^{r-1} ks_k \equiv 0 \mod r) \text{ and } (|\mathbf{s}| \leq j)} \Pr[\mathbf{N}_j = \mathbf{s}] \cdot \left(\{*\prod\}_{i=1}^{r-1} (\mathbb{P}^i)^{*s_i})\right) * (\mathbb{P}^0)^{*(j-|\mathbf{s}|)}$$

Using this fact, we can bound $||\mathbb{P}_t^0 - \mathbb{P}_{t+r}^0||$ for $t = t_q$ as follows:

$$\begin{aligned} ||\mathbb{P}_t^0 - \mathbb{P}_{t+r}^0|| &\leq \Pr[\mathbf{N}_t \neq \mathbf{N}_{t+r}] + \Pr[|\mathbf{N}_t| \geq 4qr^2 \log r\sqrt{C} \log(1/\delta)] \\ &+ \sum_{\substack{\mathbf{s} \text{ such that} \\ (\sum_{k=1}^{r-1} ks_k \equiv 0 \mod r) \\ (|\mathbf{s}| \leq 4qr^2 \log r\sqrt{C} \log(1/\delta))}} \Pr[\mathbf{N}_t = \mathbf{s}] \cdot \left\| (\{*\prod\}_{i=1}^{r-1} (\mathbb{P}^i)^{*s_i}) * [(\mathbb{P}^0)^{*(t-|\mathbf{s}|)} - (\mathbb{P}^0)^{*(t+r-|\mathbf{s}|)}] \right\| \end{aligned}$$

The first summand is equal to the probability that a non-trivial step in the first coordinate (i.e step x with $x_1 \neq 0$) was made in one of the times $t+1, \ldots, t+r$ and this is at most $2\delta q/\sqrt{C}rb(q)$ (because $\mathbb{P}(V_0) \geq 1 - 2\delta q/\sqrt{C}r^2 b(q)$).

To upper bound the second summand, we observe that $|\mathbf{N}_t| = \sum_{i=1}^{r-1} N_t^i$ is a binomial random variable with parameters $t = t_q$ and $p \leq 2\delta q/\sqrt{C}r^2 b(q)$. Thus, by a standard Chernoff bound, we get that the second summand is also very small, so that the sum of the first two summands is at most $\frac{\delta}{S} \cdot \frac{q}{2b(q)}$ for large enough C.

Now consider the third summand. Since $|\mathbf{s}| \leq 4qr^2 \log r\sqrt{C} \log(1/\delta)$, it follows that $t_q - |\mathbf{s}| \geq t_{q-1}$ and the induction hypothesis implies:

$$\begin{aligned} \left\| \left(\{* \prod\}_{i=1}^{r-1} (\mathbb{P}^i)^{*s_i} \right) * \left[(\mathbb{P}^0)^{*(t-|\mathbf{s}|)} - (\mathbb{P}^0)^{*(t+r-|\mathbf{s}|} \right] \right\| &\leq \left\| (\mathbb{P}^0)^{*(t-|\mathbf{s}|)} - (\mathbb{P}^0)^{*(t+r-|\mathbf{s}|)} \right\| \\ &\leq \frac{\delta}{S} \cdot S_{q-1} \end{aligned}$$

The first inequality follows from the fact that $\{* \prod\}_{i=1}^{r-1} (\mathbb{P}^i)^{*s_i}$ is a distribution. Therefore, the expression $\frac{\delta}{S} \cdot S_{q-1}$ is an upper bound for the third summand and the proof is complete.

3.6.3 s-sparse polynomials are far from longer affine forms

Recall that the *length* of a monomial is the number of distinct variables that occur in it (so for example $x_1^2 x_2^4$ has length two). Recall that an *affine* function is simply a degree-1 polynomial.

Let $f : \mathbb{F}^n \rightarrow \mathbb{F}$ be any function. We say that the *influence* of variable x_i on f is

$$\mathrm{Inf}_f(i) \stackrel{\mathrm{def}}{=} \Pr_{x_1,\ldots,x_n,y \in \mathbb{F}} [f(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n) \neq f(x_1,\ldots,x_{i-1},y,x_{i+1},\ldots,x_n)].$$

If f is a single monomial of length ℓ that contains the variable x_1, then the influence of x_1 on f is $(1-\frac{1}{P})^\ell$ (the probability that the other $\ell - 1$ variables besides x_1 all take nonzero values is $(1-\frac{1}{P})^{\ell-1}$, and then there is a $1-\frac{1}{P}$ probability that the value of x_1 changes when we re-randomize). Similarly, if g is an s-sparse polynomial in which x_1 occurs in r monomials of length $\ell_1, \ldots, \ell_r$, then the influence of x_1 is at most

$$\left(1-\frac{1}{P}\right)^{\ell_1} + \cdots + \left(1-\frac{1}{P}\right)^{\ell_r}.$$

The *total influence* of f is the sum of the influences of all variables. Each monomial of length ℓ in a polynomial g contributes at most $\ell(1-\frac{1}{P})^\ell$ to the total influence of f (i.e. if a polynomial has k monomials of lengths $\ell_1, \ldots, \ell_k$ then the total influence of g is at most $\ell_1(1-\frac{1}{P})^{\ell_1} + \cdots + \ell_k(1-\frac{1}{P})^{\ell_k}$.

Note that each variable in an affine function of the form (3.3) has influence $1 - \frac{1}{P}$, and the total influence of such a function is precisely $(s + r)(1 - \frac{1}{P})$.

The following fact will be useful:

Fact 53. *Let $f, g : \mathbb{F}^n \to \mathbb{F}$ be two functions such that for some variable x_i we have $|\mathrm{Inf}_f(i) - \mathrm{Inf}_g(i)| = \tau$. Then f is $\tau/2$-far from g.*

Proof. We may assume without loss of generality that $\mathrm{Inf}_g(i) = \mathrm{Inf}_f(i) + \tau$. Let x denote a uniform random input from $\mathbb{F}$ and let x' denote x with the i-th coordinate re-randomized. We have

$$\Pr_{x,x'}[g(x) \neq g(x')] \;\leq\; \Pr_{x,x'}[g(x) \neq f(x)] + \Pr_{x,x'}[f(x) \neq f(x')] + \Pr_{x,x'}[f(x') \neq g(x')].$$

Rearranging, we get

$$\begin{aligned} \tau &= \Pr_{x,x'}[g(x) \neq g(x')] - \Pr_{x,x'}[f(x) \neq f(x')] \\ &\leq \Pr_{x,x'}[g(x) \neq f(x)] + \Pr_{x,x'}[f(x') \neq g(x')] = 2 \Pr_{x,x'}[g(x) \neq f(x)] \end{aligned}$$

where the final inequality holds since both x and x' are uniformly distributed. This gives the fact. □

Finally, recall that in any polynomial $g(x_1, \ldots, x_n)$ over $\mathbb{F}$, we may assume without loss of generality that no variable's degree in any monomial is greater than $P - 1$. (The multiplicative group is of size $P - 1$ and hence $\alpha^P = \alpha$ for every $\alpha \in \mathbb{F}$.)

Proof of Theorem 42.

The high-level idea of the proof of Theorem 42 is as follows. Let M be a particular monomial in g, and consider what happens when g is hit with a restriction that fixes all variables that do not occur in M. M itself is not affected by the restriction, but it is possible for a longer monomial to "collapse" onto M and obliterate it (i.e. if M is $x_1 x_2^2$ and g contains another monomial $M' = -x_1 x_2^2 x_3^3$, then a restriction that fixes $x_3 \leftarrow 1$ would cause M' to collapse onto M and in fact obliterate M). We show that g must have a short monomial M (which, however, has degree at least 2) with the following property: for a constant fraction of all possible restrictions of variables not in M, no longer monomial collapses onto M. This implies that for a constant fraction of all such restrictions ρ, the induced polynomial g_ρ is "substantially" different from any affine function (since g_ρ – a polynomial of degree at least two – is not identical to any affine function, it must be "substantially" different since there are only length(M) surviving variables), and hence g itself must be "far" from any affine function.

Now we give the actual proof. Let g be an s-sparse polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ and let $A(x)$ be a fixed affine function given by equation (3.3). We will show that g must be $\Phi(P)$-far from A and thus prove the theorem.

First note that without loss of generality we may assume g has no term of degree 1. (Suppose g has t such terms. Let g' be the polynomial obtained by subtracting off these terms. Then g' is $(s - t)$-sparse and is $\Phi(P)$-close to the affine function $A'(x)$ obtained by subtracting off the same terms; this affine function has at least $s + r - t$ nonconstant terms. So we can run the following argument on g' with $s - t$ playing the role of "s" in the lemma.)

Now we observe that g must satisfy

$$\text{Inf}_g(1) + \cdots + \text{Inf}_g(s) \geq (1 - 4\Phi(P))s(1 - \frac{1}{P}). \tag{3.13}$$

If this were not the case, then some variable x_i in $x_1, \ldots, x_s$ would necessarily have influence at most $(1 - 4\Phi(P))(1 - \frac{1}{P})$ on g. Since the influence of x_i on (3.3) is $1 - \frac{1}{P}$, by Fact 53 this would mean that g is at least $2\Phi(P)(1 - \frac{1}{P}) \geq \Phi(P)$-far from (3.3), and we would be done.

Notation. We will henceforth refer to monomials in g of length less than P^2 as *short* monomials, and we write S to denote the set of all short monomials in g. For $P^2 \leq \ell \leq P^8$, we refer to monomials in g of length ℓ as *intermediate* monomials, and we write I to denote the set of all intermediate monomials in g. Finally, for $\ell > P^8$ we refer to monomials in g of length ℓ as *long* monomials, and we write L to denote the set of all long monomials.

Observe that

- Each monomial in g that is intermediate or long contributes at most $1/4$ to $\mathrm{Inf}_g(1) + \cdots + \mathrm{Inf}_g(s)$. This is because each monomial of length $\ell \geq P^2$ contributes at most $\ell(1-\frac{1}{P})^\ell$ to this sum, and for integer ℓ the value $\max_{\ell \geq P^2} \ell(1-\frac{1}{P})^\ell$ is achieved at $\ell = P^2$ where the value is at most $1/4$ (the upper bound holds for all integer $P \geq 2$).

- Each short monomial in g contributes at most P/e to $\mathrm{Inf}_g(1) + \cdots + \mathrm{Inf}_g(s)$. This is because $\max_{\ell \geq 1} \ell(1-\frac{1}{P})^\ell \leq P/e$ (the max is achieved around $\ell \approx P$).

Since the RHS of (3.13) is at least $(1-\frac{1.2}{P})s$, we have the following inequalities:

$$\frac{|I|+|L|}{4} + \frac{|S|P}{e} \geq \left(1 - \frac{1.2}{P}\right) s \qquad \text{and} \qquad |I|+|L| \leq s$$

(the second inequality holds simply because there are at most s long monomials). These inequalities straightforwardly yield $|S| \geq \frac{s}{3P}$.

Let m_ℓ denote the number of monomials in g that have length exactly ℓ. Note that we have $\sum_{\ell > P^8} m_\ell = |L| \leq s$.

Given two monomials M_1, M_2 that occur in g, we say that M_1 *covers* M_2 if all variables in M_1 are also in M_2 (note we do not care about the degrees of the variables in these monomials). We refer to such a pair (M_1, M_2) as a *coverage*; more precisely, if M_1 is of length ℓ we refer to the pair (M_1, M_2) as an *ℓ-coverage*. (One can view each ℓ-coverage as an edge in a bipartite graph.)

Let $S' \subseteq S$ be the set of those monomials M in S which are "maximal" in the sense that no other monomial $M' \in S$ (with $M' \neq M$) covers M.

Claim 54. *We have* $|S'| \geq s/(3P^{P^2})$.

Proof. Since S is finite it is clear that S' is nonempty; suppose the elements of S' are $M_1, \ldots, M_k$. Each of the (at least $s/(3P)$ many) elements of S is covered by some M_i. But each M_i is of length ℓ for some $\ell \leq P^2 - 1$, and hence can cover at most P^ℓ monomials (any monomial covered by M_i is specified by giving ℓ exponents, each between 0 and $P-1$, for the ℓ variables in M_i). □

Fix any $\ell \geq P^2$. Each fixed monomial of length ℓ participates in at most $\binom{\ell}{P^2} P^{P^2} \leq (\ell P)^{P^2}$ many ℓ-coverages of monomials in S'. (There are $\binom{\ell}{P^2}$ ways to choose a subset of P^2 variables, and once chosen, each variable may take any exponent between 0 and $P-1$.) Consequently, the length-ℓ monomials in g collectively participate in at most $m_\ell(\ell P)^{P^2}$ many ℓ-coverages of variables in S' in total. By Claim 54, it follows that

$$\mathop{\mathbf{E}}_{M \in S'}[\#\ \ell\text{-coverages } M \text{ is in}] \leq \frac{m_\ell(\ell P)^{P^2}}{s/(3P^{P^2})} = \frac{3m_\ell \ell^{P^2} P^{2P^2}}{s}.$$

By Markov's inequality, we have

$$\Pr_{M \in S'}[\#\ \ell\text{-coverages } M \text{ is in} \geq 3m_\ell \ell^{P^2+2} P^{2P^2}/s] \leq 1/\ell^2.$$

So for each $\ell \geq P^2$, we have that at most a $1/\ell^2$ fraction of monomials in S' are covered by at least $3m_\ell \ell^{P^2+2} P^{2P^2}/s$ many length-ℓ monomials. Since $\sum_{\ell \geq P^2} 1/\ell^2 < 1/2$, we have that at least half of the monomials in S' have the following property:

- For all $\ell \geq P^2$, at most $3m_\ell \ell^{P^2+2} P^{2P^2}/s$ many length-ℓ monomials cover M. $\quad (\dagger)$

Fix M to be some particular monomial with property $(\dagger)$. Since M belongs to S', we know that no short monomial in g covers M; we now show that for a constant fraction of all restrictions ρ of variables outside of M, no intermediate or long monomial in g_ρ covers M. (Once this is accomplished, we will be almost done.)

First observe that for any value ℓ with $P^2 \leq \ell \leq P^8$, using the fact that m_ℓ/s is at most 1, we have that at most

$$3\ell^{P^2+2} P^{2P^2} \leq 3P^{10P^2+16} \leq P^{10P^2+18}$$

many length-ℓ monomials cover M. So in total there are at most $(P^8 - P^2 + 1)P^{10P^2+18} \leq P^{10P^2+26}$ many intermediate monomials that cover M; let T denote the set of these intermediate monomials. Each intermediate monomial in T has length strictly greater than the length of M, so each such monomial contains at least one variable that is not in M. Let V be a set of at most P^{10P^2+26} variables such that each monomial in T contains at least one variable from V, and let ρ_1 be the restriction that sets all variables in V to 0 and leaves all other variables unfixed. Note that for each long monomial in g, applying ρ_1 either kills the monomial (because some variable is set to 0) or leaves it unchanged (no variable in the monomial is set) in g_{ρ_1}. Thus the result of applying ρ_1 is that no intermediate monomial in g_{ρ_1} covers M.

Now let ρ_2 denote a random restriction over the remaining variables which leaves free precisely those variables that occur in M and fixes all other variables independently to uniformly chosen elements of $\mathbb{F}$. Suppose M' is a long monomial (of length $\ell > P^8$) from g that survived into g_{ρ_1}. It must be the case that M' contains at least $\ell - P^2$ variables that are neither in M nor in V, and consequently the probability that M' is not killed by ρ_2 (i.e. the probability that all variables in M' that are not in M are set to nonzero values under ρ_2) is at most $(1-\frac{1}{P})^{\ell-P^2}$. Consequently the expected number of length-ℓ monomials in g_{ρ_1} that cover M and are not killed by ρ_2 is at most $3m_\ell \ell^{P^2} P^{2P^2}(1-\frac{1}{P})^{\ell-P^2}/s$. Summing over all $\ell > P^8$, we have

$$\begin{aligned} &\mathop{\mathbf{E}}_{\rho_2}[\#\text{ long monomials that cover } M \text{ and survive } \rho_1\rho_2] && (3.14)\\ &\quad\leq \sum_{\ell>P^8} \frac{3m_\ell \ell^{P^2} P^{2P^2}(1-\frac{1}{P})^{\ell-P^2}}{s} \\ &\quad\leq \left(\sum_{\ell>P^8} \frac{m_\ell}{s}\right) \cdot \max_{\ell\geq P^8} 3\ell^{P^2} P^{2P^2}(1-\frac{1}{P})^{\ell-P^2}. && (3.15)\end{aligned}$$

We have $\sum_{\ell>P^8} \frac{m_\ell}{s} \leq 1$. A routine exercise shows that for all $P \geq 2$, the max in (3.15) is achieved at $\ell = P^8$ where the value is at most $1/2$ (in fact it is far smaller). So (3.14) is certainly at most $1/2$, and we have

$$\mathop{\mathbf{E}}_{\rho_2}[\#\text{ long monomials that cover } M \text{ and survive } \rho_1\rho_2] \leq 1/2.$$

So the probability that any long monomial that covers M survives $\rho_1\rho_2$ is at most $1/2$. Since we already showed that no short or intermediate monomial in $g_{\rho_1\rho_2}$ covers M, it follows that with probability at least $1/2$ over the random choice of ρ_2, no monomial in $g_{\rho_1\rho_2}$ covers M except for M itself.

Now let ρ denote a truly random restriction that assigns all variables not in M uniformly at random and keeps all variables in M free. Since the variables in V will be assigned according to ρ_2 with probability $1/P^{P^{10P^2+26}}$, we have that with probability at least $1/(2P^{P^{10P^2+26}}) > 1/(P^{P^{10P^2+26}+1})$ over the random choice of ρ, no monomial in g_ρ covers M. Suppose ρ is such a restriction. Since M itself clearly survives the restriction ρ, we have that the function g_ρ (a function on length$(M) \leq P^2 - 1$ many variables) is different from the function A_ρ – this is simply because the polynomial g_ρ contains the monomial M, which is not of degree 1, whereas all monomials in A_ρ have degree 1. Hence the functions g_ρ and A_ρ differ on at least one of the (at most) P^{P^2-1} possible inputs.

So, we have shown that for at least a $1/(P^{P^{10P^2+26}+1})$ fraction of all restrictions of the variables not occurring in M, the error of g under the restriction in computing A is at least $1/P^{P^2-1}$. This implies that the overall error of g in computing A is at least

$$1/(P^{P^{10P^2+26}+P^2}) = \Phi(P)$$

and we are done with the proof of Theorem 42. □

3.7 Lower Bounds for Boolean Function Classes

By adapting techniques of Chockler and Gutfreund [13], we can also obtain $\tilde{\Omega}(\log s)$ lower bounds for many of the other testing problems listed in Table 1.1. More precisely, we prove lower bounds on the query complexity of testing size-s decision trees, size-s branching programs, s-term DNF, and size-s Boolean formulas (Theorem 55), and Boolean functions with Fourier degree at most d (Theorem 58).

Theorem 55. *Let $\epsilon = 1/1000$. Any ϵ-testing algorithm for any of the following classes of functions over $\{0,1\}^n$ must make $\Omega(\log s/\log\log s)$ queries: (i) size-s decision trees; (ii) size-s branching programs; (iii) s-term DNF; (iv) size-s Boolean formulas.*

Proof. The proof combines a counting argument with the result of Chockler and Gutfreund [13] showing that $\Omega(J/k)$ queries are required to distinguish between J-juntas and $(J+k)$-juntas over $\{0,1\}^n$. More precisely, consider the following distributions:

1. D_{NO} is the uniform distribution over all functions (on n variables) that depend on (at most) the first $(J+k)$ variables.

2. D_{YES} is the distribution obtained in the following way. Choose a k-element subset $\mathcal{I}_k$ uniformly and randomly from the set $\{1, \ldots, J+k\}$. Then choose a uniformly random function from the set of all functions on n variables that depend on (at most) the variables indexed by the set $[J+k] \setminus \mathcal{I}_k$.

Chockler and Gutfreund show that with very high probability a random draw from D_{NO} is far from every J-junta, whereas clearly every draw from D_{YES} is a J-junta. Given any putative testing algorithm, the distributions $D_{\text{YES}}, D_{\text{NO}}$ over functions induce two distributions C_{YES}, C_{NO} over "query-answer histories". Chockler and Gutfreund show that for any (even adaptive) algorithm that makes fewer than $\Omega(J/k)$ queries, the statistical difference between C_{YES} and C_{NO} will be at most $1/6$. This implies that any successful testing algorithm must make $\Omega(J/k)$ queries.

We adapt this argument to prove Theorem 55 as follows. Let $\mathcal{C}_s$ be a class of functions for which we would like to prove a lower bound (e.g. $\mathcal{C}_s$ could be the class of all Boolean functions over n variables that are computed by decision trees of size at most s). We choose J (as a function of s) such that any J-junta is a function in $\mathcal{C}_s$; with this choice the distribution D_{YES} described above is indeed a distribution over functions in the class. We choose k (as a function of J) so that with very high probability, a random function drawn from D_{NO} (i.e. a random function over the first $J+k$ variables) is ϵ-far from every function in $\mathcal{C}_s$. This gives an $\Omega(J/k)$ lower bound for testing whether a black-box function is in $\mathcal{C}_s$ or is ϵ-far from every function in $\mathcal{C}_s$.

For all of the classes addressed in Proposition 55 we can take $J = \log_2 s$ and $k = \Theta(\log J)$. We work through the analysis for size-s decision trees, sketch the analysis for size-s branching programs, and leave the (very similar) analysis for s-term DNF and size-s Boolean formulas to the interested reader.

Decision Trees (of size s): We set $J = \log_2 s$ and $k = \log_2 J$. It is clear that any J-junta can be expressed as a size-s decision tree.

Lemma 56. *Fix $\epsilon = 1/1000$. With very high probability, a random $(J + \log J)$-junta over the first $(J + \log J)$ variables is ϵ-far from any size-s decision tree over the first $(J + \log J)$ variables.*

Proof. For any size-s decision tree over the first $(J + \log J)$ variables, the number of $(J + \log J)$-juntas (over these variables) ϵ-close to it equals $\sum_{i=0}^{\epsilon \cdot 2^{J+\log J}} \binom{2^{J+\log J}}{i}$. For $\epsilon = 1/1000$, this is at most $2^{0.1 \cdot 2^{J+\log J}} = 2^{(J/10)2^J}$ (recall that the sum of the binomial coefficients $\sum_{k=0}^{n/\alpha} \binom{n}{k}$ is $O(C(\alpha)^n)$, where $C(\alpha) = \alpha^{1/\alpha}(\frac{\alpha}{\alpha-1})^{\frac{\alpha-1}{\alpha}}$.)

Now we upper bound the number of size-s decision trees over the first $J + \log J$ variables. There are at most $4^s = 2^{2 \cdot 2^J})$ distinct decision tree topologies for trees with s leaves. For each topology there are at most $(J + \log J)^s \leq 2^{2s \log \log s} = 2^{(2 \log J)2^J}$ different labellings of the nodes.

Thus, the number of $(J + \log J)$-juntas that are ϵ-close to *any* decision tree of size s (over the first $J + \log J$ variables) is at most $2^{(J/10 + 2 \log J)2^J}$. This is a vanishingly small fraction of the total number of $(J + \log J)$-juntas over the first $(J + \log J)$ variables, which is $2^{2^{J+\log J}} = 2^{J \cdot 2^J}$. □

We are not quite done, since we need that with very high probability a random function from D_{NO} is ϵ-far from *every* size-s decision tree, not just from size-s decision trees over the first $(J + \log J)$ variables. This follows easily from the previous lemma:

Corollary 57. *For $\epsilon = 1/1000$, with very high probability a random $(J + \log J)$-junta over the first $(J + \log J)$ variables is ϵ-far from any size-s decision tree (over n variables).*

Proof. Let f be any $(J + \log J)$-junta over the set $\{x_1, \dots, x_{J+\log J}\}$. Suppose that g is a size-s decision tree over $\{x_1, \dots, x_n\}$ that is ϵ-close to f. It is not hard to show that then there exists a size-s decision tree g' over the relevant variables $\{x_1, \dots, x_{J+\log J}\}$ that is ϵ-close to f as well (g' can be obtained from g by fixing all the irrelevant variables to the values that maximize g's agreement with f). □

We have thus established part (i) of Theorem 55.

Branching Programs: We only sketch the required analysis. We set $J = \log_2 s$ and $k = 10 \log_2 J$. Any J-junta can be expressed as a size-s branching program. Simple counting arguments show that for $\epsilon = 1/1000$, a random $(J+k)$-junta over $\{x_1, \dots, x_{J+k}\}$ is with high probability ϵ-far from every size-s Branching Program over $\{x_1, \dots, x_{J+k}\}$. An analogue of Corollary 57 completes the argument.

This completes the proof of Theorem 55. □

Remark: We note that these simple arguments do not seem to give any non-trivial testing lower bound for the class of Boolean circuits of size s. It would be interesting to obtain lower bounds for this class.

Finally, we point out the following:

Theorem 58. *Let $0 < \epsilon < 1/2$. Any non-adaptive ϵ-testing algorithm for the class of Boolean functions over $\{0,1\}^n$ with Fourier degree d must make $\tilde{\Omega}(\sqrt{d})$ queries.*

Proof. Consider the following two distributions over Boolean functions on $\{-1,1\}^n$:

1. D_{NO} is the uniform distribution over all $\binom{n}{d+2}$ parities of exactly $d+2$ variables from $x_1, \dots, x_n$;

2. D_{YES} is the uniform distribution over all $\binom{n}{d}$ parities of exactly d variables from $x_1, \dots, x_n$.

Every function in the D_{YES} distribution clearly has Fourier degree, whereas every function in the D_{NO} distribution has distance precisely $1/2$ from every function with Fourier degree d (this follows immediately from Parseval's identity). Fischer *et al.* showed that any non-adaptive algorithm for distinguishing draws from D_{YES} versus D_{NO} must make $\tilde{\Omega}(\sqrt{d})$ draws; this immediately gives the desired result. □

3.8 Conclusion

There are many open questions raised by the work in this chapter. One is whether our lower bounds can be strengthened. Can poly(s) query lower bounds be obtained for classes such as size-s decision trees, s-term DNF, etc?

On the upper-bound side, our results are all achieved via a single generic algorithm that is not geared toward any particular class of functions. For many classes of interest, the query complexity of this algorithm is $\mathrm{poly}(s, 1/\epsilon)$, but the running time is exponential in s. This raises the natural question: can we also improve the running time for any of these classes? One approach to achieving better runtimes is to replace our "implicit learning" step with a more efficient proper learning algorithm (the current learning algorithm simply gathers random examples and exhaustively checks for a consistent hypothesis in the concept class $\mathcal{C}(\tau^\star)_{J(\tau^\star)}$). For some specific concept classes, proper learning is known to be NP-hard, but for other classes, such as s-sparse $GF(2)$ polynomials, polynomial-time proper learning algorithms are known. In next chapter, we leverage this fact to get a tester for s-sparse $GF(2)$ polynomials that is both query *and* time efficient.

Chapter 4

Efficiently Testing Sparse *GF*(2) Polynomials

4.1 Introduction

In the previous chapter, we gave a general technique called "testing by implicit learning," which we used to test a variety of different function classes that were not previously known to be testable. Intuitively, these classes correspond to functions with "concise representations," such as s-term DNFs, size-s Boolean formulas, size-s Boolean circuits, and s-sparse polynomials over constant-size finite fields. For each of these classes, the testing algorithm in the last chapter made only poly$(s, 1/\epsilon)$ queries (independent of n).

The main drawback of the previous algorithm is that for each of the classes mentioned above, the algorithm's running time is $2^{\omega(s)}$ as a function of s, and $\omega(\text{poly}(1/\epsilon))$ as a function of ϵ.[1] Thus, a natural question is whether any of these classes can be tested with both time complexity and query complexity poly$(s, 1/\epsilon)$.

In this chapter we focus on the class of *s-sparse polynomials over $GF(2)$*. Polynomials over $GF(2)$ (equivalently, parities of ANDs of input variables) are a simple and well-studied representation for Boolean functions. It is well known that every Boolean

[1] We note that the algorithm also has a linear running time dependence on n, the number of input variables; this is in some sense inevitable since the algorithm must set n bit values just to pose a black-box query to f. Our algorithm has running time linear in n for the same reason. For the rest of the chapter we discuss the running time only as a function of s and ϵ.

function has a unique representation as a multilinear polynomial over $GF(2)$, so the sparsity (number of monomials) of this polynomial is a very natural measure of the complexity of f. Sparse $GF(2)$ polynomials have been studied by many authors from a range of different perspectives such as learning [8, 27, 55, 10, 12], approximation and interpolation [35, 31, 53], the complexity of (approximate) counting [20, 36, 41], and property testing [17].

The main result of this chapter is a testing algorithm for s-sparse $GF(2)$ polynomials that is both time-efficient and query-efficient:

Theorem 59. *There is a poly$(s, 1/\epsilon)$-query algorithm with the following performance guarantee: given parameters s, ϵ and black-box access to any $f : \{0,1\}^n \rightarrow \{0,1\}$, it runs in time* $\mathrm{poly}(s, 1/\epsilon)$ *and tests whether f is an s-sparse $GF(2)$ polynomial versus ϵ-far from every s-sparse polynomial.*

This answers the question left open by the previous chapter, by exhibiting an interesting and natural class of functions with "concise representations" that can be tested efficiently, both in terms of query complexity and running time.

We obtain our main result by extending the "testing by implicit learning" approach. The "implicit learning" step from the previous chapter used a naive brute-force search for a consistent hypothesis; here we employ a sophisticated proper learning algorithm due to Schapire and Sellie [55]. It is much more difficult to "implicitly" run the [55] algorithm than the brute-force search. One of the main technical contributions in this chapter is a new structural theorem about how s-sparse $GF(2)$ polynomials are affected by certain carefully chosen restrictions; this is an essential ingredient that enables us to use the [55] algorithm. We elaborate on this below.

Techniques. In the last chapter we showed that for many classes of functions defined by a size parameter s, it is possible to "implicitly" run a (very naive) proper learning algorithm over a number of variables that is independent of n, and thus obtain an overall query complexity independent of n. More precisely, weobserved that for many classes $\mathcal{C}$ every $f \in \mathcal{C}$ is "very close" to a function $f' \in \mathcal{C}$ for which the number r of relevant variables is polynomial in s and independent of n; roughly speaking, the relevant variables for f'

are the variables that have high influence in f. (For example, if f is an s-sparse $GF(2)$ polynomial, an easy argument shows that there is a function f' – obtained by discarding from f all monomials of degree more than $\log(s/\tau)$ – that is τ-close to f and depends on at most $r = s\log(s/\tau)$ variables.) They then showed how, using ideas of Fischer et al. [26] for testing juntas, it is possible to construct a sample of uniform random examples over $\{0,1\}^r$ which with high probability are all labeled according to f'. At this point, the proper learning algorithm we employed was a naive brute-force search. Our algorithm tried all possible functions in $\mathcal{C}$ over r (as opposed to n) variables, to see if any were consistent with the labeled sample. Thus we obtained a testing algorithm with overall query complexity poly(s/ϵ) but whose running time was dominated by the brute-force search. For the class of s-sparse $GF(2)$ polynomials, our algorithm used $\tilde{O}(s^4/\epsilon^2)$ queries but had running time at least $2^{\omega(s)} \cdot (1/\epsilon)^{\log\log(1/\epsilon)}$.

The high-level idea of this chapter is to employ a much more sophisticated – and efficient – proper learning algorithm than brute-force search. In particular we would like to use a proper learning algorithm which, when applied to learn a function over only r variables, runs in time polynomial in r and in the size parameter s. For the class of s-sparse $GF(2)$ polynomials, precisely such an algorithm was given by Schapire and Sellie [55]. Their algorithm, which we describe in Section 4.2.2, is computationally efficient and generates a hypothesis h which is an s-sparse $GF(2)$ polynomial. But this power comes at a price: the algorithm requires access to a *membership query* oracle, i.e. a black-box oracle for the function being learned. Thus, in order to run the Schapire/Sellie algorithm in the "testing by implicit learning" framework, it is necessary to simulate membership queries to an approximating function $f' \in \mathcal{C}$ which is close to f but depends on only r variables. This is significantly more challenging than generating uniform random examples labeled according to f', which is all that was required before

To see why membership queries to f' are more difficult to simulate than uniform random examples, recall that f and the f' described above (obtained from f by discarding high-degree monomials) are τ-close. Intuitively this is extremely close, disagreeing only on a $1/m$ fraction of inputs for an m that is much larger than the number of random examples required for learning f' via brute-force search (this number is "small" – independent of

n – because f' depends on only r variables). Thus before, it sufficed to use f, the function to which we actually have black-box access, rather than f' to label the random examples used for learning f'; since f and f' are so close, and the examples are uniformly random, with high probability all the labels will also be correct for f'. However, now that membership queries are required, things are no longer so simple. For any given f' which is close to f, one can no longer assume that the learning algorithm's queries to f' are uniformly distributed and hence unlikely to hit the error region – indeed, it is possible that the learning algorithm's membership queries to f' are clustered on the few inputs where f and f' disagree.

In order to successfully simulate membership queries, we must somehow consistently answer queries according to a particular f', even though we only have oracle access to f. Moreover this must be done implicitly in a query-efficient way, since explicitly identifying even a single variable relevant to f' requires at least $\Omega(\log n)$ queries. This is the main technical challenge we address.

We meet this challenge by showing that for any s-sparse polynomial f, an approximating f' can be obtained as a restriction of f by setting certain carefully chosen subsets of variables to zero. Roughly speaking, this restriction is obtained by randomly partitioning all of the input variables into r subsets and zeroing out all subsets whose variables have small "collective influence" (more precisely, small variation in the sense of [26]). It is important that the restriction sets these variables to zero, rather than a random assignment; intuitively this is because setting a variable to zero "kills" all monomials that contain the variable, whereas setting it to 1 does not. Our main technical theorem (Theorem 64, given in Section 5.5.1) shows that this f' is indeed close to f and has at most one of its relevant variables in each of the surviving subsets. We moreover show that these relevant variables for f' all have high influence in f (the converse is not true; examples can be given which show that not every variable that has "high influence" in f will in general become a relevant variable for f'). This property is important in enabling our simulation of membership queries. In addition to the crucial role that Theorem 64 plays in the completeness proof for our test, we feel that the new insights the theorem gives into how sparse polynomials "simplify" under (appropriately defined) random restrictions may be of independent interest.

Organization. In Section 4.2.2 we describe in detail the "learning component" of the algorithm. In Section 4.3 we state Theorem 64, which provides intuition behind the algorithm and serves as the main technical tool in the completeness proof. In Section 4.4, we present our testing algorithm, **Test-Sparse-Poly**, along with a high-level description and sketch of correctness. The proof of Theorem 64 is presented in section 4.5, while the completeness and soundness proofs are given in sections 4.6 and 4.7, respectively.

4.2 Notation and Background

In this chapter we require some additional notation and background.

4.2.1 Low-influence, high-influence, and well-structured subsets

First we define the notion of low- and high-influence subsets with respect to a partition of the set $[n]$ and a parameter $\alpha > 0$.

Definition 60. *For* $f : \{0,1\}^n \to \{-1,1\}$, *a partition of* $[n]$ *into* $\{I_j\}_{j=1}^r$ *and a parameter* $\alpha > 0$, *define* $L(\alpha) \overset{\text{def}}{=} \{j \in [r] \mid \mathrm{Inf}_f(I_j) < \alpha\}$ *(low-influence subsets) and* $H(\alpha) \overset{\text{def}}{=} [r] \setminus L(\alpha)$ *(high-influence subsets). For* $j \in [r]$ *and* $i \in I_j$, *if* $\mathrm{Inf}_f(i) \geq \alpha$ *we say that the variable* x_i *is a* high-influence element *of* I_j.

Next, the notion of a *well-structured* subset will be important for us:

Definition 61. *For* $f : \{0,1\}^n \to \{-1,1\}$ *and parameters* $\alpha > \Delta > 0$, *we say that a subset* $I \subseteq [n]$ *of coordinates is* (α, Δ)-well structured *if there is an* $i \in I$ *such that* $\mathrm{Inf}_f(i) \geq \alpha$ *and* $\mathrm{Inf}_f(I \setminus \{i\}) \leq \Delta$.

Note that since $\alpha > \Delta$, by monotonicity, the $i \in I$ in the above definition is unique. Hence, a well-structured subset contains a single high-influence coordinate, while the remaining coordinates have small total influence.

4.2.2 Background on Schapire and Sellie's algorithm

In [55] Schapire and Sellie gave an algorithm, which we refer to as **LearnPoly**, for exactly learning s-sparse $GF(2)$ polynomials using membership queries (i.e. black-box queries) and equivalence queries. Their algorithm is *proper*; this means that every equivalence query the algorithm makes (including the final hypothesis of the algorithm) is an s-sparse polynomial. (We shall see that it is indeed crucial for our purposes that the algorithm is proper.) Recall that in an equivalence query the learning algorithm proposes a hypothesis h to the oracle: if h is logically equivalent to the target function being learned then the response is "correct" and learning ends successfully, otherwise the response is "no" and the learner is given a counterexample x such that $h(x) \neq f(x)$.

Schapire and Sellie proved the following about their algorithm:

Theorem 62. *[[55], Theorem 10] Algorithm* **LearnPoly** *is a proper exact learning algorithm for the class of s-sparse $GF(2)$ polynomials over $\{0,1\}^n$. The algorithm runs in poly(n, s) time and makes at most* $\text{poly}(n, s)$ *membership queries and at most* $ns + 2$ *equivalence queries.*

We can easily also characterize the behavior of **LearnPoly** if it is run on a function f that is not an s-sparse polynomial. In this case, since the algorithm is proper all of its equivalence queries have s-sparse polynomials as their hypotheses, and consequently no equivalence query will ever be answered "correct." So if the $(ns + 2)$-th equivalence query is not answered "correct," the algorithm may infer that the target function is not an s-sparse polynomial, and it returns "not s-sparse."

A well-known result due to Angluin [2] says that in a Probably Approximately Correct or PAC setting (where there is a distribution $\mathcal{D}$ over examples and the goal is to construct an ϵ-accurate hypothesis with respect to that distribution), equivalence queries can be straightforwardly simulated using random examples. This is done simply by drawing a sufficiently large sample of random examples for each equivalence query and evaluting both the hypothesis h and the target function f on each point in the sample. This either yields a counterexample (which simulates an equivalence query), or if no counterexample is obtained then simple arguments show that for a large enough ($O(\log(1/\delta)/\epsilon)$-size) sample, with probability $1-\delta$ the functions f and h must be ϵ-close under the distribution $\mathcal{D}$, which is the success criterion for PAC learning. This directly gives the following corollary of Theorem 62:

Corollary 63. *There is a uniform distribution membership query proper learning algorithm, which we call* **LearnPoly**$'(s,n,\epsilon,\delta)$*, which makes* $Q(s,n,\epsilon,\delta) \stackrel{\text{def}}{=} \text{poly}(s,n,1/\epsilon, \log(1/\delta))$ *membership queries and runs in* $\text{poly}(Q)$ *time to learn s-sparse polynomials over* $\{0,1\}^n$ *to accuracy ϵ and confidence $1-\delta$ under the uniform distribution.*

4.3 On restrictions which simplify sparse polynomials

This section presents Theorem 64, which gives the intuition behind our testing algorithm, and lies at the heart of the completeness proof. We give the full proof of Theorem 64 in section 4.5.

Roughly speaking, the theorem says the following: consider any s-sparse $GF(2)$ polynomial p. Suppose that its coordinates are randomly partitioned into $r = \text{poly}(s)$ many subsets $\{I_j\}_{j=1}^r$. The first two statements say that w.h.p. a randomly chosen "threshold value" $\alpha \approx 1/\text{poly}(s)$ will have the property that no single coordinate i, $i \in [n]$, or subset I_j, $j \in [r]$, has $\text{Inf}_p(i)$ or $\text{Inf}_p(I_j)$ "too close" to α. Moreover, the high-influence subsets (w.r.t. α) are precisely those that contain a single high influence element i (i.e. $\text{Inf}_p(i) \geq \alpha$), and in fact each such subset I_j is well-structured (part 3). Also, the number of such high-influence subsets is small (part 4). Finally, let p' be the restriction of p obtained by setting all variables in the low-influence subsets to 0. Then, p' has a nice structure: it has at most one relevant variable per high-influence subset (part 5), and it is close to p (part 6).

