Testing Fourier dimensionality and sparsity

Parikshit Gopalan, Ryan O'Donnell, Rocco A. Servedio,
Amir Shpilka, and Karl Wimmer

parik@microsoft.com, {odonnell,wimmer}@cs.cmu.edu,
rocco@cs.columbia.edu, shpilka@cs.technion.ac.il

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 first give an efficient algorithm for testing whether the Fourier spectrum of a Boolean function is supported in a low-dimensional subspace of $\mathbb{F}_2^n$ (equivalently, for testing whether f is a junta over a small number of parities). We next give an efficient algorithm for testing whether a Boolean function has a sparse Fourier spectrum (small number of nonzero coefficients). 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. Finally, we give an "implicit learning" algorithm that lets us test *any* sub-property of Fourier concision.

Our technical contributions include new structural results about sparse Boolean functions and new analysis of the pairwise independent hashing of Fourier coefficients from [13].

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 [1, 5, 3, 4] and [18, 19] 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 [10, 6], dictators [7, 23], low-degree $\mathbb{F}_2$-polynomials [2, 24], juntas [15, 9], and halfspaces [22]. 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 [12], showing that any class of functions sufficiently well-approximated by juntas is locally testable.

It is natural to think that general characterizations of testability for Boolean functions might come from analyzing the Fourier spectrum (see e.g. [14, 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 [21, 20, 17] can be characterized in terms of the Fourier spectrum.

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 the linear functions, juntas, and low-degree polynomials whose testability has been intensively studied, and thus the testability of these classes is a natural question in its own right. Building on our testing algorithms, we are able to give an "implicit learner" (in the sense of [12]), 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; see Appendix 6 for these applications.

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

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

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). 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 a function of k characters $\chi_{\alpha_1}, \ldots, \chi_{\alpha_k}$, i.e. f is a junta over k parity functions. We write $\dim(f)$ to denote the smallest k for which f is k-dimensional. A function f is said to be *s-sparse* if $|\mathrm{Spec}(f)| \leq s$. We write $\mathrm{sp}(f)$ 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 $\dim(f)$, $\mathrm{sp}(f)$. For any Boolean function f, we have

$$\deg_2(f) \leq \log \mathrm{sp}(f) \leq \dim(f), \tag{2}$$

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. [8, 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 on n variables has $\deg_2(f) = 2$ but $\log \mathrm{sp}(f) = n$. For the second inequality, the addressing function 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:

$$\{k\text{-dimensional}\} \subseteq \{2^k\text{-sparse}\} \subseteq \{\mathbb{F}_2 - \text{degree-}k\}\} \tag{3}$$

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. [2] 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 *any* sub-class of k-dimensional functions using $2^{O(k)}$ queries.[1]

1.2 Our results and techniques

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

Theorem 1. [Testing k-dimensionality – informal] *There is a nonadaptive $O(k2^{2k}/\epsilon)$-query algorithm for ϵ-testing 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. Because of space constraints, this algorithm is given in Appendix 5.

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

Theorem 2. [Testing s-sparsity – informal] *There is a nonadaptive $\mathrm{poly}(s, 1/\epsilon)$-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.*

[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.

The high-level idea behind our tester is that of "hashing" the Fourier coefficients, following [13]. 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 $\mathrm{Spec}(f)$ 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 $\mathrm{Spec}(f)$ 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.

A similar argument to the one used for k-dimensionality shows that if f passes this test, most of its Fourier mass lies on a few coefficients. However, unlike in the low-dimensionality test, 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. 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 prove that truncating the Fourier expansion to these coefficients and rounding them to nearby granular values gives a sparse *Boolean*-valued function (Theorem 6). Thus our sparsity test and its analysis depart significantly from the tests for juntas [15] and from our test for low-dimensionality.

