TESTING FOURIER DIMENSIONALITY AND SPARSITY*

PARIKSHIT GOPALAN†, RYAN O'DONNELL‡, ROCCO A. SERVEDIO§, AMIR SHPILKA¶, AND KARL WIMMER‖

Abstract. We present a range of new results for testing properties of Boolean functions that are defined in terms of the Fourier spectrum. Broadly speaking, our results show that the property of a Boolean function having a concise Fourier representation is locally testable.

We give the first efficient algorithms for testing whether a Boolean function has a sparse Fourier spectrum (small number of nonzero coefficients) and for testing whether the Fourier spectrum of a Boolean function is supported in a low-dimensional subspace of $\mathbb{F}_2^n$. In both cases we also prove lower bounds showing that any testing algorithm — even an adaptive one — must have query complexity within a polynomial factor of our algorithms, which are nonadaptive. Building on these results, we give an "implicit learning" algorithm that lets us test *any* sub-property of Fourier concision. We also present some applications of these results to exact learning and decoding.

Our technical contributions include new structural results about sparse Boolean functions and new analysis of the pairwise independent hashing of Fourier coefficients from [FGKP06].

Key words. Property testing, Fourier spectrum, discrete Fourier analysis, local testability

AMS subject classifications. 94C10, 06E30, 68R99, 42C10

1. Introduction. Recent years have witnessed broad research interest in the local testability of mathematical objects such as graphs, error-correcting codes and Boolean functions. One of the goals of this study is to understand the minimal conditions required to make a property locally testable. For graphs and codes, works such as [AFNS06, AT08, AS08a, AS08b] and [KS07, KS08] have given fairly general characterizations of when a property is testable. For Boolean functions, however, testability is less well understood. On one hand, there are a fair number of testing algorithms for specific classes of functions such as $\mathbb{F}_2$-linear functions [BLR93, BCH+96], dictators [BGS98, PRS02], low-degree $\mathbb{F}_2$-polynomials [AKK+05, Sam07], juntas [FKR+04, Bla08, Bla09], and halfspaces [MORS09]. But there is not much by way of general characterizations of what makes a property of Boolean functions testable. Perhaps the only example is the work of [DLM+07], showing that any class of functions sufficiently well-approximated by juntas is locally testable.

It is reasonable to think that analyzing the Fourier spectrum might help us identify fairly general classes of Boolean functions that can be tested efficiently (see e.g. [Fis01, Section 9.1]). For one thing, many of the known tests — for linearity, dictators, juntas, and halfspaces — involve a careful analysis of the Fourier spectrum. Further intuition comes from learning theory, where the class of functions that are learnable using many of the well-known algorithms [LMN93, KM93, Jac97] can be characterized in terms of the Fourier spectrum.

*A PRELIMINARY VERSION OF THIS WORK APPEARED IN THE *PROCEEDINGS OF THE 36TH INTERNATIONAL COLLOQUIUM ON AUTOMATA, LANGAUGES AND PROGRAMMING (ICALP)* [GOS+09].

†MSR-Silicon Valley, 1065 La Avenida, Mountain View, CA 94043 (parik@microsoft.com). Work done while the author was a postdoc at the University of Washington.

‡Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 (odonnell@cs.cmu.edu). Supported by NSF grants CCF-0747250, CCF-0915893, and DMS-0635607, BSF grant 2008477, and a Sloan fellowship.

§Department of Computer Science, Columbia University, New York, NY 10027 (rocco@cs.columbia.edu). Supported in part by NSF award CCF-0347282, by NSF award CCF-0523664, by DARPA award HR0011-08-1-0069, and by a Sloan Foundation Fellowship.

¶ Faculty of Computer Science, Technion-Israel Institute of Technology, Haifa, Israel and Microsoft Research, Cambridge MA, USA (shpilka@cs.technion.ac.il). This research was partially supported by the Israel Science Foundation (grant number 339/10).

‖Department of Mathematics and Computer Science, Duquesne University, Pittsburgh, PA 15282 (wimmerk@duq.edu). Some of this work was done while the author was a student at Carnegie Mellon University.

In this paper we make some progress toward this goal, by giving efficient algorithms for testing Boolean functions that have *low-dimensional* or *sparse* Fourier representations. These are two natural ways to formalize what it means for a Boolean function to have a "concise" Fourier representation; thus, roughly speaking our results show that the property of having a concise Fourier representation is efficiently testable. Further, as we explain below, Boolean functions with low-dimensional or sparse Fourier representations are closely related to linear functions, juntas, and low-degree polynomials whose testability has been intensively studied, and thus the testability of these classes is an interesting question in its own right. Building on our testing algorithms, we are able to give an "implicit learner" (in the sense of [DLM+07]), which determines the "truth table" of a sparse Fourier spectrum without actually knowing the identities of the underlying Fourier characters. This lets us test *any* sub-property of having a concise Fourier representation. We view this as a step toward the goal of a more unified understanding of the testability of Boolean functions.

Our algorithms rely on new structural results on Boolean functions with sparse and close-to-sparse Fourier spectrums, which may find applications elsewhere. As one such application, we show that the well-known Kushilevitz-Mansour algorithm is in fact an exact proper learning algorithm for Boolean functions with sparse Fourier representations. As another application, we give polynomial-time unique-decoding algorithms for sparse functions and k-dimensional functions.

1.1. The Fourier spectrum, dimensionality, and sparsity. We are concerned with testing various properties defined in terms of the *Fourier representation* of Boolean functions $f : \mathbb{F}_2^n \to \{-1, 1\}$. Input bits will be treated as $0, 1 \in \mathbb{F}_2$, the field with two elements; output bits will be treated as $-1, 1 \in \mathbb{R}$. Every Boolean function $f : \mathbb{F}_2^n \to \mathbb{R}$ has a unique representation as

$$f(x) = \sum_{\alpha \in \mathbb{F}_2^n} \hat{f}(\alpha)\chi_\alpha(x) \text{ where } \chi_\alpha(x) \stackrel{\text{def}}{=} (-1)^{\langle \alpha, x \rangle} = (-1)^{\sum_{i=1}^n \alpha_i x_i}. \tag{1.1}$$

The coefficients $\hat{f}(\alpha)$ are the *Fourier coefficients* of f, and the functions $\chi_\alpha(\cdot)$ are sometimes referred to as *linear functions* or *characters*. In addition to treating input strings x as lying in $\mathbb{F}_2^n$, we also index the characters by vectors $\alpha \in \mathbb{F}_2^n$. This is to emphasize the fact that we are concerned with the linear-algebraic structure. We write $\mathrm{Spec}(f)$ for the Fourier spectrum of f, i.e. the set $\{\alpha \in \mathbb{F}_2^n : \hat{f}(\alpha) \neq 0\}$.

Dimensionality and sparsity (and degree).

We begin by defining the notions of low-dimensionality and sparsity.

DEFINITION 1.1. *A function $f : \mathbb{F}_2^n \to \{-1, 1\}$ is said to be* k-dimensional *if $\mathrm{Spec}(f)$ lies in a k-dimensional subspace of $\mathbb{F}_2^n$. An equivalent definition is that f is k-dimensional if it is $f(x) = g(\chi_{\alpha_1}(x), \ldots, \chi_{\alpha_k}(x))$ where g is any k-variable Boolean function and $\chi_{\alpha_1}, \ldots, \chi_{\alpha_k}$ are parity functions. We write $\dim(f)$ to denote the smallest k for which f is k-dimensional.*

DEFINITION 1.2. *A function f is said to be* s-sparse *if $|\mathrm{Spec}(f)| \leq s$. We write $\mathrm{spar}(f)$ to denote $|\mathrm{Spec}(f)|$, i.e. the smallest s for which f is s-sparse.*

We recall the notion of the $\mathbb{F}_2$*-degree* of a Boolean function, $\deg_2(f)$, which is the degree of the unique multilinear $\mathbb{F}_2$-polynomial representation for f when viewed as a function $\mathbb{F}_2^n \to \mathbb{F}_2$. This should not be confused with the real-degree/Fourier-degree. For example, $\deg_2(\chi_\alpha) = 1$ for all $\alpha \neq 0$. Let us note some relations between $\dim(f)$ and $\mathrm{spar}(f)$. For any Boolean function f, we have

$$\deg_2(f) \leq \log \mathrm{spar}(f) \leq \dim(f), \tag{1.2}$$

except that the first inequality fails when $\deg_2(f) = 1$. (Throughout this paper, $\log$ always means $\log_2$.) The first inequality above is not difficult (see e.g. [BC99, Lemma 3]) and the second one is essentially immediate. Either of the above inequalities can be quite loose; for the first inequality, the inner product function $(-1)^{x_1x_2+x_3x_4+\cdots+x_{n-1}x_n}$ on n variables has $\deg_2(f) = 2$ but $\log \operatorname{spar}(f) = n$. For the second inequality, the "address function" (see Section 2.1 of [BdW02]) with $\frac{1}{2}\log s$ addressing variables and $s^{1/2}$ addressee variables can be shown to be s-sparse but has $\dim(f) \geq s^{1/2}$. (It is trivially true that $\dim(f) \leq s$ for any s-sparse function.)

We may rephrase these bounds as containments between classes of functions:

$$\{k\text{-dimensional}\} \subseteq \{2^k\text{-sparse}\} \subseteq \{\mathbb{F}_2 - \text{degree-}k\} \tag{1.3}$$

where the right containment is proper for $k > 1$ and the left is proper for k larger than some small constant such as 6. Alon et al. [AKK+05] gave essentially matching upper and lower bounds for testing the class of $\mathbb{F}_2$-degree-k functions, showing that $2^{\Theta(k)}$ nonadaptive queries are necessary and sufficient. We show that $2^{\Theta(k)}$ queries are also necessary and sufficient for testing each of the first two classes as well; in fact, by our implicit learning result, we can test a broad range of sub-classes of k-dimensional functions using $2^{O(k)}$ queries.[1]

1.2. Our results and techniques.

1.2.1. Testing Sparsity. We give an algorithm for testing whether a function is s-sparse. Its query complexity is $\operatorname{poly}(s)$, which is optimal up to the degree of the polynomial:

THEOREM 1.3. **[Testing s-sparsity – informal]** *There is a nonadaptive $\operatorname{poly}(s, 1/\epsilon)$-query algorithm for ϵ-testing whether f is s-sparse. Moreover, any algorithm (adaptive, even) for 0.49-testing this property must make $\Omega(\sqrt{s})$ queries.*

The high-level idea behind our tester is that of "hashing" the Fourier coefficients, following [FGKP06]. We choose a random subspace H of $\mathbb{F}_2^n$ with codimension $O(s^2)$. This partitions all the Fourier coefficients into the cosets (affine subspaces) defined by H. If f is s-sparse, then each vector in $\operatorname{Spec}(f)$ is likely to land in a distinct coset. We define the "projection" of f to a coset $r + H$ to be the real-valued function given by zeroing out all Fourier coefficients not in $r + H$. Given query access to f, one can obtain approximate query access to a projection of f by a certain averaging. Now if each vector in $\operatorname{Spec}(f)$ is hashed to a different coset, then each projection function will have sparsity either 1 or 0, so we can try to test that at most s of the projection functions have sparsity 1, and the rest have sparsity 0. Our main theorem shows that this test in fact succeeds, but the analysis is non-trivial as sketched below.

The first step is to show that if f passes this test, most of its Fourier mass lies on a few coefficients. However, this is not *a priori* enough to conclude that f is close to a sparse Boolean function. The obvious way to get a Boolean function close to f would be to truncate the Fourier spectrum to its s largest coefficients and then take the sign, but taking the sign could destroy the sparsity and give a function which is not at all sparse.

We circumvent this obstacle by using some new structural theorems about sparse Boolean functions, coupled with a more delicate truncation procudure.We show that if most of the Fourier mass of a function f lies on its largest s coefficients, then these coefficients are close to being "$\lceil \log s \rceil$–granular," i.e. close to integer multiples of $1/2^{\lceil \log s \rceil}$. We then show that truncating the Fourier expansion to these coefficients and rounding them to nearby granular values gives a sparse *Boolean*-valued function (Theorem 3.4). Thus our sparsity test and its analysis depart significantly from the Fourier based test for juntas [FKR+04].

[1] We remind the reader that efficient testability does not translate downward: if C_1 is a class of functions that is efficiently testable and $C_2 \subsetneq C_1$, the class C_2 need not be efficiently testable.

1.2.2. Testing Low-Dimensionality. We give nearly matching upper and lower bounds for testing whether a function is k-dimensional:

THEOREM 1.4. **[Testing k-dimensionality – informal]** *There is a nonadaptive algorithm that makes $O(k2^{2k}/\epsilon)$ queries for ϵ-testing (see Definition 2.1) whether f is k-dimensional. Moreover, any algorithm (adaptive, even) for 0.49-testing this property must make $\Omega(2^{k/2})$ queries.*

We outline the basic idea behind our dimensionality test. Given $h \in \mathbb{F}_2^n$, we say that $f : \mathbb{F}_2^n \to \mathbb{R}$ is *h-invariant* if it satisfies $f(x+h) = f(x)$ for all $x \in \mathbb{F}_2^n$. We define the subspace $\mathrm{Inv}(f) = \{h : f \text{ is } h\text{-invariant}\}$. If f is truly k-dimensional, then $\mathrm{Inv}(f)$ has codimension k; we use this as the characterization of k-dimensional functions. We estimate the size of $\mathrm{Inv}(f)$ by randomly sampling vectors h and testing if they belong to $\mathrm{Inv}(f)$. We reject if the fraction of such h is much smaller than 2^{-k}.

The crux of our soundness analysis is to show that if a function passes the test with good probability, most of its Fourier spectrum is concentrated on a k-dimensional subspace. From this we conclude that it must in fact be close to a k-dimensional function.