Theorem 64. *Let* $p : \{0,1\}^n \rightarrow \{-1,1\}$ *be an s-sparse polynomial. Fix* $\tau \in (0,1)$ *and* Δ *such that* $\Delta \leq \Delta_0 \stackrel{\text{def}}{=} \tau/(1600s^3\log(8s^3/\tau))$ *and* $\Delta = \text{poly}(\tau/s)$*. Let* $r \stackrel{\text{def}}{=} 4Cs/\Delta$*, for a suitably large constant C. Let* $\{I_j\}_{j=1}^r$ *be a random partition of* $[n]$*. Choose α uniformly at random from the set* $\mathcal{A}(\tau,\Delta) \stackrel{\text{def}}{=} \{\frac{\tau}{4s^2} + (8\ell-4)\Delta : \ell \in [K]\}$ *where K is the largest integer such that* $8K\Delta \leq \frac{\tau}{4s^2}$*. Then with probability at least* $9/10$ *(over the choice of α and* $\{I_j\}_{j=1}^r$*), all of the following statements hold:*

1. *Every variable* x_i, $i \in [n]$, *has* $\mathrm{Inf}_p(i) \notin [\alpha - 4\Delta, \alpha + 4\Delta]$.

2. *Every subset* I_j, $j \in [r]$, *has* $\mathrm{Inf}_p(I_j) \notin [\alpha - 3\Delta, \alpha + 4\Delta]$.

3. *For every* $j \in H(\alpha)$, I_j *is* (α, Δ)*-well structured.*

4. $|H(\alpha)| \leq s \log(8s^3/\tau)$.

Let $p' \stackrel{\mathrm{def}}{=} p|_{0 \leftarrow \cup_{j \in L(\alpha)} I_j}$ *(the restriction obtained by fixing all variables in low-influence subsets to* 0*).*

5. *For every* $j \in H(\alpha)$, p' *has at most one relevant variable in* I_j *(hence* p' *is a* $|H(\alpha)|$*-junta).*

6. *The function* p' *is* τ*-close to* p.

Theorem 64 naturally suggests a testing algorithm, whereby we attempt to partition the coordinates of a function f into "high-influence" subsets and "low-influence" subsets, then zero-out the variables in low-influence subsets and implicitly learn the remaining function f' on only $\mathrm{poly}(s, 1/\epsilon)$ many variables. This is exactly the approach we take in the next section.

4.4 The testing algorithm Test-Sparse-Poly

In this section we present our main testing algorithm and give high-level sketches of the arguments establishing its completeness and soundness. The algorithm, called **Test-Sparse-Poly**, takes as input the values $s, \epsilon > 0$ and black-box access to $f : \{0,1\}^n \rightarrow \{-1,1\}$. It is presented in full in Figure 1.

The first thing **Test-Sparse-Poly** does (Step 2) is randomly partition the coordinates into $r = \tilde{O}(s^4/\tau)$ subsets. In Steps 3 and 4 the algorithm attempts to distinguish subsets that contain a high-influence variable from subsets that do not; this is done by using the independence test to estimate the influence of each subset (see Lemma 11).

Once the high-influence and low-influence subsets have been identified, intuitively we would like to focus our attention on the high-influence variables. Thus, Step 5 of the

Algorithm **Test-Sparse-Poly**(f, s, ϵ)

Input: Black-box access to $f : \{0,1\}^n \rightarrow \{-1,1\}$; sparsity parameter $s \geq 1$; error parameter $\epsilon > 0$

Output: **"yes"** if f is an s-sparse $GF(2)$ polynomial, **"no"** if f is ϵ-far from every s-sparse $GF(2)$ polynomial

1. Let $\tau = \Theta(\epsilon), \Delta = \Theta(\text{poly}(\tau, 1/s)), r = \Theta(s/\Delta), \delta = \Theta(\text{poly}(\tau, 1/s))$.[a]
2. Set $\{I_j\}_{j=1}^{r}$ to be a random partition of $[n]$.
3. Choose α uniformly at random from the set $\mathcal{A}(\tau, \Delta) \overset{\text{def}}{=} \{\frac{\tau}{4s^2} + (8\ell - 4)\Delta : 1 \leq \ell \leq K\}$ where K is the largest integer such that $8K\Delta \leq \frac{\tau}{4s^2}$.
4. For each subset $I_1, \ldots, I_r$ run the independence test $M \overset{\text{def}}{=} \frac{2}{\Delta^2} \ln(200r)$ times and let $\widetilde{\text{Inf}}_f(I_j)$ denote $2 \times$ (fraction of the M runs on I_j that the test rejects). If any subset I_j has $\widetilde{\text{Inf}}_f(I_j) \in [\alpha - 2\Delta, \alpha + 3\Delta]$ then exit and return "**no**," otherwise continue.
5. Let $\widetilde{L}(\alpha) \subseteq [r]$ denote $\{j \in [r] : \widetilde{\text{Inf}}_f(I_j) < \alpha - 2\Delta < \alpha\}$ and let $\widetilde{H}(\alpha)$ denote $[r] \setminus \widetilde{L}(\alpha)$. Let $\widetilde{f}' : \{0,1\}^n \rightarrow \{-1,1\}$ denote the function $f|_{0 \leftarrow \cup_{j \in \widetilde{L}(\alpha)} I_j}$.
6. Draw a sample of $m \overset{\text{def}}{=} \frac{2}{\epsilon} \ln 12$ uniform random examples from $\{0,1\}^n$ and evaluate both $\widetilde{f}'$ and f on each of these examples. If f and $\widetilde{f}'$ disagree on any of the m examples then exit and return "**no**." If they agree on all examples then continue.
7. Run the learning algorithm **LearnPoly**$'(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100)$ from [55] using **SimMQ**$(f, \widetilde{H}(\alpha), \{I_j\}_{j \in \widetilde{H}(\alpha)}, \alpha, \Delta, z, \delta/Q(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100))$ to simulate each membership query on a string $z \in \{0,1\}^{|\widetilde{H}(\alpha)|}$ that **LearnPoly**$'$ makes. If **LearnPoly**$'$ returns "not s-sparse" then exit and return "**no**." Otherwise the algorithm terminates successfully; in this case return "**yes**."

[a] More precisely, we set $\tau = \epsilon/600, \Delta = \min\{\Delta_0, (\tau/8s^2)(\delta/\ln(2/\delta))\}$, $r = 4Cs/\Delta$ (for a suitable constant C from Theorem 64), where $\Delta_0 \overset{\text{def}}{=} \tau/(1600s^3 \log(8s^3/\tau))$ and $\delta \overset{\text{def}}{=} 1/\left(100s \log(8s^3/\tau) Q(s, s\log(8s^3/\tau), \epsilon/4, 1/100)\right)$

Figure 4-1: The algorithm **Test-Sparse-Poly**.

Algorithm **Set-High-Influence-Variable**$(f, I, \alpha, \Delta, b, \delta)$
Input: Black-box access to $f : \{0,1\}^n \rightarrow \{-1,1\}$; (α, Δ)-well-structured set $I \subseteq [n]$; bit $b \in \{0,1\}$; failure parameter δ.
Output: assignment $w \in \{0,1\}^I$ to the variables in I such that $w_i = b$ with probability $1 - \delta$

1. Draw x uniformly from $\{0,1\}^I$. Define $I^0 \stackrel{\text{def}}{=} \{j \in I : x_j = 0\}$ and $I^1 \stackrel{\text{def}}{=} \{j \in I : x_j = 1\}$.
2. Apply $c = \frac{2}{\alpha} \ln(\frac{2}{\delta})$ iterations of the *independence test* to (f, I^0). If any of the c iterations reject, mark I^0. Do the same for (f, I^1).
3. If both or neither of I^0 and I^1 are marked, stop and output "fail".
4. If I^b is marked then return the assignment $w = x$. Otherwise return the assignment $w = \overline{x}$ (the bitwise negation of x).

Figure 4-2: The subroutine **Set-High-Influence-Variable**.

algorithm defines a function $\widetilde{f}'$ which "zeroes out" all of the variables in all low-influence subsets. Step 6 of **Test-Sparse-Poly** checks that f is close to $\widetilde{f}'$

The final step of **Test-Sparse-Poly** is to run the algorithm **LearnPoly**$'$ of [55] to learn a sparse polynomial, which we call $\widetilde{f}''$, which is isomorphic to $\widetilde{f}'$ but is defined only over the high-influence variables of f (recall that if f is indeed s-sparse, there is at most one from each high-influence subset). The overall **Test-Sparse-Poly** algorithm accepts f if and only if **LearnPoly**$'$ successfully returns a final hypothesis (i.e. does not halt and output "fail"). The membership queries that the [55] algorithm requires are simulated using the **SimMQ** procedure, which in turn uses a subroutine called **Set-High-Influence-Variables**.

The procedure **Set-High-Influence-Variable (SHIV)** is presented in Figure 4-2. The idea of this procedure is that when it is run on a well-structured subset of variables I, it returns an assignment in which the high-influence variable is set to the desired bit value. Intuitively, the executions of the independence test in the procedure are used to determine whether the high-influence variable $i \in I$ is set to 0 or 1 under the assignment x. Depending on whether this setting agrees with the desired value, the algorithm either returns x or the bitwise negation of x (this is slightly different from **Construct-Sample**, the analogous subroutine from Chapter 3, which is content with a random x and thus never needs to negate coordinates).

```
Algorithm **SimMQ**$(f, H, \{I_j\}_{j \in H}, \alpha, \Delta, z, \delta)$
**Input:** Black-box access to $f : \{0,1\}^n \rightarrow \{-1,1\}$; subset $H \subseteq [r]$; disjoint subsets $\{I_j\}_{j \in H}$ of $[n]$; parameters $\alpha > \Delta$; string $z \in \{0,1\}^{|H|}$; failure probability $\delta$
**Output:** bit $b$ which, with probability $1 - \delta$ is the value of $f'$ on a random assignment $x$ in which each high-influence variable $i \in I_j$ $(j \in H)$ is set according to $z$

  1. For each $j \in H$, call **Set-High-Influence-Variable**$(f, I_j, \alpha, \Delta, z_j, \delta/|H|)$ and get back an assignment (call it $w^j$) to the variables in $I_j$.
  2. Construct $x \in \{0,1\}^n$ as follows: for each $j \in H$, set the variables in $I_j$ according to $w^j$. This defines $x_i$ for all $i \in \cup_{j \in H} I_j$. Set $x_i = 0$ for all other $i \in [n]$.
  3. Return $b = f(x)$.
```

Figure 4-3: The subroutine **SimMQ**.

Figure 4-3 gives the **SimMQ** procedure. When run on a function f and a collection $\{I_j\}_{j \in H}$ of disjoint well-structured subsets of variables, **SimMQ** takes as input a string z of length $|H|$ which specifies a desired setting for each high-influence variable in each I_j $(j \in H)$. **SimMQ** constructs a random assignment $x \in \{0,1\}^n$ such that the high-influence variable in each I_j $(j \in H)$ is set in the desired way in x, and it returns the value $f'(x)$.

4.4.1 Time and Query Complexity of Test-Sparse-Poly

As stated in Figure 4-1, the **Test-Sparse-Poly** algorithm runs **LearnPoly′**$(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100)$ using **SimMQ**$(f, \widetilde{H}(\alpha), \{I_j\}_{j \in \widetilde{H}(\alpha)}, \alpha, \Delta, z, 1/(100Q(s, |\widetilde{H}(\alpha)|, z, 1/100)))$ to simulate each membership query on an input string $z \in \{0,1\}^{|\widetilde{H}(\alpha)|}$. Thus the algorithm is being run over a domain of $|\widetilde{H}(\alpha)|$ variables. Since we certainly have $|\widetilde{H}(\alpha)| \leq r \leq \mathrm{poly}(s, \frac{1}{\epsilon})$, Corollary 63 gives that **LearnPoly′** makes at most $\mathrm{poly}(s, \frac{1}{\epsilon})$ many calls to **SimMQ**. From this point, by inspection of **SimMQ**, **SHIV** and **Test-Sparse-Poly**, it is straightforward to verify that **Test-Sparse-Poly** indeed makes $\mathrm{poly}(s, \frac{1}{\epsilon})$ many queries to f and runs in time $\mathrm{poly}(s, \frac{1}{\epsilon})$ as claimed in Theorem 59. Thus, to prove Theorem 59 it remains only to establish correctness of the test.

4.4.2 Sketch of completeness

The main tool behind our completeness argument is Theorem 64. Suppose f is indeed an s-sparse polynomial. Then Theorem 64 guarantees that a randomly chosen α will w.h.p. yield a "gap" such that subsets with a high-influence variable have influence above the gap, and subsets with no high-influence variable have influence below the gap. This means that the estimates of each subset's influence (obtained by the algorithm in step 4) are accurate enough to effectively separate the high-influence subsets from the low-influence ones in step 5. Thus, the function $\widetilde{f}'$ defined by the algorithm will w.h.p be equal to the function p' from Theorem 64.

Assuming that f is an s-sparse polynomial (and that $\widetilde{f}'$ is equal to p'), Theorem 64 additionally implies that the function $\widetilde{f}'$ will be close to the original function (so Step 6 will pass), that $\widetilde{f}'$ only depends on $\mathrm{poly}(s, 1/\epsilon)$ many variables, and that all of the subsets I_j that "survive" into $\widetilde{f}'$ are well-structured. As we show in section 4.6, this condition is sufficient to ensure that **SimMQ** can successfully simulate membership queries to $\widetilde{f}''$. Thus, for f an s-sparse polynomial, the **LearnPoly**′ algorithm can run successfully, and the test will accept.

4.4.3 Sketch of soundness

Here, we briefly argue that if **Test-Sparse-Poly** accepts f with high probability, then f must be close to some s-sparse polynomial (we give the full proof in section 4.7). Note that if f passes Step 4, then **Test-Sparse-Poly** must have obtained a partition of variables into "high-influence" subsets and "low-influence" subsets. If f passes Step 6, then it must moreover be the case that f is close to the function $\widetilde{f}'$ obtained by zeroing out the low-influence subsets.

In the last step, **Test-Sparse-Poly** attempts to run the **LearnPoly**′ algorithm using $\widetilde{f}'$ and the high-influence subsets; in the course of doing this, it makes calls to **SimMQ**. Since f could be an arbitrary function, we do not know whether each high-influence subset has at most one variable relevant to $\widetilde{f}'$ (as would be the case, by Theorem 64, if f were an s-sparse polynomial). However, we are able to show (Lemma 78) that, if with high probability all calls to the **SimMQ** routine are answered without its ever returning "fail," then $\widetilde{f}'$ must be close to a junta g whose relevant variables are the individual "highest-influence" variables in each of the high-influence subsets. Now, given that **LearnPoly′** halts successfully, it must be the case that it constructs a final hypothesis h that is itself an s-sparse polynomial and that agrees with many calls to **SimMQ** on random examples. Lemma 79 states that, in this event, h must be close to g, hence close to $\widetilde{f}'$, and hence close to f.

4.5 Proof of Theorem 64

In Section 4.5.1 we prove some useful preliminary lemmas about the influence of individual variables in sparse polynomials. In Section 4.5.2 we extend this analysis to get high-probability statements about influence of subsets $\{I_j\}_{j=1}^r$ in a random partition. We put the pieces together to finish the proof of Theorem 64 in Section 4.5.3.

Throughout this section the parameters τ, Δ, r and α are all as defined in Theorem 64.

4.5.1 The influence of variables in s-sparse polynomials

We start with a simple lemma stating that only a small number of variables can have large influence:

Lemma 65. *Let* $p : \{0,1\}^n \rightarrow \{-1,1\}$ *be an s-sparse polynomial. For any* $\delta > 0$*, there are at most* $s \log(2s/\delta)$ *many variables* x_i *that have* $\mathrm{Inf}_p(i) \geq \delta$.

Proof. Any variable x_i with $\mathrm{Inf}_p(i) \geq \delta$ must occur in some term of length at most $\log(2s/\delta)$. (Otherwise each occurrence of x_i would contribute less than δ/s to the influence of the i-th coordinate, and since there are at most s terms this would imply $\mathrm{Inf}_p(i) < s \cdot (\delta/s) = \delta$.) Since at most $s \log(2s/\delta)$ distinct variables can occur in terms of length at most $\log(2s/\delta)$, the lemma follows. □ □

Lemma 66. *With probability at least* $96/100$ *over the choice of* α*, no variable* x_i *has* $\mathrm{Inf}_p(i) \in [\alpha - 4\Delta, \alpha + 4\Delta]$.

Proof. The uniform random variable α has support $\mathcal{A}(\tau, \Delta)$ of size $\geq 50s\log(8s^3/\tau)$. Each possible value of α defines the interval of influences $[\alpha - 4\Delta, \alpha + 4\Delta]$. Note that $\alpha - 4\Delta \geq \tau/(4s^2)$. In other words, the only variables which could lie in $[\alpha - 4\Delta, \alpha + 4\Delta]$ are those with influence at least $\tau/(4s^2)$. By Lemma 65 there are at most $k \overset{\text{def}}{=} s\log(8s^3/\tau)$ such candidate variables. Since we have at least $50k$ intervals (two consecutive such intervals overlap at a single point) and at most k candidate variables, by the pigeonhole principle, at least $48k$ intervals will be empty. □ □

Lemma 65 is based on the observation that, in a sparse polynomial, a variable with "high" influence (influence) must occur in some "short" term. The following lemma is in some sense a quantitative converse: it states that a variable with "small" influence can only appear in "long" terms.

Lemma 67. *Let* $p : \{0,1\}^n \to \{-1,1\}$ *be an s-sparse polynomial. Suppose that i is such that* $\mathrm{Inf}_p(i) < \tau/(s^2+s)$. *Then the variable* x_i *appears only in terms of length greater than* $\log(s/\tau)$.

Proof. By contradiction. Assuming that x_i appears in some term of length at most $\log(s/\tau)$, we will show that $\mathrm{Inf}_p(i) \geq \tau/(s^2+s)$. Let T be a shortest term that x_i appears in. The function p can be uniquely decomposed as follows: $p(x_1, x_2, \dots, x_n) = x_i \cdot (T' + p_1) + p_2$, where $T = x_i \cdot T'$, the term T' has length less than $\log(s/\tau)$ and does not depend on x_i, and p_1, p_2 are s-sparse polynomials that do not depend on x_i. Observe that since T is a shortest term that contains x_i, the polynomial p_1 does not contain the constant term 1.

Since T' contains fewer than $\log(s/\tau)$ many variables, it evaluates to 1 on at least a τ/s fraction of all inputs. The partial assignment that sets all the variables in T' to 1 induces an s-sparse polynomial p_1' (the restriction of p_1 according to the partial assignment). Now observe that p_1' still does not contain the constant term 1 (for since each term in p_1 is of length at least the length of T', no term in p_1 is a subset of the variables in T'). We now recall the following (nontrivial) result of Karpinski and Luby [36]:

Claim 68 ([36], Corollary 1). *Let g be an s-sparse multivariate* $GF(2)$ *polynomial which does not contain the constant-*1 *term. Then* $g(x) = 0$ *for at least a* $1/(s+1)$ *fraction of all inputs.*

Applying this corollary to the polynomial p_1', we have that p_1' is 0 on at least a $1/(s+1)$ fraction of its inputs. Therefore, the polynomial $T' + p_1$ is 1 on at least a $(\tau/s) \cdot 1/(s+1)$ fraction of all inputs in $\{0,1\}^n$; this in turn implies that $\mathrm{Inf}_p(i) \geq (\tau/s) \cdot 1/(s+1) = \tau/(s^2+s)$. □ □

By a simple application of Lemma 67 we can show that setting low-influence variables to zero does not change the polynomial by much:

Lemma 69. *Let $p : \{0,1\}^n \to \{-1,1\}$ be an s-sparse polynomial. Let g be a function obtained from p by setting to 0 some subset of variables all of which have $\mathrm{Inf}_p(i) < \tau/(2s^2)$. Then g and p are τ-close.*

Proof. Setting a variable to 0 removes all the terms that contain it from p. By Lemma 67, doing this only removes terms of length greater than $\log(s/\tau)$. Removing one such term changes the function on at most a τ/s fraction of the inputs. Since there are at most s terms in total, the lemma follows by a union bound. □ □

4.5.2 Partitioning variables into random subsets

The following lemma is at the heart of Theorem 64. The lemma states that when we randomly partition the variables (coordinates) into subsets, (*i*) each subset gets at most one "high-influence" variable (the term "high-influence" here means relative to an appropriate threshold value $t \ll \alpha$), and (*ii*) the remaining (low-influence) variables (w.r.t. t) have a "very small" contribution to the subset's total influence.

The first part of the lemma follows easily from a birthday–paradox type argument, since there are many more subsets than high-influence variables. As intuition for the second part, we note that in expectation, the total influence of each subset is very small. A more careful argument lets us argue that the total contribution of the low-influence variables in a given subset is unlikely to highly exceed its expectation.

Lemma 70. *Fix a value of α satisfying the first statement of Theorem 64. Let $t \overset{\text{def}}{=} \Delta\tau/(4C's)$, where C' is a suitably large constant. Then with probability $99/100$ over the random partition the following statements hold true:*

- *For every $j \in [r]$, I_j contains at most one variable x_i with* $\mathrm{Inf}_p(i) > t$.

- *Let $I_j^{\leq t} \stackrel{\mathrm{def}}{=} \{i \in I_j \mid \mathrm{Inf}_p(i) \leq t\}$. Then, for all $j \in [r]$,* $\mathrm{Inf}_p(I_j^{\leq t}) \leq \Delta$.

Proof. We show that each statement of the lemma fails independently with probability at most $1/200$ from which the lemma follows.

By Lemma 65 there are at most $b = s\log(2s/t)$ coordinates in $[n]$ with influence more than t. A standard argument yields that the probability there exists a subset I_j with more than one such variable is at most b^2/r. It is easy to verify that this is less than $1/200$, as long as C is large enough relative to C'. Therefore, with probability at least $199/200$, every subset contains at most one variable with influence greater than t. So the first statement fails with probability no more than $1/200$.

Now for the second statement. Consider a fixed subset I_j. We analyze the contribution of variables in $I_j^{\leq t}$ to the total influence $\mathrm{Inf}_p(I_j)$. We will show that with high probability the contribution of these variables is at most Δ.

Let $S = \{i \in [n] \mid \mathrm{Inf}_p(i) \leq t\}$ and renumber the coordinates such that $S = [k']$. Each variable x_i, $i \in S$, is contained in I_j independently with probability $1/r$. Let $X_1, \ldots, X_{k'}$ be the corresponding independent Bernoulli random variables. Recall that, by sub-additivity, the influence of $I_j^{\leq t}$ is upper bounded by $X = \sum_{i=1}^{k'} \mathrm{Inf}_p(i) \cdot X_i$. It thus suffices to upper bound the probability $\Pr[X > \Delta]$. Note that $\mathbb{E}[X] = \sum_{i=1}^{k'} \mathrm{Inf}_p(i) \cdot \mathbb{E}[X_i] = (1/r) \cdot \sum_{i=1}^{k'} \mathrm{Inf}_p(i) \leq (s/r)$, since $\sum_{i=1}^{k'} \mathrm{Inf}_p(i) \leq \sum_{i=1}^{n} \mathrm{Inf}_p(i) \leq s$. The last inequality follows from the following simple fact (the proof of which is left for the reader).

Fact 71. *Let $p : \{0,1\}^n \to \{-1,1\}$ be an s-sparse polynomial. Then* $\sum_{i=1}^{n} \mathrm{Inf}_p(i) \leq s$.

To finish the proof, we need the following version of the Chernoff bound:

Fact 72 ([46]). *For $k' \in \mathbb{N}^*$, let $\alpha_1, \ldots, \alpha_{k'} \in [0,1]$ and let $X_1, \ldots, X_{k'}$ be independent Bernoulli trials. Let $X' = \sum_{i=1}^{k'} \alpha_i X_i$ and $\mu \stackrel{\mathrm{def}}{=} \mathbb{E}[X'] \geq 0$. Then for any $\gamma > 1$ we have* $\Pr[X' > \gamma \cdot \mu] < (\frac{e^{\gamma-1}}{\gamma^\gamma})^\mu$.

We apply the above bound for the X_i's with $\alpha_i = \mathrm{Inf}_p(i)/t \in [0,1]$. (Recall that the coordinates in S have influence at most t.) We have $\mu = \mathbb{E}[X'] = \mathbb{E}[X]/t \leq s/(rt) = C's/C\tau$, and we are interested in the event $\{X > \Delta\} \equiv \{X' > \Delta/t\}$. Note that $\Delta/t =$

$4C's/\tau$. Hence, $\gamma \geq 4C$ and the above bound implies that $\Pr[X > \Delta] < \left(e/(4C)\right)^{4C's/\tau} < (1/4C^4)^{C's/\tau}$.

Therefore, for a fixed subset I_j, we have $\Pr[\mathrm{Inf}_p(I_j^{\leq t}) > \Delta] < (1/4C^4)^{C's/\tau}$. By a union bound, we conclude that this happens in every subset with failure probability at most $r \cdot (1/4C^4)^{C's/\tau}$. This is less than $1/200$ as long as C' is a large enough absolute constant (independent of C), which completes the proof. □ □

Next we show that by "zeroing out" the variables in low-influence subsets, we are likely to "kill" all terms in p that contain a low-influence variable.

Lemma 73. *With probability at least* $99/100$ *over the random partition, every monomial of* p *containing a variable with influence at most* α *has at least one of its variables in* $\cup_{j \in L(\alpha)} I_j$.

Proof. By Lemma 65 there are at most $b = s\log(8s^3/\tau)$ variables with influence more than α. Thus, no matter the partition, at most b subsets from $\{I_j\}_{j=1}^r$ contain such variables. Fix a low-influence variable (influence at most α) from every monomial containing such a variable. For each fixed variable, the probability that it ends up in the same subset as a high-influence variable is at most b/r. Union bounding over each of the (at most s) monomials, the failure probability of the lemma is upper bounded by $sb/r < 1/100$. □ □

4.5.3 Proof of Theorem 64

Proof. (Theorem 64) We prove each statement in turn. The first statement of the theorem is implied by Lemma 66. (Note that, as expected, the validity of this statement does not depend on the random partition.)

We claim that statements 2-5 essentially follow from Lemma 70. (In contrast, the validity of these statements crucially depends on the random partition.)

Let us first prove the third statement. We want to show that (w.h.p. over the choice of α and $\{I_j\}_{j=1}^r$) for every $j \in H(\alpha)$, (*i*) there exists a *unique* $i_j \in I_j$ such that $\mathrm{Inf}_p(i_j) \geq \alpha$ and (*ii*) that $\mathrm{Inf}_p(I_j \setminus \{i_j\}) \leq \Delta$. Fix some $j \in H(\alpha)$. By Lemma 70, for a given value of α satisfying the first statement of the theorem, we have: (*i'*) I_j contains at most one variable x_{i_j} with $\mathrm{Inf}_p(i_j) > t$ and *(ii')* $\mathrm{Inf}_p(I_j \setminus \{i_j\}) \leq \Delta$. Since $t < \tau/4s^2 < \alpha$ (with probability 1), *(i')* clearly implies that, if I_j has a high-influence element (w.r.t. α), then it is unique. In fact, we claim that $\mathrm{Inf}_p(i_j) \geq \alpha$. For otherwise, by sub-additivity of influence, we would have $\mathrm{Inf}_p(I_j) \leq \mathrm{Inf}_p(I_j \setminus \{i_j\}) + \mathrm{Inf}_p(i_j) \leq \Delta + \alpha - 4\Delta = \alpha - 3\Delta < \alpha$, which contradicts the assumption that $j \in H(\alpha)$. Note that we have used the fact that α satisfies the first statement of the theorem, that is $\mathrm{Inf}_p(i_j) < \alpha \Rightarrow \mathrm{Inf}_p(i_j) < \alpha - 4\Delta$. Hence, for a "good" value of α (one satisfying the first statement of the theorem), the third statement is satisfied with probability at least $99/100$ over the random partition. By Lemma 66, a "good" value of α is chosen with probability $96/100$. By independence, the conclusions of Lemma 66 and Lemma 70 hold simultaneously with probability more than $9/10$.

We now establish the second statement. We assume as before that α is a "good" value. Consider a fixed subset I_j, $j \in [r]$. If $j \in H(\alpha)$ (i.e. I_j is a high-influence subset) then, with probability at least $99/100$ (over the random partition), there exists $i_j \in I_j$ such that $\mathrm{Inf}_p(i_j) \geq \alpha + 4\Delta$. The monotonicity of influence yields $\mathrm{Inf}_p(I_j) \geq \mathrm{Inf}_p(i_j) \geq \alpha + 4\Delta$. If $j \in L(\alpha)$ then I_j contains no high-influence variable, i.e. its maximum influence element has influence at most $\alpha - 4\Delta$ and by the second part of Lemma 70 the remaining variables contribute at most Δ to its total influence. Hence, by sub-additivity we have that $\mathrm{Inf}_p(I_j) \leq \alpha - 3\Delta$. Since a "good" value of α is chosen with probability $96/100$, the desired statement follows.

The fourth statement follows from the aforementioned and the fact that there exist at most $s\log(8s^3/\tau)$ variables with influence at least α (as follows from Lemma 65, given that $\alpha > \tau/(4s^2)$).

Now for the fifth statement. Lemma 73 and monotonicity imply that the only variables that remain relevant in p' are (some of) those with high influence (at least α) in p, and, as argued above, each high-influence subset I_j contains at most one such variable. By a union bound, the conclusion of Lemma 73 holds simultaneously with the conclusions of Lemma 66 and Lemma 70 with probability at least $9/10$.

The sixth statement (that p and p' are τ-close) is a consequence of Lemma 69 (since p' is obtained from p by setting to 0 variables with influence less than $\alpha < \tau/(2s^2)$). This concludes the proof of Theorem 64. □ □

4.6 Completeness of the test

In this section we show that **Test-Sparse-Poly** is complete:

Theorem 74. *Suppose f is an s-sparse $GF(2)$ polynomial. Then* **Test-Sparse-Poly** *accepts f with probability at least* $2/3$.

Proof. Fix f to be an s-sparse $GF(2)$ polynomial over $\{0,1\}^n$. By the choice of the Δ and r parameters in Step 1 of **Test-Sparse-Poly** we may apply Theorem 64, so with failure probability at most $1/10$ over the choice of α and $I_1, \dots, I_r$ in Steps 2 and 3, statements 1–6 of Theorem 64 all hold. We shall write f' to denote $f|_{0 \leftarrow \cup_{j \in L(\alpha)} I_j}$. Note that at each successive stage of the proof we shall assume that the "failure probability" events do not occur, i.e. henceforth we shall assume that statements 1–6 all hold for f; we take a union bound over all failure probabilities at the end of the proof.

Now consider the M executions of the independence test for a given fixed I_j in Step 4. Lemma 11 gives that each run rejects with probability $\frac{1}{2}\mathrm{Inf}_f(I_j)$. A standard Hoeffding bound implies that for the algorithm's choice of $M = \frac{2}{\Delta^2}\ln(200r)$, the value $\widetilde{\mathrm{Inf}}_f(I_j)$ obtained in Step 4 is within $\pm\Delta$ of the true value $\mathrm{Inf}_f(I_j)$ with failure probability at most $\frac{1}{100r}$. A union bound over all $j \in [r]$ gives that with failure probability at most $1/100$, we have that each $\widetilde{\mathrm{Inf}}_f(I_j)$ is within an additive $\pm\Delta$ of the true value $\mathrm{Inf}_f(I_j)$. This means that (by statement 2 of Theorem 64) every I_j has $\widetilde{\mathrm{Inf}}_f(I_j) \notin [\alpha - 2\Delta, \alpha + 3\Delta]$, and hence in Step 5 of the test, the sets $\widetilde{L}(\alpha)$ and $\widetilde{H}(\alpha)$ are identical to $L(\alpha)$ and $H(\alpha)$ respectively, which in turn means that the function $\widetilde{f}'$ defined in Step 5 is identical to f' defined above.

We now turn to Step 6 of the test. By statement 6 of Theorem 64 we have that f and f' disagree on at most a τ fraction of inputs. A union bound over the m random examples drawn in Step 6 implies that with failure probability at most $\tau m < 1/100$ the test proceeds to Step 7.

By statement 3 of Theorem 64 we have that each I_j, $j \in \widetilde{H}(\alpha) \equiv H(\alpha)$, contains precisely one high-influence element i_j (i.e. which satisfies $\mathrm{Inf}_f(i_j) \geq \alpha$), and these are all of the high-influence elements. Consider the set of these $|\widetilde{H}(\alpha)|$ high-influence variables; statement 5 of Theorem 64 implies that these are the only variables which f' can depend on (it is possible that it does not depend on some of these variables). Let us write f'' to denote the function $f'' : \{0,1\}^{|\widetilde{H}(\alpha)|} \to \{-1,1\}$ corresponding to f' but whose input variables are these $|\widetilde{H}(\alpha)|$ high-influence variables in f, one per I_j for each $j \in \widetilde{H}(\alpha)$. We thus have that f'' is isomorphic to f' (obtained from f' by discarding irrelevant variables).

The main idea behind the completeness proof is that in Step 7 of **Test-Sparse-Poly**, the learning algorithm **LearnPoly$'$** is being run with target function f''. Since f'' is isomorphic to f', which is an s-sparse polynomial (since it is a restriction of an s-sparse polynomial f), with high probability **LearnPoly$'$** will run successfully and the test will accept. To show that this is what actually happens, we must show that with high probability each call to **SimMQ** which **LearnPoly$'$** makes correctly simulates the corresponding membership query to f''. This is established by the following lemmas:

Lemma 75. *Let f, I, α, Δ be such that I is (α, Δ)-well-structured with $\Delta \leq \alpha\delta/(2\ln(2/\delta))$. Then with probability at least $1-\delta$, the output of* **SHIV**$(f, I, \alpha, \Delta, b, \delta)$ *is an assignment $w \in \{0,1\}^I$ which has $w_i = b$.*

Proof. We assume that I^b contains the high-influence variable i (the other case being very similar). Recall that by Lemma 11, each run of the independence test on I^b rejects with probability $\frac{1}{2}\mathrm{Inf}_f(I^b)$; by Lemma 13 (monotonicity) this is at least $\frac{1}{2}\mathrm{Inf}_f(i) \geq \alpha/2$. So the probability that I^b is not marked even once after c iterations of the independence test is at most $(1-\alpha/2)^c \leq \delta/2$, by our choice of c. Similarly, the probability that $I^{\bar{b}}$ is ever marked during c iterations of the independence test is at most $c(\Delta/2) \leq \delta/2$, by the condition of the lemma. Thus, the probability of failing at step 3 of **SHIV** is at most δ, and since $i \in I^b$, the assignment w sets variable i correctly in step 4. □ □

Lemma 76. *With total failure probability at most $1/100$, all the $Q(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100)$ calls to* **SimMQ**$(f,\ \widetilde{H}(\alpha),\ \{I_j\}_{j\in\widetilde{H}(\alpha)},\ \alpha,\ \Delta,\ z,\ 1/(100Q(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100)))$ *that* **LearnPoly$'$** *makes in Step 7 of* **Test-Sparse-Poly** *return the correct value of $f''(z)$.*

Proof. Consider a single call to the procedure **SimMQ**$(f,\ \widetilde{H}(\alpha),\ \{I_j\}_{j\in\widetilde{H}(\alpha)},\ \alpha,\ \Delta,\ z,\ 1/(100Q(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100)))$ made by **LearnPoly$'$**. We show that with failure probability at most $\delta' \overset{\mathrm{def}}{=} 1/(100Q(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100)$ this call returns the value $f''(z)$, and the lemma then follows by a union bound over the $Q(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100)$ many calls to **SimMQ**.

This call to **SimMQ** makes $|\widetilde{H}(\alpha)|$ calls to **SHIV**$(f, I_j, \alpha, \Delta, z_j, \delta'/\widetilde{H}(\alpha)|)$, one for each $j \in \widetilde{H}(\alpha)$. Consider any fixed $j \in \widetilde{H}(\alpha)$. Statement 3 of Theorem 64 gives that I_j ($j \in \widetilde{H}(\alpha)$) is (α, Δ)-well-structured. Since $\alpha > \frac{\tau}{4s^2}$, it is easy to check the condition of Lemma 75 holds where the role of "δ" in that inequality is played by $\delta'/|\widetilde{H}(\alpha)|$, so we may apply Lemma 75 and conclude that with failure probability at most $\delta'/|\widetilde{H}(\alpha)|$ (recall that by statement 4 of Theorem 64 we have $|\widetilde{H}(\alpha)| \leq s \log(8s^3/\tau)$), **SHIV** returns an assignment to the variables in I_j which sets the high-influence variable to z_j as required. By a union bound, the overall failure probability that any I_j ($j \in \widetilde{H}(\alpha)$) has its high-influence variable not set according to z is at most δ'. Now statement 5 and the discussion preceding this lemma (the isomorphism between f' and f'') give that **SimMQ** sets all of the variables that are relevant in f' correctly according to z in the assignment x it constructs in Step 2. Since this assignment x sets all variables in $\cup_{j \in \widetilde{L}} I_j$ to 0, the bit $b = f(x)$ that is returned is the correct value of $f''(z)$, with failure probability at most δ' as required. □ □

With Lemma 76 in hand, we have that with failure probability at most $1/100$, the execution of **LearnPoly'**$(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100)$ in Step 7 of **Test-Sparse-Poly** correctly simulates all membership queries. As a consequence, Corollary 63 thus gives us that **LearnPoly'**$(s, |\widetilde{H}(\alpha)|, \epsilon/4, 1/100))$ returns "not s-sparse" with probability at most $1/100$. Summing all the failure probabilities over the entire execution of the algorithm, the overall probability that **Test-Sparse-Poly** does not output "yes" is at most

$$\overbrace{1/10}^{\text{Theorem 64}} + \overbrace{1/100}^{\text{Step 4}} + \overbrace{1/100}^{\text{Step 6}} + \overbrace{1/100}^{\text{Lemma 76}} + \overbrace{1/100}^{\text{Corollary 63}} < 1/5,$$

and the completeness theorem is proved. (Theorem 74) ■ □

4.7 Soundness of the Test

In this section we prove the soundness of **Test-Sparse-Poly**:

Theorem 77. *If f is ϵ-far from any s-sparse polynomial, then **Test-Sparse-Poly** accepts with probability at most* $1/3$.

Proof. To prove the soundness of the test, we start by assuming that the function f has progressed to step 5, so there are subsets $I_1, \ldots, I_r$ and $\widetilde{H}(\alpha)$ satisfying $\widetilde{\mathrm{Inf}}_f(I_j) > \alpha + 2\Delta$ for all $j \in \widetilde{H}(\alpha)$. As in the proof of completeness, we have that the actual influences of all subsets should be close to the estimates, i.e. that $\mathrm{Inf}_f(I_j) > \alpha + \Delta$ for all $j \in \widetilde{H}(\alpha)$ except with with probability at most $1/100$. We may then complete the proof in two parts by establishing the following:

- If f and $\widetilde{f}'$ are ϵ_a-far, step 6 will accept with probability at most δ_a.
- If $\widetilde{f}'$ is ϵ_b-far from every s-sparse polynomial, step 7 will accept with probability at most δ_b.

Establishing these statements with $\epsilon_a = \epsilon_b = \epsilon/2$, $\delta_a = 1/12$ and $\delta_b = 1/6$ will allow us to complete the proof (and we may assume throughout the rest of the proof that $\mathrm{Inf}_f(I_j) > \alpha$ for each $j \in \widetilde{H}(\alpha)$).