Testing subclasses of k-dimensional functions. Finally, 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 $f(x) = g(\chi_{\alpha_1}(x), \ldots, \chi_{\alpha_k}(x))$ 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 $\mathcal{C}' = \{$ all linear threshold functions over k Boolean variables$\}$.

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

Theorem 3. [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 3 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 3 extends the approach of Theorem 2 with ideas from the "testing by implicit learning" work of [12]. Briefly, by hashing the Fourier coefficients 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 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 $g \in \mathcal{C}'$. We give a more detailed overview at the start of Section 7.

Organization of the paper. We give standard preliminaries and an explanation of our techniques for hashing the Fourier spectrum in Section 2. Section 3 gives our new structural theorems about sparse Boolean functions, and Section 4 uses these theorems to give our test for s-sparse functions. Because of space constraints, our results for testing k-dimensional functions, for unique-decoding, for testing induced subclasses of k-dimensional functions, and our lower bounds are given in Appendices 5-8 respectively.

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 (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 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 $\mathrm{sp}(f) = |\mathrm{Spec}(f)|$. 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 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 [13] in a learning-theoretic context, plays an important role in our tests.

Definition 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:*

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

In other words, we have $\mathrm{P}_{r+H} f = A_{r+H} * f$, where $A_{r+H} \stackrel{def}{=} \sum_{\alpha \in r+H} \chi_\alpha$.

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 [13]):

Fact 4 $\mathrm{P}_{r+H} f(x) = \mathbf{E}_{y \in H^\perp}[\chi_r(y) f(x+y)]$.

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 1. *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 4. 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 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 4, 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. *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 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 3. *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 multiset of $C(b)$'s, which has cardinality 2^t.

Remark 1. 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. 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 3. *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 $\mathbf{Pr}[\alpha, \alpha'$ 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 3 in mind, we give the following simple deviation bound for the sum of pairwise independent random variables:

Proposition 4. *Let $X = \sum_{i=1}^n X_i$, where the X_i's are pairwise independent random variables satisfying $0 \leq X_i \leq \tau$. Assume $\mu = \mathbf{E}[X] > 0$. Then for any $\epsilon > 0$, we have $\mathbf{Pr}[X \leq (1-\epsilon)\mu] \leq \frac{\tau}{\epsilon^2\mu}$.*

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

Finally, it is slightly annoying that Part 1 of Proposition 3 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 4. *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 5. *For a random permuted t-dimensional coset structure, Proposition 3 continues to hold, with Part 1 even holding for $\alpha = 0$.*

Proof. Use Proposition 3 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; we also present a learning application in Section 6.

Definition 5. *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 $\sum_{\alpha \in S} \hat{f}(\alpha)^2 \leq \mu^2$.*

Definition 6. *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 $|v - j/2^k| \leq \mu$ 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 1 *[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.*

Proof. Pick a set of $k = \lceil \log s \rceil + 1$ equations $A\alpha = b$ at random. Let $A^\perp \subset \{0,1\}^n$ be the set of solutions to $A\alpha = 0$. Define H to be the coset of $A^\perp$ of solutions to $A\alpha = b$. We have

$$\mathrm{P}_H f(x) = \sum_{\alpha \in H} \hat{f}(\alpha)\chi_\alpha(x).$$

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

$$\mathbf{Pr}_{A,b}[\exists j \neq i \text{ such that } A\alpha_j = b \,|\, A\alpha_i = b] < \frac{1}{2}. \tag{4}$$

Now consider the coefficients from S. We have

$$\mathbf{E}_{A,b}\left[\sum_{\beta \in S \cap H} \hat{f}(\beta)^2 \middle| A\alpha_i = b\right] = \sum_{\beta \in S} \mathbf{Pr}[\beta \in H | A\alpha_i = b]\hat{f}(\beta)^2 \leq 2^{-k}\mu^2 \leq \frac{\mu^2}{2s}.$$