1.2.3. Testing subclasses of k-dimensional functions. Building on these results, we show that a broad range of subclasses of k-dimensional functions are also testable with $2^{O(k)}$ queries. Recall that k-dimensional functions are all functions that can be expressed as $f(x) = g(\chi_{\alpha_1}(x), \ldots, \chi_{\alpha_k}(x))$ where g is any k-variable Boolean function. We say that a class $\mathcal{C}$ is an *induced subclass of k-dimensional functions* if there is some collection $\mathcal{C}'$ of k-variable Boolean functions such that $\mathcal{C}$ is the class of all functions $f = g(\chi_{\alpha_1}, \ldots, \chi_{\alpha_k})$ where g is any function in $\mathcal{C}'$ and $\chi_{\alpha_1}, \ldots, \chi_{\alpha_k}$ are any linear functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ as before. For example, let $\mathcal{C}$ be the class of all k-sparse polynomial threshold functions over $\{-1,1\}^n$; i.e., each function in $\mathcal{C}$ is the sign of a *real* polynomial with at most k nonzero terms. This is an induced subclass of k-dimensional functions, corresponding to the collection $\mathcal{C}' = \{\text{all linear threshold functions over } k \text{ Boolean variables}\}$.

We show that any induced subclass of k-dimensional functions can be tested:

THEOREM 1.5. **[Testing induced subclasses of k-dimensional functions – informal]** *Let $\mathcal{C}$ be any induced subclass of k-dimensional functions. There is a nonadaptive $\mathrm{poly}(2^k, 1/\epsilon)$-query algorithm for ϵ-testing $\mathcal{C}$.*

We note that the upper bound of Theorem 1.5 is essentially best possible in general, by the $2^{\Omega(k)}$ lower bound for testing the whole class of k-dimensional functions.

Our algorithm for Theorem 1.5 extends the approach of Theorem 1.3 with ideas from the "testing by implicit learning" work of [DLM+07]. Briefly, by hashing the Fourier coefficients of a k-dimensional f we are able to construct a matrix of size $2^k \times 2^k$ whose entries are the values taken by the characters χ_α that are in the spectrum of f. This matrix, together with a vector of the corresponding values of f, serves as a data set for "implicit learning" (we say the learning is "implicit" since we do not actually know the names of the relevant characters). Our test inspects sub-matrices of this matrix and tries to find one which, together with the vector of f-values, matches the truth table of some k-variable function $g \in \mathcal{C}'$.

1.2.4. Applications to Exact Learning and Unique Decoding. The soundness of our tests is proved by (implicitly) giving an algorithm that reconstructs a nearby sparse/low-dimensional function. We make these algorithms explicit, and show that they are in fact tolerant to rather high levels of noise. We show that they work up to the *unique decoding radius* for these classes, which is the best one could hope for. As an application, we show that the well-known Kushilevitz-Mansour algorithm is in fact an exact proper learning algorithm for Boolean functions with sparse Fourier representations, and moreover it can handle some amount of adversarial noise in the input.

Organization of the paper. This paper is organized as follows: new structural results, then testing algorithms, then lower bounds and lastly some applications. We give standard preliminaries and an explanation of our techniques for hashing the Fourier spectrum in Section 2. In Section 3 we prove our new structural theorems about sparse Boolean functions, and Section 4 uses these theorems to analyze our test for s-sparse functions. We analyze a natural algorithm for testing k-dimensional functions in Section 5. We give a different algorithm whose analysis extends to testing induced subclasses of k-dimensional functions in Section 6. We present our lower bounds in Section 7 and conclude with applications to exact learning and unique-decoding in Section 8.

2. Preliminaries. Throughout the paper we view Boolean functions as mappings from $\mathbb{F}_2^n$ to $\{-1,1\}$. We will also consider functions which map from $\mathbb{F}_2^n$ to $\mathbb{R}$. Such functions have a unique Fourier expansion as in Equation (1.1). For $\mathcal{A}$ a collection of vectors $\alpha \in \mathbb{F}_2^n$, we write $\mathrm{wt}(\mathcal{A})$ to denote the "Fourier weight" $\mathrm{wt}(\mathcal{A}) = \sum_{\alpha\in\mathcal{A}} \hat{f}(\alpha)^2$ on the elements of $\mathcal{A}$. This notation suppresses the dependence on f, but it will always be clear from context. We frequently use Parseval's identity: $\mathrm{wt}(\mathbb{F}_2^n) = \sum_{\alpha\in\mathbb{F}_2^n} \hat{f}(\alpha)^2 = \|f\|_2^2 \stackrel{\text{def}}{=} \mathbf{E}_{x\in\mathbb{F}_2^n}[f(x)^2]$. Here and elsewhere, an expectation or probability over "$x \in X$" refers to the uniform distribution on X.

As defined in the previous section, the sparsity of f is $\mathrm{spar}(f) = |\mathrm{Spec}(f)|$. We may concisely restate the definition of dimension as $\dim(f) = \dim(\mathrm{span}(\mathrm{Spec}(f)))$.

Given two Boolean functions f and g, we say that f and g are ϵ-*close* if $\mathbf{Pr}_{x\in\mathbb{F}_2^n}[f(x) \neq g(x)] \leq \epsilon$ and say they are ϵ-*far* if $\mathbf{Pr}_{x\in\mathbb{F}_2^n}[f(x) \neq g(x)] \geq \epsilon$. We use the standard definition of property testing:

DEFINITION 2.1. *Let $\mathcal{C}$ be a class of functions mapping $\mathbb{F}_2^n$ to $\{-1,1\}$. A* property tester for $\mathcal{C}$ *is an oracle algorithm $\mathcal{A}$ which is given a distance parameter $\epsilon > 0$ and oracle access to a function $f : \mathbb{F}_2^n \to \{-1,1\}$ and satisfies the following conditions:*

1. *if $f \in \mathcal{C}$ then $\mathcal{A}$ outputs "accept" with probability at least $2/3$;*
2. *if f is ϵ-far from every $g \in \mathcal{C}$ then $\mathcal{A}$ outputs "accept" with probability at most $1/3$.*

We also say that $\mathcal{A}$ ϵ-tests $\mathcal{C}$. The main interest is in the number of queries *the testing algorithm makes.*

All of our testing upper and lower bounds allow "two-sided error" as described above. Our lower bounds are for adaptive query algorithms and our upper bounds are via nonadaptive query algorithms.

2.1. Projections of the Fourier spectrum. The idea of "isolating" or "hashing" Fourier coefficients by projection, as done in [FGKP06] in a learning-theoretic context, plays an important role in our tests.

DEFINITION 2.2. *Given a subspace $H \leq \mathbb{F}_2^n$ and a coset $r + H$, define the* projection operator P_{r+H} *on functions $f : \mathbb{F}_2^n \to \mathbb{R}$ as follows:*

$$\widehat{\mathrm{P}_{r+H} f}(\alpha) \stackrel{\text{def}}{=} \begin{cases} \widehat{f}(\alpha) & \text{if } \alpha \in r + H, \\ 0 & \text{otherwise.} \end{cases}$$

In other words, we have $\mathrm{P}_{r+H} f = A_{r+H} * f$, where $A_{r+H} \stackrel{\text{def}}{=} \sum_{\alpha\in r+H} \chi_\alpha$ and $*$ is the convolution operator: $f * g(x) = \mathbf{E}_y[f(x+y)\cdot g(y)]$.

Clearly $A_{r+H} = \chi_r \cdot \sum_{h\in H} \chi_h$, and it is a simple and well-known fact that $\sum_{h\in H} \chi_h = |H| \cdot \mathbf{1}_{H^\perp}$. Thus we conclude the following (see also Lemma 1 of [FGKP06]):

FACT 2.3. $\mathrm{P}_{r+H} f(x) = \mathbf{E}_{y\in H^\perp}[\chi_r(y) f(x+y)]$.

We now show that for any coset $r+H$, we can approximately determine both $\mathrm{P}_{r+H}f(x)$ and $\|\mathrm{P}_{r+H}f\|_2^2$.

PROPOSITION 2.4. *For any $x \in \mathbb{F}_2^n$, the value $\mathrm{P}_{r+H}f(x)$ can be estimated to within $\pm\tau$ with confidence $1-\delta$ using $O(\log(1/\delta)/\tau^2)$ queries to f.*

Proof. Empirically estimate the right-hand side in Fact 2.3. Since the quantity inside the expectation is bounded in $[-1,1]$, the result follows from a Chernoff bound. □

Recall that $\mathrm{wt}(r+H) = \sum_{\alpha \in r+H} \hat{f}(\alpha)^2 = \|\mathrm{P}_{r+H}f\|_2^2$. We have:

FACT 2.5. $\mathrm{wt}(r+H) = \mathbf{E}_{x \in \mathbb{F}_2^n, z \in H^\perp}[\chi_r(z)f(x)f(x+z)]$.

Proof. Using Parseval and Fact 2.3, we have

$$\mathrm{wt}(r+H) = \mathop{\mathbf{E}}_{w \in \mathbb{F}_2^n}[(\mathrm{P}_{r+H}f(w))^2] = \mathop{\mathbf{E}}_{w \in \mathbb{F}_2^n, y_1, y_2 \in H^\perp}[\chi_r(y_1)f(w+y_1)\chi_r(y_2)f(w+y_2)],$$

which reduces to the desired equality upon writing $x = w + y_1$, $z = y_1 + y_2$. □

PROPOSITION 2.6. *The value $\mathrm{wt}(r+H)$ can be estimated to within $\pm\tau$ with confidence $1-\delta$ using $O(\log(1/\delta)/\tau^2)$ queries to f.*

Proof. Empirically estimate the right-hand side in Fact 2.5. Since the quantity inside the expectation is bounded in $[-1,1]$, the result follows from a Chernoff bound. □

2.2. Hashing to a random coset structure. In this section we present our technique for pairwise independently hashing the Fourier characters.

DEFINITION 2.7. *For $t \in \mathbb{N}$, we define a* random t-dimensional coset structure $(H, \mathcal{C})$ *as follows: We choose vectors $\beta_1, \ldots, \beta_t \in \mathbb{F}_2^n$ independently and uniformly at random and set $H = \mathrm{span}\{\beta_1, \ldots, \beta_t\}^\perp$. For each $b \in \mathbb{F}_2^t$ we define the "bucket"*

$$C(b) \stackrel{def}{=} \{\alpha \in \mathbb{F}_2^n : \langle \alpha, \beta_i \rangle = b_i \text{ for all } i\}.$$

We take $\mathcal{C}$ to be the (multi)set of $C(b)$'s, which has cardinality 2^t.

REMARK 2.8. *Given such a random coset structure, if the β_i's are linearly independent then the buckets $C(b)$ are precisely the cosets in $\mathbb{F}_2^n/H$, and the coset-projection function $\mathrm{P}_{C(b)}f$ is defined according to Definition 2.2. In the (usually unlikely) case that the β_i's are linearly* dependent*, some of the $C(b)$'s will be cosets in $\mathbb{F}_2^n/H$ and some of them will be empty. For the empty buckets $C(b)$ we define $\mathrm{P}_{C(b)}f$ to be identically 0. It is algorithmically easy to distinguish empty buckets from genuine coset buckets.*

We now derive some simple but important facts about this random hashing process:

PROPOSITION 2.9. *Let $(H, \mathcal{C})$ be a random t-dimensional coset structure. Define the indicator random variable $I_{\alpha \to b}$ for the event that $\alpha \in C(b)$.*

1. *For each $\alpha \in \mathbb{F}_2^n \setminus \{0\}$ and each b we have $\mathbf{Pr}[\alpha \in C(b)] = \mathbf{E}[I_{\alpha \to b}] = 2^{-t}$.*
2. *Let $\alpha, \alpha' \in \mathbb{F}_2^n$ be distinct. Then $\mathbf{Pr}[\alpha, \alpha' \text{ belong to the same bucket}] = 2^{-t}$.*
3. *Fix any set $S \subseteq \mathbb{F}_2^n$ with $|S| \leq s+1$. If $t \geq 2\log s + \log(1/\delta)$ then except with probability at most δ, all vectors in S fall into different buckets.*
4. *For each b, the collection of random variables $(I_{\alpha \to b})_{\alpha \in \mathbb{F}_2^n}$ is pairwise independent.*

Proof. Part 1 is because for any $\alpha \neq 0$, each $\langle \alpha, \beta_i \rangle$ is an independent uniformly random bit. Part 2 is because each $\langle \alpha - \alpha', \beta_i \rangle$ is an independent uniformly random bit, and hence the probability that $\langle \alpha, \beta_i \rangle = \langle \alpha', \beta_i \rangle$ for all i is 2^{-t}. Part 3 follows from Part 2 and taking a union bound over the at most $\binom{s+1}{2} \leq s^2$ distinct pairs in S. For Part 4, assume first that $\alpha \neq \alpha'$ are both nonzero. Then from the fact that α and α' are linearly independent, it follows that $\mathbf{Pr}[\alpha, \alpha' \in C(b)] = 2^{-2t}$ as required. On the other hand, if one of $\alpha \neq \alpha'$ is zero, then $\mathbf{Pr}[\alpha, \alpha' \in C(b)] = \mathbf{Pr}[\alpha \in C(b)]\mathbf{Pr}[\alpha' \in C(b)]$ follows immediately by checking the two cases $b = 0$, $b \neq 0$. □

With Proposition 2.9 in mind, we give the following simple deviation bound for the sum of pairwise independent random variables:

PROPOSITION 2.10. *Let $X = \sum_{i=1}^{n} X_i$, where the X_i's are pairwise independent random variables satisfying $0 \le X_i \le \tau$. Assume $\mu = \mathbf{E}[X] > 0$. Then for any $\epsilon > 0$, we have $\mathbf{Pr}[X \le (1-\epsilon)\mu] \le \frac{\tau}{\epsilon^2 \mu}$.*

Proof. By pairwise independence, we have $\mathbf{Var}[X] = \sum \mathbf{Var}[X_i] \le \sum \mathbf{E}[X_i^2] \le \sum \tau \mathbf{E}[X_i] = \tau\mu$. The result now follows from Chebyshev's inequality. □

Finally, it is slightly annoying that Part 1 of Proposition 2.9 fails for $\alpha = 0$ (because 0 is always hashed to $C(0)$). However we can easily handle this issue by renaming the buckets with a simple random permutation.