The first statement follows immediately by our choice of $m = \frac{1}{\epsilon_a} \ln \frac{1}{\delta_a}$ with $\epsilon_a = \epsilon/2$ and $\delta_a = 1/12$ in Step 6. Our main task is to establish the second statement, which we do using Lemmas 78 and 79 stated below. Intuitively, we would like to show that if **LearnPoly**$'$ outputs a hypothesis h (which must be an s-sparse polynomial since **LearnPoly**$'$ is proper) with probability greater than 1/6, then $\widetilde{f}'$ is close to a junta isomorphic to h. To do this, we establish that if **LearnPoly**$'$ succeeds with high probability, then the last hypothesis on which an equivalence query is performed in **LearnPoly**$'$ is a function which is close to $\widetilde{f}'$. Our proof uses two lemmas: Lemma 79 tells us that this holds if the high influence subsets satisfy a certain structure, and Lemma 78 tells us that if **LearnPoly**$'$ succeeds with high probability then the subsets indeed satisfy this structure. We now state these lemmas formally and complete the proof of the theorem, deferring the proofs of the lemmas until later.

Recall that the algorithm **LearnPoly**$'$ will make repeated calls to **SimMQ** which in turn makes repeated calls to **SHIV**. Lemma 78 states that if, with probability greater than δ_2, all of these calls to **SHIV** return without failure, then the subsets associated with $\widetilde{H}(\alpha)$ have a special structure.

Lemma 78. *Let $J \subset [n]$ be a subset of variables obtained by including the highest-influence element in I_j for each $j \in \widetilde{H}(\alpha)$ (breaking ties arbitrarily). Suppose that $k > 300|\widetilde{H}(\alpha)|/\epsilon_2$ queries are made to **SimMQ**. Suppose moreover that* $\Pr[$ *every call to **SHIV** that is made during these k queries returns without outputting 'fail'*$]$ *is greater than δ_2 for $\delta_2 = 1/\Omega(k)$. Then the following both hold:*

- *Every subset I_j for $j \in \widetilde{H}(\alpha)$ satisfies* $\mathrm{Inf}_f(I_j \setminus J) \leq 2\epsilon_2/|\widetilde{H}(\alpha)|$*; and*
- *The function $\widetilde{f}'$ is ϵ_2-close to the junta $g : \{0,1\}^{|\widetilde{H}(\alpha)|} \to \{-1,1\}$ defined as as:*

$$g(x) \stackrel{\text{def}}{=} \underset{z}{\mathbf{Plur}}[\widetilde{f}'(x_J z_{\overline{J}})].$$

Given that the subsets associated with $\widetilde{H}(\alpha)$ have this special structure, Lemma 79 tells us that the hypothesis output by **LearnPoly**$'$ should be close to the junta g.

Lemma 79. *Define Q_E as the maximum number of calls to **SimMQ** that that will be made by **LearnPoly**$'$ in all of its equivalence queries. Suppose that for every $j \in \widetilde{H}(\alpha)$, it holds that* $\mathrm{Inf}_f(I_j \setminus J) < 2\epsilon_2/|\widetilde{H}(\alpha)|$ *with $\epsilon_2 < \frac{\alpha}{800Q_E}$. Then the probability that **LearnPoly**$'$ outputs a hypothesis h which is $\epsilon/4$-far from the junta g is at most $\delta_3 = 1/100$.*

We now show that Lemmas 78 and 79 suffice to prove the desired result. Suppose that **LearnPoly**$'$ accepts with probability at least $\delta_b = 1/6$. Assume **LearnPoly**$'$ makes at least k queries to **SimMQ** (we address this in the next paragraph); then it follows from Lemma 78 that the bins associated with $\widetilde{H}(\alpha)$ satisfy the conditions of Lemma 79 and that $\widetilde{f}'$ is ϵ_2-close to the junta g. Now applying Lemma 79, we have that with failure probability at most $1/100$, **LearnPoly**$'$ outputs a hypothesis which is $\epsilon/4$-close to g. But then $\widetilde{f}'$ must be $(\epsilon_2 + \epsilon/4)$-close to this hypothesis, which is an s-sparse polynomial.

We need to establish that **LearnPoly**$'$ indeed makes $k > 300|\widetilde{H}(\alpha)|/\epsilon_2$ **SimMQ** queries for an ϵ_2 that satisfies the condition on ϵ_2 in Lemma 79. (Note that if **LearnPoly**$'$ does not actually make this many queries, we can simply have it make artificial calls to **SHIV** to achieve this. An easy extension of our completeness proof handles this slight extension of the algorithm; we omit the details.) Since we need $\epsilon_2 < \alpha/800Q_E$ and Theorem 62 gives us that $Q_E = (|\widetilde{H}(\alpha)|s + 2) \cdot \frac{4}{\epsilon} \ln 300(|\widetilde{H}(\alpha)|s + 2)$ (each equivalence query is simulated using $\frac{4}{\epsilon}\ln 300(|\widetilde{H}(\alpha)|s+2)$ random examples), an easy computation shows that it suffices to take $k = \mathrm{poly}(s, 1/\epsilon)$, and the proof of Theorem 77 is complete. □ □

Before proving Lemma 79 and Lemma 78, we prove the following about the behavior of **SHIV** when it is called with parameters α, Δ that do not quite match the real values α', Δ' for which I is (α', Δ')-well-structured:

Lemma 80. *If I is (α', Δ')-well-structured, then the probability that* ***SHIV***$(f, I, \alpha, \Delta, b, \delta)$ *passes (i.e. does not output "fail") and sets the high influence variable incorrectly is at most* $(\delta/2)^{\alpha'/\alpha} \cdot (1/\alpha) \cdot \Delta' \cdot \ln(2/\delta)$.

Proof. The only way for **SHIV** to pass with an incorrect setting of the high-influence variable i is if it fails to mark the subset containing i for c iterations of the independence test, and marks the other subset at least once. Since $Vr(i) > \alpha'$ and $Vr(I \setminus i) < \Delta'$, the probability of this occurring is at most $(1-\alpha'/2)^c \cdot \Delta' \cdot c/2$. Since **SHIV** is called with failure parameter δ, c is set to $\frac{2}{\alpha}\ln\frac{2}{\delta}$. □ □

We now give a proof of Lemma 79, followed by a proof of Lemma 78.

Proof. (Lemma 79) By assumption each $\mathrm{Inf}_f(I_j \setminus J) \leq 2\epsilon_2/|\widetilde{H}(\alpha)|$ and $\mathrm{Inf}_f(I_j) > \alpha$, so subadditivity of influence gives us that for each $j \in \widetilde{H}(\alpha)$, there exists an $i \in I_j$ such that $\mathrm{Inf}_f(i) > \alpha - 2\epsilon_2/|\widetilde{H}(\alpha)|$. Thus for every each call to **SHIV** made by **SimMQ**, the conditions of Lemma 80 are satisfied with $\mathrm{Inf}_f(i) > \alpha - 2\epsilon_2/|\widetilde{H}(\alpha)|$ and $\mathrm{Inf}_f(I_j \setminus J) < 2\epsilon_2/|\widetilde{H}(\alpha)|$. We show that as long as $\epsilon_2 < \frac{\alpha}{800Q_E}$, the probability that any particular query z to **SimMQ** has a variable set incorrectly is at most $\delta_3/3Q_E$.

Suppose **SHIV** has been called with failure probability δ_4, then the probability given by Lemma 80 is at most:

$$(\delta_4/2)^{1-2\epsilon_2/(\alpha\cdot|\widetilde{H}(\alpha)|)} \cdot \frac{2}{\alpha}\ln\left(\frac{2}{\delta_4}\right) \cdot 2\epsilon_2/|\widetilde{H}(\alpha)|, \tag{4.1}$$

We shall show that this is at most $\delta_3/3|\widetilde{H}(\alpha)|Q_E = 1/300Q_E|\widetilde{H}(\alpha)|$. Taking $\epsilon_2 \leq$

$\alpha/800Q_E$ simplifies (4.1) to:

$$\frac{1}{300Q_E|\widetilde{H}(\alpha)|}\cdot(\delta_4/2)^{1-2\epsilon_2/(\alpha\cdot|\widetilde{H}(\alpha)|)}\cdot\frac{3}{4}\ln\frac{2}{\delta_4},$$

which is at most $1/300|\widetilde{H}(\alpha)|Q_E$ as long as

$$(2/\delta_4)^{1-2\epsilon_2/(\alpha\cdot|\widetilde{H}(\alpha)|)}>\frac{3}{4}\ln\frac{2}{\delta_4},$$

which certainly holds for our choice of ϵ_2 and the setting of $\delta_4 = 1/100k|\widetilde{H}(\alpha)|$. Each call to **SimMQ** uses $|\widetilde{H}(\alpha)|$ calls to **SHIV**, so a union bound gives that each random query to **SimMQ** returns an incorrect assignment with probability at most $1/300Q_E$.

Now, since $\widetilde{f}'$ and g are ϵ_2-close and ϵ_2 satisfies $\epsilon_2 Q_E \leq \delta_3/3$, in the uniform random samples used to simulate the final (accepting) equivalence query, **LearnPoly**$'$ will receive examples labeled correctly according to g with probability at least $1-2\delta_3/3$. Finally, note that **LearnPoly**$'$ makes at most $|\widetilde{H}(\alpha)|s+2$ equivalence queries and hence each query is simulated using $\frac{4}{\epsilon}\ln\frac{3(|\widetilde{H}(\alpha)|s+2)}{\delta_3}$ random examples (for a failure probability of $\frac{\delta_3}{|\widetilde{H}(\alpha)|s+2}$ for each equivalence query). Then **LearnPoly**$'$ will reject with probability at least $1-\delta_3/3$ unless g and h are $\epsilon/4$-close. This concludes the proof of Lemma 79. □ □

Proof. (Lemma 78) We prove that if $\mathrm{Inf}_f(I_j\setminus J) > 2\epsilon_2/|\widetilde{H}(\alpha)|$ for some $j\in\widetilde{H}(\alpha)$, then the probability that all calls to **SHIV** return successfully is at most δ_2. The closeness of $\widetilde{f}'$ and g follows easily by the subadditivity of influence and Proposition 3.2 of [26].

First, we prove a much weaker statement whose analysis and conclusion will be used to prove the proposition. We show in Proposition 81 that if the test accepts with high probability, then the influence from each variable in any subset is small. We use the bound on each variable's influence to obtain the concentration result in Proposition 82, and then complete the proof of Lemma 78.

Proposition 81. *Suppose that k calls to **SHIV** are made with a particular subset I, and let i be the variable with the highest influence in I. If $\mathrm{Inf}_f(j) > \epsilon_2/100|\widetilde{H}(\alpha)|$ for some $j\in I\setminus i$, then the probability that **SHIV** returns without outputting 'fail' for all k calls is at most $\delta^* = e^{-k/18}+e^{-c}$.*

Proof. Suppose that there exist $j, j' \in I$ with $\mathrm{Inf}_f(j) \geq \mathrm{Inf}_f(j') \geq \epsilon_2/100|\widetilde{H}(\alpha)|$. A standard Chernoff bound gives that except with probability at most $e^{-k/18}$, for at least $(1/3)k$ of the calls to **SHIV**, variables j and j' are in different partitions. In these cases, the probability **SHIV** does not output 'fail' is at most $2(1 - \epsilon_2/100|\widetilde{H}(\alpha)|)^c$, since for each of the c runs of the independence test, one of the partitions must not be marked. The probability no call outputs 'fail' is at most $e^{-k/18} + 2(1 - \epsilon_2/100|\widetilde{H}(\alpha)|)^{ck/3}$. Our choice of $k > 300|\widetilde{H}(\alpha)|/\epsilon_2$ ensures that $(1/e)^{ck\epsilon_2/300|\widetilde{H}(\alpha)|} \leq (1/e)^c$. □ □

Since in our setting $|I_j|$ may depend on n, using the monotonicity of influence with the previous claim does not give a useful bound on $\mathrm{Inf}_f(I \backslash i)$. But we see from the proof that if the influence of each partition is not much less than $\mathrm{Inf}_f(I \backslash i)$ and $\mathrm{Inf}_f(I \backslash i) > 2\epsilon_2/|\widetilde{H}(\alpha)|$, then with enough calls to **SHIV** one of these calls should output "fail." Hence the lemma will be easily proven once we establish the following proposition:

Proposition 82. *Suppose that k calls to **SHIV** are made with a particular subset I having $\mathrm{Inf}_f(I \setminus i) > 2\epsilon_2/|\widetilde{H}(\alpha)|$ and $\mathrm{Inf}_f(j) \leq \epsilon_2/100|\widetilde{H}(\alpha)|$ for every $j \in I \setminus i$. Then with probability greater than $1 - \delta^{**} = 1 - e^{-k/18}$, at least $1/3$ of the k calls to **SHIV** yield both $\mathrm{Inf}_f(I^1) > \eta \mathrm{Inf}_f(I \setminus i)/2$ and $\mathrm{Inf}_f(I^0) > \eta \mathrm{Inf}_f(I \setminus i)/2$, where $\eta = 1/e - 1/50$.*

Proof. We would like to show that a random partition of I into two parts will result in parts each of which has influence not much less than the influence of $I \setminus i$. Choosing a partition is equivalent to choosing a random subset I' of $I \backslash i$ and including i in I' or $I \backslash I'$ with equal probability. Thus it suffices to show that for random $I' \subseteq I \setminus i$, it is unlikely that $\mathrm{Inf}_f(I')$ is much smaller than $\mathrm{Inf}_f(I \setminus i)$.

This does not hold for general I, but by bounding the influence of any particular variable in I, which we have done in Proposition 81, and computing the *unique variation* (see Definition 27 from Chapter 3) of I', we may obtain a deviation bound on $\mathrm{Inf}_f(I')$.

Now $\mathrm{Inf}_f(I')$ is lower bounded by a sum of independent, non-negative random variables whose expectation is given by

$$\mathbb{E}[\sum_{j \in I'} \mathrm{Ur}_f(j)] = \sum_{j=1}^{n} (1/2)\mathrm{Ur}_f(j) = \mathrm{Inf}_f(I \setminus i)/2 \stackrel{\text{def}}{=} \mu.$$

To obtain a concentration property, we require a bound on each $\mathrm{Ur}_f(j) \leq \mathrm{Inf}_f(j)$, which is precisely what we showed in the previous proposition. Note that $\mathrm{Ur}_f(i) = 0$, and recall that we have assumed that $\mu > \epsilon_2/|\widetilde{H}(\alpha)|$ and every $j \in I \setminus i$ satisfies $\mathrm{Inf}_f(j) < \mu/100$.

Now we may use the bound from [26] in Proposition 3.5 with $\eta = 1/e - 2/100$ to obtain:

$$\Pr[\sum_{j \in I'} \mathrm{Ur}_f(j) < \eta\mu] < \exp(\frac{100}{e}(\eta e - 1)))] \leq 1/e^2.$$

Thus the probability that one of I^0 and I^1 has influence less than $\eta\mu$ is at most $1/2$. We expect that half of the k calls to **SHIV** will result in I^0 and I^1 having influence at least $\eta\mu$, so a Chernoff bound completes the proof of the claim with $\delta^{**} \leq e^{-k/18}$. This concludes the proof of Proposition 82. □ □

Finally, we proceed to prove the lemma. Suppose that there exists some I such that $\mathrm{Inf}_f(I \setminus i) > 2\epsilon_2/|\widetilde{H}(\alpha)|$. Now the probability that a particular call to **SHIV** with subset I succeeds is:

$$\Pr[\mathrm{marked}(I^0); \neg\,\mathrm{marked}(I^1)] + \Pr[\mathrm{marked}(I^1); \neg\,\mathrm{marked}(I^0)].$$

By Propositions 81 and 82, if with probability at least $\delta^* + \delta^{**}$ none of the k calls to **SHIV** return fail, then for $k/3$ runs of **SHIV** both $\mathrm{Inf}_f(I^1)$ and $\mathrm{Inf}_f(I^0)$ are at least $\eta\epsilon_2/|\widetilde{H}(\alpha)| > \epsilon_2/4|\widetilde{H}(\alpha)|$ and thus both probabilities are at most $(1 - \epsilon_2/4|\widetilde{H}(\alpha)|)^c$.

As in the analysis of the first proposition, we may conclude that every subset I which is called with **SHIV** at least k times either satisfies $\mathrm{Inf}_f(I \setminus i) < 2\epsilon_2/|\widetilde{H}(\alpha)|$ or will cause the test to reject with probability at least $1 - \delta^{**} - 2\delta^*$. Recall that $\delta^* = e^{-c} + e^{-k/18}$; since **SHIV** is set to run with failure probability at most $1/|\widetilde{H}(\alpha)|k$, we have that δ_2 is $1/\Omega(k)$. This concludes the proof of Lemma 78. □ □

4.8 Conclusion and future directions

An obvious question raised by this work is whether similar methods can be used to efficiently test s-sparse polynomials over a general finite field $\mathbb{F}$, with query and time complexity polynomial in s, $1/\epsilon$, and $|\mathbb{F}|$. The algorithm from Chapter 3 uses $\tilde{O}((s|\mathbb{F}|)^4/\epsilon^2)$ queries to test s-sparse polynomials over $\mathbb{F}$, but has running time $2^{\omega(s|\mathbb{F}|)} \cdot (1/\epsilon)^{\log\log(1/\epsilon)}$ (arising, as discussed in Section 4.1, from brute-force search for a consistent hypothesis.). One might hope to improve that algorithm by using techniques from this chapter. However, doing so requires an algorithm for properly learning s-sparse polynomials over general finite fields. To the best of our knowledge, the most efficient algorithm for doing this (given only black-box access to $f : \mathbb{F}^n \rightarrow \mathbb{F}$) is the algorithm of Bshouty [11] which requires $m = s^{O(|\mathbb{F}| \log |\mathbb{F}|)} \log n$ queries and runs in $\mathrm{poly}(m, n)$ time. (Other learning algorithms are known which do not have this exponential dependence on $|\mathbb{F}|$, but they either require evaluating the polynomial at complex roots of unity [42] or on inputs belonging to an extension field of $\mathbb{F}$ [31, 35].) It would be interesting to know whether there is a testing algorithm that simultaneously achieves a polynomial runtime (and hence query complexity) dependence on both the size parameter s and the cardinality of the field $|\mathbb{F}|$.

Another goal for future work is to apply our methods to other classes beyond just polynomials. Is it possible to combine the "testing by implicit learning" approach with other membership-query-based learning algorithms, to achieve time and query efficient testers for other natural classes?

Chapter 5

Testing Halfspaces

5.1 Introduction

A *halfspace* is a function of the form $f(x) = \mathrm{sgn}(w_1 x_1 + \cdots + w_n x_n - \theta)$. Halfspaces are also known as *threshold functions* or *linear threshold functions*; for brevity we shall often refer to them here as LTFs. More formally, we have the following:

Definition 83. *A "linear threshold function," or LTF, is a Boolean-valued function of the form $f(x) = \mathrm{sgn}(w_1 x_1 + ... + w_n x_n - \theta)$ where $w_1, ..., w_n, \theta \in \mathbb{R}$. The w_i's are called "weights," and θ is called the "threshold." The* sgn *function is* 1 *on arguments* ≥ 0, *and* -1 *otherwise.*

LTFs are a simple yet powerful class of functions, which for decades have played an important role in fields such as complexity theory, optimization, and machine learning (see e.g. [32, 64, 7, 49, 45, 57]).

In this chapter, we focus on the *halfspace testing* problem: given query access to a function, we would like to distinguish whether it is an LTF or whether it is ϵ-far from any LTF. Our main result is to show that the halfspace testing problem can be solved with a number of queries that is *independent* of n. In doing so, we establish new structural results about LTFs which essentially characterize LTFs in terms of their degree-0 and degree-1 Fourier coefficients.

We note that any learning algorithm — even one with black-box query access to f —

must make at least $\Omega(\frac{n}{\epsilon})$ queries to learn an unknown LTF to accuracy ϵ under the uniform distribution on $\{-1,1\}^n$ (this follows easily from, e.g., the results of [40]). Thus the complexity of learning is linear in n, as opposed to our testing bounds which are independent of n.

We start by describing our testing results in more detail.

Our Results. We consider the standard property testing model, in which the testing algorithm is allowed black-box query access to an unknown function f and must minimize the number of times it queries f. The algorithm must with high probability pass all functions that have the property and with high probability fail all functions that have distance at least ϵ from any function with the property. Our main algorithmic results are the following:

1. We first consider functions that map $\mathbb{R}^n \rightarrow \{-1,1\}$, where we measure the distance between functions with respect to the standard n-dimensional Gaussian distribution. In this setting we give a $\mathrm{poly}(\frac{1}{\epsilon})$ query algorithm for testing LTFs with two-sided error.

2. [Main Result.] We next consider functions that map $\{-1,1\}^n \rightarrow \{-1,1\}$, where (as is standard in property testing) we measure the distance between functions with respect to the uniform distribution over $\{-1,1\}^n$. In this setting we also give a $\mathrm{poly}(\frac{1}{\epsilon})$ query algorithm for testing LTFs with two-sided error.

Results 1 and 2 show that in two natural settings we can test a highly geometric property — whether or not the -1 and $+1$ values defined by f are linearly separable — with a number of queries that is independent of the dimension of the space. Moreover, the dependence on $\frac{1}{\epsilon}$ is only polynomial, rather than exponential or tower-type as in some other property testing algorithms.

While it is slightly unusual to consider property testing under the standard multivariate Gaussian distribution, we remark that our results are much simpler to establish in this setting because the rotational invariance essentially means that we can deal with a 1-dimensional problem. We moreover observe that it seems essentially *necessary* to solve the LTF testing problem in the Gaussian domain in order to solve the problem in the standard

$\{-1,1\}^n$ uniform distribution framework; to see this, observe that an unknown function $f : \{-1,1\}^n \to \{-1,1\}$ to be tested could in fact have the structure

$$f(x_1,\ldots,x_{dm}) = \tilde{f}\left(\frac{x_1+\cdots+x_m}{\sqrt{m}},\ldots,\frac{x_{(d-1)m+1}+\cdots+x_{dm}}{\sqrt{m}}\right),$$

in which case the arguments to $\tilde{f}$ behave very much like d independent standard Gaussian random variables.

We note that the assumption that our testing algorithm has query access to f (as opposed to, say, access only to random labeled examples) is necessary to achieve a complexity independent of n. Any LTF testing algorithm with access only to uniform random examples $(x, f(x))$ for $f : \{-1,1\}^n \to \{-1,1\}$ must use at least $\Omega(\log n)$ examples (an easy argument shows that with fewer examples, the distribution on examples labeled according to a truly random function is statistically indistinguishable from the distribution on examples labeled according to a randomly chosen variable from $\{x_1,\ldots,x_n\}$).

Characterizations and Techniques. In Chapter 3, we described the "implicit learning" approach to testing. However, that approach does not apply directly to LTFs, since in general implicit learning requires that the functions in question be "well approximated" by juntas, and LTFs clearly are not. To test LTFs we must take a new tack.

We establish new structural results about LTFs which essentially characterize LTFs in terms of their degree-0 and degree-1 Fourier coefficients. For functions mapping $\{-1,1\}^n$ to $\{-1,1\}$ it has long been known [14] that any linear threshold function f is *completely specified* by the $n+1$ parameters consisting of its degree-0 and degree-1 Fourier coefficients (also referred to as its *Chow parameters*). While this specification has been used to *learn* LTFs in various contexts [5, 28, 56], it is not clear how it can be used to construct efficient *testers* (for one thing this specification involves $n+1$ parameters, and in testing we want a query complexity independent of n). Intuitively, we get around this difficulty by giving new characterizations of LTFs as those functions that satisfy a particular relationship between just *two* parameters, namely the degree-0 Fourier coefficient and the sum of the squared degree-1 Fourier coefficients. Moreover, our characterizations are robust in that if a function approximately satisfies the relationship, then it must be close to an LTF. This is what makes the characterizations useful for testing.

In Section 5.3 we consider functions mapping $\mathbb{R}^n$ to $\{-1, 1\}$, where we view $\mathbb{R}^n$ as endowed with the standard n-dimensional Gaussian distribution. Our characterization is particularly clean in this setting and illustrates the essential approach that also underlies the much more involved Boolean case. On one hand, it is not hard to show that for every LTF f, the sum of the squares of the degree-1 Hermite coefficients of f is equal to a particular function of the mean of f — regardless of *which* LTF f is. We call this function W; it is essentially the square of the "Gaussian isoperimetric" function.

Conversely, Theorem 100 shows that if $f : \mathbb{R}^n \to \{-1, 1\}$ is *any* function for which the sum of the squares of the degree-1 Hermite coefficients is within $\pm\epsilon^3$ of $W(\mathbf{E}[f])$, then f must be $O(\epsilon)$-close to an LTF — in fact to an LTF whose n weights are the n degree-1 Hermite coefficients of f. The value $\mathbf{E}[f]$ can clearly be estimated by sampling, and moreover it can be shown that a simple approach of sampling f on pairs of correlated inputs can be used to obtain an accurate estimate of the sum of the squares of the degree-1 Hermite coefficients. We thus obtain a simple and efficient test for LTFs under the Gaussian distribution and thereby establish Result 1.

In Section 5.4 we take a step toward handling general LTFs over $\{-1, 1\}^n$ by developing an analogous characterization and testing algorithm for the class of *balanced regular* LTFs over $\{-1, 1\}^n$; these are LTFs with $\mathbf{E}[f] = 0$ for which all degree-1 Fourier coefficients are small. The heart of this characterization is a pair of results, Theorems 112 and 113, which give Boolean-cube analogues of our characterization of Gaussian LTFs. Theorem 112 states that the sum of the squares of the degree-1 Fourier coefficients of any balanced regular LTF is approximately $W(0) = \frac{2}{\pi}$. Theorem 113 states that any function f whose degree-1 Fourier coefficients are all small and whose squares sum to roughly $\frac{2}{\pi}$ is in fact close to an LTF — in fact, to one whose weights are the degree-1 Fourier coefficients of f. Similar to the Gaussian setting, we can estimate $\mathbf{E}[f]$ by uniform sampling and can estimate the sum of squares of degree-1 Fourier coefficients by sampling f on pairs of correlated inputs. An additional algorithmic step is also required here, namely checking that all the degree-1 Fourier coefficients of f are indeed small; it turns out that this can be done by estimating the sum of *fourth* powers of the degree-1 Fourier coefficients, which can again be obtained by sampling f on (4-tuples of) correlated inputs.

The general case of testing arbitrary LTFs over $\{-1,1\}^n$ is substantially more complex and is dealt with in Section 5.5. Very roughly speaking, the algorithm has three main conceptual steps:

- First the algorithm implicitly identifies a set of $O(1)$ many variables that have "large" degree-1 Fourier coefficients. Even a single such variable cannot be explicitly identified using $o(\log n)$ queries; we perform the implicit identification using $O(1)$ queries by adapting an algorithmic technique from [25]. This is similar to the implicit learning approach from Chapter 3.

- Second, the algorithm analyzes the regular subfunctions that are obtained by restricting these implicitly identified variables; in particular, it checks that there is a single set of weights for the unrestricted variables such that the different restrictions can all be expressed as LTFs with these weights (but different thresholds) over the unrestricted variables. Roughly speaking, this is done using a generalized version of the regular LTF test that tests whether a *pair* of functions are close to LTFs over the same linear form but with different thresholds. The key technical ingredients enabling this are Theorems 127 and 128, which generalize Theorems 112 and 113 in two ways (to pairs of functions, and to functions which may have nonzero expectation).

- Finally, the algorithm checks that there exists a single set of weights for the restricted variables that is compatible with the different biases of the different restricted functions. If this is the case then the overall function is close to the LTF obtained by combining these two sets of weights for the unrestricted and restricted variables. (Intuitively, since there are only $O(1)$ restricted variables there are only $O(1)$ possible sets of weights to check here.)

Outline of the Chapter. In Section 5.2 we describe a subroutine for estimating sums of powers of Fourier and Hermite coefficients, based on the notion of Noise Stability. Section 5.3 contains our algorithm for testing general LTFs over Gaussian Space. Section 5.4 contains an algorithm for testing *balanced, regular* LTFs over $\{-1,1\}^n$, a "warm-up" to our main result. Finally, Section 5.5 contains our main result, a general algorithm for testing LTFs over $\{-1,1\}^n$

5.2 Tools for Estimating Sums of Powers of Fourier and Hermite Coefficients

In this section we show how to estimate the sum $\sum_{i=1}^n \hat{f}(i)^2$ for functions over a boolean domain, and the sum $\sum_{i=1}^n \hat{f}(e_i)^2$ for functions over gaussian space. This subroutine lies at the heart of our testing algorithms. We actually prove a more general theorem, showing how to estimate $\sum_{i=1}^n \hat{f}(i)^p$ for any integer $p \geq 2$. Estimating the special case of $\sum_{i=1}^n \hat{f}(i)^4$ allows us to distinguish whether a function has a single large $|\hat{f}(i)|$, or whether all $|\hat{f}(i)|$ are small. The main results in this section are Corollary 90 (along with its analogue for Gaussian space, Lemma 93), and Lemma 92.

5.2.1 Noise Stability

Definition 84. *(Noise stability for Boolean functions.) Let $f, g : \{-1,1\}^n \rightarrow \{-1,1\}$, let $\eta \in [0,1]$, and let (x,y) be a pair of η-correlated random inputs — i.e., x is a uniformly random string and y is formed by setting $y_i = x_i$ with probability η and letting y_i be uniform otherwise, independently for each i. We define*

$$\mathbb{S}_\eta(f,g) = \mathbf{E}[f(x)g(y)].$$

Fact 85. *In the above setting, $\mathbb{S}_\eta(f,g) = \sum_{S\subseteq[n]} \hat{f}(S)\hat{g}(S)\eta^{|S|}$.*

Definition 86. *(Noise stability for Gaussian functions.) Let $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$ be in $L^2(\mathbb{R}^n)$ with respect to the Gaussian measure, let $\eta \in [0,1]$, and let (x,y) be a pair of η-correlated n-dimensional Gaussians. I.e., each pair of coordinates (x_i, y_i) is chosen independently as follows: x_i is a standard 1-dimensional Gaussian, and $y_i = \eta x_i + \sqrt{1-\eta^2} \cdot z_i$, where z_i*

is an independent standard Gaussian. We define

$$\mathbb{S}_\eta(f,g) = \mathbf{E}[f(x)g(y)].$$

Fact 87. *In the above setting,* $\mathbb{S}_\eta(f,g) = \sum_{S\in\mathbb{N}^n} \hat{f}(S)\hat{g}(S)\eta^{|S|}$*, where* $|S|$ *denotes* $\sum_{i=1}^n S_i$*.*

5.2.2 Estimating sums of powers of Fourier coefficients

For $x = (x_1, \dots, x_n)$ and $S \subseteq [n]$ we write x_S for the monomial $\prod_{i\in S} x_i$. The following lemma generalizes Fact 85:

Lemma 88. *Fix* $p \geq 2$*. Let* $f_1, \dots, f_p$ *be* p *functions* $f_i : \{-1,1\}^n \to \{-1,1\}$*. Fix any set* $T \subseteq [n]$*. Let* $x^1, \dots, x^{p-1}$ *be independent uniform random strings in* $\{-1,1\}^n$ *and let* y *be a random string whose bits are independently chosen with* $\Pr[y_i = 1] = \frac{1}{2}$ *for* $i \notin T$ *and* $\Pr[y_i = 1] = \frac{1}{2} + \frac{1}{2}\eta$ *for* $i \in T$*. Let* $\odot$ *denote coordinate-wise multiplication. Then*

$$\mathbf{E}[f_1(x^1)f_2(x^2)\cdots f_{p-1}(x^{p-1})f_p(x^1\odot x^2\odot\cdots\odot x^{p-1}\odot y)] = \sum_{S\subseteq T} \eta^{|S|}\hat{f}_1(S)\hat{f}_2(S)\cdots\hat{f}_p(S).$$

Proof. We have

$$\begin{aligned}
&\mathbf{E}[(\prod_{i=1}^{p-1} f_i(x^i))f_p(x^1\odot x^2\odot\cdots\odot x^{p-1}\odot y)]\\
=\;&\mathbf{E}[\sum_{S_1,\dots,S_p\subseteq[n]} \left(\prod_{i=1}^{p-1}\hat{f}_i(S_i)(x^i)_{S_i}\right)\hat{f}_p(S_p)(x^1\odot x^2\odot\cdots\odot x^{p-1}\odot y)_{S_p}]\\
=\;&\sum_{S_1,\dots,S_p\subseteq[n]} \left(\prod_{i=1}^{p}\hat{f}_i(S_i)\right)\cdot\mathbf{E}[(x^1)_{S_1\Delta S_p}\cdots(x^{p-1})_{S_{p-1}\Delta S_p}(y)_{S_p}]
\end{aligned}$$

Now recalling that $x^1, \dots, x^{p-1}$ and y are all independent and the definition of y, we have that the only nonzero terms in the above sum occur when $S_1 = \cdots = S_{p-1} = S_p \subseteq T$; in this case the expectation is $\eta^{|S_p|}$. This proves the lemma. □

Lemma 89. *Let* $p \geq 2$*, and suppose we have black-box access to* $f_1, \dots, f_p : \{-1,1\}^n \to \{-1,1\}$*. Then for any* $T \subseteq [n]$*, we can estimate the sum of products of degree-1 Fourier* *coefficients*

$$\sum_{i \in T} \hat{f}_1(i) \cdots \hat{f}_p(i)$$

to within an additive η, with confidence $1 - \delta$, using $O(p \cdot \log(1/\delta)/\eta^4)$ queries.

Proof. Let $x^1, \dots, x^p$ be independent uniform random strings in $\{-1, 1\}^n$ and let y be as in the previous lemma. Empirically estimate

$$\mathbf{E}[f_1(x^1) f_2(x^2) \cdots f_p(x^p)] \text{ and} \tag{5.1}$$

$$\mathbf{E}[f_1(x^1) f_2(x^2) \cdots f_{p-1}(x^{p-1}) f_p(x^1 \odot x^2 \odot \cdots \odot x^{p-1} \odot y)] \tag{5.2}$$

to within an additive $\pm\eta^2$, using $O(\log(1/\delta)/\eta^4)$ samples for each random variable (and hence $O(p \cdot \log(1/\delta)/\eta^4)$ queries overall). By the previous lemma these two quantities are exactly equal to

$$\hat{f}_1(\emptyset) \cdots \hat{f}_p(\emptyset) \qquad \text{and} \qquad \sum_{S \subseteq T} \eta^{|S|} \hat{f}_1(S) \hat{f}_2(S) \cdots \hat{f}_p(S)$$

respectively. Subtracting the former estimate from the latter yields

$$\sum_{|S|>0, S \subseteq T} \eta^{|S|} \hat{f}_1(S) \cdots \hat{f}_p(S)$$

to within an additive $O(\eta^2)$, and this itself is within η^2 of

$$\sum_{|S|=1, S \subseteq T} \eta \hat{f}_1(S) \cdots \hat{f}_p(S)$$

because the difference is

$$\begin{aligned}
&\sum_{|S|>1, S \subseteq T} \eta^{|S|} \hat{f}_1(S) \cdots \hat{f}_p(S) \\
&\quad \leq \eta^2 \sum_{|S|>1, S \subseteq T} |\hat{f}_1(S) \cdots \hat{f}_p(S)| \\
&\quad \leq \eta^2 \sqrt{\sum_{|S|>1, S \subseteq T} \hat{f}_1(S)^2} \sqrt{\sum_{|S|>1, S \subseteq T} (\hat{f}_2(S) \cdots \hat{f}_p(S))^2} && (5.3) \\
&\quad \leq \eta^2 \cdot 1 \cdot \sqrt{\sum_{|S|>1, S \subseteq T} \hat{f}_2(S)^2} \leq \eta^2 && (5.4)
\end{aligned}$$

where (5.3) is Cauchy-Schwarz and (5.4) uses the fact that the sum of the squares of the Fourier coefficients of a Boolean function is at most 1. Thus we have $\eta \cdot \sum_{i \in T} \hat{f}_1(i) \cdots \hat{f}_p(i)$ to within an additive $O(\eta^2)$; dividing by η gives us the required estimate within $O(\eta)$. □

Taking all f_i's to be the same function f, we have

Corollary 90. *Fix $p \geq 2$ and fix any $T \subseteq [n]$. Given black-box access to $f : \{-1,1\}^n \to \{-1,1\}$, we can estimate $\sum_{i \in T} \hat{f}(i)^p$ to an additive $\pm\eta$, with confidence $1 - \delta$, using $O(p \cdot \log(1/\delta)/\eta^4)$ queries.*

Proposition 91. *If every $i \in T$ has $|\hat{f}(i)| < \alpha$, then $\sum_{i \in T} \hat{f}(i)^4 < \alpha^2 \sum_{i \in T} \hat{f}(i)^2 \leq \alpha^2$.*

Lemma 92. *Fix any $T \subseteq [n]$. There is a $O(\log(1/\delta)/\tau^{16})$-query test* **Non-Regular**(τ, δ, T) *which, given query access to $f : \{-1,1\}^n \to \{-1,1\}$, behaves as follows: with probability $1 - \delta$,*

- *If $|\hat{f}(i)| \geq \tau$ for some $i \in T$ then the test accepts;*
- *If every $i \in T$ has $|\hat{f}(i)| < \tau^2/4$ then the test rejects.*

Proof. The test is to estimate $\sum_{i \in T} \hat{f}(i)^4$ to within an additive $\pm\tau^4/4$ and then accept if and only if the estimate is at least $\tau^4/2$. If $|\hat{f}(i)| \geq \tau$ for some i then clearly $\sum_{i=1}^n \hat{f}(i)^4 \geq \tau^4$ so the test will accept since the estimate will be at least $3\tau^4/4$. On the other hand, if each $i \in T$ has $|\hat{f}(i)| < \tau^2/4$, then $\sum_{i \in T} \hat{f}(i)^4 < \tau^4/16$ by Proposition 91 and so the test will reject since the estimate will be less than $5\tau^4/16$. □

5.2.3 Estimating sums of powers of Hermite coefficients

Here we let $\hat{f}(e_i)$ denote the i-th degree-1 Hermite coefficient of $f : \mathbb{R}^n \to \mathbb{R}$ as described in Section 5.3.

For the Gaussian distribution we require only the following lemma, which can be proved in a straightforward way following the arguments in Section 5.2.2 and using Fact 87.

Lemma 93. *Given black-box access to $f : \mathbb{R}^n \to \{-1,1\}$, we can estimate $\sum_{i=1}^n \hat{f}(e_i)^2$ to within an additive η, with confidence $1 - \delta$, using $O(\log(1/\delta)/\eta^4)$ queries.*

5.3 A Tester for General LTFs over $\mathbb{R}^n$

In this section we consider functions f that map $\mathbb{R}^n$ to $\{-1,1\}$, where we view $\mathbb{R}^n$ as endowed with the standard n-dimensional Gaussian distribution. A draw of x from this distribution over $\mathbb{R}^n$ is obtained by drawing each coordinate x_i independently from the standard one-dimensional Gaussian distribution with mean zero and variance 1.

Our main result in this section is an algorithm for testing whether a function f is an LTF vs ϵ-far from all LTFs in this Gaussian setting. The algorithm itself is surprisingly simple. It first estimates f's mean, then estimates the sum of the squares of f's degree-1 hermite coefficients. Finally it checks that this latter sum is equal to a particular function W of the mean.

The tester and the analysis in this section can be viewed as a "warmup" for the results in later sections. Thus, it is worth saying a few words here about why the Gaussian setting is so much easier to analyze. Let $f : \mathbb{R}^n \rightarrow \{-1,1\}$ be an LTF, $f(x) = \operatorname{sgn}(w \cdot x - \theta)$, and assume by normalization that $\|w\| = 1$. Now note the n-dimensional Gaussian distribution is spherically symmetric, as is the class of LTFs. Thus there is a sense in which all LTFs with a given threshold θ are "the same" in the Gaussian setting. (This is very much untrue in the discrete setting of $\{-1,1\}^n$.) We can thus derive Hermite-analytic facts about all LTFs by studying one particular LTF; say, $f(x) = \operatorname{sgn}(e_1 \cdot x - \theta)$. In this case, the picture is essentially 1-dimensional; i.e., we can think of simply $h_\theta : \mathbb{R} \rightarrow \{-1,1\}$ defined by $h_\theta(x) = \operatorname{sgn}(x - \theta)$, where x is a single standard Gaussian, and the only parameter is $\theta \in \mathbb{R}$. In the following sections we derive some simple facts about this function, then give the details of our tester.