Hence by Markov's inequality,

$$\mathbf{Pr}_{A,b}\left[\sum_{\beta \in S \cap H} \hat{f}(\beta)^2 \geq \frac{\mu^2}{s} \middle| A\alpha_i = b\right] \leq \frac{1}{2}. \tag{5}$$

Thus by applying the union bound to Equations 4 and 5, 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 A, b. Fixing this choice, we have

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

But by Fact 4 we also have $\mathrm{P}_H f(x) = \mathbf{E}_{y \in A}[\chi_b(y) f(x+y)]$. Thus the function $\mathrm{P}_H f(x)$ is the average of a Boolean function over 2^k points, hence it is $(k-1)$-granular.

We now consider the function

$$g(x) = \sum_{\beta \in S \cap H} \hat{f}(\beta)\chi_\beta(x).$$

Since $\mathbf{E}_x[g(x)^2] \leq \frac{\mu^2}{s}$, for some $x \in \{\pm 1\}^n$ we have $g(x)^2 \leq \frac{\mu^2}{s}$, hence $g(x) \leq \frac{\mu}{\sqrt{s}}$. Fixing this x, we have $\mathrm{P}_H f(x) = \hat{f}(\alpha_i)\chi_{\alpha_i}(x) + g(x)$, and hence $|\hat{f}(\alpha_i)| = |\mathrm{P}_H f(x) - g(x)|$. Since $\mathrm{P}_H f(x)$ is $(k-1)$-granular and $|g(x)| \leq \frac{\mu}{\sqrt{s}}$, 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 6. *[Soundness Theorem.] Let $f : \mathbb{F}_2^n \to \{-1, 1\}$ be $\mu \leq \frac{1}{20s^2}$ close to s-sparse in ℓ_2. Then there is an s-sparse Boolean function $F : \mathbb{F}_2^n \to \{-1, 1\}$ within Hamming distance $\frac{\mu^2}{2}$.*

Proof. Let $B = \{\alpha_1, \cdots, \alpha_s\}$ be the s largest Fourier coefficients of f and let $k = \lceil \log s \rceil$. By Lemma 1, 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 $\beta \in S$. 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 $\{-1, 1\}$. In this case, 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 functions on $\mathbb{F}_2^n$ we have

$$1 = f^2 = F^2 + 2FG + G^2 = F^2 + G(2f - G). \tag{6}$$

Writing $H = G(2f - G)$, from Fact 7 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 (6) 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 7 *Let $f, g : \mathbb{F}_2^n \to \mathbb{R}$. Then $|\widehat{fg}(\alpha)| \leq \|f\|_2 \|g\|_2$ for every α.*

Proof. Using Cauchy-Schwartz and Parseval,

$$|\widehat{fg}(\alpha)| = |\sum_\beta \widehat{f}(\beta)\widehat{g}(\alpha + \beta)| \leq \sqrt{\sum_\beta \widehat{f}(\beta)^2}\sqrt{\sum_\beta \widehat{g}(\alpha + \beta)^2} = \|f\|_2 \|g\|_2. \qquad \square$$

4 Testing s-sparsity

The following is our algorithm for testing whether $f : \mathbb{F}_2^n \to \{-1, 1\}$ is s-sparse:

TESTING s-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.
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 6 now shows that f is close to a sparse function. Our theorem about the test is the following:

Theorem 8. *Algorithm 4 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 8 follows immediately from Proposition 2 and the fact that there are $2^t = O(s^2)$ buckets. In the remainder of this section we present the completeness (Lemma 1) and the soundness (Lemma 4) of the test. We begin with the completeness, which is straightforward.