DEFINITION 2.11. *In a random* permuted *t-dimensional coset structure, we additionally choose a random $z \in \mathbb{F}_2^t$ and rename $C(b)$ by $C(b+z)$.*

PROPOSITION 2.12. *For a random permuted t-dimensional coset structure, Proposition 2.9 continues to hold, with Part 1 even holding for $\alpha = 0$.*

Proof. Use Proposition 2.9 and the fact that adding a random z permutes the buckets. □

3. Structural theorems about s-sparse functions. In this section we prove structural theorems about close-to-sparse Boolean functions. These theorems are crucial to the analysis of our test for s-sparsity.

DEFINITION 3.1. *Let $B = \{\alpha_1, \cdots, \alpha_s\}$ denote the (subsets of $[n]$ with the) s-largest Fourier coefficients of f, and let $S = \bar{B}$ be its complement. We say that f is* μ-close to s-sparse in ℓ_2 *if $\sum_{\alpha \in S} \hat{f}(\alpha)^2 \le \mu^2$.*

DEFINITION 3.2. *We say a rational number* has granularity $k \in \mathbb{N}$, *or is* k-granular, *if it is of the form (integer)$/2^k$. We say a function $f : \mathbb{F}_2^n \to \mathbb{R}$ is* k-granular *if $\widehat{f}(\alpha)$ is k-granular for every α. We say that a number v is* μ-close to k-granular *if $|v - j/2^k| \le \mu$ for some integer j.*

The following structural result is the key theorem for the completeness of our sparsity test; it says that in any function that is close to being sparse in ℓ_2, all the large Fourier coefficients are close to being granular.

THEOREM 3.3. *[Completeness Theorem.] If f is μ-close to s-sparse in ℓ_2, then each $\hat{f}(\alpha)$ for $\alpha \in B$ is $\frac{\mu}{\sqrt{s}}$-close to $\lceil \log s \rceil$-granular (here B is defined as in Definition 3.1).*

Proof. Fix $k = \lceil \log s \rceil + 1$, and let $(H, \mathcal{C})$ denote a random permuted k-dimensional coset structure (defined by choosing vectors $\beta_1, \ldots, \beta_k \in \mathbb{F}_2^n$ and a random shift vector $z \in \mathbb{F}_2^k$ as described in Definitions 2.7 and 2.11). For $b \in \mathbb{F}_2^k$ we have

$$\mathrm{P}_{C(b)} f(x) = \sum_{\alpha \in C(b)} \hat{f}(\alpha) \chi_\alpha(x).$$

Fix $\alpha_i \in B$. We will show that with non-zero probability (over the choice of $(H, \mathcal{C})$ and a uniform choice of $b \in \mathbb{F}_2^k$) the following two events happen together: the vector α_i is the unique coefficient in $B \cap C(b)$, and the ℓ_2 Fourier mass of the set $S \cap C(b)$ is bounded by $\frac{\mu^2}{s}$. Clearly we have $\mathbf{Pr}_{(H,\mathcal{C}),b}[\alpha_i \in C(b)] = 2^{-k}$. Let us condition on this event. By pairwise independence, for any $j \ne i$ we have $\mathbf{Pr}_{(H,\mathcal{C}),b}[\alpha_j \in C(b) \mid \alpha_i \in C(b)] = 2^{-k} \le \frac{1}{2s}$. Thus $\mathbf{E}_{(H,\mathcal{C}),b}\left[|\{j \ne i \text{ such that } \alpha_j \in C(b)\}| \;\middle|\; \alpha_i \in C(b)\right] = \frac{(s-1)}{2^k} < \frac{1}{2}$. Hence by Markov's inequality we get that

$$\mathbf{Pr}_{(H,\mathcal{C}),b}[\exists j \ne i \text{ such that } \alpha_j \in C(b) \mid \alpha_i \in C(b)] < \frac{1}{2}. \tag{3.1}$$

Now consider the coefficients from S. We have

$$\begin{aligned}
&\mathbf{E}_{(H,\mathcal{C}),b}\left[\sum_{\beta\in S\cap C(b)}\hat{f}(\beta)^2 \mid \alpha_i \in C(b)\right] \\
&= \sum_{\beta\in S}\mathbf{Pr}_{(H,\mathcal{C}),b}[\beta\in C(b)\mid \alpha_i\in C(b)]\hat{f}(\beta)^2 \\
&\leq 2^{-k}\mu^2 \leq \frac{\mu^2}{2s}.
\end{aligned}$$

Hence by Markov's inequality,

$$\mathbf{Pr}_{(H,\mathcal{C}),b}\left[\sum_{\beta\in S\cap C(b)}\hat{f}(\beta)^2 \geq \frac{\mu^2}{s} \mid \alpha_i\in C(b)\right] \leq \frac{1}{2}. \tag{3.2}$$

Thus by applying the union bound to Equations 3.1 and 3.2, we have both the desired events (α_i being the unique solution from B, and small ℓ_2 mass from S) happening with non-zero probability over the choice of $(H,\mathcal{C}), b$. Fixing this choice, we have

$$\mathrm{P}_{C(b)}f(x) = \hat{f}(\alpha_i)\chi_{\alpha_i}(x) + \sum_{\beta\in S\cap C(b)}\hat{f}(\beta)\chi_\beta(x) \text{ where } \sum_{\beta\in S\cap C(b)}\hat{f}(\beta)^2 \leq \frac{\mu^2}{s}.$$

But by Fact 2.3 (and writing the coset $C(b)$ as $r+H$ for a suitable r), we also have $\mathrm{P}_{C(b)}f(x) = \mathbf{E}_{y\in H^\perp}[\chi_r(y)f(x+y)]$. Thus the function $\mathrm{P}_{C(b)}f(x)$ is the average of a Boolean function over 2^k points, hence it is $(k-1)$-granular.

We now consider the function $g(x) = \sum_{\beta\in S\cap C(b)}\hat{f}(\beta)\chi_\beta(x)$. Since $\mathbf{E}_x[g(x)^2] \leq \frac{\mu^2}{s}$, for some $x_0\in\mathbb{F}_2^n$ we have $g(x_0)^2 \leq \frac{\mu^2}{s}$, hence $|g(x_0)| \leq \frac{\mu}{\sqrt{s}}$. Fixing this x_0, we have $\mathrm{P}_{C(b)}f(x_0) = \hat{f}(\alpha_i)\chi_{\alpha_i}(x_0) + g(x_0)$, and hence $|\hat{f}(\alpha_i)| = |\mathrm{P}_{C(b)}f(x_0) - g(x_0)|$. Since $\mathrm{P}_{C(b)}f(x_0)$ is $(k-1)$-granular and $|g(x_0)| \leq \frac{\mu}{\sqrt{s}}$, the claim follows. □

Thus, if f has its Fourier mass concentrated on s coefficients, then it is close in ℓ_2 to an s-sparse, $\lceil\log s\rceil$ granular real-valued function. We next show that this real-valued function must in fact be Boolean.

THEOREM 3.4. *[Soundness Theorem.] Let $f:\mathbb{F}_2^n\to\{-1,1\}$ be μ-close to s-sparse in ℓ_2, where $\mu \leq \frac{1}{20s^2}$. Then there is an s-sparse Boolean function $F:\mathbb{F}_2^n\to\{-1,1\}$ within Hamming distance at most $\frac{\mu^2}{2}$ from f.*

Proof. Let $B = \{\alpha_1,\cdots,\alpha_s\}$ be the s largest Fourier coefficients of f and let $k = \lceil\log s\rceil$. By Theorem 3.3, each $\hat{f}(\alpha_i)$ is $\frac{\mu}{\sqrt{s}}$ close to k-granular. So we can write

$$\hat{f}(\alpha_i) = \hat{F}(\alpha_i) + \hat{G}(\alpha_i)$$

where $\hat{F}(\alpha_i)$ is k-granular and $|\hat{G}(\alpha_i)| \leq \frac{\mu}{\sqrt{s}}$. Set $\hat{F}(\beta) = 0$ and $\hat{G}(\beta) = \hat{f}(\beta)$ for $\beta\in S = \bar{B}$. Thus we have $f(x) = F(x) + G(x)$, further F is s-sparse and k-granular, while

$$\mathbf{E}[G(x)^2] \leq s\frac{\mu^2}{s} + \mu^2 \leq 2\mu^2.$$

It suffices to show that F's range is $\{-1,1\}$, for if this is the case then G's range must be $\{-2,0,2\}$, the value $G(x)^2$ is exactly 4 whenever f and F differ, and therefore f and F satisfy

$$\mathbf{Pr}_x[f(x)\neq F(x)] = \mathbf{Pr}[|G(x)|=2] = \frac{1}{4}\mathbf{E}_x[G(x)^2] \leq \frac{\mu^2}{2}.$$

As f is a Boolean function on $\mathbb{F}_2^n$ we have

$$1 = f^2 = F^2 + 2FG + G^2 = F^2 + G(2f - G). \tag{3.3}$$

Writing $H = G(2f - G)$, from Fact 3.5 below we have that for all α,

$$|\widehat{H}(\alpha)| \leq \|G\|_2 \|2f - G\|_2 \leq \|G\|_2(\|2f\|_2 + \|G\|_2) \leq 2\sqrt{2}\mu + 2\mu^2 < 4\mu \leq \frac{1}{5s^2}.$$

On the other hand, since F has granularity k it is easy to see that F^2 has granularity $2k$; in particular, $|\widehat{F^2}(\alpha)|$ is either an integer or at least $2^{-2k} \geq \frac{1}{4s^2}$-far from being an integer. But for (3.3) to hold as a functional identity, we must have $\widehat{F^2}(0) + \widehat{H}(0) = 1$ and $\widehat{F^2}(\alpha) + \widehat{H}(\alpha) = 0$ for all $\alpha \neq 0$. It follows then that we must have $\widehat{F^2}(0) = 1$ and $\widehat{F^2}(\alpha) = 0$ for all $\alpha \neq 0$; i.e., $F^2 = 1$ and hence F has range $\{-1, 1\}$, as claimed. □

FACT 3.5. *Let $f, g : \mathbb{F}_2^n \to \mathbb{R}$. Then $|\widehat{fg}(\alpha)| \leq \|f\|_2 \|g\|_2$ for every α.*

Proof. Any Fourier coefficient of fg is upper bounded by $\|fg\|_2$, and this is at most $\|f\|_2\|g\|_2$. □

4. Testing s-sparsity. This section presents our algorithm for testing whether $f : \mathbb{F}_2^n \to \{-1, 1\}$ is s-sparse, Algorithm **Test-Sparsity**.

Algorithm Test-Sparsity
Inputs: s, ϵ
Parameters: $\mu = \min(\sqrt{2\epsilon}, \frac{1}{20s^2})$, $t = \lceil 2\log s + \log 100 \rceil$, $\tau = \frac{\mu^2}{100 \cdot 2^t}$.

1. Choose a random permuted t-dimensional coset structure $(H, \mathcal{C})$.
2. For each bucket $C \in \mathcal{C}$, estimate $\mathrm{wt}(C) = \sum_{\alpha \in C} \hat{f}(\alpha)^2$ to accuracy $\pm\tau$ with confidence $1 - (1/100)2^{-t}$, using Proposition 2.6.
3. Let $\mathcal{L}$ be the set of buckets where the estimate is at least 2τ. If $|\mathcal{L}| \geq s + 1$, reject.

Roughly speaking, Step 1 pairwise independently hashes the Fourier coefficients of f into $\Theta(s^2)$ buckets. If f is s-sparse then at most s buckets have nonzero weight and the test accepts. On the other hand, if f passes the test with high probability then we show that almost all the Fourier mass of f is concentrated on at most s nonzero coefficients (one for each bucket in $\mathcal{L}$). Theorem 3.4 now shows that f is close to a sparse function. Our theorem about the test is the following:

THEOREM 4.1. *Algorithm* **Test-Sparsity** *ϵ-tests whether $f : \mathbb{F}_2^n \to \{-1, 1\}$ is s-sparse (with confidence $3/4$), making $O\left(\frac{s^6 \log s}{\epsilon^2} + s^{14} \log s\right)$ nonadaptive queries.*

The query complexity of Theorem 4.1 follows immediately from Proposition 2.6 and the fact that there are $2^t = O(s^2)$ buckets. In the remainder of this section we present the completeness (Lemma 4.2) and the soundness (Lemma 4.5) of the test. We begin with the completeness, which is straightforward.