5.3.1 Gaussian LTF facts.

In this section we will use Hermite analysis on functions.

Definition 94. *We write ϕ for the p.d.f. of a standard Gaussian; i.e.,* $\phi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$.

Definition 95. *Let $h_\theta : \mathbb{R} \rightarrow \{-1,1\}$ denote the function of one Gaussian random variable x given by* $h_\theta(x) = \operatorname{sgn}(x - \theta)$.

Definition 96. *The function* $\mu : \mathbb{R} \cup \{\pm\infty\} \to [-1, 1]$ *is defined as* $\mu(\theta) = \widehat{h_\theta}(0) = \mathbf{E}[h_\theta]$. *Explicitly,* $\mu(\theta) = -1 + 2\int_\theta^\infty \phi$.

Note that μ is a monotone strictly decreasing function, and it follows that μ is invertible. Note also that by an easy explicit calculation, we have that $\widehat{h_\theta}(1) = \mathbf{E}[h_\theta(x)x] = 2\phi(\theta)$.

Definition 97. *We define the function* $W : [-1, 1] \to [0, 2/\pi]$ *by* $W(\nu) = (2\phi(\mu^{-1}(\nu)))^2$. *Equivalently,* W *is defined so that* $W(\mathbf{E}[h_\theta]) = \widehat{h_\theta}(1)^2$.

The intuition for W is that it "tells us what the squared degree-1 Hermite coefficient should be, given the mean." We remark that W is a function symmetric about 0, with a peak at $W(0) = \frac{2}{\pi}$.

Proposition 98. *If* x *denotes a standard Gaussian random variable, then*

1. $\mathbf{E}[|x - \theta|] = 2\phi(\theta) - \theta\mu(\theta)$.
2. $|\mu'| \leq \sqrt{2/\pi}$ *everywhere, and* $|W'| < 1$ *everywhere.*
3. *If* $|\nu| = 1 - \eta$ *then* $W(\nu) = \Theta(\eta^2 \log(1/\eta))$.

Proof. The first statement is because both equal $\mathbf{E}[h_\theta(x)(x - \theta)]$. The bound on μ's derivative holds because $\mu' = -2\phi$. The bound on W's derivative is because $W'(\nu) = 4\phi(\theta)\theta$, where $\theta = \mu^{-1}(\nu)$, and this expression is maximized at $\theta = \pm 1$, where it is $.96788\cdots < 1$. Finally, the last statement can be straightforwardly derived from the fact that $1 - \mu(\theta) \sim 2\phi(\theta)/|\theta|$ for $|\theta| \geq 1$. □

Having understood the degree-0 and degree-1 Hermite coefficients for the "1 dimensional" LTF $f : \mathbb{R}^n \to \{-1, 1\}$ given by $f(x) = \mathrm{sgn}(x_1 - \theta)$, we can immediately derive analogues for general LTFs:

Proposition 99. *Let* $f : \mathbb{R}^n \to \{-1, 1\}$ *be the LTF* $f(x) = \mathrm{sgn}(w \cdot x - \theta)$, *where* $w \in \mathbb{R}^n$. *Assume without loss of generality that* $\|w\| = 1$ *(we can do so, since the sign of* $w \cdot x - \theta$ *is unchanged when multiplied by any positive constant). Then:*

1. $\hat{f}(0) = \mathbf{E}[f] = \mu(\theta)$.
2. $\hat{f}(e_i) = \sqrt{W(\mathbf{E}[f])}w_i$.
3. $\sum_{i=1}^n \hat{f}(e_i)^2 = W(\mathbf{E}[f])$.

Proof. The first statement follows from the definition of $\mu(\theta)$. The third statement follows from the second, which we will prove. We have $\hat{f}(e_i) = \mathbf{E}_x[\mathrm{sgn}(w \cdot x - \theta)x_i]$. Now $w \cdot x$ is distributed as a standard 1-dimensional Gaussian. Further, $w \cdot x$ and x_i are jointly Gaussian with covariance $\mathbf{E}[(w \cdot x)x_i] = w_i$. Hence $(w \cdot x, x_i)$ has the same distribution as $(y, w_i y + \sqrt{1 - w_i^2} \cdot z)$ where y and z are independent standard 1-dimensional Gaussians. Thus

$$\begin{aligned}
\mathop{\mathbf{E}}_x[\mathrm{sgn}(w \cdot x - \theta)x_1] &= \mathbf{E}[\mathrm{sgn}(y - \theta)(w_i y + \sqrt{1 - w_i^2} \cdot z)] \\
&= w_i \widehat{h_\theta}(1) + \mathbf{E}[\mathrm{sgn}(y - \theta)\sqrt{1 - w_i^2} \cdot z] \\
&= w_i \sqrt{W(\mathbf{E}[h_\theta])} + 0 \\
&= \sqrt{W(\mathbf{E}[f])} w_i
\end{aligned}$$

as desired. □

The second item in the above proposition leads us to an interesting observation: if $f(x) = \mathrm{sgn}(w_1 x_1 + \cdots + w_n x_n - \theta)$ is any LTF, then its vector of degree-1 Hermite coefficients, $(\hat{f}(e_1), \ldots, \hat{f}(e_n))$, is parallel to its vector of weights, $(w_1, \ldots, w_n)$.

5.3.2 The Tester.

We now give a simple algorithm and prove that it accepts any LTF with probability at least $2/3$ and rejects any function that is $O(\epsilon)$-far from all LTFs with probability at least $2/3$. The algorithm is nonadaptive and has two-sided error; the analysis of the two-sided confidence error is standard and will be omitted.

Given an input parameter $\epsilon > 0$, the algorithm works as follows:

1. Let $\tilde{\mu}$ denote an estimate of $\mathbf{E}[f]$ that is accurate to within additive accuracy $\pm\epsilon^3$.

2. Let $\tilde{\sigma}^2$ denote an estimate of $\sum_{i=1}^n \hat{f}(e_i)^2$ that is accurate to within additive accuracy $\pm\epsilon^3$.

3. If $|\tilde{\sigma}^2 - W(\tilde{\mu})| \leq 2\epsilon^3$ then output "yes," otherwise output "no."

The first step can be performed simply by making $O(1/\epsilon^6)$ independent draws from the Gaussian distribution, querying f on each draw, and letting $\tilde{\mu}$ be the corresponding empirical estimate of $\mathbf{E}[f]$; the result will be $\pm\epsilon^3$-accurate with high probability. The second step of estimating $\sum_{i=1}^n \hat{f}(e_i)^2$ was described in section 5.2.

We now analyze the correctness of the test. The "yes" case is quite easy: Since $\tilde{\mu}$ is within $\pm\epsilon^3$ of $\mathbf{E}[f]$, and since $|W'| \leq 1$ for all x (by Proposition 98 item 2), we conclude that $W(\tilde{\mu})$ is within $\pm\epsilon^3$ of the true value $W(\mathbf{E}[f])$. But since f is an LTF, this value is precisely $\sum_{i=1}^n \hat{f}(e_i)^2$, by Proposition 99 item 3. Now $\tilde{\sigma}^2$ is within $\pm\epsilon^3$ of $\sum_{i=1}^n \hat{f}(e_i)^2$, and so the test indeed outputs "yes".

As for the "no" case, the following theorem implies that any function f which passes the test with high probability is $O(\epsilon)$-close to an LTF (either a constant function ± 1 or a specific LTF defined by $\mathbf{E}[f]$ and f's degree-1 Hermite coefficients):

Theorem 100. *Assume that $|\mathbf{E}[f]| \leq 1 - \epsilon$. If $|\sum_{i=1}^n \hat{f}(e_i)^2 - W(\mathbf{E}[f])| \leq 4\epsilon^3$, then f is $O(\epsilon)$-close to an LTF (in fact to an LTF whose coefficients are the Hermite coefficients $\hat{f}(e_i)$).*

Proof. Let $\sigma = \sqrt{\sum_i \hat{f}(e_i)^2}$, let $t = \mu^{-1}(\mathbf{E}[f])$, and let $h(x) = \frac{1}{\sigma}\sum \hat{f}(e_i)x_i - t$. We will show that f and the LTF $\mathrm{sgn}(h)$ are $O(\epsilon)$-close, by showing that both functions are correlated similarly with h. We have

$$\mathbf{E}[fh] = \frac{1}{\sigma}\sum_i \hat{f}(e_i)^2 - t\,\mathbf{E}[f] = \sigma - t\,\mathbf{E}[f],$$

where the first equality uses Plancherel. On the other hand, by Proposition 98 (item 1), we have

$$\mathbf{E}[|h|] = 2\phi(t) - t\mu(t) = 2\phi(\mu^{-1}(\mathbf{E}[f])) - t\,\mathbf{E}[f] = \sqrt{W(\mathbf{E}[f])} - t\,\mathbf{E}[f], \text{ and thus}$$
$$\mathbf{E}[h(\mathrm{sgn}(h) - f)] = \mathbf{E}[|h| - fh] = \sqrt{W(\mathbf{E}[f])} - \sigma \leq \frac{4\epsilon^3}{\sqrt{W(\mathbf{E}[f])}} \leq C\epsilon^2,$$

where $C > 0$ is some universal constant. Here the first inequality follows from the fact:

Fact 101. *Suppose A and B are nonnegative and $|A - B| \leq \eta$. Then $|\sqrt{A} - \sqrt{B}| \leq \eta/\sqrt{B}$.*

Proof. $|\sqrt{A}-\sqrt{B}| = \frac{|A-B|}{\sqrt{A}+\sqrt{B}} \leq \frac{\eta}{\sqrt{B}}$. □

with $W(\mathbf{E}[f])$, σ^2, $4\epsilon^3$ in place of A, B, and η. The second follows from the assumption that $|\mathbf{E}[f]| \leq 1-\epsilon$, which by Proposition 98 (item 3) implies that $\sqrt{W(\mathbf{E}[f])} \geq \Omega(\epsilon)$.

Note that for any x, the value $h(x)(\mathrm{sgn}(h(x)) - f(x))$ equals $2|h(x)|$ if f and $\mathrm{sgn}(h)$ disagree on x, and zero otherwise. So given that $\mathbf{E}[h(\mathrm{sgn}(h) - f)] \leq C\epsilon^2$, the value of $\Pr[f(x) \neq \mathrm{sgn}(h(x))]$ is greatest if the points of disagreement are those on which h is smallest. Let p denote $\Pr[f \neq \mathrm{sgn}(h)]$. Recall that h is defined as a linear combination of x_i's. Since each x_i is chosen according to a gaussian distribution, and a linear combination of gaussian random variables is itself a gaussian (with variance equal to the sum of the square of the weights, in this case 1), it is easy to see that $\Pr[|h| \leq p/2] \leq \frac{1}{\sqrt{2\pi}}p \leq p/2$. It follows that f and $\mathrm{sgn}(h)$ disagree on a set of measure at least $p/2$, over which $|h|$ is at least $p/2$. Thus, $\mathbf{E}[h(\mathrm{sgn}(h) - f)] \geq 2 \cdot (p/2) \cdot (p/2) = p^2/2$. Combining this with the above, it follows that $p \leq \sqrt{2C} \cdot \epsilon$, and we are done. □

5.4 A Tester for Balanced Regular LTFs over $\{-1,1\}^n$

It is natural to hope that an algorithm similar to the one we employed in the Gaussian case — estimating the sum of squares of the degree-1 Fourier coefficients of the function, and checking that it matches up with W of the function's mean — can be used for LTFs over $\{-1,1\}^n$ as well. It turns out that LTFs which are what we call "regular" — i.e., they have all their degree-1 Fourier coefficients small in magnitude — are amenable to the basic approach from Section 5.3, but LTFs which have large degree-1 Fourier coefficients pose significant additional complications. For intuition, consider $\mathrm{Maj}(x) = \mathrm{sgn}(x_1 + \cdots + x_n)$ as an example of a highly regular halfspace and $\mathrm{sgn}(x_1)$ as an example of a halfspace which is highly non-regular. In the first case, the argument $x_1 + \cdots + x_n$ behaves very much like a Gaussian random variable so it is not too surprising that the Gaussian approach can be made to work; but in the second case, the ±1-valued random variable x_1 is very unlike a Gaussian.

We defer testing general LTF's over $\{-1,1\}^n$ to Section 5.5, and in this section we present a tester for *balanced, regular* LTFs.

Definition 102. *We say that* $f : \{-1,1\}^n \rightarrow \{-1,1\}$ *is* τ-regular *if* $|\hat{f}(i)| \leq \tau$ *for all* $i \in [n]$.

Definition 103. *We say that an LTF* $f : \{-1,1\}^n \rightarrow \{-1,1\}$ *is "balanced" if it has threshold zero and* $\mathbf{E}[f] = 0$. *We define* $\mathrm{LTF}_{n,\tau}$ *to be the class of all balanced,* τ*-regular LTFs.*

The balanced regular LTF subcase gives an important conceptual ingredient in the testing algorithm for general LTFs and admits a relatively self-contained presentation. As we discuss in Section 5.5, though, significant additional work is required to get rid of either the "balanced" or "regular" restriction.

The following theorem shows that we can test the class $\mathrm{LTF}_{n,\tau}$ with a constant number of queries:

Theorem 104. *Fix any* $\tau > 0$. *There is an* $O(1/\tau^8)$*-query algorithm* A *that satisfies the following property: Let* ϵ *be any value* $\epsilon \geq C\tau^{1/6}$, *where* C *is an absolute constant. Then if* A *is run with input* ϵ *and black-box access to any* $f : \{-1,1\}^n \rightarrow \{-1,1\}$,

- *if* $f \in \mathrm{LTF}_{n,\tau}$ *then* A *outputs "yes" with probability at least* $2/3$*;*

- *if* f *is* ϵ*-far from every function in* $\mathrm{LTF}_{n,\tau}$ *then* A *outputs "no" with probability at least* $2/3$.

The algorithm A in Theorem 104 has two steps. The purpose of Step 1 is to check that f is roughly τ-regular; if it is not, then the test rejects since f is certainly not a τ-regular halfspace. In Step 2, A checks that $\sum_{i=1}^n \hat{f}(i)^2 \approx \frac{2}{\pi}$. This check is based on the idea (see Section 5.4.2) that for *any* regular function f, the degree-1 Fourier weight is close to $\frac{2}{\pi}$ if and only if f is close to being an LTF. (Note the correspondence between this statement and the results of Section 5.3 in the case $\mathbf{E}[f] = 0$.)

We now describe algorithm A, which takes as input a parameter $\epsilon \geq C\tau^{1/6}$:

1. First A estimates $\sum_{i=1}^n \hat{f}(i)^4$ to within an additive $\pm\tau^2$. If the estimate is greater than $2\tau^2$ then A halts and outputs "no," otherwise it continues.

2. Next A estimates $\sum_{i=1}^n \hat{f}(i)^2$ to within an additive $\pm C_1\tau^{1/3}$ (where $C_1 > 0$ is an absolute constant specified below). If this estimate is within an additive $\pm 2C_1\tau^{1/3}$ of $\frac{2}{\pi}$ then A outputs "yes", otherwise it outputs "no."

A description of how the sums of powers of degree-1 Fourier coefficients can be estimated was given in Section 5.2, see Corollary 90 in particular.

In Section 5.4.1 we discuss how regular LTFs over $\{-1,1\}^n$ can be approximated by functions of the form $\mathrm{sgn}(X-\theta)$ where X is a single Gaussian random variable. In Section 5.4.2, we prove two theorems showing that balanced regular LTFs are essentially characterized by the property $\sum_{i=1}^n \hat{f}(i)^2 \approx \frac{2}{\pi}$. In Section 5.4.3 we prove correctness of the test.

5.4.1 Approximating Regular LTFs as Gaussian Threshold Functions.

In this section we show that regular LTFs over $\{-1,1\}^n$ behave essentially like functions of the form $\mathrm{sgn}(X-\theta)$, where X is a single Gaussian random variable. In sections 5.4.2 and 5.4.3 we will be particularly interested in the case when $\theta = 0$, however in later sections we will be interested in arbitrary θ, hence we prove more general versions of the theorems here.

Before getting started, we make a notational note. Throughout the rest of this chapeter we will be dealing with approximations, and therefore it will be convenient to have a quick way to indicate when a is an approximation of b to "within an $O(\eta)$ factor." Thus we make the following definition:

Definition 105. *For $a, b \in \mathbb{R}$ we write $a \overset{\eta}{\approx} b$ to indicate that $|a-b| \leq O(\eta)$.*

Now we state the well-known Berry-Esseen theorem, a version of the Central Limit Theorem with error bounds (see, e.g., [21]):

Theorem 106. *Let $\ell(x) = c_1x_1 + \cdots + c_nx_n$ be a linear form over the random ± 1 bits x_i. Let τ be such that $|c_i| \leq \tau$ for all i, and write $\sigma = \sqrt{\sum c_i^2}$. Write F for the c.d.f. of $\ell(x)/\sigma$; i.e., $F(t) = \Pr[\ell(x)/\sigma \leq t]$. Then for all $t \in \mathbb{R}$,*

$$|F(t) - \Phi(t)| \leq O(\tau/\sigma) \cdot \frac{1}{1+|t|^3},$$

where Φ denotes the c.d.f. of X, a standard Gaussian random variable. In particular, if $A \subseteq \mathbb{R}$ is any interval then $\Pr[\ell(x)/\sigma \in A] \overset{\tau/\sigma}{\approx} \Pr[X \in A]$.

We will sometimes find it useful to quote a special case of the Berry-Essen theorem (with a sharper constant). The following can be found in [52]:

Theorem 107. *In the setup of Theorem 106, for any $\lambda \geq \tau$ and any $\theta \in \mathbb{R}$ it holds that* $\Pr[|\ell(x) - \theta| \leq \lambda] \leq 6\lambda/\sigma$.

The following is an almost immediate consequence of the Berry-Esseen theorem:

Proposition 108. *Let $f(x) = \mathrm{sgn}(c \cdot x - \theta)$ be an LTF such that $\sum_i c_i^2 = 1$ and $|c_i| \leq \tau$ for all i. Then we have $\mathbf{E}[f] \overset{\tau}{\approx} \mu(\theta)$, where μ is the function defined in Definition 96.*

Next we prove the following more difficult statement, which gives an approximation for the expected magnitude of the linear form $c \cdot x - \theta$ itself:

Proposition 109. *Let $\ell(x) = \sum c_i x_i$ be a linear form over $\{-1, 1\}^n$ and assume $|c_i| \leq \tau$ for all i. let $\sigma = \sqrt{\sum c_i^2}$ and let $\theta \in \mathbb{R}$. Then*

$$\mathbf{E}[|\ell - \theta|] \overset{\tau}{\approx} \mathbf{E}[|\sigma X - \theta|],$$

where X is a standard Gaussian random variable.

Proof. The result is certainly true if $\sigma = 0$, so we may assume $\sigma > 0$. Using the fact that $\mathbf{E}[R] = \int_0^\infty \Pr[R > s]\, ds$ for any nonnegative random variable R for which $\mathbf{E}[R] < \infty$, we have that

$$\begin{aligned} \mathbf{E}[|\ell - \theta|] &= \int_0^\infty \Pr[|\ell - \theta| > s]\, ds \\ &= \int_0^\infty \Pr[\ell > \theta + s] + \Pr[\ell < \theta - s]\, ds \\ &= \int_0^\infty (1 - F((\theta + s)/\sigma) + F((\theta - s)/\sigma)\, ds \qquad (5.5) \end{aligned}$$

where we have written F for the c.d.f. of $\ell(x)/\sigma$. We shall apply Berry-Esseen to $\ell(x)$. Berry-Esseen tells us that for all $z \in \mathbb{R}$ we have $|F(z) - \Phi(z)| \leq O(\tau/\sigma)/(1 + |z|^3)$. Note that

$$\begin{aligned}\sum_{i=1}^{n} \mathbf{E}[|cx_i|^3] &= \sum_{i=1}^{n} |c_i|^3 \\ &\leq \tau \sum_{i=1}^{n} c_i^2 \\ &= \tau\sigma^2\end{aligned}$$

It follows that (5.5) $\leq (A) + (B)$, where

$$(A) = \int_0^\infty 1 - \Phi((\theta + s)/\sigma) + \Phi((\theta - s)/\sigma)\, ds$$

and

$$(B) = O(\tau/\sigma) \cdot \int_0^\infty \left(\frac{1}{1 + |(\theta + s)/\sigma|^3} + \frac{1}{1 + |(\theta - s)/\sigma|^3}\right) ds.$$

It is easy to see that

$$(B) = O(\tau/\sigma) \cdot \int_{-\infty}^{\infty} \frac{1}{1 + |x/\sigma|^3}\, dx = O(\tau).$$

For (A), observe that (A) can be re-expressed as

$$\int_0^\infty \Pr[X > (\theta + s)/\sigma] + \Pr[X < (\theta - s)/\sigma] ds = \int_0^\infty \Pr[|\sigma X - \theta| > s]\, ds.$$

Again using the fact that $\mathbf{E}[R] = \int_0^\infty \Pr[R > s]\, ds$ for any nonnegative random variable R for which $\mathbf{E}[R] < \infty$, this equals $\mathbf{E}[|\sigma X - \theta|]$. This gives the desired bound. □

Multidimensional Berry-Esseen

We now discuss a multidimensional generalization of the Berry-Esseen Theorem (Theorem 106) that will be useful in our tester for general LTFs over $\{-1, 1\}^n$. The argument here is very similar to an argument in [39], and is included only for completeness. It won't be used until Section 5.5, thus for now the reader may safely skip ahead to Section 5.4.2

The following theorem appears as Theorem 16 in [39] and Corollary 16.3 in [4]

Theorem 110. *Let $X_1, ..., X_n$ be independent random variables taking values in $\mathbb{R}^k$ satisfying:*

- $E[X_j] = 0, j = 1...n$
- $n^{-1} \sum_{j=1}^{n} Cov(X_j) = V$*, where Cov denotes the variance-covariance matrix*
- λ *is the smallest eigenvalue of V,* Λ *is the largest eigenvalue of V*
- $\rho_3 = n^{-1} \sum_{j=1}^{n} \mathbf{E}[\|X_j\|^3] < \infty$

Let Q_n denote the distribution of $n^{-1/2}(X_1 + \cdots + X_n)$, let $\Phi_{0,V}$ denote the distribution of the k-dimensional Gaussian with mean 0 and variance-covariance matrix V, and let $\eta = C\lambda^{-3/2}\rho_3 n^{-1/2}$, where C is a certain universal constant.

Then for any Borel set A,

$$|Q_n(A) - \Phi_{0,V}(A)| \leq \eta + B(A)$$

where B(A) is the following measure of the boundary of A : $B(A) = 2 \sup_{y \in \mathbb{R}^k} \Phi_{0,V}((\partial A)^{\eta'} + y)$, where $\eta' = \Lambda^{1/2}\eta$ and $(\partial A)^{\eta'}$ denotes the set of points within distance η' of the topological boundary of A.

The following application of Theorem 110 will be useful for our purposes. The argument is the same as that used in the proof of Proposition 10.1 in [39].

Theorem 111. *Let $\ell(x) = c_1x_1 + \cdots + c_nx_n$ be a linear form such that $\sum {c_i}^2 = 1$. Let τ be such that $|c_i| \leq \tau$ for all i. Let (x, y) be a pair of ρ-correlated random binary strings. Then for any intervals $I_1 \subseteq \mathbb{R}$ and $I_2 \subseteq \mathbb{R}$ we have*

$$Pr[(\ell(x), \ell(y)) \in (A, B)] \overset{\tau}{\approx} \Pr[(X, Y) \in (I_1, I_2)]$$

where (X, Y) is a pair of ρ-correlated Gaussians.

Proof. We will apply Theorem 110. First, for $i = 1, ...n$ we define the random variables $L_i = (\sqrt{n}c_ix_i, \sqrt{n}c_iy_i)$. It is easy to see that $\mathbf{E}[L_i] = (0, 0)$ and $Cov(L_i) = nc_i^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$

for each i. Thus $V = n^{-1}\sum_{j=1}^{n} Cov(L_j) = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. The eigenvalues of V are $\lambda = 1-\rho$ and $\Lambda = 1+\rho$.

Note that $\|L_i\| = \sqrt{2n}|c_i|$ with probability 1, so

$$\begin{aligned} \rho_3 &= n^{-1}\sum_{j=1}^{n} \mathbf{E}[\|X_j\|^3] \\ &= 2^{3/2}n^{1/2}\sum |c_i|^3 \\ &\leq 2^{3/2}n^{1/2} \cdot \max_i|c_i| \cdot \sum |c_i|^2 \\ &\leq 2^{3/2}n^{1/2}\tau \end{aligned}$$

Thus η is $O((1-\rho)^{-3/2}\tau)$. If $|\rho|$ bounded away from 1, then this is $O(\tau)$.

It is easy to check that the topological boundary of $I_1 \times I_2$ is $O(\eta')$. Since $\eta' = (1+\rho)^{1/2}\eta$, this is also $O(\tau)$. Thus $|Pr[(\ell(x), \ell(y)) \in (A, B)] \overset{\tau}{\approx} \Pr[(X, Y) \in (I_1, I_2)]| \leq O(\tau)$ and the theorem is proved.

□

5.4.2 Two theorems about $\mathrm{LTF}_{n,\tau}$.

The first theorem of this section tells us that any $f \in \mathrm{LTF}_{n,\tau}$ has sum of squares of degree-1 Fourier coefficients very close to $\frac{2}{\pi}$. The next theorem is a sort of dual; it states that any Boolean function f whose degree-1 Fourier coefficients are all small and have sum of squares $\approx \frac{2}{\pi}$ is close to being a balanced regular LTF (in fact, to the LTF whose weights equal f's degree-1 Fourier coefficients). Note the similarity in spirit between these results and the characterization of LTFs with respect to the Gaussian distribution that was provided by Proposition 99 item 3 and Theorem 100.

Theorem 112. *Let $f \in \mathrm{LTF}_{n,\tau}$. Then* $\left|\sum_{i=1}^{n} \hat{f}(i)^2 - \frac{2}{\pi}\right| \leq O(\tau^{2/3})$.

Proof. Let $\rho > 0$ be small (chosen later). Theorem 5 of [39] states that for $f \in \mathrm{LTF}_{n,\tau}$ and $\rho \in [-1, 1]$ we have

$$S_\rho(f, f) = 1 - \frac{2}{\pi}\arccos(\rho) \pm O(\tau(1-\rho)^{-3/2})$$

Combining this with Fact 85, and substituting $\arccos(\rho) = \frac{\pi}{2} - \arcsin(\rho)$, we have

$$\sum_S \rho^{|S|} \hat{f}(S)^2 = \frac{2}{\pi} \arcsin \rho \pm O(\tau).$$

On the LHS side we have that $\hat{f}(S) = 0$ for all even $|S|$ since f is an odd function, and therefore, $|\sum_S \rho^{|S|}\hat{f}(S)^2 - \rho \sum_{|S|=1} \hat{f}(S)^2| \leq \rho^3 \sum_{|S|\geq 3} \hat{f}(S)^2 \leq \rho^3$. On the RHS, by a Taylor expansion we have $\frac{2}{\pi} \arcsin \rho = \frac{2}{\pi}\rho + O(\rho^3)$. We thus conclude

$$\rho \sum_{i=1}^{n} \hat{f}(i)^2 = \frac{2}{\pi}\rho \pm O(\rho^3 + \tau).$$

Dividing by ρ and optimizing with $\rho = \Theta(\tau^{1/3})$ completes the proof. □

Theorem 113. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be any function such that $|\hat{f}(i)| \leq \tau$ for all i and $|\sum_{i=1}^{n} \hat{f}(i)^2 - \frac{2}{\pi}| \leq \gamma$. Write $\ell(x) := \sum_{i=1}^{n} \hat{f}(i)x_i$. Then f and* $\mathrm{sgn}(\ell(x))$ *are $O(\sqrt{\gamma + \tau})$-close.*

Proof. First note that if $\gamma > 1/3$ then the claimed bound is trivially true, so we will prove the theorem assuming $\gamma \leq 1/3$. Let $L := \sqrt{\sum_{i=1}^{n} \hat{f}(i)^2}$; note that by our assumption on γ we have $L \geq \frac{1}{2}$. We have:

$$\begin{aligned} (2/\pi) - \gamma &\leq \sum_{i=1}^{n} \hat{f}(i)^2 = \mathbf{E}[f\ell] \leq \mathbf{E}[|\ell|] && (5.6) \\ &\leq \sqrt{2/\pi} \cdot L + O(\tau) && (5.7) \\ &\leq \sqrt{2/\pi}\sqrt{2/\pi + \gamma} + O(\tau) \leq (2/\pi) + O(\gamma) + O(\tau). \end{aligned}$$

The equality in (5.6) is Plancherel's identity, and the latter inequality is because f is a ±1-valued function. The inequality (5.7) holds for the following reason: $\ell(x)$ is a linear form over random ±1's in which all the coefficients are at most τ in absolute value. Hence we expect it to act like a Gaussian (up to $O(\tau)$ error) with standard deviation L, which would have expected absolute value $\sqrt{2/\pi} \cdot L$. See Proposition 109 for the precise justification. Comparing the overall left- and right-hand sides, we conclude that $\mathbf{E}[|\ell|] - \mathbf{E}[f\ell] \leq O(\gamma) + O(\tau)$.

Let ϵ denote the fraction of points in $\{-1,1\}^n$ on which f and $\mathrm{sgn}(\ell)$ disagree. Given that there is a ϵ fraction of disagreement, the value $\mathbf{E}[|\ell|] - \mathbf{E}[f\ell]$ is smallest if the disagreement points are precisely those points on which $|\ell(x)|$ takes the smallest value. Now again we use the fact that ℓ should act like a Gaussian with standard deviation L, up to some error $O(\tau/L) \leq O(2\tau)$; we can assume this error is at most $\epsilon/4$, since if $\epsilon \leq O(\tau)$ then the theorem already holds. Hence we have (see Theorem 106 for precise justification)

$$\Pr[|\ell| \leq \epsilon/8] = \Pr[|\ell/L| \leq \epsilon/8L] \leq \Pr[|N(0,1)| \leq \epsilon/8L] + \epsilon/4 \leq \epsilon/8L + \epsilon/4 \leq \epsilon/2,$$

since $L \geq 1/2$. It follows that at least an $\epsilon/2$ fraction of inputs x have both $f(x) \neq \mathrm{sgn}(\ell(x))$ and $|\ell(x)| > \epsilon/8$. This implies that $\mathbf{E}[|\ell|] - \mathbf{E}[f\ell] \geq 2 \cdot (\epsilon/2) \cdot (\epsilon/8) = \epsilon^2/8$. Combining this with the previous bound $\mathbf{E}[|\ell|] - \mathbf{E}[f\ell] \leq O(\gamma) + O(\tau)$, we get $\epsilon^2/8 \leq O(\gamma) + O(\tau)$ which gives the desired result. □

5.4.3 Proving correctness of the test.

First observe that for any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, if $|\hat{f}(i)| \leq \tau$ for all i then $\sum_{i \in T} \hat{f}(i)^4 \leq \tau^2 \sum_{i \in T} \hat{f}(i)^2 \leq \tau^2$, using Parseval. On the other hand, if $|\hat{f}(i)| \geq 2\tau^{1/2}$ for some i, then $\sum_{i=1}^n \hat{f}(i)^4$ is certainly at least $16\tau^2$.

Suppose first that the function f being tested belongs to $\mathrm{LTF}_{n,\tau}$. As explained above, in this case f will with high probability pass Step 1 and continue to Step 2. By Theorem 112 the true value of $\sum_{i=1}^n \hat{f}(i)^2$ is within an additive $O(\tau^{2/3})$ of $\frac{2}{\pi}$; since this additive $O(\tau^{2/3})$ term is at most $C_1\tau^{1/3}$ for some constant C_1, the algorithm outputs "yes" with high probability. So the algorithm behaves correctly on functions in $\mathrm{LTF}_{n,\tau}$.

Now suppose $f : \{-1,1\}^n \to \{-1,1\}$ is such that the algorithm outputs "yes" with high probability; we show that f must be ϵ-close to some function in $\mathrm{LTF}_{n,\tau}$. Since there is a low probability that A outputs "no" in Step 1 on f, it must be the case that each $|\hat{f}(i)|$ is at most $2\tau^{1/2}$. Since f outputs "yes" with high probability in Step 2, it must be the case that $\sum_{i=1}^n \hat{f}(i)^2$ is within an additive $O(\tau^{1/3})$ of $\frac{2}{\pi}$. Plugging in $2\tau^{1/2}$ for "τ" and $O(\tau^{1/3})$ for "γ" in Theorem 113, we have that f is $C\tau^{1/6}$-close to $\mathrm{sgn}(\ell(x))$ where C is some absolute constant. This proves the correctness of A.

To analyze the query complexity, note that Corollary 90 tells us that Step 1 requires

$O(1/\tau^8)$ queries, and Step 2 only $O(1/\tau^{4/3})$, so the total query complexity is $O(1/\tau^8)$. This completes the proof of Theorem 104. □

5.5 A Tester for General LTFs over $\{-1, 1\}^n$

In this section we give our main result, a constant-query tester for general halfspaces over $\{-1, 1\}^n$. We start with a very high-level overview of our approach.

As we saw in Section 5.4, it is possible to test a function f for being close to a balanced τ-regular LTF. The key observation was that such functions have $\sum_{i=1}^n \hat{f}(i)^2$ approximately equal to $\frac{2}{\pi}$ if and only if they are close to LTFs. Furthermore, in this case, the functions are actually close to being the sign of their degree-1 Fourier part. It remains to extend the test described there to handle general LTFs, which may be unbalanced and/or non-regular. We will first discuss how to remove the balancedness condition, and then how to remove the regularity condition.

For handling unbalanced regular LTFs, a clear approach suggests itself, using the $W(\cdot)$ function as in Section 5.3. This is to try to show that for f an arbitrary τ-regular function, the following holds: $\sum_{i=1}^n \hat{f}(i)^2$ is approximately equal to $W(\mathbf{E}[f])$ if and only if f is close to an LTF — in particular, close to an LTF whose linear form is the degree-1 Fourier part of f. The "only if" direction here is not too much more difficult than Theorem 113 (see Theorem 128 in Section 5.5.2), although the result degrades as the function's mean gets close to 1 or -1. However the "if" direction turns out to present significant difficulty.

In the proof of Theorem 112, the special case of mean-zero, we appealed to a result from [39]. This results said that for balanced, regular LTFs, the sum $\sum_S \rho^{|S|}\hat{f}(S)^2$ is close to $\frac{2}{\pi}\arcsin\rho$. [39] proved this result using two propositions. First they showed showed that balanced LTFs with small weights must have $\sum_S \rho^{|S|}\hat{f}(S)^2$ close to $\frac{2}{\pi}\arcsin\rho$. Then they showed that balanced, regular LTFs must have small weights. While it is not too hard to appropriately generalize the first of [39]'s arguments to unbalanced LTFs, generalizing the second is considerably more complicated. It requires us to upper-bound the weights of an LTF as a function of both the regularity parameter and the mean of the function. We do this with Theorem 118, which we prove in Section 5.5.1:[1]

We now discuss removing the regularity condition; this requires additional analytic work and moreover requires that several new algorithmic ingredients be added to the test. Given any Boolean function f, Parseval's inequality implies that $J := \{i : |\hat{f}(i)| \geq \tau^2\}$ has cardinality at most $1/\tau^4$. Let us pretend for now that the testing algorithm could somehow know the set J. (If we allowed the algorithm $\Theta(\log n)$ many queries, it *could* in fact exactly identify some set like J. However with constantly many queries this is not possible. We ignore this problem for the time being, and will discuss how to get around it at the end of this section.) If the set J is known, then the testing algorithm can set the variables in J to fixed values, and consider the induced function over the remaining variables that results.

Our algorithm first checks whether it is the case that for all but an ϵ fraction of restrictions ρ of J, the restricted induced f_ρ is ϵ-close to a constant function. If this is the case, then f is an LTF if and only if f is close to an LTF which depends only on the variables in J. So in this case the tester simply enumerates over "all" LTFs over J and checks whether f seems close to any of them. (Note that since J is of constant size there are at most constantly many LTFs to check here.)

It remains to deal with the case that for at least an ϵ fraction of restrictions of J, the restricted function is ϵ-far from a constant function. In this case, it can be shown using Theorem 118 that if f is an LTF then in fact *every* restriction of the variables in J yields a regular subfunction. So it can use the testing procedure for (general mean) regular LTFs already described to check that for most restrictions π, the restricted function f_π is close to an LTF — indeed, close to an LTF whose linear form is its own degree-1 Fourier part.

This is a good start, but it is not enough. At this point the tester is confident that most restricted functions f_π are close to LTFs whose linear forms are their own degree-1 Fourier parts — but in a true LTF, all of these restricted functions are expressible using a *common* linear form. Thus the tester needs to test *pairwise consistency* among the linear parts of the different f_π's.

To do this, recall that our approach for testing whether the regular function f_π is close

[1] Readers familiar with the notion of influence (Definition 10) will recall that for any LTF f we have $\mathrm{Inf}_f(i) = |\hat{f}(i)|$ for each i. Thus Theorem 118 may roughly be viewed as saying that "every not-too-biased LTF with a large weight has an influential variable."

to an LTF will be to check that there is near-equality in the inequality $\sum_{|S|=1} \widehat{f_\pi}(S)^2 \leq W(\mathbf{E}[f_\pi])$. If this holds for both f_π and $f_{\pi'}$, the algorithm can further check that the degree-1 parts of f_π and $f_{\pi'}$ are essentially parallel (i.e., equivalent) by testing that near-equality holds in the Cauchy-Schwarz inequality $\sum_{|S|=1} \widehat{f_\pi}(S)\widehat{f_{\pi'}}(S) \leq \sqrt{W(\mathbf{E}[f_\pi])}\sqrt{W(\mathbf{E}[f_{\pi'}])}$. Thus to become convinced that most restricted f_π's are close to LTFs over the *same* linear form, the tester can pick any particular π, call it π^*, and check that $\sum_{|S|=1} \widehat{f_{\pi^*}}(S)\widehat{f_\pi}(S) \approx \sqrt{W(\mathbf{E}[f_{\pi^*}])} \cdot \sqrt{W(\mathbf{E}[f_\pi])}$ for most other π's. (At this point there is one caveat. As mentioned earlier, the general-mean LTF tests degrade when the function being tested has mean close to 1 or -1. For the above-described test to work, f_{π^*} needs to have mean somewhat bounded away from 1 and -1, so it is important that the algorithm uses a restriction π^* that has $|\,\mathbf{E}[f]|$ bounded away from 1. Fortunately, finding such a restriction is not a problem since we are in the case in which at least an ϵ fraction of restrictions have this property.)

Now the algorithm has tested that there is a single linear form ℓ (with small weights) such that for most restrictions π to J, f_π is close to being expressible as an LTF with linear form ℓ. It only remains for the tester to check that the thresholds — or essentially equivalently, for small-weight linear forms, the means — of these restricted functions are consistent with some arbitrary weight linear form on the variables in J. It can be shown that there are at most $2^{\mathrm{poly}(|J|)}$ essentially different such linear forms $w \cdot \pi - \theta$, and thus the tester can just enumerate all of them and check whether for most π's it holds that $\mathbf{E}[f_\pi]$ is close to the mean of the threshold function $\mathrm{sgn}(\ell - (\theta - w \cdot \pi))$. This will happen for one such linear form if and only if f is close to being expressible as the LTF $h(\pi, x) = \mathrm{sgn}(w \cdot \pi + \ell - \theta)$.

This completes the sketch of the testing algorithm, modulo the explanation of how the tester can get around "knowing" what the set J is. Looking carefully at what the tester needs to do with J, it turns out that it suffices for it to be able to query f on random strings and correlated tuples of strings, subject to given restrictions π to J. This can be done essentially by borrowing a technique from the paper [25] (see the discussion after Theorem 132 in Section 5.5.4).