Lemma 1. *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. Similar to Section 5 we partition the Fourier coefficients of f into two sets: B of big coefficients and S of small coefficients. (The 0-character does not play a special role as it does in Section 5.) Formally, let

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

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

Lemma 2. *If $|B| \geq s+1$ then the test rejects with probability at least $3/4$.*

Proof. Proposition 5(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 3. *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 5. 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 4 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. *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 2 we have $|B| \leq s$, by Lemma 3 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 6 to conclude that f must be $\frac{\mu^2}{2} \leq \epsilon$-close in Hamming distance to an s-sparse Boolean function. □

References

1. ALON, N., FISCHER, E., NEWMAN, I., AND SHAPIRA, A. A combinatorial characterization of the testable graph properties: It's all about regularity. In *Proc. STOC* (2006).
2. ALON, N., KAUFMAN, T., KRIVELEVICH, M., LITSYN, S., AND RON, D. Testing low-degree polynomials over GF(2). In *Proc. RANDOM* (2003), pp. 188–199.
3. ALON, N., AND SHAPIRA, A. A characterization of the (natural) graph properties testable with one-sided error. In *Proc. FOCS'05* (2005), pp. 429–438.
4. ALON, N., AND SHAPIRA, A. Every monotone graph property is testable. In *Proc. STOC 2005* (2005), pp. 128–137.
5. AUSTIN, T., AND TAO, T. On the testability and repair of hereditary hypergraph properties. *Submitted to Random Structures and Algorithms* (2008).
6. BELLARE, M., COPPERSMITH, D., HASTAD, J., KIWI, M., AND SUDAN, M. Linearity testing in characteristic two. *IEEE Trans. on Information Theory 42*, 6 (1996), 1781–1795.
7. BELLARE, M., GOLDREICH, O., AND SUDAN, M. Free bits, pcps and non-approximability-towards tight results. *SIAM J. Comput. 27(3)* (1998), 804–915.
8. BERNASCONI, A., AND CODENOTTI, B. Spectral analysis of boolean functions as a graph eigenvalue problem. *IEEE Trans. Computers 48*, 3 (1999), 345–351.
9. BLAIS, E. Improved bounds for testing juntas. In *To appear in RANDOM'08* (2008).
10. BLUM, M., LUBY, M., AND RUBINFELD, R. Self-testing/correcting with applications to numerical problems. *J. Comp. Sys. Sci. 47* (1993), 549–595. Earlier version in STOC'90.
11. BUHRMAN, H., FORTNOW, L., NEWMAN, I., AND ROHRIG, H. Quantum property testing. *SIAM Journal on Computing 37*, 5 (2008), 1387–1400.
12. DIAKONIKOLAS, I., LEE, H., MATULEF, K., ONAK, K., RUBINFELD, R., SERVEDIO, R., AND WAN, A. Testing for concise representations. In *Proc. FOCS)* (2007), pp. 549–558.
13. FELDMAN, V., GOPALAN, P., KHOT, S., AND PONNUSWAMI, A. New results for learning noisy parities and halfspaces. In *Proc. FOCS* (2006), pp. 563–576.
14. FISCHER, E. The art of uninformed decisions: A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science 75* (2001), 97–126.
15. FISCHER, E., KINDLER, G., RON, D., SAFRA, S., AND SAMORODNITSKY, A. Testing juntas. *J. Computer & System Sciences 68*, 4 (2004), 753–787.
16. GOPALAN, P., KHOT, S., AND SAKET, R. Hardness of reconstructing multivariate polynomials over finite fields. In *Proc.FOCS* (2007), pp. 349–359.
17. JACKSON, J. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences 55* (1997), 414–440.
18. KAUFMAN, T., AND SUDAN, M. Sparse random linear codes are locally decodable and testable. In *Proc. FOCS* (2007), pp. 590–600.
19. KAUFMAN, T., AND SUDAN, M. Algebraic property testing: the role of invariance. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)* (2008), pp. 403–412.
20. KUSHILEVITZ, E., AND MANSOUR, Y. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing 22*, 6 (Dec. 1993), 1331–1348.
21. LINIAL, N., MANSOUR, Y., AND NISAN, N. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM 40*, 3 (1993), 607–620.
22. MATULEF, K., O'DONNELL, R., RUBINFELD, R., AND SERVEDIO, R. Testing Halfspaces. Tech. Rep. 128, Electronic Colloquium in Computational Complexity, 2007.
23. PARNAS, M., RON, D., AND SAMORODNITSKY, A. Testing basic boolean formulae. *SIAM J. Disc. Math. 16* (2002), 20–46.
24. SAMORODNITSKY, A. Low-degree tests at large distances. In *Proc.* 39^{th} *ACM Symposium on the Theory of Computing (STOC'07)* (2007), pp. 506–515.