LEMMA 4.2. *If f is s-sparse then the test accepts with probability at least 0.9.*

Proof. Write $f = \sum_{i=1}^{s'} \hat{f}(\alpha_i)\chi_{\alpha_i}$, where each $\hat{f}(\alpha_i) \neq 0$ and $s' \leq s$. Since there are 2^t buckets, all of the estimates in Step 2 are indeed τ-accurate, except with probability at most $1/100$. If the estimates are indeed accurate, the only buckets with weight at least τ are

those that contain a nonzero Fourier coefficient, which are at most s in number. So f passes the test with probability at least 0.9. □

We now analyze the soundness. We partition the Fourier coefficients of f into two sets: B of big coefficients and S of small coefficients. Formally, let

$$B \stackrel{\text{def}}{=} \{\alpha : \hat{f}(\alpha)^2 \geq 3\tau\}, \qquad S \stackrel{\text{def}}{=} \{\alpha : \hat{f}(\alpha)^2 < 3\tau\}. \tag{4.1}$$

We observe that if there are too many big coefficients the test will probably reject:

LEMMA 4.3. *If $|B| \geq s+1$ then the test rejects with probability at least $3/4$.*

Proof. Proposition 2.12(3) implies that after Step 1, except with probability at most $1/100$ there are at least $s+1$ buckets C containing an element of B. In Step 2, except with probability at most $1/100$, we get an estimate of at least $3\tau - \tau \geq 2\tau$ for each such bucket. Then $|\mathcal{L}|$ will be at least $s+1$ in Step 3. Hence the overall rejection probability is at least $1 - 2/100$. □

Next we show that if the weight on small coefficients, $\mathrm{wt}(S) = \sum_{\alpha \in S} \hat{f}(\alpha)^2$, is too large then the test will probably reject:

LEMMA 4.4. *If $\mathrm{wt}(S) \geq \mu^2$ then the test rejects with probability at least $3/4$.*

Proof. Suppose that indeed $\mathrm{wt}(S) \geq \mu^2$. Fix a bucket index b and define the random variable $M_b := \mathrm{wt}(C(b) \cap S) = \sum_{\alpha \in C(b) \cap S} \hat{f}(\alpha)^2 = \sum_{\alpha \in S} \hat{f}(\alpha)^2 \cdot I_{\alpha \to b}$. Here the randomness is from the choice of $(H, \mathcal{C})$, and we have used the pairwise independent indicator random variables defined in Proposition 2.12(4). Let us say that the bucket $C(b)$ is *good* if $M_b \geq \frac{1}{2}\mathbf{E}[M_b]$. We have $\mathbf{E}[M_b] = 2^{-t}\,\mathrm{wt}(S) \geq 100\tau > 0$, and by Proposition 2.10 we deduce $\mathbf{Pr}[M_b \leq \frac{1}{2}\mathbf{E}[M_b]] \leq \frac{3\tau}{(1/2)^2\mathbf{E}[M_b]} \leq 3/25$. Thus the expected fraction of bad buckets is at most $3/25$, so by Markov's inequality there are at most $(3/5)2^t$ bad buckets except with probability at most $1/5$. But if there are at least $(2/5)2^t$ good buckets, we have at least $(2/5)(100s^2) \geq s+1$ buckets b with $\mathrm{wt}(C(b) \cap S) \geq \frac{1}{2}\mathbf{E}[M_b] \geq 50\tau$. Assuming all estimates in Step 2 of the test are accurate to within $\pm\tau$ (which fails with probability at most $1/100$), Step 3 of the test will reject. Thus we reject except with probability at most $1/5 + 1/100 < 1/4$. □

Now we put together the pieces to establish soundness of the test:

LEMMA 4.5. *Suppose the test accepts f with probability exceeding $1/4$. Then f is ϵ-close to an s-sparse Boolean function.*

Proof. Assuming the test accepts f with probability exceeding $1/4$, by Lemma 4.3 we have $|B| \leq s$, by Lemma 4.4 we have $\mathrm{wt}(S) \leq \mu^2$. Thus f is $\mu \leq \frac{1}{20s^2}$ close in ℓ_2 to being s-sparse. We now apply the soundness theorem, Theorem 3.4 to conclude that f must be $\frac{\mu^2}{2} \leq \epsilon$-close in Hamming distance to an s-sparse Boolean function.

We note that the proof of Theorem 3.4 in fact shows that f is $\frac{\mu^2}{2}$-close to the function $F = \sum_{\beta \in B} \tilde{f}(\beta)\chi_\beta$, where each $\tilde{f}(\beta)$ is the $\lceil \log s \rceil$-granular value obtained by rounding $\hat{f}(\beta)$; this will be useful for us in Section 6. □

5. Testing k-dimensionality. In this section we give our algorithm for testing whether a Boolean function is k-dimensional. The test is inspired by the following notion of invariance:

DEFINITION 5.1. *If $f : \mathbb{F}_2^n \to \mathbb{R}$ satisfies $f(x+h) = f(x)$ for all $x \in \mathbb{F}_2^n$, we say that f is h-*invariant*. We define*

$$\mathrm{Inv}(f) \stackrel{def}{=} \{h : f \text{ is } h\text{-invariant}\},$$

which is clearly a subspace of $\mathbb{F}_2^n$. We may view f as a function on $\mathbb{F}_2^n/\mathrm{Inv}(f)$.

The following fact is easily verified (see e.g. [GKS07]):

FACT 5.2. *For any* $f : \mathbb{F}_2^n \to \mathbb{R}$, *we have* $\mathrm{span}(\mathrm{Spec}(f)) = \mathrm{Inv}(f)^\perp$. *Hence we also have* $\dim(f) = \mathrm{codim}(\mathrm{Inv}(f))$.

Recalling that $\dim(f) = \dim(\mathrm{span}(\mathrm{Spec}(f)))$, Fact 5.2 naturally suggests that we test k-dimensionality by estimating the probability that a randomly chosen $h \in \mathbb{F}_2^n$ belongs to $\mathrm{Inv}(f)$. This probability is at least 2^{-k} if f is k-dimensional, and is at most $2^{-(k+1)}$ if f is not k-dimensional. If we could perfectly determine whether a vector h belongs to $\mathrm{Inv}(f)$ with q queries, we would get a nonadaptive test making $O(2^k) \cdot q$ queries. In lieu of a perfect decision on whether $h \in \mathrm{Inv}(f)$, we instead check that $f(x+h) = f(x)$ for $\tilde{O}(2^k)/\epsilon$ many randomly chosen x's. A formal statement of our test is given as Algorithm **Test-Dimensionality**.

Algorithm Test-Dimensionality
Inputs: k, ϵ.
Additional parameter settings: $\ell = O(1) \cdot 2^k$, $m = O(1) \cdot k2^k/\epsilon$

1. Pick $h_1, \ldots, h_\ell \in \mathbb{F}_2^n$ independently and uniformly at random.
2. For each h_i,
   Pick $x_1, \ldots, x_m \in \mathbb{F}_2^n$ independently and uniformly at random.
   If $f(x_j + h_i) = f(x_j)$ for all x_j, add h_i to the multiset H.
3. If $|H|/\ell \geq (9/10)2^{-k}$, accept; otherwise, reject.

Our theorem about this test is the following:

THEOREM 5.3. *Algorithm* **Test-Dimensionality** ϵ*-tests whether* $f : \mathbb{F}_2^n \to \{-1, 1\}$ *has dimension* k, *making* $O(k2^{2k}/\epsilon)$ *nonadaptive queries.*

The query complexity in Theorem 5.3 is immediate. It remains to present the completeness (Lemma 5.4) and the soundness (Lemma 5.9) of the test. We begin with the completeness, which is straightforward:

LEMMA 5.4. *If* f *is* k*-dimensional then the test accepts with probability at least* $2/3$.

Proof. Clearly any $h_i \in \mathrm{Inv}(f)$ will be added to H. Thus the expected fraction of h_i's added to H is at least $2^{-\mathrm{codim}(\mathrm{Inv}(f))}$, which is at least 2^{-k} if f is k-dimensional. A Chernoff bound then shows that the actual fraction will be at least $(9/10)2^{-k}$ except with probability at most $1/3$, assuming the $O(1)$ in the definition of ℓ is suitably large. □

The idea behind the soundness proof is to look at the "essential spectrum" of f, i.e., all of the (nonzero) characters α such that $|\hat{f}(\alpha)|$ is relatively big. We will show that if the test passes with reasonable probability then these characters span a space of dimension at most k (Lemma 5.7), and also have most of the Fourier weight (Lemma 5.8). Formally, let

$$B \stackrel{\text{def}}{=} \{\alpha \neq 0 : \ \hat{f}(\alpha)^2 \geq (1/100)\epsilon 2^{-k}\}, \qquad S \stackrel{\text{def}}{=} \{\alpha \neq 0 : \ \hat{f}(\alpha)^2 < (1/100)\epsilon 2^{-k}\}.$$

To prove the two lemmas mentioned, we make use of the following notation and fact:

DEFINITION 5.5. *For* $h \in \mathbb{F}_2^n$, *we abbreviate by* $h^\perp$ *the subspace* $\{0, h\}^\perp$. *(This space has codimension* 1 *unless* $h = 0$.*)*

FACT 5.6.

$$\Pr_{x \in \mathbb{F}_2^n}[f(x+h) = f(x)] = \sum_{\alpha \in h^\perp} \hat{f}(\alpha)^2.$$

Proof. This follows easily from Fact 2.5, taking $r = 0$ and $H = h^\perp$. □

First we show that if $\mathrm{span}(B)$ has dimension exceeding k, the test probably rejects:

LEMMA 5.7. *If* $\dim(\mathrm{span}(B)) \geq k+1$ *then the test rejects with probability at least* $2/3$.

Proof. Our goal will be to show that the probability a single random h is added to H is at most $(3/4)2^{-k}$. Having shown this, a Chernoff bound will show that we reject in Step 5 with probability at least $2/3$, provided we take the $O(1)$ in the definition of ℓ large enough.

To this end, define $\mathrm{WeakInv}(f) = \mathrm{span}(B)^{\perp}$, a subspace of $\mathbb{F}_2^n$ with codimension at least $k+1$ by assumption. The probability that a random h lies in $\mathrm{WeakInv}(f)$ is thus at most $(1/2)2^{-k}$. We will complete the proof by showing that if $h \notin \mathrm{WeakInv}(f)$, the probability it is added to H in Steps 3–4 is at most $(1/4)2^{-k}$.

So suppose $h \notin \mathrm{WeakInv}(f)$. By definition, this means that $\alpha^* \notin h^{\perp}$ for at least one $\alpha^* \in B$. Then Fact 5.6 implies that

$$\mathop{\mathbf{Pr}}_{x \in \mathbb{F}_2^n}[f(x+h) \neq f(x)] = \sum_{\alpha \notin h^{\perp}} \hat{f}(\alpha)^2 \geq \hat{f}(\alpha^*)^2 \geq (1/100)\epsilon 2^{-k}.$$

Hence the probability h is added to H in Steps 3–4 is at most $(1-(1/100)\epsilon 2^{-k})^m \leq \exp(-k \cdot O(1)/100)$. Taking the $O(1)$ in the definition of m sufficiently large, this is indeed at most $(1/4)2^{-k}$, as required. □

Next we show that if the weight on small coefficients, $\mathrm{wt}(S) = \sum_{\alpha \in S} \hat{f}(\alpha)^2$, is too large then the test will probably reject. The intuition is that we expect half of the weight in $\mathcal{S}$ to fall outside a given $h^{\perp}$, making it unlikely that h is added to H if this weight is big. We convert the expectation result to a high-probability result using Proposition 2.10.

LEMMA 5.8. *If* $\mathrm{wt}(S) > \epsilon$ *then the test rejects with probability at least* $2/3$.

Proof. As in Lemma 5.7, it suffices to show that the probability a single random h is added to H is at most $(3/4)2^{-k}$. So let h be uniformly random and define $D = \{\alpha : \langle \alpha, h \rangle = 1\}$, the complement of $h^{\perp}$. Define the random variable

$$M = \mathrm{wt}(D \cap S) = \sum_{\alpha \in S} \hat{f}(\alpha)^2 \cdot I_{\alpha \to 1}.$$

Here $I_{\alpha \to 1}$ is the indicator random variable for α falling into D. Thinking of h as forming a random 1-dimensional coset structure, we have $D = C(1)$ and the notation is consistent with Proposition 2.9. Recalling that $0 \notin S$, it follows from that proposition that $\mathbf{E}[M] = (1/2)\,\mathrm{wt}(S) > \epsilon/2$ and that the random variables $(I_{\alpha \to 1})_{\alpha \in S}$ are pairwise independent. Thus Proposition 2.10 implies that

$$\mathbf{Pr}[M \leq \tfrac{1}{2}\mathbf{E}[M]] \leq \frac{(1/100)\epsilon 2^{-k}}{(1/2)^2 \mathbf{E}[M]} \leq (8/100)2^{-k}.$$

On the other hand, if $M > \frac{1}{2}\mathbf{E}[M]$ then by Fact 5.6 we have

$$\mathop{\mathbf{Pr}}_{x \in \mathbb{F}_2^n}[f(x+h) \neq f(x)] = \mathrm{wt}(D) \geq M > \tfrac{1}{2}\mathbf{E}[M] > \epsilon/4.$$

In this case, m is more than large enough to imply that h will be added to H in Steps 3–4 with probability at most $(1/4)2^{-k}$ (as in Lemma 5.7). Overall, the probability that a single random h is added to H is at most $(8/100)2^{-k} + (1/4)2^{-k} < (3/4)2^{-k}$, as desired. □

We can now establish the soundness of the test:

LEMMA 5.9. *Suppose the test accepts* f *with probability exceeding* $1/3$. *Then* f *is* ϵ*-close to a* k*-dimensional function.*

Proof. Assuming the test accepts f with probability exceeding $1/3$, Lemma 5.7 implies that $\dim(\mathrm{span}(B)) \leq k$, and Lemma 5.8 implies that $\mathrm{wt}(S) \leq \epsilon$. Define $F : \mathbb{F}_2^n \to \mathbb{R}$ by

$$F(x) = \hat{f}(0) + \sum_{\alpha \in B} \hat{f}(\alpha)\chi_\alpha(x).$$

Clearly F is k-dimensional, and $\|f - F\|_2^2 = \mathrm{wt}(S) \leq \epsilon$. If we now define $g : \mathbb{F}_2^n \to \{-1, 1\}$ by $g = \mathrm{sgn}(F)$, then g is k-dimensional (since it is a function of the k characters F is a function of) and g is ϵ-close to f. Indeed $\mathbf{Pr}[f(x) \neq g(x)] \leq \mathbf{E}[|f(x) - F(x)|^2] = \|f - F\|_2^2 \leq \epsilon$. □

6. Testing induced subclasses of k-dimensional functions. Let $\mathcal{C}$ be any fixed induced subclass of k-dimensional functions, defined by a class $\mathcal{C}'$ of k-variable Boolean functions (recall Section 1.2.3). In this section we show that $\mathcal{C}$ is ϵ-testable using $\mathrm{poly}(2^k, 1/\epsilon)$ queries.