In the remainder of this section we make all these ideas precise and prove the following, which is our main result:

Theorem 114. *There is an algorithm* **Test-LTF** *for testing whether an arbitrary black-box*

$f : \{-1, 1\}^n \to \{-1, 1\}$ *is an LTF versus* ϵ*-far from any LTF. The algorithm has two-sided error and makes at most* $\text{poly}(1/\epsilon)$ *queries to* f.

Remark 115. *The algorithm described above is adaptive. We note that similar to [25], the algorithm can be made nonadaptive with a polynomial factor increase in the query complexity (see Remark 134 in Section 5.5.4).*

Section 5.5.1 gives the proof of Theorem 118. Section 5.5.2 gives two theorems essentially characterizing LTFs; these theorems are the main tools in proving the correctness of our test. Section 5.5.3 gives an overview of the algorithm, which is presented in Sections 5.5.4 and 5.5.5. Section 5.5.6 proves correctness of the test.

5.5.1 On the structure of LTFs: relating weights, influences and biases

In this section we explore the relationship between the weights of an LTF and the influences of the LTF's variables. Intuition tells us that these two quantities should be directly related. In particular, if we assume that the weights of LTFs are appropriately normalized, then LTFs without any large weights should not have any highly influential variables, and LTFs without any highly influential variables should not have any large weights. This intuition is in fact correct, however proving the former statement turns out to be much easier than the latter.

To start, we state the following very simple fact (an explicit proof appears in, e.g.,[22]).

Fact 116. *Let* $f = \text{sgn}(w_1x_1 + \cdots + w_nx_n - \theta)$ *be an LTF such that* $|w_1| \geq |w_i|$ *for all* $i \in [n]$. *Then* $|\text{Inf}_f(1)| \geq |\text{Inf}_f(i)|$ *for all* $i \in [n]$.

Using this fact together with the Berry-Esseen theorem we can prove an upper bound on the influences of LTFs with bounded weights:

Theorem 117. *Let* $f(x) = \text{sgn}(\sum_{i=1}^n w_ix_i - \theta)$ *be an LTF such that* $\sum_i w_i^2 = 1$ *and* $\delta \geq |w_i|$ *for all* i. *Then* f *is* $O(\delta)$*-regular; i.e.,* $\text{Inf}_f(i) \leq O(\delta)$ *for all* i.

Proof. Without loss of generality we may assume that $\delta = |w_1| \geq |w_i|$ for all i. By Fact 116 we need to show that $\text{Inf}_f(1) \leq O(\delta)$. Now observe that

$$\text{Inf}_f(1) = \Pr\big[|w_2x_2 + \cdots + w_nx_n - \theta| \leq \delta\big]. \tag{5.8}$$

If $\delta \geq 1/2$ then clearly $\mathrm{Inf}_f(1) \leq 2\delta$ so we may assume $\delta < 1/2$. By the Berry-Esseen theorem, the probability (5.8) above is within an additive $O(\delta/\sqrt{1-\delta^2}) = O(\delta)$ of the probability that $|X - \theta| \leq \delta$, where X is a mean-zero Gaussian with variance $1 - \delta^2$. This latter probability is at most $O(\delta/\sqrt{1-\delta^2}) = O(\delta)$, so indeed we have $\mathrm{Inf}_f(1) \leq O(\delta)$. □

Proving a converse to this theorem is significantly harder. We would like to show that in an LTF, the variable with largest (normalized) weight also has high influence. However, any lower bound on the size of that variable's influence must depend not only on the size of the associated weight, but also on the mean of the LTF (if the LTF is very biased, it may contain a variable with large weight but low influence, since the LTF is nearly constant). We quantify this dependence in the following theorem, which says that an LTF's most influential variable has influence at least polynomial in the size of the largest weight and the LTF's bias.

Theorem 118. *Let $f(x) = \mathrm{sgn}(w_1x_1 + \cdots + w_nx_n - \theta)$ be an LTF such that $\sum_i w_i^2 = 1$ and $\delta := |w_1| \geq |w_i|$ for all $i \in [n]$. Let $0 \leq \epsilon \leq 1$ be such that $|\mathbf{E}[f]| \leq 1 - \epsilon$. Then $|\hat{f}(1)| \geq \Omega(\delta\epsilon^6 \log(1/\epsilon))$.*

The remainder of Section 5.5.1 is devoted to proving Theorem 118. We note that even the $\theta = 0$ case of the theorem, corresponding to $\epsilon = 1$, is somewhat tricky to prove. It appeared first as Proposition 10.2 of [39]. A substantially more intricate proof is required for the general statement; indeed, the arguments of [39] occur in somewhat modified form as Cases 1.a and 1.b of our proof below.

It is an interesting open question whether the dependence on ϵ in Theorem 118 can be improved. It is easy to give an upper bound on $\mathrm{Inf}_f(1)$ in terms of either δ or ϵ: it is immediate that $\mathrm{Inf}_f(1) \leq O(\epsilon)$, and from Theorem 117 we have that $\mathrm{Inf}_f(1) \leq O(\delta)$. However there is a gap between $O(\delta + \epsilon)$ and $\Omega(\delta\epsilon^6 \log(1/\epsilon))$. We suspect that $\Theta(\delta\epsilon)$ may be the optimal bound for Theorem 118.

Useful tools for proving Theorem 118.

We first observe that

$$\mathrm{Inf}_f(1) = \Pr\big[|w_2x_2 + \cdots + w_nx_n - \theta| \le \delta\big]. \tag{5.9}$$

We shall prove Theorem 118 by lower bounding the right hand side of (5.9).

At many points in the proof of Theorem 118 we will use the following fact, which is a simple consequence of "Poincaré's inequality. "

Fact 119. *Let* $g : \{-1,1\}^\ell \to \{-1,1\}$ *be an LTF* $g(x) = \mathrm{sgn}(\sum_{i=1}^{\ell} w_ix_i - \theta)$ *with* $|w_1| \ge |w_i|$ *for all* $i = 1, \dots, \ell$. *Then* $\mathrm{Inf}_g(1) \ge \mathbf{V}[g]/\ell$.

Proof. Poincaré's inequality says that the sum of a function's influences is at least its variance, i.e. that $\sum_{i=1}^{\ell} \mathrm{Inf}_g(i) \ge \mathbf{V}[g]$ for any Boolean function g. Since $|w_1| \ge |w_i|$ for all i (Fact 116), we have $\mathrm{Inf}_g(1) \ge \mathrm{Inf}_g(i)$, and the fact follows. □

The following easily verified fact is also useful:

Fact 120. *Let* $g : \{-1,1\}^\ell \to \{-1,1\}$ *be an LTF* $g(x) = \mathrm{sgn}(\sum_{i=1}^{\ell} w_ix_i - \theta)$ *with* $|w_1| > |\theta|$. *Then* $\mathbf{V}[g] = \Omega(1)$.

Proof. Since $|w_1| > |\theta|$, one of the two restrictions obtained by fixing the first variable outputs 1 at least half the time, and the other outputs -1 at least half the time. This implies that $1/4 \le \Pr[g(x) = 1] < 3/4$, which gives $\mathbf{V}[g] = \Omega(1)$. □

We will also often use the Berry-Esseen theorem, Theorem 106. For definiteness, we will write C for the implicit constant in the $O(\cdot)$ of the statement, and we note that for every interval A we in fact have $|\Pr[\ell(x)/\sigma \in A] - \Pr[X \in A]| \le 2C\tau/\sigma$.

Finally, we will also use the Hoeffding bound:

Theorem 121. *Fix any* $0 \ne w \in \mathbb{R}^n$ *and write* $\|w\|$ *for* $\sqrt{w_1^2 + \cdots + w_n^2}$. *For any* $\gamma > 0$, *we have*

$$\Pr_{x\in\{-1,1\}^n}[w \cdot x \ge \gamma\|w\|] \le e^{-\gamma^2/2} \quad \textit{and} \quad \Pr_{x\in\{-1,1\}^n}[w \cdot x \le -\gamma\|w\|] \le e^{-\gamma^2/2}.$$

The idea behind Theorem 118.

We give a high-level outline of the proof before delving into the technical details. Here and throughout the proof we suppose for convenience that $\delta = |w_1| \geq |w_2| \geq \cdots \geq |w_n| \geq 0$.

We first consider the case (Case 1) that the biggest weight δ is small relative to ϵ. We show that with probability $\Omega(\epsilon^2)$, the "tail" $w_\beta x_\beta + \cdots + w_n x_n$ of the linear form (for a suitably chosen β) takes a value in $[\theta - 1, \theta + 1]$; this means that the effective threshold for the "head" $w_2 x_2 + \cdots + w_{\beta-1} x_{\beta-1}$ is in the range $[-1, 1]$. In this event, a modified version of the [39] proof shows that the probability that $w_2 x_2 + \cdots + w_{\beta-1} x_{\beta-1}$ lies within $\pm\delta$ of the effective threshold is $\Omega(\delta)$; this gives us an overall probability bound of $\Omega(\delta\epsilon^2)$ for (5.9) in Case 1.

We next consider the case (Case 2) that the biggest weight δ is large. We define the "critical index" of the sequence $w_1, \ldots, w_n$ to be the first index $k \in [n]$ at which the Berry-Esseen theorem applied to the sequence $w_k, \ldots, w_n$ has a small error term; see Definition 125 below. (This quantity was implicitly defined and used in [56].) We proceed to consider different cases depending on the size of the critical index.

Case 2.a deals with the situation when the critical index k is "large" (specifically larger than $\Theta(\log(1/\epsilon)/\epsilon^4)$. Intuitively, in this case the weights $w_1, \ldots, w_k$ decrease exponentially and the value $\sum_{j \geq k'} w_j^2$ is very small, where $k' = \Theta(\log(1/\epsilon)/\epsilon^4)$. The rough idea in this case is that the effective number of relevant variables is at most k', so we can use Fact 119 to get a lower bound on $\mathrm{Inf}(1)$. (There are various subcases here for technical reasons but this is the main idea behind all of them.)

Case 2.b deals with the situation when the critical index k is "small" (smaller than $\Theta(\log(1/\epsilon)/\epsilon^4)$). Intuitively, in this case the value $\sigma_k \overset{\text{def}}{=} \sqrt{\sum_{j \geq k} w_j^2}$ is large, so the random variable $w_k x_k + \cdots + w_n x_n$ behaves like a Gaussian random variable $N(0, \sigma_k)$ (recall that since k is the critical index, the Berry-Esseen error is "small"). Now there are several different subcases depending on the relative sizes of σ_k and θ, and on the relative sizes of δ and θ. In some of these cases we argue that "many" restrictions of the tail variables $x_k, \ldots, x_n$ yield a resulting LTF which has "large" variance; in these cases we can use Fact 119 to argue that for any such restriction the influence of x_1 is large, so the overall influence of x_1 cannot be too small. In the other cases we use the Berry-Esseen theorem to approximate the random variable $w_k x_k + \cdots + w_n x_n$ by a Gaussian $N(0, \sigma_k)$, and use properties of the Gaussian to argue that the analogue to expression (5.9) (with a Gaussian in place of $w_k x_k + \cdots + w_n x_n$) is not too small.

The detailed proof of Theorem 118.

We suppose without loss of generality that $\mathbf{E}[f] = -1 + \epsilon$, i.e. that $\theta \geq 0$. We have the following two useful facts:

Fact 122. *We have* $0 \leq \theta \leq \sqrt{2\ln(2/\epsilon)}$.

Proof. The lower bound is by assumption, and the upper bound follows from the Hoeffding bound and the fact that $\mathbf{E}[f] = -1 + \epsilon$. □

Fact 123. *Let S be any subset of variables $x_1, \ldots, x_n$. For at least an $\epsilon/4$ fraction of restrictions ρ that fix the variables in S and leave other variables free, we have* $\mathbf{E}[f_\rho] \geq -1 + \epsilon/4$.

Proof. If this were not the case then we would have $\mathbf{E}[f] < (\epsilon/4) \cdot 1 + (1 - \epsilon/4)(-1 + \epsilon/4) < -1 + \epsilon$, which contradicts the fact that $\mathbf{E}[f] = -1 + \epsilon$. □

Now we consider the cases outlined in the previous subsection. Recall that C is the absolute constant in the Berry-Esseen theorem; we shall suppose w.l.o.g. that C is a positive integer. Let $C_1 > 0$ be a suitably large (relative to C) absolute constant to be chosen later.

Case 1: $\delta \leq \epsilon^2/C_1$. We will show that in Case 1 we actually have $\mathrm{Inf}_f(1) = \Omega(\delta\epsilon^2)$.

Let us define $T \overset{\mathrm{def}}{=} \{\beta, \ldots, n\}$ where $\beta \in [n]$ is the last value such that $\sum_{i=\beta}^n w_i^2 \geq \frac{1}{2}$. Since each $|w_i|$ is at most $\epsilon^2/C_1 \leq 1/C_1$ (because we are in Case 1), we certainly have that $\sum_{i \in T} w_i^2 \in [\frac{1}{2}, \frac{3}{4}]$ by choosing C_1 suitably large.

We first show that the tail sum $\sum_{i \in T} w_i x_i$ lands in the interval $[\theta - 1, \theta + 1]$ with fairly high probability:

Lemma 124. *We have*

$$\Pr\left[\sum_{i \in T} w_i x_i \in [\theta - 1, \theta + 1]\right] \geq \epsilon^2/18.$$

Proof. Let σ_T denote $\left(\sum_{i \in T} w_i^2\right)^{1/2}$. As noted above we have $\sqrt{4/3} \le \sigma_T^{-1} \le \sqrt{2}$. We thus have

$$\begin{aligned}
\Pr\left[\sum_{i \in T} w_i x_i \in [\theta - 1, \theta + 1]\right] &= \Pr\left[\sigma_T^{-1} \sum_{i \in T} w_i x_i \in \sigma_T^{-1}[\theta - 1, \theta + 1]\right] \\
&\ge \Phi([\sigma_T^{-1}\theta - \sigma_T^{-1}, \sigma_T^{-1}\theta + \sigma_T^{-1}]) - 2C\delta\sigma_T^{-1} (5.10) \\
&> \Phi([\sigma_T^{-1}\theta - \sigma_T^{-1}, \sigma_T^{-1}\theta + \sigma_T^{-1}]) - 2\sqrt{2}C\delta (5.11)
\end{aligned}$$

where (5.10) follows from the Berry-Esseen theorem using the fact that each $|w_i| \le \delta$.

If $0 \le \theta \le 1$, then clearly the interval $[\sigma_T^{-1}\theta - \sigma_T^{-1}, \sigma_T^{-1}\theta + \sigma_T^{-1}]$ contains the interval $[0, 1]$. Since $\Phi([0, 1]) \ge \frac{1}{3}$, the bound $\delta \le \epsilon^2/C_1$ easily gives that (5.11) is at least $\epsilon^2/18$ as required, for a suitably large choice of C_1.

If $\theta > 1$, then using our bounds on σ_T^{-1} we have that

$$\begin{aligned}
\Phi([\sigma_T^{-1}\theta - \sigma_T^{-1}, \sigma_T^{-1}\theta + \sigma_T^{-1}]) &\ge \Phi([\sqrt{2} \cdot \theta - \sqrt{4/3}, \sqrt{2} \cdot \theta + \sqrt{4/3}) \\
&> \Phi([\sqrt{2} \cdot \theta - \sqrt{4/3}, \sqrt{2} \cdot \theta]) \\
&> \sqrt{4/3} \cdot \phi(\sqrt{2} \cdot \theta) \\
&\ge \sqrt{4/3} \cdot \phi(2\sqrt{\ln(2/\epsilon)}) \qquad (5.12) \\
&= \sqrt{\frac{4}{3}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \frac{\epsilon^2}{4} > \frac{\epsilon^2}{9}. \qquad (5.13)
\end{aligned}$$

Here (5.12) follows from Fact 122 and the fact that ϕ is decreasing, and (5.13) follows from definition of $\phi(\cdot)$. Since $\delta \le \epsilon^2/C_1$, again with a suitably large choice of C_1 we easily have $2\sqrt{2}C\delta \le \epsilon^2/18$, and thus (5.11) is at least $\epsilon^2/18$ as required and the lemma is proved. □

Now consider any fixed setting of $x_\beta, \ldots, x_n$ such that the tail $\sum_{i \in T} w_i x_i$ comes out in the interval $[\theta - 1, \theta + 1]$, say $\sum_{i \in T} w_i x_i = \theta - \tau$ where $|\tau| \le 1$. We show that the head $w_2 x_2 + \cdots + w_{\beta-1} x_{\beta-1}$ lies in $[\tau - \delta, \tau + \delta]$ with probability $\Omega(\delta)$; with Lemma 124, this implies that the overall probability (5.9) is $\Omega(\delta\epsilon^2)$.

Let $\alpha \stackrel{\text{def}}{=} C_1^2/8$, let $S \stackrel{\text{def}}{=} \{\alpha, \ldots, \beta - 1\}$, and let $R \stackrel{\text{def}}{=} \{2, \ldots, \alpha - 1\}$. Since $\delta \le \epsilon^2/C_1$, we have that $\sum_{i=1}^{\alpha-1} w_i^2 \le 1/8$, so consequently $1/8 \le \sum_{i \in S} w_i^2 \le 1/2$. Letting σ_S denote $(\sum_{i \in S} w_i^2)^{1/2}$, we have $\sqrt{2} \le \sigma_S^{-1} \le 2\sqrt{2}$.

We now consider two cases depending on the magnitude of w_α. Let $C_2 \overset{\text{def}}{=} C_1/4$.

Case 1.a: $|w_\alpha| \leq \delta/C_2$. In this case we use the Berry-Esseen theorem on S to obtain

$$\begin{aligned}\Pr\left[\sum_{i\in S} w_i x_i \in [\tau-\delta,\tau+\delta]\right] &= \Pr\left[\sigma_S^{-1}\sum_{i\in S} w_i x_i \in \sigma_S^{-1}[\tau-\delta,\tau+\delta]\right]\\ &\geq \Phi([\sigma_S^{-1}\tau-\sigma_S^{-1}\delta,\sigma_S^{-1}\tau+\sigma_S^{-1}\delta]) - 2C(\delta/C_2)\sigma_S^{-1}.\end{aligned} \tag{5.14}$$

Using our bounds on τ and σ_S^{-1}, we have that the $\Phi(\cdot)$ term of (5.14) is at least $(\sqrt{2}\delta)\cdot\phi(2\sqrt{2}) > \delta/100$. Since the error term $2C(\delta/C_2)\sigma_S^{-1}$ is at most $\delta/200$ for a suitably large choice of C_1 relative to C (recall that $C_2 = C_1/4$), we have (5.14) $\geq \delta/200$. Now for any setting of $x_\alpha,\dots,x_{\beta-1}$ such that $\sum_{i\in S} w_i x_i$ lies in $[\tau-\delta,\tau+\delta]$, since each of $|w_2|,\dots,|w_{\alpha-1}|$ is at most δ there is (at least one) corresponding setting of $x_2,\dots,x_{\alpha-1}$ such that $\sum_{i\in(R\cup S)} w_i x_i$ also lies in $[\tau-\delta,\tau+\delta]$. (Intuitively, one can think of successively setting each bit $x_{\alpha-1},x_{\alpha-2},\dots,x_j,\dots,x_2$ in such a way as to always keep $\sum_{i=j}^{\beta-1} w_i x_i$ in $[\tau-\delta,\tau+\delta]$). So the overall probability that $w_2x_2+\cdots+w_{\beta-1}x_{\beta-1}$ lies in $[\tau-\delta,\tau+\delta]$ is at least $(\delta/200)\cdot 2^{-\alpha+2} = \Omega(\delta)$, and we are done with Case 1.a.

Case 1.b: $w_\alpha > \delta/C_2$. Similar to Case 2 of [39], we again use the Berry-Esseen theorem on S, now using the bound that $|w_i| \leq \delta$ for each $i \in S$ and bounding the probability of a larger interval $[\tau - C_2\delta, \tau + C_2\delta]$:

$$\begin{aligned}&\Pr\left[\sum_{i\in S} w_i x_i \in [\tau-C_2\delta,\tau+C_2\delta]\right]\\ &\quad= \Pr\left[\sigma_S^{-1}\sum_{i\in S} w_i x_i \in \sigma_S^{-1}[\tau-C_2\delta,\tau+C_2\delta]\right]\\ &\quad\geq \Phi([\sigma_S^{-1}\tau-\sigma_S^{-1}C_2\delta,\sigma_S^{-1}\tau+\sigma_S^{-1}C_2\delta]) - 2C\delta\sigma_S^{-1} \qquad (5.15)\\ &\quad\geq \Phi([2\sqrt{2}-\sqrt{2}C_2\delta,2\sqrt{2}]) - 4\sqrt{2}C\delta \qquad (5.16)\end{aligned}$$

In (5.15) we have used the Berry-Esseen theorem and in (5.16) we have used our bounds on σ_S^{-1} and τ. Now recalling that $\delta \leq \epsilon^2/C_1 \leq 1/C_1$ and $C_2 = C_1/4$, we have $\sqrt{2}C_2\delta < 2\sqrt{2}$, and hence

$$(5.16) \geq \sqrt{2}C_2\delta \cdot \phi(2\sqrt{2}) - 4\sqrt{2}C\delta > C\delta \tag{5.17}$$

where the second inequality follows by choosing C_1 (and hence C_2) to be a sufficiently large constant multiple of C. Now for any setting of $x_\alpha, \ldots, x_{\beta-1}$ such that $\sum_{i \in S} w_i x_i = t$ lies in $[\tau - C_2\delta, \tau + C_2\delta]$, since $\delta/C_2 \leq |w_2|, \ldots, |w_{\alpha-1}| \leq \delta$, there is at least one setting of the bits $x_2, \ldots, x_{\alpha-1}$ for which $t + \sum_{i=2}^{\alpha-1} w_i x_i$ lies in $[\tau - \delta, \tau + \delta]$. (Since, as is easily verified from the definitions of α and C_2, we have $(\alpha - 2)\delta/C_2 \geq C_2\delta$, the magnitude of $w_2, \ldots, w_{\alpha-1}$ is large enough to get from $\tau - C_2\delta$ to τ; and since each $|w_i|$ is at most δ, once the interval $[\tau - \delta, \tau + \delta]$ is reached a suitable choice of signs will keep the sum in the right interval.) So in Case 1.b. the overall probability that $w_2x_2 + \cdots + w_{\beta-1}x_{\beta-1}$ lies in $[\tau - \delta, \tau + \delta]$ is at least $C\delta \cdot 2^{-\alpha+2} = \Omega(\delta)$, and we are done with Case 1.b..

We turn to the remaining case in which δ is "large:"

Case 2: $\delta > \epsilon^2/C_1$. Let us introdu ce the following definition which is implicit in [56]:

Definition 125. *Let $w_1, \ldots, w_n$ be a sequence of values such that $|w_1| \geq \cdots \geq |w_n| \geq 0$. The* critical index *of the sequence is the smallest value of $k \in [n]$ such that*

$$\frac{C|w_k|}{\sqrt{\sum_{j=k}^n w_j^2}} \leq C_3\delta\epsilon^2. \tag{5.18}$$

Here $C_3 > 0$ is a (suitably small) absolute constant specified below. (Note that the LHS value $C|w_k|/\sqrt{\sum_{j=k}^n w_j^2}$ is an upper bound on the Berry-Esseen error when the theorem is applied to $w_kx_k + \cdots + w_nx_n$.)

Throughout the rest of the proof we write k to denote the critical index of $w_1, \ldots, w_n$. Observe that $k > 1$ since we have

$$\frac{C|w_1|}{\sqrt{\sum_{j=1}^n w_j^2}} = C\delta > \frac{C\epsilon^2}{C_1} \geq \frac{C\delta\epsilon^2}{C_1} > C_3\delta\epsilon^2$$

where the final bound holds for a suitably small constant choice of C_3.

We first consider the case that the critical index k is large. In the following $C_4 > 0$ denotes a suitably large absolute constant.

Case 2.a: $k > C_4 \ln(1/\epsilon)/\epsilon^4 + 1$. In this case we define $k' \stackrel{\text{def}}{=} \lceil C_4 \ln(1/\epsilon)/\epsilon^4 \rceil + 1$. Let us also define $\sigma_{k'} \stackrel{\text{def}}{=} \sqrt{\sum_{j=k'}^n w_j^2}$. The following claim shows that $\sigma_{k'}$ is small:

Claim 126. *We have* $\sigma_{k'} \leq \frac{\epsilon^3}{10C_1}$.

Proof. For $i \in [n]$ let us write W_i to denote $\sum_{j=i}^n w_j^2$; note that $W_1 = 1$ and $W_i = w_i^2 + W_{i+1}$. For ease of notation let us write ζ to denote $\delta\epsilon^2 C_3/C$.

Since we are in Case 2.a, for any $1 \leq i < k'$ we have $w_i^2 > \zeta W_i = \zeta w_i^2 + \zeta W_{i+1}$, or equivalently $(1-\zeta)w_i^2 > \zeta W_{i+1}$. Adding $(1-\zeta)W_{i+1}$ to both sides gives $(1-\zeta)(w_i^2 + W_{i+1}) = (1-\zeta)W_i > W_{i+1}$. So consequently we have

$$W_{k'} < (1-\zeta)^{k'-1} \leq (1-\zeta)^{C_4 \ln(1/\epsilon)/\epsilon^4} \leq (1 - \epsilon^4 C_3/(CC_1))^{C_4 \ln(1/\epsilon)/\epsilon^4} \leq \left(\frac{\epsilon^3}{10C_1}\right)^2,$$

where in the third inequality we used $\delta > \epsilon^2/C_1$ (which holds since we are in Case 2) and the fourth inequality holds for a suitable choice of the absolute constant C_4. This proves the claim. □

At this point we know δ is "large" (at least ϵ^2/C_1) and $\sigma_{k'}$ is "small" (at most $\frac{\epsilon^3}{10C_1}$). We consider two cases depending on whether θ is large or small.

Case 2.a.i: $\theta < \epsilon^2/(2C_1)$. In this case we have $0 \leq \theta < \delta/2$. Since $4\sigma_{k'} < \epsilon^2/(2C_1) < \delta/2$, the Hoeffding bound gives that a random restriction that fixes variables $x_{k'}, \dots, x_n$ gives $|w_{k'}x_{k'} + \cdots + w_n x_n| > 4\sigma_{k'}$ with probability at most $e^{-8} < 1/100$. Consequently we have that for at least $99/100$ of all restrictions ρ to $x_{k'}, \dots, x_n$, the resulting function f_ρ (on variables $x_1, \dots, x_{k'-1}$) is $f_\rho(x) = \text{sgn}(w_1 x_1 + \cdots + w_{k'-1}x_{k'-1} - \theta_\rho)$ where $-\delta/2 \leq \theta_\rho < \delta$. Facts 119 and 120 now imply that each such f_ρ has $\text{Inf}_{f_\rho}(1) = \Omega(1)/k' = \Omega(1) \cdot \epsilon^4/\ln(1/\epsilon)$, so consequently $\text{Inf}_f(1)$ is also $\Omega(1) \cdot \epsilon^4/\log(1/\epsilon)$, which certainly suffices for Theorem 118. This concludes Case 2.a.i.

Case 2.a.ii: $\theta \geq \epsilon^2/(2C_1)$. We now apply the Hoeffding bound (Theorem 121) to $w_{k'}x_{k'} + \cdots + w_n x_n$ with $\gamma = 2\sqrt{\ln(8/\epsilon)}$. This gives that $w_{k'}x_{k'} + \cdots + w_n x_n < -2\sqrt{\ln(8/\epsilon)} \cdot \sigma_{k'}$

with probability at most $\epsilon^2/8$. Since $2\sqrt{\ln(8/\epsilon)} \cdot \sigma_{k'} < \epsilon^2/(2C_1) \leq \theta$, we have that for at least a $1 - \epsilon^2/8$ fraction of all restrictions ρ to $x_{k'}, \ldots, x_n$, the resulting function f_ρ (on variables $x_1, \ldots, x_{k'-1}$) is $f_\rho(x) = \mathrm{sgn}(w_1 x_1 + \cdots + w_{k'-1}x_{k'-1} - \theta_\rho)$ where $\theta_\rho > 0$. i.e. $\mathbf{E}[f_\rho] < 0$. Together with Fact 123, this implies that for at least an $\epsilon/4 - \epsilon^2/8 > \epsilon/8$ fraction of restrictions ρ, we have $-1 + \epsilon/4 \leq \mathbf{E}[f_\rho] < 0$. Each such f_ρ has $\mathbf{V}[f_\rho] = \Omega(\epsilon)$, so by Fact 119 has $\mathrm{Inf}_{f_\rho}(1) = \Omega(\epsilon)/k' = \Omega(\epsilon^5/\log(1/\epsilon))$. Consequently we have that $\mathrm{Inf}_f(1) = \Omega(\epsilon^6/\log(1/\epsilon))$ which is certainly $\Omega(\delta\epsilon^6/\log(1/\epsilon))$. This concludes Case 2.a.ii.

Case 2.b: $k \leq C_4 \log(1/\epsilon)/\epsilon^4 + 1$. We now define $\sigma_k \stackrel{\mathrm{def}}{=} \sqrt{\sum_{j=k}^n w_j^2}$ and work with this quantity. First we consider a subcase in which σ_k is "small" relative to θ; this case can be handled using essentially the same arguments as Case 2.a.ii.

Case 2.b.i: $\sigma_k < \theta/(2\sqrt{\ln(8/\epsilon)})$. As above, the Hoeffding bound (now applied to $w_k x_k + \cdots + w_n x_n$) gives that $w_k x_k + \cdots + w_n x_n < -2\sqrt{\ln(8/\epsilon)} \cdot \sigma_k$ with probability at most $\epsilon^2/8$, so for at least a $1 - \epsilon^2/8$ fraction of restrictions ρ to $x_k, \ldots, x_n$ we have $\mathbf{E}[f_\rho] < 0$. Using Fact 123, the argument from Case 2.a.ii again gives that $\mathrm{Inf}_f(1) = \Omega(\epsilon^6/\log(1/\epsilon))$, and we are done with Case 2.b.i.

Case 2.b.ii: $\sigma_k \geq \theta/(2\sqrt{\ln(8/\epsilon)})$. In this case we shall show that $N(0, \sigma_k)$, the zero-mean Gaussian distribution with variance σ_k, assigns at least $2C_3\delta\epsilon^2$ probability weight to the interval $[\theta - \delta/2, \theta + \delta/2]$. In other words, writing Φ_{σ_k} to denote the c.d.f. of $N(0, \sigma_k)$, we shall show

$$\Phi_{\sigma_k}([\theta - \delta/2, \theta + \delta/2]) \geq 3C_3\delta\epsilon^2. \tag{5.19}$$

Given (5.19), by the Berry-Esseen theorem and the definition of the critical index we obtain

$$\Pr\left[\sum_{i=k}^n w_k \in [\theta - \delta/2, \theta + \delta/2]\right] \geq 3C_3\delta\epsilon^2 - 2C_3\delta\epsilon^2 = C_3\delta\epsilon^2. \tag{5.20}$$

For any restriction ρ that gives $w_k x_k + \cdots + w_n x_n \in [\theta - \delta/2, \theta + \delta/2]$, Fact 120 gives $\mathbf{V}[f_\rho] = \Omega(1)$ and hence Fact 119 gives $\mathrm{Inf}_{f_\rho}(1) = \Omega(1)/k = \Omega(\epsilon^4/\log(1/\epsilon))$. By (5.20) we thus have $\mathrm{Inf}_f(1) = \Omega(C_3\delta\epsilon^6 \log(1/\epsilon))$, which is the desired result.

We turn to proving (5.19). Let ϕ_{σ_k} denote the c.d.f. of $N(0, \sigma_k)$, i.e. $\phi_{\sigma_k}(x) \stackrel{\mathrm{def}}{=}$

$(1/\sigma_k\sqrt{2\pi})e^{-x^2/2\sigma_k^2}$. We first observe that since $\sigma_k \geq \theta/(2\sqrt{\ln 8/\epsilon})$, we have

$$\phi_{\sigma_k}(\theta) \geq \Omega(1/\sigma_k) \cdot \epsilon^2 \geq 6C_3\epsilon^2, \tag{5.21}$$

where the second bound holds for a suitably small choice of the absolute constant C_3 and uses $\sigma_k \leq 1$.

We consider two different cases depending on the relative sizes of δ and θ.

Case 2.b.ii.A: $\delta/2 \geq \theta$. In this case we have that $[0, \delta/2] \subseteq [\theta-\delta/2, \theta+\delta/2]$ and it suffices to show that $\Phi_{\sigma_k}([0, \delta/2]) \geq 3\delta\epsilon^2 C_3$.

If $\delta \geq \sigma_k$, then we have

$$\Phi_{\sigma_k}([0, \delta/2]) \geq \Phi_{\sigma_k}([0, \sigma_k/2]) \geq 3C_3 \geq 3C_3\delta\epsilon^2$$

by a suitable choice of the absolute constant C_3. On the other hand, if $\delta < \sigma_k$ then we have

$$\Phi_{\sigma_k}([0, \delta/2]) \geq (\delta/2)\phi_{\sigma_k}(\delta/2) \geq (\delta/2)\phi_{\sigma_k}(\sigma_k/2) \geq 3C_3\delta \geq 3C_3\delta\epsilon^2$$

for a suitable choice of the absolute constant C_3. This gives Case 2.b.ii.A.

Case 2.b.ii.B: $\delta/2 < \theta$. In this case we have

$$\Phi_{\sigma_k}([\theta - \delta/2, \theta + \delta/2]) \geq \Phi_{\sigma_k}([\theta - \delta/2, \theta]) \geq (\delta/2) \cdot \phi_{\sigma_k}(\theta) \geq 3C_3\delta\epsilon^2$$

where the final inequality is obtained using (5.21). This concludes Case 2.b.ii.B, and with it the proof of Theorem 118. □

5.5.2 Two theorems about LTFs

In this section we prove two theorems that essentially characterize LTFs. These theorems are the analogues of Theorems 112 and 113 in Section 5.4.2.

The following is the main theorem used in proving the completeness of our test. Roughly speaking, it says that if f_1 and f_2 are two regular LTFs with the same weights (but possibly different thresholds), then the the inner product of their degree-1 Fourier coefficients is essentially determined by their means.

Theorem 127. *Let f_1 be a τ-regular LTF. Then*

$$\left| \sum_{i=1}^{n} \widehat{f_1}(i)^2 - W(\mathbf{E}[f_1]) \right| \leq \tau^{1/6}. \tag{5.22}$$

Further, suppose $f_2 : \{-1,1\}^n \rightarrow \{-1,1\}$ is another τ-regular LTFs that can be expressed using the same linear form as f_1; i.e., $f_k(x) = \operatorname{sgn}(w \cdot x - \theta_k)$ for some w, θ_1, θ_2. Then

$$\left| \left(\sum_{i=1}^{n} \widehat{f_1}(i) \widehat{f_2}(i) \right)^2 - W(\mathbf{E}[f_1]) W(\mathbf{E}[f_2]) \right| \leq \tau^{1/6}. \tag{5.23}$$

(We assume in this theorem that τ is less than a sufficiently small constant.)

Proof. We first dispense with the case that $|\mathbf{E}[f_1]| \geq 1 - \tau^{1/10}$. In this case, Proposition 2.2 of Talagrand [59] implies that $\sum_{i=1}^{n} \widehat{f_1}(i)^2 \leq O(\tau^{2/10} \log(1/\tau))$, and Proposition 98 (item 3) implies that $W(\mathbf{E}[f_1]) \leq O(\tau^{2/10} \log(1/\tau))$. Thus

$$\left| \sum_{i=1}^{n} \widehat{f_1}(i)^2 - W(\mathbf{E}[f_1]) \right| \leq O(\tau^{1/5} \log(1/\tau)) \leq \tau^{1/6},$$

so (5.22) indeed holds. Further, in this case we have

$$\left(\sum_{i=1}^{n} \widehat{f_1}(i) \widehat{f_2}(i) \right)^2 \overset{\text{Cauchy-Schwarz}}{\leq} \left(\sum_{i=1}^{n} \widehat{f_1}(i)^2 \right) \left(\sum_{i=1}^{n} \widehat{f_2}(i)^2 \right) \leq O(\tau^{1/5} \log(1/\tau)) \cdot 1,$$

and also $W(\mathbf{E}[f_1]) W(\mathbf{E}[f_2]) \leq O(\tau^{1/5} \log(1/\tau)) \cdot \frac{2}{\pi}$. Thus (5.23) holds as well.

We may now assume that $|\mathbf{E}[f_1]| < 1 - \tau^{1/10}$. Without loss of generality, assume that the linear form w defining f_1 (and f_2) has $\|w\| = 1$ and $|w_1| \geq |w_i|$ for all i. Then from Theorem 118 it follows that

$$\tau \geq \operatorname{Inf}_{f_1}(1) \geq \Omega(|w_1| \tau^{6/10} \log(1/\tau))$$

which implies that $|w_1| \leq O(\tau^{2/5})$. Note that by Proposition 108, this implies that

$$\mathbf{E}[f_k] \overset{\tau^{2/5}}{\approx} \mu(\theta_k), \quad k = 1, 2. \tag{5.24}$$

Let (x, y) denote a pair of η-correlated random binary strings, where $\eta = \tau^{1/5}$. By definition of $\mathbb{S}_\eta$, we have

$$\mathbb{S}_\eta(f_1, f_2) = 2\Pr[(w \cdot x, w \cdot y) \in A \cup B] - 1,$$

where $A = [\theta_1, \infty) \times [\theta_2, \infty)$ and $B = (-\infty, \theta_1] \times (-\infty, \theta_2]$. Using a multidimensional version of the Berry-Esseen theorem (see Theorem 111 in Section 5.4.1), the fact that $|w_i| \leq O(\tau^{2/5})$ holds for all i implies

$$\Pr[(w \cdot x, w \cdot y) \in A \cup B] \overset{\tau^{2/5}}{\approx} \Pr[(X, Y) \in A \cup B],$$

where (X, Y) is a pair of η-correlated standard Gaussians. (Note that the error in the above approximation also depends multiplicatively on constant powers of $1 + \eta$ and of $1 - \eta$, but these are just constants, since $|\eta|$ is bounded away from 1.) It follows that

$$\mathbb{S}_\eta(f_1, f_2) \overset{\tau^{2/5}}{\approx} \mathbb{S}_\eta(h_{\theta_1}, h_{\theta_2}), \tag{5.25}$$

where $h_{\theta_k} : \mathbb{R} \to \{-1, 1\}$ is the function of one Gaussian variable $h_{\theta_k}(X) = \text{sgn}(X - \theta_k)$.