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 7. *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. [16]):

Fact 9 *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 9 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 follows.

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

Our theorem about this test is the following:

Theorem 10. *Algorithm 5 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 10 is immediate. It remains to present the completeness (Lemma 5) and the soundness (Lemma 8) of the test. We begin with the completeness, which is straightforward:

Lemma 5. *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 6), and also have most of the Fourier weight (Lemma 7). 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 8. *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 11

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

Proof. This follows easily from Fact 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 6. *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 11 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 4.

Lemma 7. *If $\mathrm{wt}(S) > \epsilon$ then the test rejects with probability at least $2/3$.*

Proof. As in Lemma 6, 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 3. 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 4 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 11 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 6). 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 8. *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 $1/3$, Lemmas 6 and 7 imply that both $\dim(\mathrm{span}(B)) \leq k$ and $\mathrm{wt}(S) \leq \epsilon$. 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 (a well-known consequence of $\|f - F\|_2^2 \leq \epsilon$). □

6 Applications to Unique Decoding.

The soundness of both our tests is proved by (implicitly) giving an algorithm that reconstructs a nearby sparse/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 $\deg_2(f) \leq \log \mathrm{sp}(f)$ implies that one could use known unique-decoding algorithms for $\mathbb{F}_2$ polynomials of degree $\log s$ to unique decode sparse functions. However, the running time of such an approach is $O(n^{\log s})$ whereas we will achieve running time of $\mathrm{poly}(n, s)$. Similarly, in the low-dimensional case, we achieve a running time of $\mathrm{poly}(n, 2^k)$ as opposed to $O(n^k)$.

6.1 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 Lemma 1 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 12. *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 we can choose an α such that $\widehat{f}(\alpha) = 0$. Now for an arbitrary $\beta \neq \alpha$ we will show that $\widehat{f}(\beta)$ has granularity $\lfloor \log s \rfloor - 1$, completing the proof. Since $\beta \neq \alpha$ we can pick $i \in [n]$ such that $\alpha_i + \beta_i + 1 = 0$. Consider now the function $g : \mathbb{F}_2^{[n]\setminus i} \to \{-1, 1\}$ defined by

$$g(x) = f(x_1, \ldots, x_{i-1}, \langle x, \alpha + \beta + e_i \rangle, x_{i+1}, \ldots, x_n).$$

It is easy to check that for each $\gamma \in \mathbb{F}_2^{[n]\setminus i}$, we have $\widehat{g}(\gamma) = \widehat{f}(\gamma) + \widehat{f}(\gamma + \alpha + \beta)$, and in particular $\widehat{g}(\alpha) = \widehat{f}(\alpha) + \widehat{f}(\beta) = \widehat{f}(\beta)$. 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 $\widehat{g}(\alpha)$ 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 12 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 12 implies that a variant of the membership query learning algorithm of [20] can be used to *exactly* reconstruct the Fourier representation of any s-sparse function f in poly(n, s) time. Specifically, using [20] one can find and approximate to within $\pm 1/(3s)$ all Fourier coefficients of f with $|\hat{f}(\alpha)| \geq 1/s$. By Theorem 12, 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 [20] implied that an exactly correct hypothesis could be obtained in 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 13. *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 $f(x) = g(x) + n(x)$ where $|n(x)| = 2$ at η fraction of points x, and $\eta(x) = 0$ otherwise. It follows that $\hat{n}(\alpha) \leq 2\eta$ for all $\alpha \subseteq [n]$.