6.1. Overview. Let us give a brief overview of our method for testing membership in $\mathcal{C}$. The first step is to run the s-sparsity test from Section 4 with s set to 2^k (note that since every function in $\mathcal{C}$ is 2^k-sparse, every function in $\mathcal{C}$ will pass this step with high probability.) By Lemma 4.5, if f passes this step with high probability then f is close to the Boolean function

$$F = \sum_{\beta \in B} \tilde{f}(\beta)\chi_\beta, \tag{6.1}$$

which is both s-sparse and k-dimensional (recall from Equation 4.1 that B is the set of "big" Fourier coefficients of f). In fact, the sparsity test "isolates" the elements of B by placing each $\beta \in B$ into its own distinct bucket.

The key to our approach is the following: for any given bucket that the sparsity test identifies (which contains exactly one element $\beta \in B$), we can efficiently simulate query access to the function χ_β. This is done using Proposition 2.4 and a simple form of linear self-correction; we emphasize that it is accomplished without revealing the actual identity of any β in $\mathrm{Spec}(F)$ (indeed, this would require a number of queries dependent on n). The fact that the actual identity of each χ_β is never revealed but we are nonetheless able to obtain the value of $\chi_\beta(x)$ is similar to the "implicit learning" approach of [DLM+07].

We select $O(k2^k)$ random points x and perform the simulated queries described above on these points for all of the buckets that the sparsity test identifies. By doing this, we obtain a complete "implicit truth table" for F which is likely to contain no errors. Roughly speaking, this is a table whose rows are indexed by query points x; each row's entries give the values of $\chi_\beta(x)$ for each $\beta \in B$, along with a final value that gives $F(x)$ (see Section 6.2 for a precise definition). With this implicit truth table in hand it is easy to test that f is k-dimensional (see Section 6.3).

It remains to check whether f corresponds to a junta g (over parity functions) that belongs to $\mathcal{C}'$. Because F, like f, is a k-dimensional function, F can be written as a k-junta over k of the $\{\chi_\beta\}_{\beta \in B}$ functions, i.e. $F = g(\chi_{\alpha_1}, \ldots, \chi_{\alpha_k})$ where each α_i belongs to B. (There may be many different ways of doing this; we will try them all.) For each such g, with the implicit truth table for F in hand we can check — deterministically and without making any further queries — whether g belongs to $\mathcal{C}'$. This concludes the high-level overview of our method.

The organization of this section is as follows. In Section 6.2 we present an augmented version of our sparsity test, and argue that this augmented test produces an "implicit truth table" as described earlier. In Section 6.3 we show that we can use this implicit truth table to test that f is k-dimensional, and show that if f passes this test with high probability then the implicit truth table is "complete." Finally, we explain how to use a complete implicit truth

table to test membership of f in $\mathcal{C}$ in Section 6.4.

REMARK 6.1. *Before entering into details in the following subsections, we re-emphasize that our test for membership in $\mathcal{C}$ will begin by first running the sparsity test Algorithm* **Test-Sparsity** *with $s = 2^k$. (Recall again that k-dimensional functions are 2^k-sparse.) All of our subsequent analysis throughout Section 6 will therefore assume that f is a function which Algorithm* **Test-Sparsity** *accepts with probability exceeding $1/4$. Consequently Lemma 4.5 and its proof give us useful information about f and other assorted quantities; we collect this information here and recall relevant parameter settings for ease of future reference.*

- *Parameter settings: we have $\mu = \min(\sqrt{2\epsilon}, \frac{1}{20s^2})$, $t = \lceil 2\log s + \log 100 \rceil$, $\tau = \frac{\mu^2}{100 \cdot 2^t}$.*
- *The set B, defined as $B = \{\alpha : \hat{f}(\alpha)^2 \geq 3\tau\}$ (recall Equation 4.1), satisfies $|B| \leq s$. The set S, defined as $S = \{\alpha : \hat{f}(\alpha)^2 < 3\tau\}$, satisfies $\mathrm{wt}(S) \leq \mu^2$.*
- *For each $\beta \in B$ the value $\hat{f}(\beta)$ is within $\mu/\sqrt{s}$ of a nonzero $\lceil \log s \rceil$-granular number $\tilde{f}(\beta)$ (by Theorem 3.3). Consequently each $\hat{f}(\beta)$ has magnitude at least $1/(4s)$ and has the same sign as $\tilde{f}(\beta)$.*
- *The function F defined in Equation 6.1 is s-sparse and Boolean, and f is $\frac{\mu^2}{2}$-close to F in Hamming distance.*

6.2. Building an implicit truth table. We first give a precise definition of an "implicit truth table." We then present our algorithm **Build-Implicit-Truth-Table** and argue that then with high probability it correctly builds an implicit truth table.

DEFINITION 6.2. *Let $\mathcal{M}$ be a list of strings $x \in \mathbb{F}_2^n$. The* implicit truth table for F corresponding to $\mathcal{M}$ *consists of a matrix $\mathcal{Q} \in \{-1,1\}^{\mathcal{M} \times |B|}$ and a vector $\mathcal{F} \in \{-1,1\}^{\mathcal{M}}$. We call $|\mathcal{M}|$ the* size *of the implicit truth table. The rows of the matrix $\mathcal{Q}$ are indexed by the elements of $\mathcal{M}$ and the columns by the elements of B, and the (x, β) entry is equal to $\chi_\beta(x)$ for all $x \in \mathcal{M}$ and $\beta \in B$. The entries of vector $\mathcal{F}$ are indexed by the elements of $\mathcal{M}$ and are defined by $\mathcal{F}_x = F(x)$.*

We observe that $\mathcal{F}_x$ is uniquely determined by the x-row of $\mathcal{Q}$, in the sense that if $x, y \in \mathbb{F}_2^n$ give rise to identical rows of $\mathcal{Q}$ then $\mathcal{F}_x$ equals $\mathcal{F}_y$. (This is simply because the values $\chi_\beta(x)$ and $\chi_\beta(y)$ are identical for each $\beta \in B$, and thus $F(x) = F(y)$ by Equation 6.1.)

Our algorithm **Build-Implicit-Truth-Table** is given below. We stress that the first three steps of the test could be replaced simply by "Run Algorithm **Test-Sparsity**(s, ϵ)" – the parameters and the code are completely identical to **Test-Sparsity**. We have reproduced the code of **Test-Sparsity** here to make **Build-Implicit-Truth-Table** more self-contained and readable.

The high-level idea of **Build-Implicit-Truth-Table** is as follows. After running **Test-Sparsity** in Steps 1-3, the buckets that will be used are identified as $\mathcal{L}'$ in Step 4. Step 5 draws the list $\mathcal{M}$ of strings that define the implicit truth table $(\mathcal{Q}, \mathcal{F})$ being constructed, and Step 6 constructs the vector $\mathcal{F}$. Steps 7-10 use linear self-correction and Proposition 2.4 to build the matrix $\mathcal{Q}$. The main result of this subsection, establishing correctness of **Build-Implicit-Truth-Table**, is Theorem 6.6.

Algorithm Build-Implicit-Truth-Table
Inputs: s, ϵ, $m \le O(s^2)$
Parameters: $\mu = \min(\sqrt{2\epsilon}, \frac{1}{20s^2})$, $t = \lceil 2\log s + \log 100 \rceil$, $\tau = \frac{\mu^2}{100 \cdot 2^t}$.

1. Choose a random permuted t-dimensional coset structure $(H, \mathcal{C})$.
2. For each bucket $C \in \mathcal{C}$, estimate $\mathrm{wt}(C) = \sum_{\alpha \in C} \hat{f}(\alpha)^2$ to accuracy $\pm\tau$ with confidence $1 - (1/100)2^{-t}$, using Proposition 2.6.
3. Let $\mathcal{L}$ be the set of buckets where the estimate is at least 2τ. If $|\mathcal{L}| \ge s+1$, reject.
4. Let $\mathcal{L}' \subseteq \mathcal{L}$ be the buckets whose Step 2 estimate is at least $1/(32s^2)$.
5. Draw a list $\mathcal{M} = \{x_1, x_2, \ldots, x_m\}$ of m uniformly random strings from $\mathbb{F}_2^n$.
6. Define the length-m column vector $\mathcal{F}$ by querying f on each $x \in \mathcal{M}$ and setting $\mathcal{F}_x = f(x)$.
7. Draw a list $\mathcal{M}' = \{y_1, y_2, \ldots, y_m\}$ of m uniformly random strings from $\mathbb{F}_2^n$.
8. Define the $m \times |\mathcal{L}'|$ matrix $\mathcal{Q}'$ as follows: For each $i \in [m]$ and $C \in \mathcal{L}'$, estimate $\mathrm{P}_C f(y_i)$ to within $\pm 1/(4s)$ with confidence $1 - 1/(200sm)$, using Proposition 2.4; set $\mathcal{Q}'_{i,C}$ to be the sign of the estimate.
9. Define the $m \times |\mathcal{L}'|$ matrix $\mathcal{Q}''$ as follows: For each $i \in [m]$ and $C \in \mathcal{L}'$, estimate $\mathrm{P}_C f(x_i + y_i)$ to within $\pm 1/(4s)$ with confidence $1 - 1/(200sm)$, using Proposition 2.4; set $\mathcal{Q}''_{i,C}$ to be the sign of the estimate.
10. Define the $m \times |\mathcal{L}'|$ matrix $\mathcal{Q}$ as follows: For each $i \in [m]$ and $C \in \mathcal{L}'$, set $\mathcal{Q}_{i,C} = \mathcal{Q}'_{i,C} \cdot \mathcal{Q}''_{i,C}$. Output $(\mathcal{M}, \mathcal{Q}, \mathcal{F})$.

REMARK 6.3. *The query complexity of Algorithm* **Build-Implicit-Truth-Table** *differs from that of Algorithm* **Test-Sparsity** *by at most a constant factor. To see this, note that although that Steps* 4 *through* 10 *are described as being adaptive, they could easily be done nonadaptively by estimating* $\mathrm{P}_C f(y_i)$ *and* $\mathrm{P}_C f(x_i + y_i)$ *for* every *bucket* $C \in \mathcal{C}$. *Even this would require query complexity only* $m + O(s^2) \cdot m \cdot O(s^2 \log s) \le O(s^6 \log s)$, *which is asymptotically no more than the query complexity of Algorithm* **Test-Sparsity**.

We proceed with our analysis of **Build-Implicit-Truth-Table.**

LEMMA 6.4. *Define* $c : B \to \mathcal{L}'$ *such that* $c(\beta)$ *is the bucket containing* β. *Except with probability at most* $2/100$, *the mapping* c *is a* 1-1 *correspondence.*

Proof. By Proposition 2.12(3), except with failure probability at most 1/100 after Step 1 all $\beta \in B$ fall into different buckets, so the function c is injective. After Step 2, except with failure probability $1/100$ the estimates are all accurate to within $\pm\tau$ (so the total failure probability incurred is $2/100$). To see that the range of c is contained in $\mathcal{L}'$, note that for each $\beta \in B$ we have $|\hat{f}(\beta)| \ge 1/(4s)$ (see Remark 6.1); hence the bucket containing β has weight at least $1/(16s^2) \ge 1/(32s^2) + \tau$ and therefore this bucket will be put into $\mathcal{L}'$ in Step 4. To show that c is an onto map we need to verify that each bucket in $\mathcal{L}'$ contains a vector from B. Since $\mathrm{wt}(S) \le \mu^2 \le 1/(400s^4) < 1/(32s^2) - \tau$ (again see Remark 6.1), even if all vectors $\alpha \notin B$ landed in the same bucket, if that bucket did not contain any vector from B then it would not be added into $\mathcal{L}'$. □

This lemma implies that we may view the columns of the matrices $\mathcal{Q}$, $\mathcal{Q}'$, and $\mathcal{Q}''$ defined in Algorithm **Build-Implicit-Truth-Table** as indexed by elements of $\{c(\beta)\}_{\beta\in B}$.

Algorithm **Build-Implicit-Truth-Table** constructs two matrices $\mathcal{Q}'$ and $\mathcal{Q}''$ for two lists of strings $(y_i)_{i\in[m]}$ and $(x_i + y_i)_{i\in[m]}$ because the most straightforward approach (simply computing $\mathrm{sgn}(\mathrm{P}_{c(\beta)}f(x_i)$ to compute $\chi_\beta(x_i)$) is off by a factor of $\mathrm{sgn}(\tilde{f}(\beta))$. In order to offset this factor, Algorithm **Build-Implicit-Truth-Table** uses linear self-correction.

LEMMA 6.5. *After Step* 9*, we have that* $\mathcal{Q}'_{i,c(\beta)} = \mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(y_i)$ *and* $\mathcal{Q}''_{i,c(\beta)} = \mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(x_i + y_i)$ *for each* $i \in [m]$ *and* $\beta \in B$ *except with probability at most* $4/100$.