Using the Fourier and Hermite expansions, we can write Equation (5.25) as follows:

$$\begin{aligned} &\widehat{f_1}(\emptyset)\widehat{f_2}(\emptyset) + \eta \cdot \left(\sum_{i=1}^{n} \widehat{f_1}(i)\widehat{f_2}(i)\right) + \sum_{|S| \geq 2} \eta^{|S|} \widehat{f_1}(S)\widehat{f_2}(S) \\ &\overset{\tau^{2/5}}{\approx} \widehat{h_{\theta_1}}(0)\widehat{h_{\theta_2}}(0) + \eta \cdot \widehat{h_{\theta_1}}(1)\widehat{h_{\theta_2}}(1) + \sum_{j \geq 2} \eta^j \widehat{h_{\theta_1}}(j)\widehat{h_{\theta_2}}(j). \end{aligned} \tag{5.26}$$

Now by Cauchy-Schwarz (and using the fact that $\eta \geq 0$) we have

$$\begin{aligned}\left|\sum_{|S|\geq 2} \eta^{|S|}\widehat{f_1}(S)\widehat{f_2}(S)\right| &\leq \sqrt{\sum_{|S|\geq 2} \eta^{|S|}\widehat{f_1}(S)^2}\sqrt{\sum_{|S|\geq 2} \eta^{|S|}\widehat{f_2}(S)^2} \\ &\leq \eta^2\sqrt{\sum_S \widehat{f_1}(S)^2}\sqrt{\sum_S \widehat{f_2}(S)^2} \\ &= \eta^2.\end{aligned}$$

The analogous result holds for h_{θ_1} and h_{θ_2}. If we substitute these into Equation (5.26) and also use

$$\widehat{h_{\theta_k}}(0) = \mathbf{E}[h_{\theta_k}] = \mu(\theta_k) \overset{\tau^{2/5}}{\approx} \mathbf{E}[f_k] = \widehat{f_k}(\emptyset)$$

which follows from Equation (5.24), we get:

$$\eta \cdot \left(\sum_{i=1}^n \widehat{f_1}(i)\widehat{f_2}(i)\right) \overset{\tau^{2/5}+\eta^2}{\approx} \eta \cdot \widehat{h_{\theta_1}}(1)\widehat{h_{\theta_2}}(1) = \eta \cdot 2\phi(\theta_1) \cdot 2\phi(\theta_2),$$

where the equality is by the comment following Definition 96. Dividing by η and using $\tau^{2/5}/\eta + \eta = 2\tau^{1/5}$ in the error estimate, we get

$$\sum_{i=1}^n \widehat{f_1}(i)\widehat{f_2}(i) \overset{\tau^{1/5}}{\approx} 2\phi(\theta_1) \cdot 2\phi(\theta_2) = \sqrt{W(\mu(\theta_1))W(\mu(\theta_2))}. \qquad (5.27)$$

Since we can apply this with f_1 and f_2 equal, we may also conclude

$$\sum_{i=1}^n \widehat{f_k}(i)^2 \overset{\tau^{1/5}}{\approx} W(\mu(\theta_k)) \qquad (5.28)$$

for each $k = 1, 2$.

Using the Mean Value Theorem, the fact that $|W'| \leq 1$ on $[-1, 1]$, and Equation (5.24), we conclude

$$\sum_{i=1}^n \widehat{f_k}(i)^2 \overset{\tau^{1/5}}{\approx} W(\mathbf{E}[f_k])$$

for each $k = 1, 2$, establishing (5.22). Similar reasoning applied to the square of Equation (5.27) yields

$$\left(\sum_{i=1}^{n} \widehat{f_1}(i)\widehat{f_2}(i)\right)^2 \overset{\tau^{1/5}}{\approx} W(\mathbf{E}[f_1])W(\mathbf{E}[f_2]),$$

implying (5.23). The proof is complete. □

The next theorem is a sort of converse of the previous theorem, and will be the main theorem we use in proving the soundness of our test. The previous theorem stated that if f and g were LTFs with the same weights, the inner product of their degree-1 fourier coefficients is close to a particular value. Roughly speaking, this theorem says that for any Boolean function g and any τ-regular Boolean function f that satisfies certain conditions, if the inner product of the degree-1 Fourier coefficients of f and g is close to the "right" value (from the preveious theorem), then g is close to an LTF (in particular the LTF whose weights are the degree-1 Fourier coefficients of f.).

Theorem 128. *Let $f, g : \{-1,1\}^n \to \{-1,1\}$, and suppose that:*

1. *f is τ-regular and $|\mathbf{E}[f]| \le 1 - \tau^{2/9}$;*

2. *$|\sum_{i=1}^{n} \hat{f}(i)^2 - W(\mathbf{E}[f])| \le \tau$;*

3. *$|(\sum_{i=1}^{n} \hat{f}(i)\hat{g}(i))^2 - W(\mathbf{E}[f])W(\mathbf{E}[g])| \le \tau$, and $\sum_{i=1}^{n} \hat{f}(i)\hat{g}(i) \ge -\tau$.*

Write $\ell(x)$ for the linear form $\sum_{i=1}^{n}(\hat{f}(i)/\sigma)x_i$, where $\sigma = \sqrt{\sum_{i=1}^{n} \hat{f}(i)^2}$. Then there exists $\theta \in \mathbb{R}$ such that $g(x)$ is $O(\tau^{1/9})$-close to the function $\mathrm{sgn}(\ell(x) - \theta)$. Moreover, we have that each coefficient $(\hat{f}(i)/\sigma)$ of $\ell(x)$ is at most $O(\tau^{7/9})$.

Proof. We may assume $|\mathbf{E}[g]| \le 1 - \tau^{1/9}$, since otherwise g is $\tau^{1/9}$-close to a constant function, which may of course be expressed in the desired form. Using this assumption, the fact that $|\mathbf{E}[f]| \le 1 - \tau^{2/9}$, and the final item in Proposition 98, it follows that

$$W(\mathbf{E}[g]) \ge \Omega(\tau^{2/9}) \quad \text{and} \quad W(\mathbf{E}[f]) \ge \Omega(\tau^{4/9}). \tag{5.29}$$

The latter above, combined with assumption 2 of the theorem, also yields

$$\sigma \ge \Omega(\tau^{2/9}). \tag{5.30}$$

Note that the second assertion of the theorem follows immediately from the τ-regularity of f and (5.30).

Let $\theta = \mu^{-1}(\mathbf{E}[g])$. We will show that g is $O(\tau^{1/9})$-close to $\mathrm{sgn}(h)$, where $h(x) = \ell(x) - \theta$, and thus prove the first assertion of the theorem.

Let us consider $\mathbf{E}[gh]$. By Plancherel and the fact that h is affine, we have

$$\mathbf{E}[gh] = \sum_{|S|\le 1} \hat{g}(S)\hat{h}(S) = \sum_{i=1}^{n} \frac{\hat{g}(i)\hat{f}(i)}{\sigma} - \theta\,\mathbf{E}[g]. \tag{5.31}$$

On the other hand,

$$\mathbf{E}[gh] \le \mathbf{E}[|h|] \overset{\tau}{\approx} \mathbf{E}[|X-\theta|] = 2\phi(\theta) - \theta\mu(\theta) = \sqrt{W(\mathbf{E}[g])} - \theta\,\mathbf{E}[g], \tag{5.32}$$

where the inequality is because g is ±1-valued, the following approximation is by Proposition 109, the following equality is by Proposition 98, and the last equality is by definition of θ. Combining Equation (5.31) and Equation (5.32) we get

$$\mathbf{E}[|h|] - \mathbf{E}[gh] \le \left(\sqrt{W(\mathbf{E}[g])} - \sum_{i=1}^{n} \frac{\hat{g}(i)\hat{f}(i)}{\sigma}\right) + O(\tau). \tag{5.33}$$

We now wish to show the parenthesized expression in (5.33) is small. Using Fact 101 and the first part of assumption 3 of the theorem, we have

$$\left|\left|\sum_{i=1}^{n} \hat{f}(i)\hat{g}(i)\right| - \sqrt{W(\mathbf{E}[f])}\sqrt{W(\mathbf{E}[g])}\right| \le \frac{\tau}{\sqrt{W(\mathbf{E}[f])}\sqrt{W(\mathbf{E}[g])}} \le O(\tau^{6/9}), \tag{5.34}$$

where we used (5.29) for the final inequality. We can remove the inner absolute value on the left of (5.34) by using the second part of assumption 3 and observing that 2τ is negligible compared with $O(\tau^{6/9})$, i.e. we obtain

$$\left|\sum_{i=1}^{n} \hat{f}(i)\hat{g}(i) - \sqrt{W(\mathbf{E}[f])}\sqrt{W(\mathbf{E}[g])}\right| \le O(\tau^{6/9}), \tag{5.35}$$

We can also use Fact 101 and the first part of assumption 2 of the theorem to get $|\sigma -$

$\sqrt{W(\mathbf{E}[f])}| \leq \tau/\sqrt{W(\mathbf{E}[f])} \leq O(\tau^{7/9})$. Since $|W(\mathbf{E}[g])| = O(1)$, we thus have

$$\left|\sigma\sqrt{W(\mathbf{E}[g])} - \sqrt{W(\mathbf{E}[f])}\sqrt{W(\mathbf{E}[g])}\right| \leq O(\tau^{7/9}). \tag{5.36}$$

Combining (5.36) and (5.35), we have

$$\left|\sum_{i=1}^{n} \hat{f}(i)\hat{g}(i) - \sigma\sqrt{W(\mathbf{E}[g])}\right| \leq O(\tau^{6/9}).$$

Dividing through by σ and using (5.30), this gives that

$$\left|\sum_{i=1}^{n} \frac{\hat{g}(i)\hat{f}(i)}{\sigma} - \sqrt{W(\mathbf{E}[g])}\right| \leq O(\tau^{4/9}).$$

Substituting this into (5.33) yields

$$\mathbf{E}[|h|] - \mathbf{E}[gh] \leq O(\tau^{4/9}). \tag{5.37}$$

Let ϵ denote the fraction of points in $\{-1,1\}^n$ on which g and $\mathrm{sgn}(h)$ disagree. Suppose first that that $\epsilon < 12\tau/\sigma$. Since $\sigma \geq \Omega(\tau^{2/9})$ by (5.30), in this case we have that $\epsilon \leq O(\tau^{7/9})$. Thus we may assume that $\epsilon \geq 12\tau/\sigma$. We may apply Theorem 107 as follows since $\epsilon\sigma/12 \geq \tau \geq \max_i |\hat{f}(i)|$:

$$\Pr[|h(x)| \leq \epsilon\sigma/12] \leq \frac{6\epsilon\sigma/12}{\sigma} = \frac{\epsilon}{2}.$$

It follows that at least an $\epsilon/2$ fraction of inputs x have both $g(x) \neq \mathrm{sgn}(h(x))$ and $|h(x)| > \epsilon\sigma/12$. This implies that $\mathbf{E}[|h|] - \mathbf{E}[gh] \geq 2 \cdot (\epsilon/2) \cdot (\epsilon\sigma/12) = \epsilon^2\sigma/12$. Combining this with the previous bound (5.37), and recalling that $\sigma \geq \Omega(\tau^{2/9})$, we get that $\epsilon^2 \leq O(\tau^{2/9})$ and thus $\epsilon \leq O(\tau^{1/9})$. This proves that g is $O(\tau^{1/9})$-close to $\mathrm{sgn}(h)$, as desired. □

5.5.3 Overview of the testing algorithm

We are given $\epsilon > 0$ and black-box access to an unknown $f : \{-1,1\}^n \to \{-1,1\}$, and our goal is to test whether f is an LTF versus ϵ-far from every LTF.

Our testing algorithm **Test-LTF** operates in three phases. The first two phases make queries to the black-box function f; the third phase is a deterministic test making no queries.

In the first phase the algorithm "isolates" a set J that consists of s "influential" coordinates. Essentially, this set J consists of those coordinates i such that $|\hat{f}(i)|$ is large. We call this phase **Isolate-Variables**; in Section 5.5.4 we present the **Isolate-Variables** algorithm and prove a theorem describing its behavior.

We note that one can show that it is possible to *identify* a set J as described above using $\Theta(\log n)$ queries using an approach based on binary search. However, since we want to use a number of queries that is independent of n, we cannot actually afford to explicitly identify the set J (note that indeed this set J is not part of the output that **Isolate-Variables** produces). The approach we use to "isolate" J without identifying it is based in part on ideas from [25].

In the second phase, the algorithm generates a set $\pi^1, \dots, \pi^M$ of i.i.d. uniform random strings in $\{-1,1\}^s$; these strings will play the role of restrictions of J. The algorithm then uses the output of **Isolate-Variables** to estimate various parameters of the restricted functions $f_{\pi^1}, \dots, f_{\pi^M}$. More specifically, for each restriction π^i, the algorithm estimates the mean $\mathbf{E}[f_{\pi^i}]$, the sum of squares of degree-1 Fourier coefficients $\sum_k \widehat{f_{\pi^i}}(k)^2$, and the sum of fourth powers of degree-1 Fourier coefficients $\sum_k \widehat{f_{\pi^i}}(k)^4$; and for each pair of restrictions π^i, π^j, the algorithm estimates the inner product of degree-1 Fourier coefficients $\sum_{k \notin J} \widehat{f_{\pi^i}}(k)\widehat{f_{\pi^i}}(k)$. We call this phase **Estimate-Parameters-Of-Restrictions**; see Section 5.5.4 where we present this algorithm and prove a theorem describing its behavior.

After these two query phases have been performed, in the third phase the algorithm does some computation on the parameters that it has obtained for the restrictions $\pi^1, \dots, \pi^M$, and either accepts or rejects. In Section 5.5.5 we give a description of the entire algorithm **Test-LTF** and prove Theorem 114.

Isolate-Variables (inputs are $\tau, \delta > 0$, and black-box access to $f : \{-1,1\}^n \to \{-1,1\}$)

1. Let $\ell = \lceil 1/(\tau^{16}\delta) \rceil$. Randomly partition the set $[n]$ into ℓ "bins" (subsets $B_1, \ldots, B_\ell$) by assigning each $i \in [n]$ to a uniformly selected B_j.
2. Run **Non-Regular**($\tau^2, \delta/\ell, B_j$) (see Lemma 92) on each set B_j and let I be the set of those bins B_j such that **Non-Regular** accepts. Let $s = |I|$.
3. Output $(B_1, \ldots, B_\ell, I)$.

Figure 5-1: The subroutine **Isolate-Variables**.

5.5.4 The querying portions of the algorithm

Isolating variables.

We require the following:

Definition 129. *Let $B_1, \ldots, B_\ell$ be a partition of $[n]$ and I be a subset of $\{B_1, \ldots, B_\ell\}$. We say that $(B_1, \ldots, B_\ell, I)$ is* isolationist *if the following conditions hold:*

1. If $\max_{i \in B_j} |\hat{f}(i)| \geq \tau^2$ then $B_j \in I$;

2. If $B_j \in I$ then $\max_{i \in B_j} |\hat{f}(i)| \geq \tau^2/4$;

3. If $B_j \in I$ then the second-largest value of $|\hat{f}(i)|$ for $i \in B_j$ is less than $\tau^4/32$.

Given $(B_1, \ldots, B_\ell, I)$ we define the set J to be

$$J := \bigcup_{B_j \in I} \{\operatorname*{argmax}_{k \in B_j} |\hat{f}(k)|\}. \tag{5.38}$$

The following lemma is useful:

Lemma 130. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be any function. With probability $1 - O(\delta)$, the sets $B_1, \ldots, B_\ell$ have the following property: for all j, the set B_j contains at most one element i such that $|\hat{f}(i)| \geq \tau^4/32$.*

Proof. Parseval's identity gives us that there are at most $1024/\tau^8$ many variables i such that $|\hat{f}(i)| \geq \tau^4/32$. For each such variable, the probability that any other such variable is assigned to its bin is at most $(1024/\tau^8)/\ell \leq 1024\tau^8\delta$. A union bound over all (at most $1024/\tau^8$ many) such variables gives that with probability at least $1 - O(\delta)$, each variable x_i with $|\hat{f}(i)| \geq \tau^4/32$ is the only variable that occurs in its bin. This gives the lemma. □

Theorem 131. *Let $f : \{-1, 1\}^n \to \{-1, 1\}$, and let $\tau, \delta > 0$ be given. Define $s_{\max} = 16/\tau^4$ and $\ell = \lceil 1/(\tau^{16}\delta) \rceil$. Then with probability $1 - O(\delta)$,*

1. *Algorithm* **Isolate-Variables** *outputs a list $(B_1, \ldots, B_\ell, I)$ that is isolationist;*
2. *The corresponding set J has $|J| = |I| \leq s_{\max}$, and J contains all coordinates $i \in [n]$ such that $|\hat{f}(i)| \geq \tau^2$.*

The algorithm makes $\widetilde{O}(1/(\delta\tau^{48}))$ queries to f.

Proof. Part (1) of the theorem follows from Lemma 130 and Lemma 92. Note that Lemma 130 contributes $O(\delta)$ to the failure probability, and since the algorithm runs **Non-Regular** ℓ times with confidence parameter set to δ/ℓ, Lemma 92 contributes another $O(\delta)$ to the failure probability.

We now show that if part (1) holds then so does part (2). Observe that since $(B_1, \ldots,$ $B_\ell,$ $I)$ is isolationist, for each $B_j \in I$ there is precisely one element that achieves the maximum value of $|\hat{f}(k)|$; thus $|J \cap B_j| = 1$ for all $B_j \in I$ and $|J| = |I|$. It is easy to see that $|J| \leq 16/\tau^4$; this follows immediately from Parseval's identity and part 2 of Definition 129.

For the query complexity, observe that **Isolate-Variables** makes $O(1/(\tau^{16}\delta))$ calls to **Non-Regular**$(\tau^2, \delta/\ell, B_j)$, each of which requires $\widetilde{O}(1/\tau^{32})$ queries to f, for an overall query complexity of

$$\widetilde{O}\left(\frac{1}{\delta\tau^{48}}\right)$$

queries. □

Estimating Parameters of Restrictions.

Theorem 132. *Let $f : \{-1, 1\}^n \to \{-1, 1\}$, $\tau, \eta, \delta > 0$, $M \in \mathbf{Z}^+$, and let $(B_1, \ldots, B_\ell, I)$ be an isolationist list where $|I| = s \leq s_{\max} = 16/\tau^4$. Then with probability at least $1 - \delta$,*

```
Estimate-Parameters-Of-Restrictions (inputs are $\tau, \eta, \delta > 0$, $M \in \mathbf{Z}^+$, an isolationist list $(B_1, \dots, B_\ell, I)$ where $|I| = s$, and black-box access to $f : \{-1,1\}^n \to \{-1,1\}$)
  0. Let $\delta' := O(\frac{\delta\eta^2}{M^2} \cdot \log(\frac{M^2}{\delta\eta^2}))$.
  1. For $i = 1, \dots, M$ let $\pi^i$ be an i.i.d. uniform string from $\{-1,1\}^s$.
  2. For $i = 1, \dots, M$ do the following:
      (a) Make $N_\mu := O(\log(1/\delta')/\eta^2)$ calls to Random-String($\pi^i, I, \delta', f$) to obtain $N_\mu$ strings $w$. Let $\widetilde{\mu}^i$ be the average value of $f(w)$ over the $N_\mu$ strings.
      (b) Make $N_\kappa := O(\log(1/\delta')/\eta^2)$ calls to Correlated-4Tuple($\pi^i, \pi^i, I, \delta', f, \eta$) to obtain $N_\kappa$ pairs of 4-tuples $(w^1, x^1, y^1, z^1), (w^2, x^2, y^2, z^2)$. Run algorithm Estimate-Sum-Of-Fourths on the output of these calls and let $\widetilde{\kappa}^i$ be the value it returns. If $\widetilde{\kappa}^i < 0$ or $\widetilde{\kappa}^i > 1$ then set $\widetilde{\kappa}^i$ to 0 or 1 respectively.
  3. For $i, j = 1, \dots, M$ do the following: Make $N_\rho := O(\log(1/\delta')/\eta^2)$ calls to Correlated-Pair($\pi^i, \pi^j, I, \delta', f, \eta$) to obtain $N_\rho$ pairs of pairs $(w^1, x^1), (w^2, x^2)$. Run algorithm Estimate-Inner-Product on the output of these calls and let $\widetilde{\rho}^{i,j}$ be the value it returns. If $|\widetilde{\rho}^{i,j}| > 1$ then set $\widetilde{\rho}^{i,j}$ to $\mathrm{sgn}(\widetilde{\rho}^{i,j})$.
  4. For $i = 1, \dots, M$, set $(\widetilde{\sigma}^i)^2$ to $(\widetilde{\rho}^{i,i})^2$.
```

Figure 5-2: The subroutine **Estimate-Parameters-Of-Restrictions** subroutine.

algorithm **Estimate-Parameters-Of-Restrictions** *outputs a list of tuples* $(\pi^1, \widetilde{\mu}^1, \widetilde{\sigma}^1, \widetilde{\kappa}^1)$, $\dots$, $(\pi^M, \widetilde{\mu}^M, \widetilde{\sigma}^M, \widetilde{\kappa}^M)$ *and a matrix* $(\widetilde{\rho}^{i,j})_{1 \le i,j \le M}$ *with the following properties:*

1. *Each* π^i *is an element of* $\{-1,1\}^s$*; further, the strings* $(\pi^i)_{i \ge 1}$ *are i.i.d. uniform elements of* $\{-1,1\}^s$.

2. *The quantities* $\widetilde{\mu}^i, \widetilde{\rho}^{i,j}$ *are real numbers in the range* $[-1,1]$, *and the quantities* $\widetilde{\sigma}^i$, $\widetilde{\kappa}^i$, *are real numbers in the range* $[0,1]$.

3. *For the set* J *corresponding to* $(B_1, \dots, B_\ell, I)$ *as in (5.38), the following properties hold. (In (a)-(d) below,* f_{π^i} *denotes the restricted function obtained by substituting* π^i*'s bits for the coordinates of* J *as follows: for each* $k = 1, \dots, s$, *the restriction assigns the value* π^i_k *to the (unique) variable in* $J \cap B_k$*.)*

   (a) *For each* $i = 1, \dots, M$,

$$|\widetilde{\mu}^i - \mathbf{E}[f_{\pi^i}]| \le \eta.$$

(b) For each $i = 1, \ldots, M$,

$$|\widetilde{\kappa}^i - \sum_{|S|=1} \widehat{f_{\pi^i}}(S)^4| \leq \eta.$$

(c) For all $1 \leq i, j \leq M$,

$$|\widetilde{\rho}^{i,j} - \sum_{|S|=1} \widehat{f_{\pi^i}}(S)\widehat{f_{\pi^j}}(S)| \leq \eta.$$

(d) For each $i = 1, \ldots, M$,

$$|(\widetilde{\sigma}^i)^2 - \sum_{|S|=1} \widehat{f_{\pi^i}}(S)^2| \leq \eta.$$

The algorithm makes $\widetilde{O}\left(\frac{M^2}{\eta^2 \tau^{36}}\right)$ queries to f.

Proof of Theorem 132.

The proof of Theorem 132 follows as a sequence of lemmas. First a word of terminology: for $x \in \{-1,1\}^n$, and π a restriction of the variables in J, we say that x *is compatible with* π if for every $j \in J$ the value of x_j is the value assigned to variable j by π.

The goal of Step 2(a) is to obtain estimates $\widetilde{\mu}^i$ of the means $\mathbf{E}[f_{\pi^i}]$ of the restricted functions f_{π^i}. Thus to execute Step 2(a) of **Estimate-Parameters-Of-Restrictions** we would like to be able to draw uniform strings $x \in \{-1,1\}^n$ conditioned on their being compatible with particular restrictions π^i of the variables in J. Similarly, to estimate sums of squares, fourth powers, etc. of degree-1 Fourier coefficients of restricted functions, recalling Section 5.2 we would like to be able to draw pairs, 4-tuples, etc. of bitwise correlated strings subject to their being compatible with the restriction

The subroutine **Correlated-4Tuple**, described below, lets us achieve this. (The subroutines **Random-Pair** and **Correlated-Pair** will be obtained as special cases of **Correlated-4Tuple**.) The basic approach, which is taken from [25], is to work with each block B_j separately: for each block we repeatedly draw correlated assignments until we find ones that agree with the restriction on the variable of J in that block. Once assignments have been independently obtained for all blocks they are combined to obtain the final desired 4-tuple of strings. (For technical reasons, the algorithm actually generates a pair of 4-tuples as seen below.)

Lemma 133. *Each time* **Correlated-4Tuple**$(\pi^1, \pi^2, I, \delta', f)$ *is invoked by the subroutine* **Estimate-Parameters-Of-Restrictions**, *with probability* $1 - O(\delta')$ *it outputs two 4-tuples* $(w^1, x^1, y^1, z^1), (w^2, x^2, y^2, z^2)$, *each in* $(\{-1,1\}^n)^4$, *such that:*

- *For* $k = 1, 2$ *we have that* w^k, x^k, y^k *and* z^k *are all compatible with* π^k *on* J*;*
- *For* $k = 1, 2$, *for each* $i \notin J$, *the bits* $(w^k)_i$, $(x^k)_i$, $(y^k)_i$ *are each independent uniform* ± 1 *values independent of everything else;*
- *For* $k = 1, 2$, *for each* $i \notin J$, *the bit* $(z^k)_i$ *is independently equal to* $(w^1)_i \odot (x^1)_i \odot (y^1)_i$ *with probability* $\frac{1}{2} + \frac{1}{2}\eta$.

Proof. We will assume that the set I is isolationist, since **Correlated-4Tuple** is only invoked by **Estimate-Parameters-Of-Restrictions** with isolationist I. Fix any $B_j \in I$, and consider a particular execution of Step 1(a). Let ℓ_j denote the unique element of $J \cap B_j$. By Definition 129 we have that $|\hat{f}(\ell_j)| \geq \tau^2/4$ and $|\hat{f}(k)| < \tau^4/32$ for all $k \in B_j$ such that $k \neq \ell_j$. Now consider the corresponding execution of Step 1(b). Assuming that **Non-Regular** does not make an error, if $\ell_j \in P$ then **Non-Regular** will accept by Lemma 92, and if $\ell_j \notin P$ then by Lemma 92 we have that **Non-Regular** will reject. It is not hard to see (using the fact that $\eta \geq 0$) that the element ℓ_j belongs to P with probability $\Theta(1)$, so the probability that $O(\log(s/\delta'))$ repetitions of 1(a) and 1(b) will pass for a given B_j without any "accept" occurring is at most $c^{O(\log(s/\delta'))}$, where c is an absolute constant less than 1. Thus the total failure probability resulting from step 2 ("stop everything and fail") is at most $s2^{-O(\log(s/\delta'))} \leq \delta'$. Since each invocation of **Non-Regular** errs with probability at most $\delta'/(s\log(s/\delta'))$ and there are $O(s\log(s/\delta))$ invocations, the total failure probability from the invocations of **Non-Regular** is at most $O(\delta')$.

Once Step 3 is reached, we have that for each j,

- Each of w^{jk}, x^{jk}, y^{jk} is a uniform independent assignment to the variables in B_j conditioned on $(w^{jk})_{\ell_j}, (x^{jk})_{\ell_j}, (y^{jk})_{\ell_j}$ each being set according to the restriction π^k;

Correlated-4Tuple (Inputs are $\pi^1, \pi^2 \in \{-1,1\}^s$, a set I of s bins, $\delta' > 0$, black-box access to $f : \{-1,1\}^n \to \{-1,1\}$, and $\eta \geq 0$. Outputs are two 4-tuples (w^1, x^1, y^1, z^1) and (w^2, x^2, y^2, z^2), each in $(\{-1,1\}^n)^4$.)

1. For each $B_j \in I$, do the following $O(\log(s/\delta'))$ times:

   (a) Draw six independent uniform assignments (call them w^{1j}, x^{1j}, y^{1j} and w^{2j}, x^{2j}, y^{2j}) to the variables in B_j. Let z^{1j} be an assignment to the same variables obtained by independently assigning each variable in B_j the same value it has in $w^{1j} \odot x^{1j} \odot y^{1j}$ with probability $\frac{1}{2} + \frac{1}{2}\eta$ and the opposite value with probability $\frac{1}{2} - \frac{1}{2}\eta$. Let z^{2j} be obtained independently exactly like z^{1j} (in particular we use $w^{1j} \odot x^{1j} \odot y^{1j}$, <u>not</u> $w^{2j} \odot x^{2j} \odot y^{2j}$, to obtain z^{2j}). Let

   $$P = \{i \in B_j : (w^{jk})_i = (x^{jk})_i = (y^{jk})_i = (z^{jk})_i = \pi_j^k \text{ for } k = 1, 2\}.$$

   i.e. P is the set of those $i \in B_j$ such that for $k = 1, 2$, assignments w^{jk}, x^{jk}, y^{jk} and z^{jk} all set bit i the same way that restriction π^k sets π_j^k.

   (b) Run **Non-Regular**$(\tau^2/4, \delta'/(s\log(s/\delta')), P, f)$.

2. If any call of **Non-Regular** above returned "accept," let $(w^{1j}, x^{1j}, y^{1j}, z^{1j})$, $(w^{2j}, x^{2j}, y^{2j}, z^{2j})$ denote the pair of assignments corresponding to the call that accepted. If no call returned "accept," stop everything and FAIL.

3. For $k = 1, 2$ let (w^k, x^k, y^k, z^k) be obtained as follows:

   - For each $i \notin \cup_{B_j \in I} B_j$, set $(w^k)_i, (x^k)_i, (y^k)_i$ independently to ± 1. Similar to 1(a) above, set both $(z^1)_i$ and $(z^2)_i$ independently to $w_i^1 \odot x_i^1 \odot y_i^1$ with probability $\frac{1}{2} + \frac{1}{2}\eta$.
   - For each bin $B_j \in I$, set the corresponding bits of w according to w^j; the corresponding bits of x according to x^j; the corresponding bits of y according to y^j; and the corresponding bits of z according to z^j.

Return the 4-tuples (w^1, x^1, y^1, z^1) and (w^2, x^2, y^2, z^2).

Figure 5-3: The subroutine **Correlated-4Tuple**.

- Each bit $z_{\ell_j}^{jk}$ is compatible with π_j^k. For each variable $i \neq \ell_j$ in B_j, the bit z_i^{jk} is independently set to $w_i^{j1} \odot x_i^{j1} \odot y_i^{j1}$ with probability $\frac{1}{2} + \frac{1}{2}\eta$.

By independence of the successive iterations of Step 1 for different B_j's, it follows that the final output strings (w^1, x^1, y^1, z^1) and (w^2, x^2, y^2, z^2) are distributed as claimed in the lemma. □

Remark 134. *The overall algorithm* **Test-LTF** *is nonadaptive because the calls to* **Non-Regular** *(which involve queries to f) in* **Correlated-4Tuple** *are only performed for those B_j which belong to I, and the set I was determined by the outcomes of earlier calls to* **Non-Regular** *(and hence earlier queries to f). The algorithm could be made nonadaptive by modifying* **Correlated-4Tuple** *to always perform Step 1 on all ℓ blocks $B_1, \ldots, B_\ell$. Once all these queries were completed for all calls to* **Correlated-4Tuple** *(and thus all queries to f for the entire algorithm were done), the algorithm could simply ignore the results of Step 1 for those sets B_j that do not belong to I. Thus, as claimed earlier, there is an nonadaptive version of the algorithm with somewhat – but only polynomially – higher query complexity (because of the extra calls to* **Non-Regular** *for sets $B_j \notin I$).*

The subroutine **Random-String**(π^i, I, δ', f) can be implemented simply by invoking the subroutine **Correlated-4Tuple**$(\pi^i, \pi^i, I, \delta, f, 0)$ to obtain a pair (w^1, x^1, y^1, z^1) and (w^2, x^2, y^2, z^2), then discarding all components but w^1. This string w^1 is uniform conditioned on being consistent with the restriction π^i. We then easily obtain:

Lemma 135. *If $(B_1, \ldots, B_\ell, I)$ is isolationist, then with probability at least $1 - \delta_1'$ (where $\delta_1' := O(MN_\mu \delta')$), each of the M values $\widetilde{\mu}^1, \ldots, \widetilde{\mu}^M$ obtained in Step 2(a) of* **Estimate-Parameters-Of-Restriction** *satisfies $|\widetilde{\mu}^i - \mathbf{E}[f_{\pi^i}]| \leq \eta$.*

Proof. Step 2(a) makes a total of MN_μ many calls to **Correlated-4Tuple**, each of which incurs failure probability $O(\delta')$. Assuming the calls to **Correlated-4Tuple** all succeed, by the choice of N_μ each of the M applications of the Chernoff bound contributes another δ' to the failure probability, for an overall failure probability as claimed. □

Now we turn to part 3(b) of Theorem 132, corresponding to Step 2(b) of **Estimate-Parameters-Of-Restrictions**. We have:

Lemma 136. *There is an algorithm* **Estimate-Sum-Of-Fourths** *with the following property: Suppose the algorithm is given as input values* $\eta, \delta > 0$*, black-box access to* f*, and the output of* N_κ *many calls to* **Correlated-4Tuple**$(\pi, \pi, I, \delta, f, \eta)$*. Then with probability* $1 - \delta$ *the algorithm outputs a value* v *such that*

$$|v - \sum_{k \in [n], k \notin J} \widehat{f_\pi}(k)^4| \leq \eta.$$

Proof. The algorithm is essentially that of Lemma 89. Consider the proof of Lemma 89 in the case where there is only one function f_π and $p = 4$. For (5.1), we would like to empirically estimate $\mathbf{E}[f_\pi(\alpha^1) f_\pi(\alpha^2) f_\pi(\alpha^3) f_\pi(\alpha^4)]$ where $\alpha^1, \ldots, \alpha^4$ are independent uniform strings conditioned on being compatible with π. Such strings can be obtained by taking each $\alpha^1 = w^1, \alpha^2 = w^2, \alpha^3 = x^1$ and $\alpha^4 = x^2$ where $(w^1, x^1, y^1, z^1), (w^2, x^2, y^2, z^2)$ is the output of a call to **Correlated-4Tuple**$(\pi, \pi, I, \delta, f, \eta)$.

For(5.2), we would like to empirically estimate

$$\mathbf{E}[f_\pi(\alpha^1) f_\pi(\alpha^2) f_\pi(\alpha^3) f_\pi(\alpha^4)]$$

where each of $\alpha^1, \alpha^2, \alpha^3$ is independent and uniform conditioned on being compatible with π, and α^4 is compatible with π and has each bit $(\alpha^4)_i$ for $i \notin J$ independently set equal to $(\alpha^1 \odot \alpha^2 \odot \alpha^3)_i$ with probability $\frac{1}{2} + \frac{1}{2}\eta$. By Lemma 133, such strings can be obtained by taking $\alpha^1 = w^1$, $\alpha^2 = x^1$, $\alpha^3 = y^1$, and $\alpha^4 = z^1$. The corollary now follows from Lemma 89. □

Observing that the two restrictions that are arguments to **Correlated-4Tuple** in Step 2(b) are both π^i, Lemma 138 directly gives us part 3(b) of Theorem 132:

Lemma 137. *If* $(B_1, \ldots, B_\ell, I)$ *is isolationist, then with probability at least* $1 - \delta_2'$ *(where* $\delta_2' := O(MN_\kappa\delta')$*), each of the* M *values* $\widetilde{\kappa}^i$ *obtained in Step 2(b) of* **Estimate-Parameters-Of-Restrictions** *satisfies* $|\widetilde{\kappa}^i - \sum_{|S|=1} \widehat{f_{\pi^i}}(S)^4| \leq \eta$.

Now we turn to parts 3(c)-(d) of Theorem 132, corresponding to Steps 3 and 4 of the algorithm. The subroutine **Correlated-Pair**$(\pi^i, \pi^j, I, \delta', f, \eta)$ works simply by invoking **Correlated-4Tuple**$(\pi^i, \pi^j, I, \delta', f, \eta)$ to obtain a pair $(w^1, x^1, y^1, z^1), (w^2, x^2, y^2, z^2)$ and outputting (u^1, z^1), (u^2, z^2) where each $u^k = (w^k \odot x^k \odot y^k)$. The following corollary of Lemma 89 describes the behavior of algorithm **Estimate-Inner-Product**:

Lemma 138. *There is an algorithm* **Estimate-Inner-Product** *with the following property: Suppose the algorithm is given as input values $\eta, \delta > 0$, black-box access to f, and the output of N_ρ many successful calls to* **Correlated-Pair**$(\pi^1, \pi^2, I, \delta, f, \eta)$. *Then with probability $1 - \delta$ the algorithm outputs a value v such that*

$$|v - \sum_{k \in [n], k \notin J} \widehat{f_{\pi^1}}(k)\widehat{f_{\pi^2}}(k)| \leq \eta.$$

Proof. Again the algorithm is essentially that of Lemma 89. Consider the proof of Lemma 89 in the case where there are $p = 2$ functions f_{π^1} and f_{π^2}. For (5.1), we would like to empirically estimate $\mathbf{E}[f_{\pi_1}(\alpha^1) f_{\pi^2}(\alpha^2)]$ where α^1, α^2 are independent uniform strings conditioned on being compatible with restrictions π^1 and π^2 respectively. Such strings can be obtained by taking each α^k to be u^k where $(u^1, z^1), (u^2, z^2)$ is the output of a call to **Correlated-Pair**$(\pi^1, \pi^2, I, \delta, f\eta)$.

For (5.2), we would like to empirically estimate $\mathbf{E}[f_{\pi_1}(\alpha^1) f_{\pi^2}(\alpha^2)]$ where α^1 is uniform conditioned on being compatible with π^1 and α^2 is compatible with π^2 and has each bit $(\alpha^2)_i$ for $i \notin J$ independently set equal to $(\alpha^1)_i$ with probability $\frac{1}{2} + \frac{1}{2}\eta$. By Lemma 133 and the definition of **Correlated-Pair**, such strings can be obtained by taking $\alpha^1 = u^1$ and $\alpha^2 = z^2$. The corollary now follows from Lemma 89. □

Lemma 138 gives us parts 3(c)-(d) of Theorem 132:

Lemma 139. *If $(B_1, \ldots, B_\ell, I)$ is isolationist, then with probability at least $1 - \delta_3'$ (where $\delta_3' := O(M^2 N_\rho \delta')$) both of the following events occur: each of the M^2 values $(\widetilde{\rho}^{i,j})^2$ obtained in Step 3 of* **Estimate-Parameters-Of-Restrictions** *satisfies*

$$|\widetilde{\rho}^{i,j} - \sum_{|S|=1} \widehat{f_{\pi^i}}(S)\widehat{f_{\pi^j}}(S)| \leq \eta$$

and each of the M values $(\widetilde{\sigma}^i)^2$ obtained in Step 4 satisfies

$$|(\widetilde{\sigma}^i)^2 - \sum_{|S|=1} \widehat{f_{\pi^i}}(S)^2| \leq \eta.$$

This essentially concludes the proof of parts 1-3 of Theorem 132. The overall failure probability is $O(\delta_1' + \delta_2' + \delta_3')$; by our initial choice of δ' this is $O(\delta)$.

It remains only to analyze the query complexity. It is not hard to see that the query complexity is dominated by Step 3. This step makes $M^2 N_\rho = \widetilde{O}(M^2/\eta^2)$ invocations to **Correlated-4Tuple**$(\pi^i, \pi^j, I, \delta', f, \eta)$; at each of these invocations **Correlated-4Tuple** makes at most

$$O(s_{\max} \log(s_{\max}/\delta') = \widetilde{O}(1/\tau^4)$$

many invocations to **Non-Regular**$(\tau^2/4, \delta', P, f)$, each of which requires

$$O(\log(s_{\max} \log(s_{\max}/\delta')/\delta')/\tau^{32})) = \widetilde{O}(1/\tau^{32})$$

queries by Lemma 92. Thus the overall number of queries is at most

$$\widetilde{O}\left(\frac{M^2}{\eta^2 \tau^{36}}\right).$$

This concludes the proof of Theorem 132. □

5.5.5 The full algorithm

We are given black-box access to $f : \{-1,1\}^n \to \{-1,1\}$, and also a "closeness parameter" $\epsilon > 0$. Our goal is to distinguish between f being an LTF and f being ϵ-far from every LTF, using $\text{poly}(1/\epsilon)$ many queries. For simplicity of exposition, we will end up distinguishing from being $O(\epsilon)$-far from every LTF. The algorithm for the test is given below, followed by a high-level conceptual explanation of the various steps it performs.

Note that all parameters described in the test are fixed polynomials in ϵ. Further, the query complexity of both **Isolate-Variables** and **Estimate-Parameters-Of-Restrictions** is polynomial in all parameters (see Theorems 131, 132). Thus the overall query complexity

Test-LTF (inputs are $\epsilon > 0$ and black-box access to $f : \{-1,1\}^n \to \{-1,1\}$)

Let $\tau = \epsilon^K$, a "regularity parameter", where K is a large universal constant to be specified later. [a] Let δ be a sufficiently small absolute constant. We will also take $\eta = \tau$ (the error parameter for **Estimate-Parameters-Of-Restrictions**), $s_{\max} = 16/\tau^4$, and $M = \text{poly}(s_{\max}) \log(1/\delta)/\epsilon^2$.

1. Run **Isolate-Variables**(τ, δ) to obtain output $(B_1, \ldots, B_\ell, I)$. This implicitly defines some set $J \subset [n]$ and explicitly defines its cardinality (the same as the cardinality of I), some s with $s \leq s_{\max}$.