But since each coefficient $\hat{g}(\alpha)$ is $(d-1)$-granular, and any two such 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 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.

6.2 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 $\deg_2(F_h) \leq \deg_2(f) - 1$ for every h. $\mathrm{Inv}(f)$ can be thought of as the subspace of vectors h so that $F_h = 0$. 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 $\mathrm{Inv}(g)$.

TESTING MEMBERSHIP IN $\mathrm{Inv}(g)$
Inputs: f, h, ϵ, δ.
Additional parameter settings: $m = \frac{2^{4k}}{\epsilon^2} \log \frac{1}{\delta}$.

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

Lemma 9. *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

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

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

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

Again the claim follows by the Chernoff bound. □

UNIQUE-DECODING LOW-DIMENSIONAL FUNCTIONS
Inputs: f, ϵ, β.
Additional parameter settings: $\ell = 4n2^k$, $m = \frac{2^{4k}}{\epsilon^2} \log \frac{1}{\beta}$.

Phase 1: Learning $\mathrm{Inv}(g)$.
Pick $h_1, \ldots, h_\ell \in \mathbb{F}_2^n$ independently and uniformly from $\mathbb{F}_2^n$.
    Run Algorithm 6.2 with $f, h_i, \epsilon, \delta = \frac{\beta}{\ell}$; if it accepts, add h_i to S.
Let $H = \mathrm{span}(S)$.

Phase 2: Learning g (as a truth-table).
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 14. *Given $f : \mathbb{F}_2^n \to \{\pm 1\}$ such that $d(f, g) < 2^{-(k+1)} - \epsilon$ and g is k-dimensional, Algorithm 6.2 recovers g with probability $1 - 3\beta$.*

We prove this claim by analyzing the two Phases separately. We prove the correctness of Phase 1 using the following simple fact.

Fact 15 *Let A be a subspace of $\mathbb{F}_2^n$. Sampling $2n$ vectors independently and uniformly from A will span all of A with probability $1 - 2^{-n}$.*

Lemma 10. *We have $H = \mathrm{Inv}(g)$ with probability $1 - 2\beta$.*

Proof. Of the $\ell = 4n2^k$ vectors h_i, at least $2n$ of them come from $\mathrm{Inv}(g)$ with probability $1 - \exp(-n) > 1 - \beta$ by the Chernoff bound. Since we pick $\delta = \frac{\beta}{\ell}$, Algorithm 6.2 correctly labels all the h_is as lying within or outside $\mathrm{Inv}(g)$, hence $S \subseteq \mathrm{Inv}(G)$. But by Fact 15, this means that S contains a basis for $\mathrm{Inv}(G)$, so the lemma follows. □

Lemma 11. *Algorithm 6.2 returns the correct value of g for every $x \in \mathbb{F}_2^n / \mathrm{Inv}(g)$ with probability $1 - 3\beta$.*

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 $x \in \mathbb{F}_2^n / \mathrm{Inv}(g)$. □

7 Testing induced subclasses of k-dimensional functions

Let C be any fixed induced subclass of k-dimensional functions. In this section we show that C is ϵ-testable using $\mathrm{poly}(2^k, 1/\epsilon)$ queries.

Let us give a brief overview of the method. From Section 5 we know that, using about 2^{2k} queries, we can test that a function f is close to some k-dimensional function F. That test, however, does not give us much information about F. On the other hand, the s-sparsity test from Section 5 (with s set to 2^k, yielding query complexity $2^{O(k)}$), *does* give us quite a good handle on the nearby sparse (and k-dimensional) F. Specifically, assuming the underlying F is

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

a successful run of the sparsity test actually obtains (approximate) query access to each of the "pieces" $\tilde{f}(\beta)\chi_\beta$. Note that it does not determine the actual identity of any β in $\mathrm{Spec}(F)$ (this would require a number of queries dependent on n); this is why we get an "implicit learning" scenario.