Proof. For each $\beta \in B$, define the function $G_\beta = \mathrm{P}_{c(\beta)}f - \tilde{f}(\beta)\chi_\beta$. Using the 1-1 correspondence between B and $\mathcal{L}'$ provided by Lemma 6.4 and the fact that coset-projection functions have disjoint Fourier support, we have

$$O(\mu^2) \geq \|f - F\|_2^2 = \sum_{\beta\in B} \|G_\beta\|_2^2 + \sum_{C\notin\mathcal{L}'} \|\mathrm{P}_C f\|_2^2 \geq \sum_{\beta\in B} \|G_\beta\|_2^2 \tag{6.2}$$

(see the final bullet of Remark 6.1 for the leftmost inequality above). Say that a string $x \in \mathbb{F}_2^n$ is *bad* for $\beta \in B$ if $|G_\beta(x)| > 1/(4s)$. Since $\|G_\beta\|_2^2 = \mathbf{E}[G_\beta^2]$, by Markov's inequality the fraction of strings bad for β is at most $(4s)^2\|G_\beta\|_2^2$. Thus we conclude that the fraction of strings x which are bad for *any* $\beta \in B$ is at most $16s^2 \sum_{\beta\in B} \|G_\beta\|_2^2 \leq O(s^2\mu^2)$, using (6.2). Since $m \leq O(s^2)$, the probability that any string y_i or $x_i + y_i$ is bad for any $\beta \in B$ is at most $O(s^4\mu^2) \leq 1/100$. So we assume all strings y_i and $x_i + y_i$ are good for all $\beta \in B$, and overall we have accumulated failure probability at most $3/100$.

Fix $i \in [m]$ and $\beta \in B$. Assuming y_i and $x_i + y_i$ are good for $\beta \in B$, it remains to show that $\mathcal{Q}'_{i,c(\beta)}$ equals $\mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(y_i)$ and $\mathcal{Q}''_{i,c(\beta)}$ equals $\mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(x_i + y_i)$. Since $\tilde{f}(\beta)$ is a nonzero $\lceil \log s\rceil$-granular number, we have $|\tilde{f}(\beta)\chi_\beta(y_i)| \geq 1/2s$. Thus if y_i is good for β we must have both that $|\mathrm{P}_{c(\beta)}f(y_i)| \geq 1/(2s)$ and that $\mathrm{sgn}(\mathrm{P}_{c(\beta)}f(y_i)) = \mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(y_i)$. Now the fact that the estimate for $\mathrm{P}_{c(\beta)}f(y_i)$ is accurate to within $\pm 1/(4s)$ except with probability at most $1/(200sm)$ means that $\mathcal{Q}'_{i,c(\beta)}$ will have the same sign as $\mathrm{P}_{c(\beta)}f(x)$, as required. A similar argument holds for $\mathcal{Q}''_{i,c(\beta)}$. Taking a union bound over all (at most s) $\beta \in B$ and all $i \in [m]$, all the lemma's claims hold except with probability at most $3/100 + 1/200 + 1/200 = 4/100$. □

Our main theorem concerning Algorithm **Build-Implicit-Truth-Table** is the following:

THEOREM 6.6. *Except with probability at most* $5/100$*, the triple* $(\mathcal{M}, \mathcal{Q},\mathcal{F})$ *output by Algorithm* **Build-Implicit-Truth-Table** *is such that* $(\mathcal{Q}, \mathcal{F})$ *is the implicit truth table for* F *corresponding to* $\mathcal{M}$.

Proof. By the previous lemma, we have that $\mathcal{Q}'_{i,c(\beta)} = \mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(y_i)$ and $\mathcal{Q}''_{i,c(\beta)} = \mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(x_i + y_i)$ for each $i \in [m]$ and $\beta \in B$ except with probability at most $4/100$. Since

$$\mathcal{Q}_{i,c(\beta)} = \mathcal{Q}'_{i,c(\beta)} \cdot \mathcal{Q}''_{i,c(\beta)} = \mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(y_i)\mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(x_i + y_i) = \chi_\beta(y_i)\chi_\beta(x_i + y_i)$$

which equals $\chi_\beta(x_i)$, this establishes the correctness of $\mathcal{Q}$.

Since f and F are $\frac{\mu^2}{2}$-close as Boolean functions (see Remark 6.1), the probability that $\mathcal{F}_x \neq F(x)$ for any $x_i \in \mathcal{M}$ is at most $m \cdot \frac{\mu^2}{2} \leq O(s^2\mu^2) \leq 1/100$. A union bound completes the proof of the theorem. □

6.3. An alternate test for k-dimensionality. We have seen that if f is such that **Test-Sparsity** (run with parameter $s = 2^k$) accepts with probability at least $1/4$, then with high

probability **Build-Implicit-Truth-Table** constructs an implicit truth table $(\mathcal{Q}, \mathcal{F})$ for F in terms of the relevant characters $\beta \in B$. We now turn our attention to k-dimensionality.

LEMMA 6.7. *Consider the matrix $\mathcal{Q}$ under the identification $1 \in \mathbb{R} \leftrightarrow 0 \in \mathbb{F}_2$ and $-1 \in \mathbb{R} \leftrightarrow 1 \in \mathbb{F}_2$. The set of all possible rows that could appear in $\mathcal{Q}$ (as the possible elements of $\mathcal{M}$ range over all of $\mathbb{F}_2^n$) forms a $\dim(F)$-dimensional subspace of $\mathbb{F}_2^{|B|}$. In the construction of $\mathcal{Q}$, each row of $\mathcal{Q}$ is uniformly distributed on this subspace.*

Proof. It suffices to prove the following: If one chooses a uniform $x \in \mathbb{F}_2^n$, the $\mathbb{F}_2$-identified vector $\langle \chi_\beta(x) \rangle_{\beta \in B}$ — i.e., $\langle \beta, x \rangle_{\beta \in B}$ — is uniformly distributed on a subspace of dimension $\dim(\mathrm{span}(B))$. Indeed, letting $A \in \mathbb{F}_2^{|B| \times n}$ be the matrix formed by stacking the $\beta \in B$ as rows, the image of A is a subspace of dimension $\mathrm{rank}(A) = \dim(\mathrm{span}(B))$. The set of x's achieving a particular vector in the image forms a coset in $\mathbb{F}_2^n / \ker(A)$; the fact that all cosets have the same cardinality completes the proof. □

DEFINITION 6.8. *We say that an implicit truth table $(\mathcal{Q}, \mathcal{F})$ for F is* complete *if $\mathcal{Q}$ contains all $2^{\dim(F)}$ possible rows at least once.*

LEMMA 6.9. *Set $m = 200k2^k$ in Algorithm* **Build-Implicit-Truth-Table**. *If F is k-dimensional then $\mathcal{Q}$ is complete except with probability at most $1/100$. Further, if F is not k-dimensional then $\mathcal{Q}$ contains more than 2^k distinct rows except with probability at most $1/100$.*

Proof. These facts follow easily from the Coupon Collector analysis and Lemma 6.7. □

Lemma 6.9 tells us that by considering the number of distinct rows of $\mathcal{Q}$, we get a test for k-dimensionality which is an alternative to the earlier algorithm **Test-Dimensionality**:

Algorithm Test-Dim-Using-Truth-Table
Inputs: k, ϵ
Additional parameter settings: $s = 2^k, m = 200k2^k$

1. Run Algorithm **Build-Implicit-Truth-Table** with parameters s, ϵ, m.
2. Reject if $\mathcal{Q}$ has more than 2^k distinct rows; otherwise accept.

THEOREM 6.10. *If f is k-dimensional then Algorithm* **Test-Dim-Using-Truth-Table** *accepts and constructs a complete implicit truth table $(\mathcal{Q}, \mathcal{F})$ with probability at least $2/3$. Further, if the test accepts with probability exceeding $1/4$ then f is ϵ-close to F, which is k-dimensional, and except with probability at most $6/100$ the pair $(\mathcal{Q}, \mathcal{F})$ constructed by the test is a complete implicit truth table for F.*

Proof. For the first statement, if f is k-dimensional then it is s-sparse, so Algorithm **Test-Sparsity** (equivalently, Steps 1-3 of Algorithm **Build-Implicit-Truth-Table**) passes with probability at least $3/4$. By Theorem 6.6 we have that $(\mathcal{Q}, \mathcal{F})$ is the implicit truth table for F except with probability at most $5/100$. We now use the fact that if f is s-sparse then the function F must be *identical* to f. (This is because both f and F, being s-sparse Boolean functions, have $\mathbb{F}_2$-degree at most $\log s$, and it is well known, by a Schwartz-Zippel variant for $\mathbb{F}_2$, that two such polynomials, at distance at most $\frac{\mu^2}{2} < 1/s$, must be identical.) Since $F = f$ is k-dimensional, any implicit truth table for F has at most 2^k distinct rows, by Lemma 6.7. By Lemma 6.9, $\mathcal{Q}$ is complete except with probability at most $1/100$. Thus the test accepts and produces a complete implicit truth table $(\mathcal{Q}, \mathcal{F})$ with probability at least $3/4 - 6/100 > 2/3$, as claimed.

For the second statement, suppose f passes Algorithm **Test-Dim-Using-Truth-Table** with probability exceeding $1/4$. Then certainly f passes Algorithm **Test-Sparsity** with probability at least $1/4$, so F is well-defined and f is ϵ-close to F by the last bullet of Remark 6.1.

Further, F must be k-dimensional as claimed, for otherwise the combination of Theorem 6.6 and Lemma 6.9 would imply that f is accepted with probability at most $6/100$. These same two lemmas imply that the $(\mathcal{Q}, \mathcal{F})$ produced by the test are a complete implicit truth table except with probability at most $6/100$. □

6.4. Testing subclasses of k-dimensionality with implicit learning. To capture every possible representation of F as a function of k parities, we require a column for each $\beta \in \mathrm{span}(B)$ instead of one for each $\beta \in B$. We obtain these columns by finding a basis for the column space of the matrix $\mathcal{Q}$ (in the $\mathbb{F}_2$-identification of $\mathcal{Q}$) and explicitly computing the $2^{\dim(F)}$ columns in the space that this basis spans; it is easy to do this algorithmically, using Gaussian elimination to find such a basis. Let $\mathcal{W}$ be this expanded version of $\mathcal{Q}$. (We could also remove duplicate rows from $\mathcal{W}$, but this is not strictly necessary.)

DEFINITION 6.11. *We define a* k-restriction *of* $(\mathcal{W}, \mathcal{F})$ *to be a pair* $(\mathcal{W}', \mathcal{F})$*, where* $\mathcal{W}'$ *is formed by taking* k *columns of* $\mathcal{W}$*.*

Since each column of $\mathcal{W}$ corresponds to χ_β for some $\beta \in \mathrm{span}(B)$, a k-restriction $(\mathcal{W}', \mathcal{F})$ may be viewed as an (attempted) truth table of a k-variable Boolean function. Each input bit in $\mathcal{W}'$ corresponds to the value of a character, and thus we may also view $(\mathcal{W}', \mathcal{F})$ as an attempted description of a k-dimensional function $g(\chi_{\alpha_1}, \ldots, \chi_{\alpha_k})$. If there are two identical rows of $\mathcal{W}'$ whose corresponding $\mathcal{F}$-values disagree then $(\mathcal{W}', \mathcal{F})$ is an inconsistent truth table that does not correspond to any Boolean function. Otherwise $(\mathcal{W}', \mathcal{F})$ is the truth table of some k-variable Boolean function g.

Now recall from the beginning of this Section that $\mathcal{C}'$ is a class of Boolean functions on k variables, and $\mathcal{C}$ – the class we are testing – is the induced subclass of k-dimensional functions on $\mathbb{F}_2^n$. We say that the k-restriction $(\mathcal{W}', \mathcal{F})$ is $\mathcal{C}'$*-consistent* if it is the truth table of some function $g \in \mathcal{C}'$.

We now give our test for testing subclasses of k-dimensionality (i.e. testing membership in $\mathcal{C}'$):

Algorithm Test-Induced-Subclass
Inputs: k, ϵ
Additional parameter settings: $s = 2^k, m = 200k2^k$

1. Run Algorithm **Test-Dim-Using-Truth-Table**.
2. Let $\mathcal{W}$ be the expanded version of $\mathcal{Q}$ as described above.
3. Accept if and only if there exists a function in $\mathcal{C}'$ that is consistent with some k'-restriction of $(\mathcal{W}, \mathcal{F})$ where $k' \leq k$.

Notice that Step 2 above uses no additional randomness and no additional queries. Any method for performing Step 2 is acceptable, even brute force search.

THEOREM 6.12. *Let* $\mathcal{C}'$ *be a class of Boolean functions on up to* k *bits; assume each function in* $\mathcal{C}'$ *depends on each of its input bits. Let* $\mathcal{C}$ *be the induced subclass of* k*-dimensional functions over* $\mathbb{F}_2^n$*. Then Algorithm* **Test-Induced-Subclass** *makes* $\mathrm{poly}(2^k, 1/\epsilon)$ *nonadaptive queries and* ϵ*-tests the class* $\mathcal{C}$*. (The running time depends on the implementation of Step 3.)*

Proof. Both the completeness and soundness can straightforwardly be verified to follow from Theorem 6.10. The main thing to note in the argument establishing completeness is that if $f = g(\chi_{\alpha_1}, \ldots, \chi_{\alpha_{k'}})$ for some $g \in \mathcal{C}'$, then although the α_i's are not necessarily in B each of them must be in $\mathrm{span}(B)$. (This uses the fact that h depends nontrivially on each of its inputs.) □

We can give some naive upper bounds regarding the running time for Step 2. Using brute force search for the right $k' \le k$ columns, we have a running time of $O(2^{k^2})T$, where T is the time required to check if a given k'-restriction is consistent with some function in $\mathcal{C}'$. Furthermore, T is certainly bounded by $O(2^{2^k})$, so for *every* induced subclass of k-dimensionality we have a running time with only linear dependence on n (but possibly doubly-exponential dependence on k). In most natural cases, T is polynomial in 2^k, leading to the improved running time of $2^{O(k^2)}$. For example, since we can determine whether a truth table is a linear threshold function in polynomial time (with linear programming), the class of k-sparse polynomial threshold functions can be tested with $\mathrm{poly}(2^k, 1/\epsilon)$ queries and $\mathrm{poly}(2^{k^2}, 1/\epsilon) \cdot n$ time. Improvement even to time $2^{O(k)}$ maybe possible for this or other natural classes; we leave this as a question for further investigation.