2. Run **Estimate-Parameters-Of-Restrictions**$(\tau, \eta, \delta, M, (B_1, \ldots, B_\ell, I), f)$. This produces a list of restrictions $\pi^i \in \{-1,1\}^s$ and real values $\widetilde{\mu}^i$, $(\widetilde{\sigma}^i)^2, \widetilde{\kappa}^i, \widetilde{\rho}^{i,j}$ where $1 \leq i, j \leq M$.

3. At this point there are two cases depending on whether or not the fraction of i's for which $|\widetilde{\mu}^i| \geq 1 - \epsilon$ is at least $1 - \epsilon$:

   (a) (The case that for at least a $1 - \epsilon$ fraction of i's, $|\widetilde{\mu}^i| \geq 1 - \epsilon$.)
   In this case, enumerate all possible length-s integer vectors w with entries up to $2^{O(s \log s)}$ in absolute value, and also all possible integer thresholds θ in the same range. For each pair (w, θ), check whether $\text{sgn}(w \cdot \pi^i - \theta) = \text{sgn}(\widetilde{\mu}^i)$ holds for at least a $1 - 20\epsilon$ fraction of the values $1 \leq i \leq M$. If this ever holds, ACCEPT. If it fails for all (w, θ), REJECT.

   (b) (The case that for at least an ϵ fraction of i's, $|\widetilde{\mu}^i| < 1 - \epsilon$.)
   In this case, pick any i^* such that $|\widetilde{\mu}^{i^*}| < 1 - \epsilon$. Then:

      i. Check that $\widetilde{\kappa}^{i^*} \leq 2\tau$. If this fails, REJECT.
      ii. Check that $|(\widetilde{\sigma}^{i^*})^2 - W(\widetilde{\mu}^{i^*})| \leq 2\tau^{1/12}$. If this fails, REJECT.
      iii. Check that both $|(\widetilde{\rho}^{i^*,i})^2 - W(\widetilde{\mu}^{i^*})W(\widetilde{\mu}^i)| \leq 2\tau^{1/12}$ and $\widetilde{\rho}^{i^*,i} \geq -\eta$ hold for all $1 \leq i \leq M$. If this fails, REJECT.
      iv. Enumerate all possible length-s vectors w whose entries are integer multiples of $\sqrt{\tau}/s$, up to $2^{O(s \log s)}\sqrt{\ln(1/\tau)}$ in absolute value, and also all possible thresholds θ with the same properties. For each pair (w, θ), check that $|\widetilde{\mu}^i - \mu(\theta - w \cdot \pi^i)| \leq 5\sqrt{\tau}$ holds for all π^i's. If this ever happens, ACCEPT. If it fails for all (w, θ), REJECT.

[a] We will eventually take $K = 108$.

Figure 5-4: The algorithm **Test-LTF**.

is $\text{poly}(1/\epsilon)$. As given, the test is adaptive, since **Estimate-Parameters-Of-Restrictions** depends on the output of **Isolate-Variables** . However, in remark 134 we discuss how the test can easily be made nonadaptive with only a polynomial blowup in query complexity.

In Section 5.5.6 we will show that indeed this test correctly distinguishes (with probability at least $2/3$) LTFs from functions that are $O(\epsilon)$-far from being LTFs. Thus our main testing result, Theorem 114, holds as claimed.

Conceptual explanation of the test.

Here we provide a high-level description of the ideas underlying the various stages of the test. The following discussion should not be viewed in the light of mathematical statements but rather as narrative exposition to aid in understanding the test and its analysis. (It may also be useful to refer back to the sketch at the beginning of Section 5.5.)

In Step 1, the idea is that J is (roughly) the set of variables i such that $|\hat{f}(i)| \geq \tau^2$.

In Step 2, each π^i is an i.i.d. uniform random restriction of the variables in J. Each value $\widetilde{\mu}^i$ is an estimate of $\mathbf{E}[f_{\pi^i}]$, each $(\widetilde{\sigma}^i)^2$ is an estimate of $\sum_k \widehat{f_{\pi^i}}(k)^2$, each $\widetilde{\kappa}^i$ is an estimate of $\sum_k \widehat{f_{\pi^i}}(k)^4$, and each $\widetilde{\rho}^{i,j}$ is an estimate of $\sum_k \widehat{f_{\pi^i}}(k)\widehat{f_{\pi^j}}(k)$.

The idea of Step 3(a) is that in this case, almost every restriction π of the variables in J causes f_π to be very close to a constant function 1 or -1. If this is the case, then f is close to an LTF if and only if it is close to an LTF which is a junta over the variables in J. Step 3(a) enumerates over every possible LTF over the variables in J and checks each one to see if it is close to f.

If the algorithm reaches Step 3(b), then a non-negligible fraction of restrictions π have $|\mathbf{E}[f_\pi]|$ bounded away from 1. We claim that when f is an LTF, this implies that at least one of those restrictions should be τ-regular, and moreover all restrictions should be $\sqrt{\tau}$-regular (these claims are argued using Proposition 142 and Theorem 118, respectively). Step 3(b)(i) verifies that one such restriction π^{i^*} is indeed $\sqrt{\tau}$-regular.

Step 3(b)(ii) checks that the sum of squares of degree-1 Fourier coefficients $\sum_k \widehat{f_{\pi^{i^*}}}(k)^2$ is close to the "correct" value $W(\mathbf{E}[f_{\pi^{i^*}}])$ that the sum should take if $f_{\pi^{i^*}}$ were a $\sqrt{\tau}$-regular LTF (see the first inequality in the conclusion of Theorem 127). If this check passes, Step 3(b)(iii) checks that every other restriction f_{π^i} is such that the inner product of its degree-1

Fourier coefficients with those of $f_{\pi^{i^*}}$, namely $\sum_{k \notin J} \widehat{f_{\pi^i}}(k)\widehat{f_{\pi^{i^*}}}(k)$, is close to the "correct" value $W(\mathbf{E}[f_{\pi^i}])W(\mathbf{E}[f_{\pi^{i^*}}])$ that it should take if f_{π^i} and $f_{\pi^{i^*}}$ were LTFs with the same linear part (see Theorem 127 again).

At this point in Step 3(b), if all these checks have passed then every restriction f_π is close to a function of the form $\mathrm{sgn}(\ell(x) - \theta_\pi)$ with the same linear part (that is based on the degree-1 Fourier coefficients of $f_{\pi^{i^*}}$, see Theorem 128). Finally, Step 3(b)(iv) exhaustively checks "all" possible weight vectors w for the variables in J to see if there is any weight vector that is consistent with all restrictions f_{π^i}. The idea is that if f passes this final check as well, then combining w with ℓ we obtain an LTF that f must be close to.

5.5.6 Proving correctness of the test

In this section we prove that the algorithm **Test-LTF** is both complete and sound. At many points in these arguments we will need that our large sample $\pi^1, \ldots, \pi^M$ of i.i.d. uniform restrictions is representative of the whole set of all 2^s restrictions, in the sense that empirical estimates of various probabilities obtained from the sample are close to the true probabilities over all restrictions. The following proposition collects the various statements of this sort that we will need. All proofs are straightforward Chernoff bounds.

Proposition 140. *After running Steps 0,1 and 2 of* **Test-LTF**, *with probability at least* $1 - O(\delta)$ *(with respect to the choice of the i.i.d.* $\pi^1, \ldots, \pi^M$*'s in* **Estimate-Parameters-Of-Restrictions***) the following all simultaneously hold:*

1. *The true fraction of restrictions* π *of* J *for which* $|\mathbf{E}[f_\pi]| \geq 1 - 2\epsilon$ *is within an additive* $\epsilon/2$ *of the fraction of the* π^i*'s for which this holds. Further, the same is true about occurrences of* $|\mathbf{E}[f_\pi]| \geq 1 - \epsilon/2$.

2. *For every pair* (w^*, θ^*), *where* w^* *is a length-*s *integer vector with entries at most* $2^{O(s \log s)}$ *in absolute value and* θ^* *is an integer in the same range, the true fraction of restrictions* π *to* J *for which*

$$|\mathbf{E}[f_\pi] - \mathrm{sgn}(w^* \cdot \pi - \theta^*)| \leq 3/5$$

is within an additive ϵ of the fraction of π^i's for which this holds. Further, the same is true about occurrences of $\operatorname{sgn}(\mathbf{E}[f_\pi]) = \operatorname{sgn}(w^* \cdot \pi - \theta^*)$.

3. *For every fixed restriction π^* to J, the true fraction of restrictions π to J for which we have*
$$|(\sum_{|S|=1} \widehat{f_{\pi^*}}(S)\widehat{f_\pi}(S))^2 - W(\mathbf{E}[f_{\pi^*}])W(\mathbf{E}[f_\pi])| \leq 3\tau^{1/12}$$
is within an ϵ fraction of the true fraction of π^i's for which this holds.

4. *For every fixed pair (w^*, θ^*), where w^* is a length-s vector with entries that are integer multiples of $\sqrt{\tau}/s$ at most $2^{O(s\log s)}\sqrt{\ln(1/\tau)}$ in absolute value and θ^* is an integer multiple of $\sqrt{\tau}/s$ in the same range, the true fraction of restrictions π to J for which*
$$|\,\mathbf{E}[f_\pi] - \mu(\theta^* - w^* \cdot \pi)| \leq 6\sqrt{\tau}$$
is within an additive ϵ of the fraction of π^i's for which this holds.

Proof. All of the claimed statements can be proved simply by using Chernoff bounds (using the fact that the π^i's are i.i.d. and M is large enough) and union bounds. For example, regarding item 4, for any particular (w^*, θ^*), a Chernoff bound implies that the true fraction and the empirical fraction differ by more than ϵ with probability at most $\exp(-\Omega(\epsilon^2 M)) \leq \delta/2^{\mathrm{poly}(s)}$, using the fact that $M \geq \mathrm{poly}(s)\log(1/\delta)/\epsilon$. Thus we may union bound over all $2^{\mathrm{poly}(s)}$ possible (w^*, θ^*) to get that the statement of item 4 holds except with probability at most δ. The other statement and the other items follow by similar or easier considerations. □

Completeness of the test.

Theorem 141. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be any LTF. Then f passes* **Test-LTF** *with probability at least* $2/3$.

Proof. Steps 1 and 2 of the test, where querying to f occurs, are the places where the test has randomness. We have that Step 1 succeeds except with probability at most δ; assuming it succeeds, the set J becomes implicitly defined according to (5.38). Step 2 also succeeds except with probability at most δ; assuming it succeeds, we obtain restrictions π^i and estimates $\widetilde{\mu}^i, (\widetilde{\sigma}^i)^2, \widetilde{\kappa}^i, \widetilde{\rho}^{i,j}$ that satisfy the conclusion of Theorem 132, with $\eta := \tau$. Finally, in Proposition 140 (which relates the empirical properties of the restrictions to the true properties), all conclusions hold except with probability at most $O(\delta)$. Thus all of these assumptions together hold with probability at least $1 - O(\delta)$, which is at least $2/3$ when we take δ to be a sufficiently small constant. Note that we have not yet used the fact that f is an LTF.

We will now show that given that all of these assumptions hold, the fact that f is an LTF implies that the deterministic part of the test, Step 3, returns ACCEPT. We consider the two cases that can occur:

Case 3(a): for at least a $1 - \epsilon$ fraction of i's, $|\widetilde{\mu}^i| \geq 1 - \epsilon$. Since Theorem 132 implies that $|\widetilde{\mu}^i - \mathbf{E}[f_{\pi^i}]| \leq \eta$, and since $\eta \ll \epsilon$, in this case we have that for at least a $1-\epsilon$ fraction of the i's it holds that $|\mathbf{E}[f_{\pi^i}]| \geq 1 - \epsilon - \eta \geq 1 - 2\epsilon$. Applying Proposition 140 item 1, we get that $|\mathbf{E}[f_\pi]| \geq 1 - 2\epsilon$ for at least a $1 - 2\epsilon$ fraction of all 2^s restrictions π on J. It follows that f is $2\epsilon \cdot \frac{1}{2} + (1 - 2\epsilon) \cdot \epsilon \leq 2\epsilon$-close to being a junta on J.

We are assuming that f is an LTF, and we know that it is 2ϵ-close to being a junta on J. We can conclude from this that f is 2ϵ-close to being an *LTF* on J. To see why, assume without loss of generality that $J = \{1, \dots, r\}$. We know that the junta over $\{-1,1\}^r$ to which f is closest is given by mapping $x_1, \dots, x_r$ to the most common value of the restricted function $f_{x_1,\dots,x_r}$. But this most common value is certainly $\text{sgn}(w_1x_1 + \cdots + w_rx_r - \theta)$, since $w_{r+1}x_{r+1} + \cdots + w_nx_n$ is centered around zero.

So we know that f is 2ϵ-close to being an LTF on J. Write this LTF as $g(\pi) = \text{sgn}(w^* \cdot \pi - \theta^*)$, where w^* is an integer vector with entries at most $2^{O(s \log s)}$ in absolute value and θ^* is also an integer in this range. (Since $|J| \leq s$, any LTF on J can be expressed thus by the well-known result of Muroga *et al.* [47].) Since f is 2ϵ-close to g, we know that for at least a $1 - 10\epsilon$ fraction of the restrictions π to J, $f_\pi(x)$ takes the value $g(\pi)$ on at least a $4/5$ fraction of inputs x. I.e., $|\mathbf{E}[f_\pi] - \text{sgn}(w^* \cdot \pi - \theta^*)| \leq 3/5$ for at least a $1 - 10\epsilon$ fraction of all π's. Using Proposition 140 item 2 we conclude that $|\mathbf{E}[f_{\pi^i}] - \text{sgn}(w^* \cdot \pi^i - \theta^*)| \leq 3/5$ for at least a $1 - 20\epsilon$ fraction of the π^i's. But for these π^i's we additionally have

$|\widetilde{\mu}^i - \mathrm{sgn}(w^* \cdot \pi^i - \theta^*)| \le 3/5 + \eta < 1$ and hence $\mathrm{sgn}(\widetilde{\mu}^i) = \mathrm{sgn}(w^* \cdot \pi^i - \theta^*)$. Thus Step 3(a) returns ACCEPT once it tries (w^*, θ^*).

Case 3(b): for at least an ϵ fraction of i's, $|\widetilde{\mu}^i| < 1 - \epsilon$. In this case we need to show that Steps i.–iv. pass.

To begin, since $|\widetilde{\mu}^i - \mathbf{E}[f_{\pi^i}]| \le \eta \ll \epsilon/2$ for all i, we have that for at least an ϵ fraction of the i's, $|\mathbf{E}[f_{\pi^i}]| \le 1 - \epsilon/2$. Thus by Proposition 140 item 1, we know that among all 2^s restrictions π of J, the true fraction of restrictions for which $|\mathbf{E}[f_{\pi^i}]| \le 1 - \epsilon/2$ is at least $\epsilon/2$.

We would also like to show that for most restrictions π of J, the resulting function f_π is regular. We do this in the following proposition:

Proposition 142. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be an LTF and let $J \supseteq \{j : |\hat{f}(i)| \ge \beta\}$. Then f_π is not (β/η)-regular for at most an η fraction of all restrictions π to J.*

Proof. Since f is an LTF, $|\hat{f}(j)| = \mathrm{Inf}_f(j)$; thus every coordinate outside J has influence at most β on f. Let k be a coordinate outside of J of maximum influence. Note that since f is an LTF, k is a coordinate of maximum influence for f_π *under every restriction π to J*; this follows from Fact 116. But $\mathrm{Inf}_f(k) = \mathrm{Avg}_\pi(\mathrm{Inf}_{f_\pi}(k)) = \mathrm{Avg}_\pi(|\widehat{f_\pi}(k)|)$ and so

$$\beta \ge \mathrm{Inf}_f(k) = \mathrm{Avg}_\pi(\text{regularity of } f_\pi).$$

The result now follows by Markov's inequality. □

Continuing case 3(b), note that J contains all coordinates j with $|\hat{f}(j)| \ge \tau^2$, so we know from Proposition 142 that f_π is τ-regular for at least a $1-\tau$ fraction of the 2^s restrictions π to J. Since $\tau \ll \epsilon/2$, we conclude that there must exist some restriction π_0 to the coordinates in J for which both $|\mathbf{E}[f_{\pi_0}]| \le 1 - \epsilon/2$ and f_{π_0} is τ-regular.

Express f as $f(\pi, x) = \mathrm{sgn}(w' \cdot \pi + \ell \cdot x - \theta')$, where π denotes the inputs in J, x denotes the inputs not in J, and ℓ is normalized so that $\|\ell\| = 1$ (note that normalization is different than the typical one we've been using, hence the use of the variables w' and θ' instead of w and θ). We've established that the LTF $f_{\pi_0}(x) = \mathrm{sgn}(\ell \cdot x - (\theta' - w' \cdot \pi_0))$

has $|\mathbf{E}[f_{\pi^0}]| \le 1 - \epsilon/2$ and is τ-regular. Applying Theorem 118, we conclude that all coefficients in ℓ are, in absolute value, at most $O(\tau/(\epsilon^6 \log(1/\epsilon))) \le \Omega(\sqrt{\tau})$; here we use the fact that $K > 12$. In particular, we've established:

Claim 143. *There is a linear form ℓ with $\|\ell\| = 1$ and all coefficients of magnitude at most $\Omega(\sqrt{\tau})$, such that the following two statements hold: 1. For every restriction π to J, the LTF f_π is expressed as $f_\pi(x) = \mathrm{sgn}(\ell \cdot x - (\theta' - w' \cdot \pi))$. 2. For every restriction π to J, f_π is $\sqrt{\tau}$-regular.*

The second statement in the claim follows immediately from the first statement and Theorem 117, taking the constant in the $\Omega(\cdot)$ to be sufficiently small.

We now show that Steps 3b(i)–(iv) all pass. Since f_π is $\sqrt{\tau}$-regular for all π, in particular $f_{\pi^{i^*}}$ is $\sqrt{\tau}$-regular. Hence $\sum_{|S|=1} \widehat{f_{\pi^{i^*}}}(S)^4 \le \tau$ (see Proposition 91) and so $\widetilde{\kappa}^{i^*} \le \tau + \eta \le 2\tau$. Thus Step 3b(i) passes.

Regarding Step 3b(ii), Claim 143 implies in particular that $f_{\pi^{i^*}}$ is $\sqrt{\tau}$-regular. Hence we may apply the first part of Theorem 127 to conclude that $\sum_{|S|=1} \widehat{f_{\pi^{i^*}}}(S)^2$ is within $\tau^{1/12}$ of $W(\mathbf{E}[f_{\pi^{i^*}}])$. The former quantity is within η of $(\widetilde{\sigma}^{i^*})^2$; the latter quantity is within η of $W(\widetilde{\mu}^{i^*})$ (using $|W'| \le 1$). Thus indeed $(\widetilde{\sigma}^{i^*})^2$ is within $\tau^{1/12} + \eta + \eta \le 2\tau^{1/12}$ of $W(\widetilde{\mu}^{i^*})$, and Step 3b(ii) passes.

The fact that the first condition in Step 3b(iii) passes follows very similarly, using the second part of Theorem 127 (a small difference being that here we can only say that $W(\mathbf{E}[f_{\pi^{i^*}}])W(\mathbf{E}[f_{\pi^i}])$ is within, say, 3η of $W(\widetilde{\mu}^{i^*})W(\widetilde{\mu}^i))$. As for the second condition in Step 3b(iii), since f is an LTF, for any pair of restrictions π, π' to J, the functions f_π and $f_{\pi'}$ are LTFs expressible using the same linear form. This implies that f_π and $f_{\pi'}$ are both unate functions with the same orientation, a condition which easily yields that $\widehat{f_\pi}(j)$ and $\widehat{f_{\pi'}}(j)$ never have opposite sign for any j. We thus have that $\sum_{|S|=1} \widehat{f_{\pi^i}}(S)\widehat{f_{\pi^{i^*}}}(S) \ge 0$ and so indeed the condition $\widetilde{\rho}^{i^*,i} \ge -\eta$ holds for all i. Thus Step 3b(iii) passes.

Finally we come to Step 3b(iv). Claim 143 tells us that for every restriction π^i, we have $f_{\pi^i}(x) = \mathrm{sgn}(\ell \cdot x - (\theta' - w' \cdot \pi^i))$, where ℓ is a linear form with 2-norm 1 and all coefficients of magnitude at most $\Omega(\sqrt{\tau})$. Applying Proposition 108 we conclude that $|\mathbf{E}[f_\pi] - \mu(\theta' - w' \cdot \pi^i)| \le \sqrt{\tau}$ holds for all i (again, ensuring the constant in the $\Omega(\cdot)$ is small enough). Using the technical Lemma 144 below, we infer that there is a vector w^* whose entries are integer multiples of $\sqrt{\tau}/s$ at most $2^{O(s \log s)}\sqrt{\ln(1/\tau)}$ in absolute value, and an integer multiple θ^* of $\sqrt{\tau}/s$, also at most $2^{O(s \log s)}\sqrt{\ln(1/\tau)}$ in absolute value, such that $|\,\mathbf{E}[f_{\pi^i}] - \mu(\theta^* - w^* \cdot \pi^i)| \leq 4\sqrt{\tau}$ holds for all π^i. By increasing the $4\sqrt{\tau}$ to $4\sqrt{\tau} + \eta \leq 5\sqrt{\tau}$, we can make the same statement with $\widetilde{\mu}^i$ in place of $\mathbf{E}[f_{\pi^i}]$. Thus Step 3(b)(iv) will return ACCEPT once it tries (w^*, θ^*). □

Lemma 144. *Suppose that* $|\,\mathbf{E}[f_\pi] - \mu(\theta' - w' \cdot \pi)| \leq \sqrt{\tau}$ *holds for some set* Π *of* π*'s. Then there is a vector* w^* *whose entries are integer multiples of* $\sqrt{\tau}/s$ *at most* $2^{O(s \log s)}\sqrt{\ln(1/\tau)}$ *in absolute value, and an integer multiple* θ^* *of* $\sqrt{\tau}/s$*, also at most* $2^{O(s \log s)}\sqrt{\ln(1/\eta)}$ *in absolute value, such that* $|\,\mathbf{E}[f_\pi] - \mu(\theta^* - w^* \cdot \pi)| \leq 4\eta^{1/6}$ *also holds for all* $\pi \in \Pi$*.*

Proof. Let us express the given estimates as

$$\left\{\mathbf{E}[f_\pi] - \sqrt{\tau} \leq \mu(\theta' - w' \cdot \pi) \leq \mathbf{E}[f_\pi] + \sqrt{\tau}\right\}_{\pi \in \Pi} \tag{5.39}$$

We would prefer all of the upper bounds $\mathbf{E}[f_\pi] + \sqrt{\tau}$ and lower bounds $\mathbf{E}[f_\pi] - \sqrt{\tau}$ in these double inequalities to have absolute value either equal to 1, or at most $1 - \sqrt{\tau}$. It is easy to see that one can get this after introducing some quantities $1 \leq K_\pi, K'_\pi \leq 2$ and writing instead

$$\left\{\mathbf{E}[f_\pi] - K_\pi\sqrt{\tau} \leq \mu(\theta' - w' \cdot \pi) \leq \mathbf{E}[f_\pi] + K'_\pi\sqrt{\tau}\right\}_{\pi \in \Pi}. \tag{5.40}$$

Using the fact that μ is a monotone function, we can apply μ^{-1} and further rewrite (5.40) as

$$\{c_\pi \leq \theta' - w' \cdot \pi \leq C_\pi\}_{\pi \in \Pi}, \tag{5.41}$$

where each $|c_\pi|, |C_\pi|$ is either ∞ (meaning the associated inequality actually drops out) or is at most $\mu^{-1}(-1 + \sqrt{\tau}) \leq O(\sqrt{\ln(1/\tau)})$. Now (5.41) may actually be thought of as a "linear program" in the entries of w' and in θ' — one which we know is feasible.

By standard results in linear programming [15] we know that if such a linear program is feasible, it has a feasible solution in which the variables take values that are not too large.

In particular, we can take as an upper bound for the variables

$$\mathcal{L} = \frac{|\max_A \det(A)|}{|\min_B \det(B)|}, \tag{5.42}$$

where B ranges over all nonsingular square submatrices of the constraint matrix and A ranges over all square submatrices of the constraint matrix with a portion of the "right-side vector" substituted in as a column. Note that the constraint matrix from (5.41) contains only ± 1's and that the right-side vector contains numbers at most $O(\sqrt{\ln(1/\tau)})$ in magnitude. Thus the minimum in the denominator of (5.42) is at least 1 and the maximum in the numerator of (5.42) is at most $O(\sqrt{\ln(1/\tau)}) \cdot (s+1)!$; hence $\mathcal{L} \leq 2^{O(s \log s)} \sqrt{\ln(1/\tau)}$.

Having made this conclusion, we may recast and slightly weaken (5.40) by saying that there exist a pair (w'', θ''), with entries all at most $\mathcal{L}$ in absolute value, such that

$$\left\{\mathbf{E}[f_\pi] - 2\sqrt{\tau} \leq \mu(\theta'' - w'' \cdot \pi) \leq \mathbf{E}[f_\pi] + 2\sqrt{\tau}\right\}_{\pi \in \Pi}$$

Finally, suppose we round the entries of w'' to the nearest integer multiples of $\sqrt{\tau}/s$ forming w^*, and we similarly round θ'' to θ^*. Then $|(\theta'' - w'' \cdot \pi) - (\theta^* - w^* \cdot \pi)| \leq 2\sqrt{\tau}$ for every π. Since $|\mu'| \leq \sqrt{2/\pi} \leq 1$ we can thus conclude that the inequalities

$$\left\{\mathbf{E}[f_\pi] - 4\sqrt{\tau} \leq \mu(\theta^* - w^* \cdot \pi) \leq \mathbf{E}[f_\pi] + 4\sqrt{\tau}\right\}_{\pi \in \Pi}$$

also hold, completing the proof. □

Soundness of the test.

Theorem 145. *Let $f : \{-1,1\}^n \rightarrow \{-1,1\}$ be a function that passes* **Test-LTF** *with probability more than $1/3$. Then f is $O(\epsilon)$-close to an LTF.*

Proof. As mentioned at the beginning of the proof of Theorem 141, for any f, with probability at least $1 - O(\delta)$ Step 1 of the algorithm succeeds (implicitly producing J), Step 2 of the algorithm succeeds (producing the π^i's, etc.), and all of the items in Proposition 140 hold. So if an f passes the test with probability more than $1/3 \geq O(\delta)$, it must be the case that f passes the deterministic portion of the test, Step 3, despite the above three conditions holding. We will show that in this case f must be $O(\epsilon)$-close to an LTF. We now divide into two cases according to whether f passes the test in Step 3(a) or Step 3(b).

Case 3(a). In this case we have that for at least a $1-\epsilon$ fraction of π^i's, $|\widetilde{\mu}^i| \geq 1-\epsilon$ and hence $|\mathbf{E}[f_{\pi^i}]| \geq 1-\epsilon-\eta \geq 1-2\epsilon$. By Proposition 140 item 1we conclude:

For at least a $1-2\epsilon$ fraction of all restrictions π to J, $|\mathbf{E}[f_\pi]| \geq 1-2\epsilon$. (5.43)

Also, since the test passed, there is some pair (w^*, θ^*) such that $\mathrm{sgn}(w^* \cdot \pi^i - \theta^*) = \mathrm{sgn}(\widetilde{\mu}^i)$ for at least a $1-20\epsilon$ fraction of the π^i's. Now except for at most an ϵ fraction of the π^i's we have $|\mathbf{E}[f_{\pi^i}]| \geq 1-2\epsilon \geq \frac{2}{3}$ and $|\widetilde{\mu}^i - \mathbf{E}[f_{\pi^i}]| \leq \eta < \frac{1}{3}$ whence $\mathrm{sgn}(\widetilde{\mu}^i) = \mathrm{sgn}(\mathbf{E}[f_{\pi^i}])$. Hence $\mathrm{sgn}(w^* \cdot \pi^i - \theta^*) = \mathrm{sgn}(\mathbf{E}[f_{\pi^i}])$ for at least a $1-20\epsilon-\epsilon \geq 1-21\epsilon$ fraction of the π^i's. By Proposition 140 item 2 we conclude:

For at least a $1-22\epsilon$ fraction of all restrictions π to J, $\mathrm{sgn}(\mathbf{E}[f_\pi]) = \mathrm{sgn}(w^* \cdot \pi - \theta^*)$. (5.44)

Combining (5.43) and (5.44), we conclude that except for a $22\epsilon + 2\epsilon \leq 24\epsilon$ fraction of restrictions π to J, f_π is ϵ-close, as a function of the bits outside J, to the constant $\mathrm{sgn}(w^* \cdot \pi - \theta^*)$. Thus f is $24\epsilon + (1-24\epsilon)\epsilon \leq 25\epsilon$-close to the J-junta LTF $\pi \mapsto \mathrm{sgn}(w^* \cdot \pi - \theta^*)$. This completes the proof in Case 3(a).

Case 3(b). In this case, write π^* for π^{i^*}. Since $|\widetilde{\mu}^{i^*}| \leq 1-\epsilon$, we have that $|\mathbf{E}[f_{\pi^*}]| \leq 1-\epsilon+\eta \leq 1-\epsilon/2$. Once we pass Step 3(b)(i) we have $\widetilde{\kappa}^{i^*} \leq 2\tau$ which implies $\sum_{|S|=1} \widehat{f_{\pi^*}}(S)^4 \leq 2\tau + \eta \leq 3\tau$. This in turn implies that f_{π^*} is $(3\tau)^{1/4} \leq 2\tau^{1/4}$-regular. Once we pass Step 3(b)(ii), we additionally have $|\sum_{|S|=1} \widehat{f_{\pi^*}}(S)^2 - W(\mathbf{E}[f_{\pi^*}])| \leq 2\tau^{1/12} + \eta + \eta \leq 3\tau^{1/12}$, where we've also used that $W(\widetilde{\mu}^{i^*})$ is within η of $W(\mathbf{E}[f_{\pi^*}])$ (since $|W'| \leq 1$).

Summarizing, f_{π^*} is $2\tau^{1/4}$-regular and satisfies

$$\begin{aligned} |\mathbf{E}[f_{\pi^*}]| &< 1-\epsilon/2\,, \text{ and} \\ \left|\sum_{|S|=1} \widehat{f_{\pi^*}}(S)^2 - W(\mathbf{E}[f_{\pi^*}])\right| &\leq 3\tau^{1/12} \end{aligned} \tag{5.45}$$

Since Step 3(b)(iii) passes we have that both $|(\widetilde{\rho}^{i^*,i})^2 - W(\widetilde{\mu}^{i^*})W(\widetilde{\mu}^i)| \leq 2\tau^{1/12}$ and $\widetilde{\rho}^{i^*,i} \geq -\eta$ hold for all i's. These conditions imply

$$|(\sum_{|S|=1} \widehat{f_{\pi^*}}(S)\widehat{f_{\pi^i}}(S))^2 - W(\mathbf{E}[f_{\pi^*}])W(\mathbf{E}[f_{\pi^i}])| \leq 2\tau^{1/12} + 4\eta \leq 3\tau^{1/12}$$

and

$$\sum_{|S|=1} \widehat{f_{\pi^*}}(S)\widehat{f_{\pi^i}}(S) \geq -2\eta$$

hold for all i. Applying Proposition 140 item 3 we conclude that for at least a $1-\epsilon$ fraction of the restrictions π to J, both

$$\begin{aligned} \left|\left(\sum_{|S|=1} \widehat{f_{\pi^*}}(S)\widehat{f_\pi}(S)\right)^2 - W(\mathbf{E}[f_{\pi^*}])W(\mathbf{E}[f_\pi])\right| &\leq 3\tau^{1/12} \text{ and} \\ \sum_{|S|=1} \widehat{f_{\pi^*}}(S)\widehat{f_{\pi^i}}(S) &\geq -2\eta \end{aligned} \tag{5.46}$$

We can use (5.45) and (5.46) in Theorem 128, with f_{π^*} playing the role of f, the good f_π's from (5.46) playing the roles of g and the "τ" parameter of Theorem 128 set to $3\tau^{1/12}$. (This requires us to ensure $K \gg 54$.) We conclude:

$$\begin{gathered} \text{There is a fixed vector } \ell \text{ with } \|\ell\| = 1 \text{ and } |\ell_j| \leq O(\tau^{7/108}) \text{ for each } j \\ \text{such that for at least a } 1-\epsilon \text{ fraction of restrictions } \pi \text{ to } J, \\ f_\pi(x) \text{ is } O(\tau^{1/108})\text{-close to the LTF } g_\pi(x) = \operatorname{sgn}(\ell \cdot x - \theta_\pi). \end{gathered} \tag{5.47}$$

We now finally use the fact that Step 3(b)(iv) passes to get a pair (w^*, θ^*) such that $|\widetilde{\mu}^i - \mu(\theta^* - w^* \cdot \pi^i)| \leq 5\sqrt{\tau} \Rightarrow |\,\mathbf{E}[f_{\pi^i}] - \mu(\theta^* - w^* \cdot \pi^i)| \leq 6\sqrt{\tau}$ holds for all π^i's. By

Proposition 140 item 4 we may conclude that:

$$\text{For at least a } 1-\epsilon/2 \text{ fraction of restrictions } \pi \text{ to } J,\ |\mathbf{E}[f_\pi] - \mu(\theta^* - w^* \cdot \pi)| \le 6\sqrt{\tau}. \tag{5.48}$$

Define the LTF $h : \{-1,1\}^n \rightarrow \{-1,1\}$ by $h(\pi, x) = \operatorname{sgn}(w^* \cdot \pi + \ell \cdot x - \theta^*)$. We will complete the proof by showing that f is $O(\tau^{1/108})$-close to h.

We have that the conclusions of (5.47) and (5.48) hold simultaneously for at least a $1 - 2\epsilon$ fraction of restrictions π; call these the "good" restrictions. For the remaining "bad" restrictions π' we will make no claim on how close to each other $f_{\pi'}$ and $h_{\pi'}$ may be. However, these bad restrictions contribute at most 2ϵ to the distance between f and h, which is negligible compared to $O(\tau^{1/108})$. Thus it suffices for us to show that for any good restriction π, we have that f_π and h_π are oh-so-close, namely, $O(\tau^{1/108})$-close. So assume π is a good restriction. In that case we have that f_π is $O(\tau^{1/108})$-close to g_π, so it suffices to show that g_π is $O(\tau^{1/108})$-close to h_π. We have $h_\pi(x) = \operatorname{sgn}(\ell \cdot x - (\theta^* - w^* \cdot \pi))$, and since $\|\ell\| = 1$ and $|\ell_j| \le O(\alpha^{7/108})$ for each j, Proposition 108 implies that $\mathbf{E}[h_\pi] \overset{\tau^{7/108}}{\approx} \mu(\theta^* - w^* \cdot \pi)$. Since π is a good restriction, using (5.48) we have that $\mathbf{E}[h_\pi] \overset{6\sqrt{\tau}}{\approx} \mathbf{E}[f_\pi]$. This certainly implies $\mathbf{E}[h_\pi] \overset{\alpha^{1/108}}{\approx} \mathbf{E}[g_\pi]$ since f_π and g_π are $O(\alpha^{1/108})$-close. But now it follows that indeed g_π is $O(\alpha^{1/108})$-close to h_π because the functions are both LTFs expressible with the same linear form and thus either $g_\pi \ge h_\pi$ pointwise or $h_\pi \ge g_\pi$ pointwise, either of which implies that the distance between the two functions is proportional to the difference of their means.

Finally, we've shown that f is $O(\tau^{1/108})$-close to an LTF. Taking $K = 108$ completes the proof. □

Chapter 6

Testing ± 1-Weight Halfspaces

6.1 Introduction

In the previous chapter, we gave an algorithm for testing whether a boolean function $f : \{-1,1\}^n \rightarrow \{-1,1\}$ is a halfpsace using only $\mathrm{poly}(1/\epsilon)$ queries. In this chapter, we consider the problem of testing whether a function f belongs to a natural subclass of halfspaces, the class of *± 1-weight halfspaces.* These are functions of the form $f(x) = \mathrm{sgn}(w_1x_1 + w_2x_2 + \cdots + w_nx_n)$ where the weights w_i all take values in $\{-1,1\}$. Included in this class is the majority function on n variables, and all 2^n "reorientations" of majority, where some variables x_i are replaced by $-x_i$. Alternatively, this can be viewed as the subclass of halfspaces where all variables have the same amount of influence on the outcome of the function, but some variables get a "positive" vote while others get a "negative" vote.

For the problem of testing ± 1-weight halfspaces, we prove two main results:

1. **Lower Bound.** We show that any nonadaptive testing algorithm which distinguishes ± 1-weight halfspaces from functions that are ϵ-far from ± 1-weight halfspaces must make at least $\Omega(\log n)$ many queries. By a standard transformation (see e.g. [24]), this also implies an $\Omega(\log \log n)$ lower bound for adaptive algorithms. Taken together with the results of the last chapter, this shows that testing this natural subclass of halfspaces is more query-intensive then testing the general class of all halfspaces.

2. **Upper Bound.** We give a nonadaptive algorithm making $O(\sqrt{n} \cdot \text{poly}(1/\epsilon))$ many queries to f, which outputs (i) YES with probability at least $2/3$ if f is a ± 1-weight halfspace (ii) NO with probability at least $2/3$ if f is ϵ-far from any ± 1-weight halfspace.

   We note that it follows from [40] that *learning* the class of ± 1-weight halfspaces requires $\Omega(n/\epsilon)$ queries. Thus, while some dependence on n is necessary for testing, our upper bound shows testing ± 1-weight halfspaces can still be done more efficiently than learning.

Although we prove our results specifically for the case of halfspaces with all weights ± 1, we remark that similar results can be obtained using our methods for other similar subclasses of halfspaces such as $\{-1, 0, 1\}$-weight halfspaces (± 1-weight halfspaces where some variables are irrelevant).

Techniques. As is standard in property testing, our lower bound is proved using Yao's method. We define two distributions D_{YES} and D_{NO} over functions, where a draw from D_{YES} is a randomly chosen ± 1-weight halfspace and a draw from D_{NO} is a halfspace whose coefficients are drawn uniformly from $\{+1, -1, +\sqrt{3}, -\sqrt{3}\}$. We show that a random draw from D_{NO} is with high probability $\Omega(1)$-far from every ± 1-weight halfspace, but that any set of $o(\log n)$ query strings cannot distinguish between a draw from D_{YES} and a draw from D_{NO}.

Our upper bound is achieved by an algorithm which uniformly selects a small set of variables and checks, for each selected variable x_i, that the magnitude of the corresponding singleton Fourier coefficient $|\hat{f}(i)|$ is close to to the right value. We show that any function that passes this test with high probability must have its degree-1 Fourier coefficients very similar to those of some ± 1-weight halfspace, and that any function whose degree-1 Fourier coefficients have this property must be close to a ± 1-weight halfspace. At a high level this approach is similar to some of what is done in the previous chapter, but here we incur a dependence on n because of the level of accuracy that is required to adequately estimate the Fourier coefficients.

6.2 A $\Omega(\log n)$ Lower Bound for ± 1-Weight Halfspaces

In this section we prove the following theorem:

Theorem 146. *There is a fixed constant $\epsilon > 0$ such that any nonadaptive ϵ-testing algorithm $\mathcal{A}$ for the class of all ± 1-weight halfspaces must make at least $(1/26)\log n$ many queries.*

To prove Theorem 146, we define two distributions D_{YES} and D_{NO} over functions. The "yes" distribution D_{YES} is uniform over all 2^n ± 1-weight halfspaces, i.e., a function f drawn from D_{YES} is $f(x) = \mathrm{sgn}(r_1 x_1 + \cdots r_n x_n)$ where each r_i is independently and uniformly chosen to be ± 1. The "no" distribution D_{NO} is similarly a distribution over halfspaces of the form $f(x) = \mathrm{sgn}(s_1 x_1 + \cdots s_n x_n)$, but each s_i is independently chosen to be $\pm\sqrt{1/2}$ or $\pm\sqrt{3/2}$ each with probability $1/4$.