We can now draw around $O(k2^k)$ random examples and obtain a complete "implicit truth table" for F (since the sparsity test ensures the "ϵ" parameter is $\leq 2^{-4k}$ anyway, we are likely to have no mistakes in this table). By this we mean a table where the rows correspond to strings x, the entries in the rows are the values of the "pieces" $\tilde{f}(\beta)\chi_\beta(x)$, and we have a value $F(x)$ for each row. With this implicit truth table for F in hand, we can check — deterministically and without queries — whether F has any particular property C.

The organization of this section is as follows. We define "implicit truth tables" formally in Section 7.1. The main work appears in Section 7.2, where we give an augmentation to the sparsity test which returns partial implicit truth tables. In Section 7.3 we point out that this augmentation lets us test for k-dimensionality as well; there is no need to additionally run the test from Section 5. In Section 7.4 we discuss how to complete and correct a partial implicit truth table. Finally, in Section 7.5, we discuss how to finish the test of any induced subclass of k-dimensionality via implicit learning.

We close this overview by mentioning that, given parameters k and ϵ, our test will always begin by running the sparsity test Algorithm 4 with $s = 2^k$. (Recall that k-dimensional functions are 2^k-sparse.) Our subsequent analysis will therefore assume that f is a function which Algorithm 4 accepts with probability exceeding $1/4$. Then the function F from Lemma 4 is well-defined, and f is $O(\epsilon_1)$-close to F. In particular, we will use the fact that if f is itself s-sparse then F is *identical* to f. This is because both f and F, being s-sparse, have $\mathbb{F}_2$-degree at most $\log s$, and it is well known (Schwartz-Zippel variant for $\mathbb{F}_2$) that two such polynomials, at distance at most $O(\epsilon_1) \leq 1/s$, must in fact be identical.

7.1 Implicit truth tables

Definition 9. *The* partial implicit truth table *for F corresponding to a list $\mathcal{M}$ of strings $x \in \mathbb{F}_2^n$ consists of a matrix $\mathcal{W} \in \{-1,1\}^{\mathcal{M} \times |B|}$ and a vector $\mathcal{F} \in \{-1,1\}^{\mathcal{M}}$. We call*

$|\mathcal{M}|$ *the* size *of the partial implicit truth table. The columns of the matrix $\mathcal{W}$ are indexed by B, and the (x, β) entry is equal to $\operatorname{sgn}(\tilde{f}(\beta))\chi_\beta(x)$ for all $x \in \mathcal{M}$ and $\beta \in B$. The vector $\mathcal{F}$ has the property that $\mathcal{F}_x = F(x)$. Note that $\mathcal{F}_x$ is uniquely determined by the x-row of $\mathcal{W}$ (since F is determined by the values $\tilde{f}(\beta)\chi_\beta(x)$).*

Definition 10. *A* random implicit truth table of size m *for F is a partial implicit truth table in which $\mathcal{M}$ is a list of m uniformly and independently drawn strings $x \in \mathbb{F}_2^n$.*

Lemma 12. *Consider the matrix $\mathcal{W}$ of a partial implicit truth table under the identification $1 \in \mathbb{R} \leftrightarrow 0 \in \mathbb{F}_2$ and $-1 \in \mathbb{R} \leftrightarrow 1 \in \mathbb{F}_2$. Then the set of possible rows forms a $\dim(F)$-dimensional coset of $\mathbb{F}_2^{|B|}$. In a random implicit truth table, each row is uniformly distributed on this coset.*

Proof. By adding the $\mathbb{F}_2$-identified vector $\langle \operatorname{sgn}(\tilde{f}(\beta)) \rangle_{\beta \in B}$ to each row, 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 $\dim(\operatorname{span}(B))$. 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 $\operatorname{rank}(A) = \dim(\operatorname{span}(B))$. And 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 11. *We call a partial implicit truth table* exhaustive *if all possible $2^{\dim(F)}$ rows occur in $\mathcal{W}$.*