7. Lower bounds. In this section we show that the query complexities of our s-sparsity test and k-dimensionality test are tight up to polynomial factors. In fact, our lower bound Theorem 7.1 is somewhat stronger. First, though, let us review some known lower bounds.

Buhrman et al. [BFNR08] implicitly considered the testability of k-dimensionality. In their Theorem 6, they showed that any adaptive $1/8$-tester for k-dimensional functions (for any $k \le n-1$) must make $\Omega(2^{k/2})$ queries. In an earlier work, Alon et al. [AKK+05] gave a lower bound for testing whether a function has degree k. Their result shows that there is some positive ϵ such that any nonadaptive ϵ-tester for having degree k must make $\Omega(2^k)$ queries.

Our lower bound combines, clarifies, and partially strengthens these two results:

THEOREM 7.1. *Fix $\tau > 0$ and let $C = C(\tau)$ be sufficiently large (one can check that $O(\log(1/\tau))$ suffices). Define the following two probability distributions on functions $f : \mathbb{F}_2^{Ck} \to \{-1, 1\}$:*

- *$\mathcal{D}_{\mathrm{yes}}$: Choose a random k-dimensional coset structure $(H, \mathcal{C})$ on the* strings *in $\mathbb{F}_2^{Ck}$ and form f by making it a randomly chosen constant from $\{-1, 1\}$ on each bucket.*
- *$\mathcal{D}_{\mathrm{no}}$: Choose a completely random function on $\mathbb{F}_2^{Ck}$ conditioned on it being $(1/2 - \tau)$-far from every function that has $\mathbb{F}_2$-degree at most k.*

Then any adaptive query algorithm which distinguishes $\mathcal{D}_{\mathrm{yes}}$ and $\mathcal{D}_{\mathrm{no}}$ with probability exceeding $1/3$ must make at least $\Omega(2^{k/2})$ queries.

Note that $\mathcal{D}_{\mathrm{yes}}$ is supported on k-dimensional functions and $\mathcal{D}_{\mathrm{no}}$ is supported on functions far from even having $\mathbb{F}_2$-degree k. Using (1.3), this result immediately gives a $\Omega(2^{k/2})$-query lower bound for adaptively $(1/2 - \tau)$-testing k-dimensionality and an $\Omega(s^{1/2})$-query lower bound for adaptively $(1/2 - \tau)$-testing s-sparsity.

Note that it suffices to prove Theorem 7.1 for *deterministic* adaptive query algorithms. This is the "easy direction" of Yao's Principle: if $\mathcal{A}$ is a randomized distinguisher, we have

$$1/3 < \mathop{\mathbf{Pr}}_{\mathcal{A}\text{'s coins}, f \sim \mathcal{D}_{\mathrm{yes}}} [\mathcal{A}_{\mathrm{coins}}(f) = \mathrm{acc}] - \mathop{\mathbf{Pr}}_{\mathcal{A}\text{'s coins}, f \sim \mathcal{D}_{\mathrm{no}}} [\mathcal{A}_{\mathrm{coins}}(f) = \mathrm{acc}]$$
$$= \mathop{\mathbf{E}}_{\mathcal{A}\text{'s coins}} \left[\mathop{\mathbf{Pr}}_{f \sim \mathcal{D}_{\mathrm{yes}}} [\mathcal{A}_{\mathrm{coins}}(f) = \mathrm{acc}] - \mathop{\mathbf{Pr}}_{f \sim \mathcal{D}_{\mathrm{no}}} [\mathcal{A}_{\mathrm{coins}}(f) = \mathrm{acc}] \right],$$

and so by averaging there exists a setting for the coins giving a deterministic distinguisher which is at least as good.

A q-query deterministic adaptive query algorithm is nothing more than a *decision tree* of depth at most q, where the internal nodes are labeled by query strings from $\mathbb{F}_2^{Ck}$ and the leaves are labeled by "accept" and "reject". In fact, we need not be concerned with leaf labels. Given a decision tree $\mathcal{T}$ with unlabeled leaves, it is well known (indeed, it is essentially by

definition) that the error of the best distinguisher one can get by labeling the leaves is precisely $\|\mathcal{L}_{\text{yes}} - \mathcal{L}_{\text{no}}\|_{TV}$. Here $\mathcal{L}_{\text{yes}}$ ($\mathcal{L}_{\text{no}}$) denotes the distribution on leaves of $\mathcal{T}$ induced by a draw from $\mathcal{D}_{\text{yes}}$ ($\mathcal{D}_{\text{no}}$), and $\|\cdot\|_{TV}$ denotes total variation distance.

Thus to prove Theorem 7.1, the following suffices: Fix a decision tree $\mathcal{T}$ with depth

$$q \le (1/10)2^{k/2}.$$

We may assume that no string appears twice on any root-to-leaf path and that the depth of every path is precisely q. We prove that

$$\|\mathcal{L}_{\text{yes}} - \mathcal{L}_{\text{no}}\|_{TV} \le 1/3, \tag{7.1}$$

and this establishes Theorem 7.1.

We will prove (7.1) via two lemmas.

LEMMA 7.2. *Let $\mathcal{D}_{\text{unif}}$ denote the uniform distribution on functions $\mathbb{F}_2^{Ck} \to \{-1,1\}$. Under $\mathcal{D}_{\text{unif}}$, the probability that f is $(1/2-\tau)$-close to having degree k is at most $1/100$.*

Proof. A statement along these lines was given in [AKK+05]; we fill in the details of the volume argument here. Fix any function $g : \mathbb{F}_2^{Ck} \to \{-1,1\}$; when $f \sim \mathcal{D}_{\text{unif}}$, the probability that it is $(1/2-\tau)$-close to g is at most $\exp(-2\tau^2 2^{Ck})$, by a standard large-deviation bound. Union-bounding over all degree-k functions g, of which there are $\sum_{i=0}^{k} 2^{\binom{Ck}{i}} \le (k+1)2^{\binom{Ck}{k}}$, gives an overall probability of at most

$$(k+1)2^{\binom{Ck}{k}} \cdot \exp(-2\tau^2 2^{Ck}) \le \exp(\binom{Ck}{k} - 2\tau^2 2^{Ck}).$$

This is certainly at most $1/100$ if we take $C = C(\tau)$ large enough. □

We can define $\mathcal{L}_{\text{unif}}$ by analogy with $\mathcal{L}_{\text{yes}}$ and $\mathcal{L}_{\text{no}}$; clearly, $\mathcal{L}_{\text{unif}}$ is the uniform distribution on the 2^q leaves of $\mathcal{T}$.

LEMMA 7.3. $\|\mathcal{L}_{\text{yes}} - \mathcal{L}_{\text{unif}}\|_{TV} \le 1/99$

Proof. This proof is similar to the one in [BFNR08], although we believe we are correcting a gap in that argument. Consider a draw $f \sim \mathcal{D}_{\text{yes}}$; recall this defines a random k-dimensional coset structure $(H, \mathcal{C})$. For a particular leaf v in $\mathcal{T}$, consider the strings appearing on the path to v. By q's definition we have $k \ge 2\log q + \log(100)$; hence Proposition 2.9(3) implies that, except with probability at most $1/100$ over the choice of $(H, \mathcal{C})$, all strings on this path to v fall into different buckets. Conditioned on this happening, the probability that f is consistent with the path to v is precisely 2^{-q}. Thus we have shown that for each leaf v,

$$\mathbf{Pr}_{\mathcal{L}_{\text{yes}}}[v] \ge (1 - 1/100)2^{-q}.$$

The lemma now follows from Proposition 7.4 below. □

PROPOSITION 7.4. *Let P be a probability distribution on a set of size m in which each element has probability at least $(1-\delta)/m$. Let U denote the uniform distribution. Then $\|P - U\|_{TV} \le \delta/(1-\delta)$.*

Proof. The unaccounted-for probability mass in P is at most δ. Hence $\|P - (1-\delta)U\|_1 \le \delta$, and therefore $\|P/(1-\delta) - U\|_1 \le \delta/(1-\delta)$. But $\|P/(1-\delta) - P\|_1 = (\delta/(1-\delta))\|P\|_1 = \delta/(1-\delta)$. Thus by the triangle inequality we have $\|P - U\|_1 \le 2\delta/(1-\delta)$, completing the proof. □

Finally, to complete the proof of (7.1) and thus Theorem 7.1, simply note that Lemma 7.2 implies $\|\mathcal{D}_{\text{no}} - \mathcal{D}_{\text{unif}}\|_{TV} \le 1/100$, hence $\|\mathcal{L}_{\text{no}} - \mathcal{L}_{\text{unif}}\|_{TV} \le 1/100$; then use Lemma 7.3 and the triangle inequality: $1/100 + 1/99 \le 1/3$.

8. Applications to Decoding and Learning. The soundness of the tests discussed so far is proved by (implicitly) giving an algorithm that reconstructs a nearby sparse or low-dimensional function. In this section, we make these algorithms explicit, and show that they are in fact tolerant to rather high levels of noise. We show that they work up to the *unique decoding radius* for these classes, which is the best one could hope for.

Note that the bound $\deg_2(f) \leq \log \operatorname{spar}(f)$ implies that one could use known unique-decoding algorithms for $\mathbb{F}_2$ polynomials of degree $\log s$ to unique decode sparse functions [GKZ08]. However, the running time of such an approach is $O(n^{\log s})$ whereas we will achieve running time of $\operatorname{poly}(n, s)$. Similarly, in the low-dimensional case, we achieve a running time of $\operatorname{poly}(n, 2^k)$ as opposed to $O(n^k)$.

8.1. A unique-decoder for sparse functions. We proved the completeness of our Sparsity tester by showing that rounding the Fourier coefficients of the function f somewhat surprisingly gives a Boolean function. In this section, we examine this rounding algorithm in detail and show that it gives a *unique-decoder* for the class of s-sparse Boolean functions which works up to half the minimum distance.

We study the granularity of s-sparse functions. Note that plugging $\mu = 0$ in Theorem 3.3 shows that every s-sparse function is $\lceil \log s \rceil$ granular, while a closer inspection of the proof reveals that one can improve this to $\lceil \log s \rceil - 1$ granular. We present a different proof which gives the optimal bound of $\lfloor \log s \rfloor - 1$.

THEOREM 8.1. *Suppose $f : \mathbb{F}_2^n \to \{-1, 1\}$ is s-sparse, $s > 1$. Then f has granularity $\lfloor \log s \rfloor - 1$. (Of course, if f is 1-sparse then it is 0-granular.)*

Proof. By induction on n. If $n = 0$ then s must be 1 and there is nothing to prove. For general $n > 0$ we consider two cases. The first is that $s = 2^n$. In this case, since every Fourier coefficient is an average of 2^n many ± 1's, it is of the form $(\text{even integer})/2^n$ and hence has granularity $n - 1 = \lfloor \log s \rfloor - 1$, as required by the theorem.

The second case is that $s < 2^n$. In this case there is an α such that $\widehat{f}(\alpha) = 0$. Since multiplying f by χ_α changes neither its sparsity nor its granularity, we may assume that $\widehat{f}(0^n) = 0$. Now for an arbitrary $\beta \neq 0^n$ we will show that $\widehat{f}(\beta)$ has granularity $\lfloor \log s \rfloor - 1$, completing the proof.

Since $\beta \neq 0^n$ we can pick $i \in [n]$ such that $\beta_i + 1 = 0$. Consider now the function $g : \mathbb{F}_2^{[n]\setminus i} \to \{-1, 1\}$ defined by

$$g(x) = f(x_1, \ldots, x_{i-1}, \sum_{j \in [n]\setminus i} x_j \beta_j, x_{i+1}, \ldots, x_n). \tag{8.1}$$

Substituting $\sum_{j \in [n]\setminus i} x_j \beta_j$ for variable x_i in the Fourier expansion of $f(x)$, it is easy to check that for each $\gamma \in \mathbb{F}_2^{[n]\setminus i}$ the coefficient of χ_γ in the Fourier expansion of (8.1) is $\widehat{f}(\gamma) + \widehat{f}(\gamma + \beta)$, and thus we have that $\widehat{g}(\gamma) = \widehat{f}(\gamma) + \widehat{f}(\gamma + \beta)$. In particular, this implies that $\widehat{g}(0^n) = \widehat{f}(0^n) + \widehat{f}(0^n + \beta) = \widehat{f}(\beta)$. Since f is s-sparse, the definition of g implies that g is also s-sparse. But now the induction hypothesis applied to g (a function on $n - 1$ variables) implies that $\widehat{g}(0^n)$ has granularity $\lfloor \log s \rfloor - 1$, and hence so does $\widehat{f}(\beta)$. □

Easy examples such as the AND function show that the granularity bound above is the best possible. By using Theorem 8.1 and Parseval's identity, one can show the interesting fact that any function $f : \mathbb{F}_2^n \to \{-1, 1\}$ has sparsity either 1, 4, or at least 8.