To show that this approach yields a lower bound we must prove two things. First, we must show that a function drawn from D_{NO} is with high probability far from any ± 1-weight halfspace. This is formalized in the following lemma:

Lemma 147. *Let f be a random function drawn from D_{NO}. With probability at least $1 - o(1)$ we have that f is ϵ-far from any ± 1-weight halfspace, where $\epsilon > 0$ is some fixed constant independent of n.*

Next, we must show that no algorithm making $o(\log n)$ queries can distinguish D_{YES} and D_{NO}. This is formalized in the following lemma:

Lemma 148. *Fix any set $x^1, \ldots, x^q$ of q query strings from $\{-1,1\}^n$. Let $\widetilde{D}_{YES}$ be the distribution over $\{-1,1\}^q$ obtained by drawing a random f from D_{YES} and evaluating it on $x^1, \ldots, x^q$. Let $\widetilde{D}_{NO}$ be the distribution over $\{-1,1\}^q$ obtained by drawing a random f from D_{NO} and evaluating it on $x^1, \ldots, x^q$. If $q = (1/26)\log n$ then $\|\widetilde{D}_{YES} - \widetilde{D}_{NO}\|_1 = o(1)$.*

We prove Lemmas 147 and 148 in subsections 6.2.1 and 6.2.2 respectively. A standard argument using Yao's method (see e.g. Section 8 of [24]) implies that the lemmas taken together prove Theorem 146.

6.2.1 Proof of Lemma 147.

Let f be drawn from D_{NO}, and let $s_1, \ldots, s_n$ denote the coefficients thus obtained. Let T_1 denote $\{i : |s_i| = \sqrt{1/2}\}$ and T_2 denote $\{i : |s_i| = \sqrt{3/2}\}$. We may assume that both $|T_1|$ and $|T_2|$ lie in the range $[n/2 - \sqrt{n \log n}, n/2 + \sqrt{n \log n}]$ since the probability that this fails to hold is $1 - o(1)$. It will be slightly more convenient for us to view f as $\mathrm{sgn}(\sqrt{2}(s_1 x_1 + \cdots + s_n x_n))$, that is, such that all coefficients are of magnitude 1 or $\sqrt{3}$.

It is easy to see that the closest ± 1-weight halfspace to f must have the same sign pattern in its coefficients that f does. Thus we may assume without loss of generality that f's coefficients are all $+1$ or $+\sqrt{3}$, and it suffices to show that f is far from the majority function $\mathrm{Maj}(x) = \mathrm{sgn}(x_1 + \cdots + x_n)$.

Let Z be the set consisting of those $z \in \{-1,1\}^{T_1}$ (i.e. assignments to the variables in T_1) which satisfy $S_{T_1} = \sum_{i \in T_1} z_i \in [\sqrt{n/2}, 2\sqrt{n/2}]$. Since we are assuming that $|T_1| \approx n/2$, using Theorem 106, we have that $|Z|/2^{|T_1|} = C_1 \pm o(1)$ for constant $C_1 = \Phi(2) - \Phi(1) > 0$.

Now fix any $z \in Z$, so $\sum_{i \in T_1} z_i$ is some value $V_z \cdot \sqrt{n/2}$ where $V_z \in [1,2]$. There are $2^{n-|T_1|}$ extensions of z to a full input $z' \in \{-1,1\}^n$. Let $C_{\mathrm{Maj}}(z)$ be the fraction of those extensions which have $\mathrm{Maj}(z') = -1$; in other words, $C_{\mathrm{Maj}}(z)$ is the fraction of strings in $\{-1,1\}^{T_2}$ which have $\sum_{i \in T_2} z_i < -V_z\sqrt{n/2}$. By Theorem 106, this fraction is $\Phi(-V_z) \pm o(1)$. Let $C_f(z)$ be the fraction of the $2^{n-|T_1|}$ extensions of z which have $f(z') = -1$. Since the variables in T_2 all have coefficient $\sqrt{3}$, $C_f(z)$ is the fraction of strings in $\{-1,1\}^{T_2}$ which have $\sum_{i \in T_2} z_i < -(V_z/\sqrt{3})\sqrt{n/2}$, which by Theorem 106 is $\Phi(-V_z/\sqrt{3}) \pm o(1)$.

There is some absolute constant $c > 0$ such that for all $z \in Z$, $|C_f(z) - C_{\mathrm{Maj}}(z)| \geq c$. Thus, for a constant fraction of all possible assignments to the variables in T_1, the functions Maj and f disagree on a constant fraction of all possible extensions of the assignment to all variables in $T_1 \cup T_2$. Consequently, we have that Maj and f disagree on a constant fraction of all assignments, and the lemma is proved. □

6.2.2 Proof of Lemma 148.

For $i = 1, \dots, n$ let $Y^i \in \{-1,1\}^q$ denote the vector of $(x_i^1, \dots, x_i^q)$, that is, the vector containing the values of the i^{th} bits of each of the queries. Alternatively, if we view the n-bit strings $x^1, \dots, x^q$ as the rows of a $q \times n$ matrix, the strings $Y^1, \dots, Y^n$ are the columns. If $f(x) = \text{sgn}(a_1 x_1 + \cdots + a_n x_n)$ is a halfspace, we write $\text{sgn}(\sum_{i=1}^n a_i Y^i)$ to denote $(f(x^1), \dots, f(x^q))$, the vector of outputs of f on $x^1, \dots, x^q$; note that the value $\text{sgn}(\sum_{i=1}^n a_i Y^i)$ is an element of $\{-1,1\}^q$.

Since the statistical distance between two distributions D_1, D_2 on a domain $\mathcal{D}$ of size N is bounded by $N \cdot \max_{x \in \mathcal{D}} |D_1(x) - D_2(x)|$, we have that the statistical distance $\|\widetilde{D}_{YES} - \widetilde{D}_{NO}\|_1$ is at most $2^q \cdot \max_{Q \in \{-1,1\}^q} |\Pr_r[\text{sgn}(\sum_{i=1}^n r_i Y^i) = Q] - \Pr_s[\text{sgn}(\sum_{i=1}^n s_i Y^i) = Q]|$. So let us fix an arbitrary $Q \in \{-1,1\}^q$; it suffices for us to bound

$$\left|\Pr_r[\text{sgn}(\sum_{i=1}^n r_i Y^i) = Q] - \Pr_s[\text{sgn}(\sum_{i=1}^n s_i Y^i) = Q]\right|. \tag{6.1}$$

Let InQ denote the indicator random variable for the quadrant Q, i.e. given $x \in \mathbb{R}^q$ the value of $\text{InQ}(x)$ is 1 if x lies in the quadrant corresponding to Q and is 0 otherwise. We have

$$(6.1) = \left|\mathbf{E}_r[\text{InQ}(\sum_{i=1}^n r_i Y^i)] - \mathbf{E}_s[\text{InQ}(\sum_{i=1}^n s_i Y^i)]\right| \tag{6.2}$$

We then note that since the Y^i vectors are of length q, there are at most 2^q possibilities in $\{-1,1\}^q$ for their values which we denote by $\widetilde{Y}^1, \dots, \widetilde{Y}^{2^q}$. We lump together those vectors which are the same: for $i = 1, \dots, 2^q$ let c_i denote the number of times that $\widetilde{Y}^i$ occurs in $Y^1, \dots, Y^n$. We then have that $\sum_{i=1}^n r_i Y^i = \sum_{i=1}^{2^q} a_i \widetilde{Y}^i$ where each a_i is an independent random variable which is a sum of c_i independent ± 1 random variables (the r_j's for those j that have $Y^j = \widetilde{Y}^i$). Similarly, we have $\sum_{i=1}^n s_i Y^i = \sum_{i=1}^{2^q} b_i \widetilde{Y}^i$ where each b_i is an independent random variable which is a sum of c_i independent variables distributed as the s_j's (these are the s_j's for those j that have $Y^j = \widetilde{Y}^i$). We thus can re-express (6.2) as

$$\left|\mathbf{E}_a[\text{InQ}(\sum_{i=1}^{2^q} a_i \widetilde{Y}^i)] - \mathbf{E}_b[\text{InQ}(\sum_{i=1}^{2^q} b_i \widetilde{Y}^i)]\right|. \tag{6.3}$$

Let us define a sequence of random variables that hybridize between $\sum_{i=1}^{2^q} a_i \widetilde{Y}^i$ and

$\sum_{i=1}^{2^q} b_i \widetilde{Y}^i$. For $1 \leq \ell \leq 2^q + 1$ define

$$Z_\ell := \sum_{i<\ell} b_i \widetilde{Y}^i \ + \ \sum_{i \geq \ell} a_i \widetilde{Y}^i, \qquad \text{so} \quad Z_1 = \sum_{i=1}^{2^q} a_i \widetilde{Y}^i \quad \text{and} \quad Z_{2^q+1} = \sum_{i=1}^{2^q} b_i \widetilde{Y}^i. \tag{6.4}$$

As is typical in hybrid arguments, by telescoping (6.3), we have that (6.3) equals

$$\begin{aligned} \left| \mathop{\mathbf{E}}_{a,b}[\sum_{\ell=1}^{2^q} \mathrm{InQ}(Z_\ell) \right. \ &- \ \left. \mathrm{InQ}(Z_{\ell+1})]| = \left| \sum_{\ell=1}^{2^q} \mathop{\mathbf{E}}_{a,b}[\mathrm{InQ}(Z_\ell) - \mathrm{InQ}(Z_{\ell+1})] \right| \\ &= \ \left| \sum_{\ell=1}^{2^q} \mathop{\mathbf{E}}_{a,b}[\mathrm{InQ}(W_\ell + a_\ell \widetilde{Y}^\ell) - \mathrm{InQ}(W_\ell + b_\ell \widetilde{Y}^\ell)] \right| \end{aligned} \tag{6.5}$$

where $W_\ell := \sum_{i<\ell} b_i \widetilde{Y}^i \ + \ \sum_{i>\ell} a_i \widetilde{Y}^i$. The RHS of (6.5) is at most

$$2^q \cdot \max_{\ell=1,\ldots,2^q} | \mathop{\mathbf{E}}_{a,b}[\mathrm{InQ}(W_\ell + a_\ell \widetilde{Y}^\ell) - \mathrm{InQ}(W_\ell + b_\ell \widetilde{Y}^\ell)]|.$$

So let us fix an arbitrary ℓ; we will bound

$$\left| \mathop{\mathbf{E}}_{a,b}[\mathrm{InQ}(W_\ell + a_\ell \widetilde{Y}^\ell) - \mathrm{InQ}(W_\ell + b_\ell \widetilde{Y}^\ell)] \right| \leq B \tag{6.6}$$

(we will specify B later), and this gives that $\|\widetilde{D}_{YES} - \widetilde{D}_{NO}\|_1 \leq 4^q B$ by the arguments above. Before continuing further, it is useful to note that W_ℓ, a_ℓ, and b_ℓ are all independent from each other.

Bounding (6.6). Let $N := (n/2^q)^{1/3}$. Without loss of generality, we may assume that the the c_i's are in monotone increasing order, that is $c_1 \leq c_2 \leq \ldots \leq c_{2^q}$. We consider two cases depending on the value of c_ℓ. If $c_\ell > N$ then we say that c_ℓ is *big*, and otherwise we say that c_ℓ is *small*. Note that each c_i is a nonnegative integer and $c_1 + \cdots + c_{2^q} = n$, so at least one c_i must be big; in fact, we know that the largest value c_{2^q} is at least $n/2^q$.

If c_ℓ is big, we argue that a_ℓ and b_ℓ are distributed quite similarly, and thus for any possible outcome of W_ℓ the LHS of (6.6) must be small. If c_ℓ is small, we consider some $k \neq \ell$ for which c_k is very big (we just saw that $k = 2^q$ is such a k) and show that for any possible outcome of a_ℓ, b_ℓ and all the other contributors to W_ℓ, the contribution to W_ℓ from this c_k makes the LHS of (6.6) small (intuitively, the contribution of c_k is so large that it "swamps" the small difference that results from considering a_ℓ versus b_ℓ).

Case 1: Bounding (6.6) when c_ℓ is big, i.e. $c_\ell > N$. Fix any possible outcome for W_ℓ in (6.6). Note that the vector $\widetilde{Y}^\ell$ has all its coordinates ± 1 and thus it is "skew" to each of the axis-aligned hyperplanes defining quadrant Q. Since Q is convex, there is some interval A (possibly half-infinite) of the real line such that for all $t \in \mathbb{R}$ we have $\mathrm{InQ}(W_\ell + t\widetilde{Y}^\ell) = 1$ if and only if $t \in A$. It follows that

$$|\Pr_{a_\ell}[\mathrm{InQ}(W_\ell + a_\ell \widetilde{Y}^\ell) = 1] - \Pr_{b_\ell}[\mathrm{InQ}(W_\ell + b_\ell \widetilde{Y}^\ell) = 1]| = |\Pr[a_\ell \in A] - \Pr[b_\ell \in A]|. \quad (6.7)$$

Now observe that as in Theorem 106, a_ℓ and b_ℓ are each sums of c_ℓ many independent zero-mean random variables (the r_j's and s_j's respectively) with the same total variance $\sigma = \sqrt{c_\ell}$ and with each $|r_j|, |s_j| \leq O(1)$. Applying Theorem 106 to both a_ℓ and b_ℓ, we get that the RHS of (6.7) is at most $O(1/\sqrt{c_\ell}) = O(1/\sqrt{N})$. Averaging the LHS of (6.7) over the distribution of values for W_ℓ, it follows that if c_ℓ is big then the LHS of (6.6) is at most $O(1/\sqrt{N})$.

Case 2: Bounding (6.6) when c_ℓ is small, i.e. $c_\ell \leq N$. We first note that every possible outcome for a_ℓ, b_ℓ results in $|a_\ell - b_\ell| \leq O(N)$. Let $k = 2^q$ and recall that $c_k \geq n/2^q$. Fix any possible outcome for a_ℓ, b_ℓ and for all other a_j, b_j such that $j \neq k$ (so the only "unfixed" randomess at this point is the choice of a_k and b_k). Let W'_ℓ denote the contribution to W_ℓ from these $2^q - 2$ fixed a_j, b_j values, so W_ℓ equals $W'_\ell + a_k \widetilde{Y}^k$ (since $k > \ell$). (Note that under this supposition there is actually no dependence on b_k now; the only randomness left is the choice of a_k.)

We have

$$\begin{aligned} & |\Pr_{a_k}[\mathrm{InQ}(W_\ell + a_\ell \widetilde{Y}^\ell) = 1] - \Pr_{a_k}[\mathrm{InQ}(W_\ell + b_\ell \widetilde{Y}^\ell) = 1]| \\ = \quad & |\Pr_{a_k}[\mathrm{InQ}(W'_\ell + a_\ell \widetilde{Y}^\ell + a_k \widetilde{Y}^k) = 1] - \Pr_{a_k}[\mathrm{InQ}(W'_\ell + b_\ell \widetilde{Y}^\ell + a_k \widetilde{Y}^k) = 1]| \quad (6.8) \end{aligned}$$

The RHS of (6.8) is at most

$$\Pr_{a_k}[\text{the vector } W'_\ell + a_\ell \widetilde{Y}^\ell + a_k \widetilde{Y}^k \text{ has any coordinate of magnitude at most } |a_\ell - b_\ell|]. \quad (6.9)$$

(If each coordinate of $W'_\ell + a_\ell \widetilde{Y}^\ell + a_k \widetilde{Y}^k$ has magnitude greater than $|a_\ell - b_\ell|$, then each corresponding coordinate of $W'_\ell + b_\ell \widetilde{Y}^\ell + a_k \widetilde{Y}^k$ must have the same sign, and so such an outcome affects each of the probabilities in (6.8) in the same way – either both points are in quadrant Q or both are not.) Since each coordinate of $\widetilde{Y}^k$ is of magnitude 1, by a union bound the probability (6.9) is at most q times

$$\max_{\text{all intervals } A \text{ of width } 2|a_\ell - b_\ell|} \Pr_{a_k}[a_k \in A]. \tag{6.10}$$

Now using the fact that $|a_\ell - b_\ell| = O(N)$, the fact that a_k is a sum of $c_k \geq n/2^q$ independent ± 1-valued variables, and Theorem 107, we have that (6.10) is at most $O(N)/\sqrt{n/2^q}$. So we have that (6.8) is at most $O(Nq\sqrt{2^q})/\sqrt{n}$. Averaging (6.8) over a suitable distribution of values for $a_1, b_1, \ldots, a_{k-1}, b_{k-1}, a_{k+1}, b_{k+1}, \ldots, a_{2^q}, b_{2^q}$, gives that the LHS of (6.6) is at most $O(Nq\sqrt{2^q})/\sqrt{n}$.

So we have seen that whether c_ℓ is big or small, the value of (6.6) is upper bounded by

$$\max\{O(1/\sqrt{N}), O(Nq\sqrt{2^q})/\sqrt{n}\}.$$

Recalling that $N = (n/2^q)^{1/3}$, this equals $O(q(2^q/n)^{1/6})$, and thus $\|\widetilde{D}_{YES} - \widetilde{D}_{NO}\|_1 \leq O(q2^{13q/6}/n^{1/6})$. Recalling that $q = (1/26)\log n$, this equals $O((\log n)/n^{1/12}) = o(1)$, and Lemma 148 is proved.

6.3 A $O(\sqrt{n})$ Upper Bound for ± 1-Weight Halfspaces

In this section we present the ± 1-**Weight Halfspace-Test** algorithm, and prove the following theorem:

Theorem 149. *For any $36/n < \epsilon < 1/2$ and any function $f : \{-1,1\}^n \rightarrow \{-1,1\}$,*

- *if f is a ± 1-weight halfspace, then* ± 1-**Weight Halfspace-Test**(f, ϵ) *passes with probability $\geq 2/3$,*
- *if f is ϵ-far from any ± 1-weight halfspace, then* ± 1-**Weight Halfspace-Test**(f, ϵ) *rejects with probability $\geq 2/3$.*

The query complexity of ± 1**-Weight Halfspace-Test**(f, ϵ) *is* $O(\sqrt{n}\frac{1}{\epsilon^6}\log\frac{1}{\epsilon})$. *The algorithm is nonadaptive and has two-sided error.*

The main tool underlying our algorithm is the following theorem, which says that if most of f's degree-1 Fourier coefficients are almost as large as those of the majority function, then f must be close to the majority function. Here we adopt the shorthand Maj_n to denote the majority function on n variables, and $\hat{\mathsf{M}}_n$ to denote the value of the degree-1 Fourier coefficients of Maj_n.

Theorem 150. *Let* $f : \{-1,1\}^n \rightarrow \{-1,1\}$ *be any Boolean function and let* $\epsilon > 36/n$. *Suppose that there is a subset of* $m \geq (1-\epsilon)n$ *variables* i *each of which satisfies* $\hat{f}(i) \geq (1-\epsilon)\hat{\mathsf{M}}_n$. *Then* $\Pr[f(x) \neq \mathrm{Maj}_n(x)] \leq 32\sqrt{\epsilon}$.

In the following subsections we prove Theorem 150 and then present our testing algorithm.

6.3.1 Proof of Theorem 150.

We start with the following well-known lemma, whose proof serves as a warmup for Theorem 150:

Lemma 151. *Every* $f : \{-1,1\}^n \rightarrow \{-1,1\}$ *satisfies* $\sum_{i=1}^n |\hat{f}(i)| \leq n\hat{\mathsf{M}}_n$.

Proof. Let $G(x) = \mathrm{sgn}(\hat{f}(1))x_1 + \cdots + \mathrm{sgn}(\hat{f}(n))x_n$ and let $g(x)$ be the ± 1-weight halfspace $g(x) = \mathrm{sgn}(G(x))$. We have

$$\sum_{i=1}^{n} |\hat{f}(i)| = \mathbf{E}[fG] \leq \mathbf{E}[|G|] = \mathbf{E}[G(x)g(x)] = \sum_{i=1}^{n} \hat{\mathsf{M}}_n,$$

where the first equality is Plancherel (using the fact that G is linear), the inequality is because f is a ± 1-valued function, the second equality is by definition of g and the third equality is Plancherel again, observing that each $\hat{g}(i)$ has magnitude $\hat{\mathsf{M}}_n$ and sign $\mathrm{sgn}(\hat{f}(i))$. □

Proof of Theorem 150. For notational convenience, we assume that the variables whose Fourier coefficients are "almost right" are $x_1, x_2, ..., x_m$. Now define $G(x) = x_1 + x_2 +$

$\cdots x_n$, so that $\mathrm{Maj}_n = \mathrm{sgn}(G)$. We are interested in the difference between the following two quantities:

$$\mathbf{E}[|G(x)|] = \mathbf{E}[G(x)\mathrm{Maj}_n(x)] = \sum_S \hat{G}(S)\widehat{\mathrm{Maj}}_n(S) = \sum_{i=1}^{n} \widehat{\mathrm{Maj}}_n(i) = n\hat{\mathsf{M}}_n,$$

$$\mathbf{E}[G(x)f(x)] = \sum_S \hat{G}(S)\hat{f}(S) = \sum_{i=1}^{n} \hat{f}(i) = \sum_{i=1}^{m} \hat{f}(i) + \sum_{i=m+1}^{n} \hat{f}(i).$$

The bottom quantity is broken into two summations. We can lower bound the first summation by $(1-\epsilon)^2 n\hat{\mathsf{M}}_n \geq (1-2\epsilon)n\hat{\mathsf{M}}_n$. This is because the first summation contains at least $(1-\epsilon)n$ terms, each of which is at least $(1-\epsilon)\hat{\mathsf{M}}_n$. Given this, Lemma 151 implies that the second summation is at least $-2\epsilon n\hat{\mathsf{M}}_n$. Thus we have

$$\mathbf{E}[G(x)f(x)] \;\geq\; (1-4\epsilon)n\hat{\mathsf{M}}_n$$

and hence

$$\mathbf{E}[|G| - Gf] \leq 4\epsilon n\hat{\mathsf{M}}_n \leq 4\epsilon\sqrt{n} \tag{6.11}$$

where we used the fact (easily verified from Parseval's equality) that $\hat{\mathsf{M}}_n \leq \frac{1}{\sqrt{n}}$.

Let p denote the fraction of points such that $f \neq \mathrm{sgn}(G)$, i.e. $f \neq \mathrm{Maj}_n$. If $p \leq 32\sqrt{\epsilon}$ then we are done, so we assume $p > 32\sqrt{\epsilon}$ and obtain a contradiction. Since $\epsilon \geq 36/n$, we have $p \geq 192/\sqrt{n}$. Let k be such that $\sqrt{\epsilon} = (4k+2)/\sqrt{n}$, so in particular $k \geq 1$. It is well known (by Stirling's approximation) that each "layer" $\{x \in \{-1,1\}^n : x_1 + \cdots + x_n = \ell\}$ of the Boolean cube contains at most a $\frac{1}{\sqrt{n}}$ fraction of $\{-1,1\}^n$, and consequently at most a $\frac{2k+1}{\sqrt{n}}$ fraction of points have $|G(x)| \leq 2k$. It follows that at least a $p/2$ fraction of points satisfy both $|G(x)| > 2k$ and $f(x) \neq \mathrm{Maj}_n(x)$. Since $|G(x)| - G(x)f(x)$ is at least $4k$ on each such point and $|G(x)| - G(x)f(x)$ is never negative, this implies that the LHS of (6.11) is at least

$$\frac{p}{2} \cdot 4k > (16\sqrt{\epsilon}) \cdot (4k) \geq (16\sqrt{\epsilon})(2k+1) = (16\sqrt{\epsilon}) \cdot \frac{\sqrt{\epsilon n}}{2} = 8\epsilon\sqrt{n},$$

but this contradicts (6.11). This proves the theorem. □

6.3.2 A Tester for ±1-Weight Halfspaces

Intuitively, our algorithm works by choosing a handful of random indices $i \in [n]$, estimating the corresponding $|\hat{f}(i)|$ values (while checking unateness in these variables), and checking that each estimate is almost as large as $\hat{\mathsf{M}}_n$. The correctness of the algorithm is based on the fact that if f is unate and most $|\hat{f}(i)|$ are large, then some *reorientation* of f (that is, a replacement of some x_i by $-x_i$) will make most $\hat{f}(i)$ large. A simple application of Theorem 150 then implies that the reorientation is close to Maj_n, and therefore that f is close to a ±1-weight halfspace.

We start with some preliminary lemmas which will assist us in estimating $|\hat{f}(i)|$ for functions that we expect to be unate.

Lemma 152.

$$\hat{f}(i) = \Pr_x[f(x^{i-}) < f(x^{i+})] - \Pr_x[f(x^{i-}) > f(x^{i+})]$$

where x^{i-} and x^{i+} denote the bit-string x with the i^{th} bit set to -1 or 1 respectively.

We refer to the first probability above as the *positive influence* of variable i and the second probability as the *negative influence* of i. Each variable in a monotone function has only positive influence. Each variable in a *unate* function has only positive influence or negative influence, but not both.

Proof. (of Lemma 152) First note that $\hat{f}(i) = \mathbf{E}_x[f(x)x_i]$, then

$$\begin{aligned}\mathop{\mathbf{E}}_x[f(x)x_i] &= \Pr_x[f(x) = 1, x_i = 1] + \Pr_x[f(x) = -1, x_i = -1] \\ &\quad - \Pr_x[f(x) = -1, x_i = 1] - \Pr_x[f(x) = 1, x_i = -1].\end{aligned}$$

Now group all x's into pairs (x^{i-}, x^{i+}) that differ in the i^{th} bit. If the value of f is the same on both elements of a pair, then the total contribution of that pair to the expectation is zero. On the other hand, if $f(x^{i-}) < f(x^{i+})$, then x^{i-} and x^{i+} each add $\frac{1}{2^n}$ to the expectation, and if $f(x^{i-}) > f(x^{i+})$, then x^{i-} and x^{i+} each subtract $\frac{1}{2^n}$. This yields the desired result. □

Lemma 153. *Let f be any Boolean function, $i \in [n]$, and let $|\hat{f}(i)| = p$. By drawing $m = \frac{3}{p\epsilon^2} \cdot \log \frac{2}{\delta}$ uniform random strings $x \in \{-1, 1\}^n$, and querying f on the values $f(x^{i+})$ and $f(x^{i-})$, with probability $1 - \delta$ we either obtain an estimate of $|\hat{f}(i)|$ accurate to within a multiplicative factor of $(1 \pm \epsilon)$, or discover that f is not unate.*

The idea of the proof is that if neither the positive influence nor the negative influence is small, random sampling will discover that f is not unate. Otherwise, $|\hat{f}(i)|$ is well approximated by either the positive or negative influence, and a standard multiplicative form of the Chernoff bound shows that m samples suffice.

Proof. (of Lemma 153) Suppose first that both the positive influence and negative influence are at least $\frac{\epsilon p}{2}$. Then the probability that we do not observe any pair with positive influence is $\leq (1 - \frac{\epsilon p}{2})^m \leq e^{-\epsilon pm/2} = e^{-(3/2\epsilon)\log(2/\delta)} < \frac{\delta}{2}$, and similarly for the negative influence. Therefore, the probability that we observe at least some positive influence and some negative influence (and therefore discover that f is not unate) is at least $1 - 2\frac{\delta}{2} = 1 - \delta$.

Now consider the case when either the positive influence or the negative influence is less than $\frac{\epsilon p}{2}$. Without loss of generality, assume that the negative influence is less than $\frac{\epsilon p}{2}$. Then the positive influence is a good estimate of $|\hat{f}(i)|$. In particular, the probability that the estimate of the positive influence is not within $(1 \pm \frac{\epsilon}{2})p$ of the true value (and therefore the estimate of $|\hat{f}(i)|$ is not within $(1 \pm \epsilon)p$), is at most $< 2e^{-mp\epsilon^2/3} = 2e^{-\log\frac{2}{\delta}} = \delta$ by the multiplicative Chernoff bound. So in this case, the probability that the estimate we receive is accurate to within a multiplicative factor of $(1 \pm \epsilon)$ is at least $1 - \delta$. This concludes the proof. □

Now we are ready to present the algorithm and prove its correctness.

Proof. (of Theorem 149) To prove that the test is correct, we need to show two things: first that it passes functions which are ±1-weight halfspaces, and second that anything it passes with high probability must be ϵ-close to a ±1-weight halfspace. To prove the first, note that if f is a ±1-weight halfspace, the only possibility for rejection is if any of the estimates of $|\hat{f}(i)|$ is less than $(1 - \frac{\epsilon'}{2})\hat{\mathsf{M}}_n$. But applying lemma 153 (with $p = \hat{\mathsf{M}}_n$, $\epsilon = \frac{\epsilon'}{2}$, $\delta = \frac{1}{6k}$), the probability that a particular estimate is wrong is $< \frac{1}{6k}$, and therefore the probability that any estimate is wrong is $< \frac{1}{6}$. Thus the probability of success is $\geq \frac{5}{6}$.

$\pm$1-**Weight Halfspace-Test** (inputs are $\epsilon > 0$ and black-box access to $f : \{-1,1\}^n \to \{-1,1\}$)

1. Let $\epsilon' = (\frac{\epsilon}{32})^2$.
2. Choose $k = \frac{1}{\epsilon'} \ln 6 = O(\frac{1}{\epsilon'})$ many random indices $i \in \{1, ..., n\}$.
3. For each i, estimate $|\hat{f}(i)|$. Do this as in Lemma 153 by drawing $m = \frac{24 \log 12k}{\hat{\mathsf{M}}_n \epsilon'^2} = O(\frac{\sqrt{n}}{\epsilon'^2} \log \frac{1}{\epsilon'})$ random x's and querying $f(x^{i+})$ and $f(x^{i-})$. If a violation of unateness is found, reject.
4. Pass if and only if each estimate is larger than $(1 - \frac{\epsilon'}{2})\hat{\mathsf{M}}_n$.

Figure 6-1: The algorithm $\pm$1-**Weight Halfspace-Test**.

The more difficult part is showing that any function which passes the test whp must be close to a $\pm$1-weight halfspace. To do this, note that if f passes the test whp then it must be the case that for all but an ϵ' fraction of variables, $|\hat{f}(i)| > (1 - \epsilon')\hat{\mathsf{M}}_n$. If this is not the case, then Step 2 will choose a "bad" variable – one for which $|\hat{f}(i)| \leq (1 - \epsilon')\hat{\mathsf{M}}_n$ – with probability at least $\frac{5}{6}$. Now we would like to show that for any bad variable i, the estimate of $|\hat{f}(i)|$ is likely to be less than $(1 - \frac{\epsilon'}{2})\hat{\mathsf{M}}_n$. Without loss of generality, assume that $|\hat{f}(i)| = (1 - \epsilon')\hat{\mathsf{M}}_n$ (if $|\hat{f}(i)|$ is less than that, then variable i will be even less likely to pass step 3). Then note that it suffices to estimate $|\hat{f}(i)|$ to within a multiplicative factor of $(1 + \frac{\epsilon}{2})$ (since $(1 + \frac{\epsilon'}{2})(1 - \epsilon')\hat{\mathsf{M}}_n < (1 - \frac{\epsilon'}{2})\hat{\mathsf{M}}_n$). Again using Lemma 153 (this time with $p = (1 - \epsilon')\hat{\mathsf{M}}_n$, $\epsilon = \frac{\epsilon'}{2}$, $\delta = \frac{1}{6k}$), we see that $\frac{12}{\hat{\mathsf{M}}\epsilon'^2(1-\epsilon')} \log 12k < \frac{24}{\hat{\mathsf{M}}\epsilon'^2} \log 12k$ samples suffice to achieve discover the variable is bad with probability $1 - \frac{1}{6k}$. The total probability of failure (the probability that we fail to choose a bad variable, or that we mis-estimate one when we do) is thus $< \frac{1}{6} + \frac{1}{6k} < \frac{1}{3}$.

The query complexity of the algorithm is $O(km) = O(\sqrt{n}\frac{1}{\epsilon'^3} \log \frac{1}{\epsilon'}) = O(\sqrt{n} \cdot \frac{1}{\epsilon^6} \log \frac{1}{\epsilon})$.

□

6.4 Conclusion

We have proven a lower bound showing that the complexity of testing $\pm$1-weight halfspaces is at least $\Omega(\log n)$ and an upper bound showing that it is at most $O(\sqrt{n} \cdot \text{poly}(\frac{1}{\epsilon}))$. An open question is to close the gap between these bounds and determine the exact dependence on n. One goal is to use some type of binary search to get a $\text{poly}\log(n)$-query adaptive testing algorithm; another is to improve our lower bound to $n^{\Omega(1)}$ for nonadaptive algorithms.

Bibliography

[1] N. Alon, T. Kaufman, M. Krivelevich, S. Litsyn, and D. Ron. Testing low-degree polynomials over GF(2). In *Proc. RANDOM*, pages 188–199, 2003.

[2] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[3] M. Ben-Or and N. Linial. Collective coin flipping. In *Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 408–416, 1985.

[4] R. Bhattacharya and R. Rao. *Normal approximation and asymptotic expansions.* Robert E. Krieger Publishing Company, 1986.

[5] A. Birkendorf, E. Dichterman, J. Jackson, N. Klasner, and H.U. Simon. On restricted-focus-of-attention learnability of Boolean functions. *Machine Learning*, 30:89–123, 1998.

[6] E. Blais. Testing juntas nearly optimally. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 2009.

[7] H. Block. The Perceptron: a model for brain functioning. *Reviews of Modern Physics*, 34:123–135, 1962.

[8] A. Blum and M. Singh. Learning functions of k terms. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (COLT)*, pages 144–153, 1990.

[9] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comp. Sys. Sci.*, 47:549–595, 1993. Earlier version in STOC'90.

[10] N. Bshouty. On learning multivariate polynomials under the uniform distribution. *Information Processing Letters*, 61(3):303–309, 1997.

[11] N. Bshouty. Simple learning algorithms using divide and conquer. *Computational Complexity*, 6:174–194, 1997.

[12] N. Bshouty and Y. Mansour. Simple Learning Algorithms for Decision Trees and Multivariate Polynomials. *SIAM J. Comput.*, 31(6):1909–1925, 2002.

[13] H. Chockler and D. Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, 90(6):301–305, 2004.

[14] C.K. Chow. On the characterization of threshold functions. In *Proceedings of the Symposium on Switching Circuit Theory and Logical Design (FOCS)*, pages 34–38, 1961.

[15] V. Chvátal. *Linear Programming*. W. H. Freeman, 1983.

[16] P. Diaconis. *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics, Hayward, CA, 1988.

[17] I. Diakonikolas, H. Lee, K. Matulef, K. Onak, R. Rubinfeld, R. Servedio, and A. Wan. Testing for concise representations. In *Proc. 48th Ann. Symposium on Computer Science (FOCS)*, pages 549–558, 2007.

[18] I. Diakonikolas, H. Lee, K. Matulef, R. Servedio, and A. Wan. Efficiently testing sparse GF(2) polynomials. In *Proc. 16th International Colloquium on Algorithms, Languages and Programming (ICALP)*, pages 502–514, 2008.

[19] B. Efron and C. Stein. The jackknife estimate of variance. *Annals of Statistics*, 9:586–596, 1981.

[20] A. Ehrenfeucht and M. Karpinski. The computational complexity of (xor,and)-counting problems. Technical report, preprint, 1989.

[21] W. Feller. *An introduction to probability theory and its applications*. John Wiley & Sons, 1968.

[22] A. Fiat and D. Pechyony. Decision trees: More theoretical justification for practical algorithms. In *Algorithmic Learning Theory, 15th International Conference (ALT 2004)*, pages 156–170, 2004.

[23] E. Fischer. The art of uninformed decisions: A primer to property testing. *Computational Complexity Column of The Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, 2001.

[24] E. Fischer. The art of uninformed decisions: A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, 2001.

[25] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 103–112, 2002.

[26] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. *Journal of Computer & System Sciences*, 68:753–787, 2004.

[27] P. Fischer and H. Simon. On learning ring-sum expansions. *SIAM Journal on Computing*, 21(1):181–192, 1992.

[28] P. Goldberg. A Bound on the Precision Required to Estimate a Boolean Perceptron from its Average Satisfying Assignment. *SIAM Journal on Discrete Mathematics*, 20:328–343, 2006.

[29] O. Goldreich, S. Goldwaser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45:653–750, 1998.

[30] P. Gopalan, R. O'Donnell, R. Servedio, A. Shpilka, and K. Wimmer. Testing Fourier dimensionality and sparsity. In *Proc. 36th International Conference on Automata, Languages, and Programming (ICALP)*, pages 500–512, 2008.

[31] D. Grigoriev, M. Karpinski, and M. Singer. Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields. *SIAM Journal on Computing*, 19(6):1059–1063, 1990.

[32] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46:129–154, 1993.

[33] C. Jutla, A. Patthak, A. Rudra, and D. Zuckerman. Testing low-degree polynomials over prime fields. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS '04)*, pages 423–432, 2004.

[34] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 68–80, 1988.

[35] M. Karpinski. Boolean circuit complexity of algebraic interpolation problems. (TR-89-027), 1989.

[36] M. Karpinski and M. Luby. Approximating the Number of Zeros of a $GF[2]$ Polynomial. *Journal of Algorithms*, 14:280–287, 1993.

[37] T. Kaufman and D. Ron. Testing polynomials over general fields. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS '04)*, pages 413–422, 2004.

[38] M. Kearns and D. Ron. Testing problems with sub-learning sample complexity. *J. Comp. Sys. Sci.*, 61:428–456, 2000.

[39] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for Max-Cut and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[40] S. Kulkarni, S. Mitter, and J. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11:23–35, 1993.

[41] Michael Luby, Boban Velickovic, and Avi Wigderson. Deterministic approximate counting of depth-2 circuits. In *Proceedings of the 2nd ISTCS*, pages 18–24, 1993.

[42] Y. Mansour. Randomized interpolation and approximation of sparse polynomials. *SIAM Journal on Computing*, 24(2):357–368, 1995.

[43] K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. Testing {-1,1}-weight halfspaces. In *Proc. 13th International Workshop on Randomization and Computation (RANDOM)*, 2009.

[44] K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. Testing halfspaces. In *Proc. 20th Annual Symposium on Discrete Algorithms (SODA)*, 2009.

[45] M. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry.* MIT Press, Cambridge, MA, 1968.

[46] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, New York, NY, 1995.

[47] S. Muroga, I. Toda, and S. Takasu. Theory of majority switching elements. *J. Franklin Institute*, 271:376–418, 1961.

[48] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. In *Proceedings of the Twenty-Fourth Annual Symposium on Theory of Computing*, pages 462–467, 1992.

[49] A. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.

[50] R. O'Donnell and R. Servedio. The Chow Parameters Problem. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 517–526, 2008.

[51] M. Parnas, D. Ron, and A. Samorodnitsky. Testing basic boolean formulae. *SIAM J. Disc. Math.*, 16:20–46, 2002.

[52] V. V. Petrov. *Limit theorems of probability theory.* Oxford Science Publications, Oxford, England, 1995.

[53] R. Roth and G. Benedek. Interpolation and approximation of sparse multivariate polynomials over $GF(2)$. *SIAM J. Comput.*, 20(2):291–314, 1991.

[54] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. on Comput.*, 25:252–271, 1996.

[55] R. Schapire and L. Sellie. Learning sparse multivariate polynomials over a field with queries and counterexamples. *J. Comput. & Syst. Sci.*, 52(2):201–213, 1996.

[56] R. Servedio. Every linear threshold function has a low-weight approximator. *Computational Complexity*, 16(2):180–209, 2007.

[57] J. Shawe-Taylor and N. Cristianini. *An introduction to support vector machines.* Cambridge University Press, 2000.

[58] D. Štefankovič. Fourier transform in computer science. Master's thesis, University of Chicago, 2000.

[59] M. Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996.

[60] A. Terras. *Fourier Analysis on Finite Groups and Applications.* Cambridge University Press, Cambridge, UK, 1999.

[61] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[62] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 314–326, 1990.

[63] A. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proceedings of the Seventeenth Annual Symposium on Foundations of Computer Science*, pages 222–227, 1977.

[64] A. Yao. On ACC and threshold circuits. In *Proceedings of the Thirty-First Annual Symposium on Foundations of Computer Science*, pages 619–627, 1990.