Application to learning theory. Theorem 8.1 implies that a variant of the membership query learning algorithm of [KM93] can be used to *exactly* reconstruct the Fourier representation of any s-sparse function f in $\operatorname{poly}(n, s)$ time. Specifically, using [KM93] one can find and

approximate to within $\pm 1/(3s)$ all Fourier coefficients of f with $|\hat{f}(\alpha)| \geq 1/s$. By Theorem 8.1, by rounding each coefficient to the nearest number of granularity $\lfloor \log s \rfloor - 1$, we exactly determine all nonzero Fourier coefficients. Prior to this, the analysis of [KM93] implied that an exactly correct hypothesis could be obtained in $\mathrm{poly}(n, s)$ time; however the hypothesis was the sign of some approximation of the Fourier spectrum of f. Using our result, we establish for the first time that sparse functions are efficiently exactly *properly* learnable.

Indeed, one can show that this version of KM gives a unique-decoder for sparse polynomials at low error rates. Recall that every s-sparse polynomial has $\mathbb{F}_2$ degree bounded by $d = \lfloor \log s \rfloor$. Thus any two sparse polynomials must differ at 2^{-d} fraction of points in the Boolean hypercube, and it is easy to see that this bound is tight. Thus, sparse functions give a code of distance 2^{-d}, so given any function $f : \mathbb{F}_2^n \to \{\pm 1\}$, there can be at most one sparse function g so that $d(f, g) < 2^{-(d+1)}$.

THEOREM 8.2. *Let $f : \mathbb{F}_2^n \to \{\pm 1\}$ be such that there exists a sparse function g so that $d(f, g) < 2^{-(d+1)}$. The function g can be recovered from f by rounding each $\hat{f}(\alpha)$ to the nearest $(d-1)$ granular number.*

Proof. One can view f as being obtained from g by changing its values at $\eta < 2^{-(d+1)}$ fraction of points on the hypercube. Thus we have $f(x) = g(x) + \nu(x)$ where $|\nu(x)| = 2$ at η fraction of points x, and $\nu(x) = 0$ otherwise. It follows that $\hat{\nu}(\alpha) \leq 2\eta < 2^{-d}$ for all $\alpha \subseteq [n]$.

But since each coefficient $\hat{g}(\alpha)$ is $(d-1)$-granular, and any two such distinct numbers are $2 \cdot 2^{-d}$ apart, the only $(d-1)$-granular number z satisfying $|z - \hat{f}(\alpha)| < 2^{-d}$ is $\hat{g}(\alpha)$. So rounding Fourier coefficients recovers the function $g(x)$. □

This also shows that by running the KM algorithm and rounding the Fourier coefficients, we can efficiently recover s-sparse polynomials in time $\mathrm{poly}(n, s, \epsilon^{-1})$ from adversarial error (mislabeled labels) of rate $\eta = 2^{-(d+1)} - \epsilon$. We identify the s largest coefficients using KM and estimate them to accuracy $\frac{\epsilon}{s}$. We then round them to the nearest $\lfloor \log s \rfloor - 1$-granular number. An argument similar to the one above shows that we recover the sparse polynomial with good probability.

8.2. A unique-decoder for low-dimensional functions. Given $f : \mathbb{F}_2^n \to \{\pm 1\}$, let $F : \mathbb{F}_2^n \to \mathbb{F}_2$ denote its representation as a polynomial over $\mathbb{F}_2$ which satisfies

$$f(x) = (-1)^{F(x)}.$$

For $h \in \mathbb{F}_2^n$ we define the directional derivative $F_h(x)$ as

$$F_h(x) = F(x + h) + F(x).$$

It is easy to see that $\deg_2(F_h) \leq \deg_2(F) - 1 = \deg_2(f) - 1$ for every h. $\mathrm{Inv}(f)$ can be thought of as the subspace of vectors h so that $F_h \equiv 0$. Further, if f is k-dimensional so that $\deg_2(f) = k$, and if $h \notin \mathrm{Inv}(f)$, then the Schwartz-Zippel lemma implies

$$\mathbf{Pr}_{x \in \mathbb{F}_2^n}[F_h(x) \neq 0] \geq 2^{-(k-1)}.$$

This gives a test for membership in $\mathrm{Inv}(f)$ which is robust to noise.

Assume that we are given $f : \mathbb{F}_2^n \to \{\pm 1\}$ so that $d(f, g) \leq 2^{-(k+1)} - \epsilon$ for some $\epsilon > 0$, and g is k-dimensional. Our goal is to recover g from f. The first step is a test for membership in $\mathrm{Inv}(g)$, given as Algorithm **Test-Inv**.

Algorithm Test-Inv
Inputs: f, h, ϵ, δ.
Additional parameter settings: $m = \frac{2^{4k}}{\epsilon^2} \log \frac{1}{\delta}$.

1. Pick $x_1, \ldots, x_m \in \mathbb{F}_2^n$ independently and uniformly at random.
2. For each j, if $f(x_j + h) \neq f(x_j)$ add x_j to the multiset S.
3. If $|S|/m \leq 2^{-k}$, accept; else reject.

LEMMA 8.3. *Every $h \in \mathrm{Inv}(g)$ passes the test with probability $1 - \delta$, whereas every $h \notin \mathrm{Inv}(g)$ passes with probability at most δ.*

Proof. Assume that $h \in \mathrm{Inv}(g)$, so that $g(x+h) = g(x)$ for every x. If $f(x+h) \neq f(x)$, then either $f(x) \neq g(x)$ or $f(x+h) \neq g(x+h)$. Thus

$$\begin{aligned}\mathbf{Pr}_x[f(x) \neq f(x+h)] &\leq \mathbf{Pr}_x[f(x) \neq g(x)] + \mathbf{Pr}_x[f(x+h) \neq g(x+h)] \\ &\leq 2(2^{-(k+1)} - \epsilon) \\ &= 2^{-k} - 2\epsilon.\end{aligned}$$

The claim follows by the Chernoff bound.

Now assume that $h \notin \mathrm{Inv}(g)$. Note that by the Schwartz-Zippel lemma,

$$\mathbf{Pr}_x[g(x) \neq g(x+h)] = \mathbf{Pr}_x[G_h(x) \neq 0] \geq 2^{-(k-1)}.$$

Thus, we have

$$\begin{aligned}\mathbf{Pr}_x[f(x) \neq f(x+h)] &\geq \mathbf{Pr}_x[g(x) \neq g(x+h)] - \\ &\qquad (\mathbf{Pr}_x[f(x) \neq g(x)] + \mathbf{Pr}_x[f(x+h) \neq g(x+h)]) \\ &\geq 2^{-(k-1)} - 2(2^{-(k+1)} - \epsilon) \\ &= 2^{-k} + 2\epsilon\end{aligned}$$

Again the claim follows by the Chernoff bound. □

Algorithm Unique-Decode
Inputs: f, ϵ, β.
Additional parameter settings: $\ell = 2(n + 8\ln(1/\beta))2^k$, $m = \frac{2^{4k}}{\epsilon^2} \log \frac{1}{\beta}$.

Phase 1: Learning $\mathrm{Inv}(g)$.

1. Pick $h_1, \ldots, h_\ell \in \mathbb{F}_2^n$ independently and uniformly from $\mathbb{F}_2^n$.
2. For each i, run Algorithm **Test-Inv** with $f, h_i, \epsilon, \delta = \frac{\beta}{\ell}$; if it accepts, add h_i to S.
3. Let $H = \mathrm{span}(S)$.

Phase 2: Learning g (as a truth-table).

4. For each $x \in \mathbb{F}_2^n / H$,
   Pick $h_1, \ldots, h_m$ independently and uniformly from H.
   Set $g(x) = \mathrm{Maj}_{h_i} f(x + h_i)$.

THEOREM 8.4. *Given $f : \mathbb{F}_2^n \to \{\pm 1\}$ such that $d(f, g) < 2^{-(k+1)} - \epsilon$ and g is k-dimensional, Algorithm* **Unique-Decode** *recovers g with probability $1 - 4\beta$.*

We prove this claim by analyzing the two Phases separately. We prove the correctness of Phase 1 using the following simple fact which is an easy consequence of Equation (1) of [LW00]:

FACT 8.5. *Let A be a subspace of $\mathbb{F}_2^n$. Sampling $n + 8\ln(1/\beta)$ vectors independently and uniformly from A will span all of A with probability at least $1 - \beta$.*

LEMMA 8.6. *We have $H = \mathrm{Inv}(g)$ with probability at least $1 - 3\beta$.*

Proof. Of the $\ell = 2(n + 8\ln(1/\beta))2^k$ vectors h_i, the expected number that lie in $\mathrm{Inv}(g)$ is at least $2(n + 8\ln(1/\beta))$. A Chernoff bound gives that at least $n + 8\ln(1/\beta)$ lie in $\mathrm{Inv}(g)$ with probability at least $1 - \beta$, and it is easy to see that conditioned on at least this many vectors being in $\mathrm{Inv}(g)$, they are independent and uniform within that subspace. Since we pick $\delta = \frac{\beta}{\ell}$, with probability at least $1 - \beta$, Algorithm **Test-Inv** correctly labels all the h_is as lying within or outside $\mathrm{Inv}(g)$, hence $S \subseteq \mathrm{Inv}(g)$. But by Fact 8.5, this means that S contains a basis for $\mathrm{Inv}(g)$ with probability at least $1 - \beta$, so the lemma follows. □

LEMMA 8.7. *Algorithm* **Unique-Decode** *returns the correct value of g for every $x \in \mathbb{F}_2^n/\mathrm{Inv}(g)$ with probability $1 - 4\beta$.*

Proof. Assume that $H = \mathrm{Inv}(g)$. Fix $x \in \mathbb{F}_2^n/\mathrm{Inv}(g)$. We have $g(x) = g(x + h)$ for every $h \in H$. The coset $x + H$ contains 2^{n-k} points, of which at most

$$2^n(2^{-(k+1)} - \epsilon) = 2^{n-k}\left(\frac{1}{2} - \frac{\epsilon}{2^k}\right).$$

are corrupted by error. Thus, the Chernoff bound implies that the majority of m samples will give the right answer with probability $\frac{\beta}{2^k}$. To complete the proof, we apply the union bound to all 2^k possible choices for $x \in \mathbb{F}_2^n/\mathrm{Inv}(g)$. □

REFERENCES

[AFNS06] N. Alon, E. Fischer, I. Newman, and A. Shapira. A combinatorial characterization of the testable graph properties: It's all about regularity. In *Proc. STOC*, 2006.

[AKK+05] N. Alon, T. Kaufman, M. Krivelevich, S. Litsyn, and D. Ron. Testing Reed-Muller Codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005.

[AS08a] Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM J. Comput.*, 37(6):1703–1727, 2008.

[AS08b] Noga Alon and Asaf Shapira. Every monotone graph property is testable. *SIAM J. Comput.*, 38(2):505–522, 2008.

[AT08] Tim Austin and Terry Tao. On the testability and repair of hereditary hypergraph properties. *Submitted to Random Structures and Algorithms*, 2008.

[BC99] A. Bernasconi and B. Codenotti. Spectral analysis of boolean functions as a graph eigenvalue problem. *IEEE Trans. Computers*, 48(3):345–351, 1999.

[BCH+96] M. Bellare, D. Coppersmith, J. Hastad, M. Kiwi, and M. Sudan. Linearity testing in characteristic two. *IEEE Trans. on Information Theory*, 42(6):1781–1795, 1996.

[BdW02] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[BFNR08] H. Buhrman, L. Fortnow, I. Newman, and H. Rohrig. Quantum property testing. *SIAM Journal on Computing*, 37(5):1387–1400, 2008.

[BGS98] M. Bellare, O. Goldreich, and M. Sudan. Free bits, pcps and non-approximability-towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998.

[Bla08] Eric Blais. Improved bounds for testing juntas. In *Proc. RANDOM*, pages 317–330, 2008.

[Bla09] Eric Blais. Testing juntas nearly optimally. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 2009.

[BLR93] 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.

[DLM+07] 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.

[FGKP06] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami. New results for learning noisy parities and halfspaces. In *Proc. FOCS*, pages 563–576, 2006.

[Fis01] 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.

[FKR+04] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. *J. Computer & System Sciences*, 68(4):753–787, 2004.

[GKS07] P. Gopalan, S. Khot, and R. Saket. Hardness of reconstructing multivariate polynomials over finite fields. In *Proc. FOCS*, pages 349–359, 2007.

[GKZ08] P. Gopalan, A. Klivans, and D. Zuckerman. List-Decoding Reed-Muller Codes over Small Fields. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 265–274, 2008.

[GOS+09] P. Gopalan, R. O'Donnell, R. Servedio, A. Shpilka, and K. Wimmer. Testing Fourier dimensionality and sparsity. In *Proc. 36th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 500–512, 2009.

[Jac97] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.

[KM93] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, December 1993.

[KS07] T. Kaufman and M. Sudan. Sparse random linear codes are locally decodable and testable. In *Proc. FOCS*, pages 590–600, 2007.

[KS08] T. Kaufman and M. Sudan. Algebraic property testing: the role of invariance. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 403–412, 2008.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[LW00] N. Linial and D. Weitz. Random Vectors of Bounded Weight and Their Linear Dependencies. Paper version of B.Sc. thesis at Hebrew University of Jerusalem; available at http://drorweitz.com/ac/, 2000.

[MORS09] K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. Testing halfspaces. In *Proc. SODA*, pages 256–264, 2009.

[PRS02] M. Parnas, D. Ron, and A. Samorodnitsky. Testing basic boolean formulae. *SIAM J. Disc. Math.*, 16:20–46, 2002.

[Sam07] A. Samorodnitsky. Low-degree tests at large distances. In *Proc.* 39^{th} *ACM Symposium on the Theory of Computing (STOC'07)*, pages 506–515, 2007.