Lemma 13. *Suppose we draw a random implicit truth table for F of size $200k2^k$. If F is k-dimensional then we get an exhaustive implicit truth table except with probability at most $1/100$. If F is not k-dimensional then we see more than 2^k distinct rows except with probability at most $1/100$.*

Proof. These facts follow from the Coupon Collector analysis and Lemma 12. □

7.2 Determining an implicit truth table

Consider the following augmentation to Algorithm 4:

TESTING s-SPARSITY WITH IMPLICIT LEARNING
Inputs: $m \leq O(s^2)$

5. Let $\mathcal{L}' \subseteq \mathcal{L}$ be the buckets whose Step 2 estimate is at least $1/(8s^2)$.
6. Define the length-m column vector $\mathcal{F}$ as follows: Draw a list $\mathcal{M}$ of m uniformly random strings from $\mathbb{F}_2^n$; query f on each $x \in \mathcal{M}$ and set $\mathcal{F}_x = f(x)$.
7. Define the $m \times |\mathcal{L}'|$ matrix $\mathcal{W}$ as follows: For each $x \in \mathcal{M}$ and $C \in \mathcal{L}'$, estimate $\mathrm{P}_C f(x)$ to within $\pm 1/(4s)$ with confidence $1 - 1/(100sm)$, using Proposition 1; set $\mathcal{W}_{x,C}$ to be the sign of the estimate.

Remark 2. This augmentation to Algorithm 4 does not increase its query complexity by more than a constant factor. To see this, note that although the above Algorithm 7.2 is described as being adaptive, we could do it nonadaptively by estimating $\mathrm{P}_C f(x)$ for *every* bucket C. Even this would require query complexity only $m + O(s^2) \cdot m \cdot O(s^2 \log s) \leq O(s^6 \log s)$, which is less than the query complexity of Algorithm 4.

Lemma 14. *After running Algorithms 4 and 7.2, the pair $(\mathcal{W}, \mathcal{F})$ is the partial implicit truth table corresponding to $\mathcal{M}$, except with probability at most $5/100$.*

Proof. Throughout this argument we freely assume that the $O(1)$ in ϵ_1's definition is sufficiently large, including in comparison to the $O(1)$ in the upper-bound on m. Analyzing $\mathcal{F}$ is easy; since f and F are $O(\epsilon_1)$-close as Boolean functions, the probability that $\mathcal{F}_x \neq F(x)$ for any $x \in \mathcal{M}$ is at most $m \cdot O(\epsilon_1) \leq O(s^2\epsilon_1) \leq 1/100$. We thus concentrate on analyzing $\mathcal{W}$. Given that f passes Algorithm 4 with probability exceeding $1/4$, the proof of Lemma 4 implies that $|B| \leq s$, $\mathrm{wt}(S) \leq \|f - F\|_2^2 \leq O(\epsilon_1)$, and each $\hat{f}(\beta)$ is within $O(\epsilon_1/s)$ of a nonzero ℓ-granular number $\tilde{f}(\beta)$. The last of these facts implies that each $\hat{f}(\beta)$ has magnitude at least $1/(2s)$ and has the same sign as $\tilde{f}(\beta)$. In running Algorithms 4 and 7.2, except with probability at most $1/100 + 1/100 + 1/100 \leq 3/100$, the following all hold: after Step 1, all $\beta \in B$ fall into different buckets (by Proposition 5(3)); after Step 2, all estimates are accurate to within $\pm\tau$; and, after Step 7, all estimates are accurate to within $\pm 1/(4s)$. Assuming all of these hold, we begin by identifying a 1-1 mapping $c : B \to \mathcal{L}'$ (recall that $\mathcal{L}'$ indexes the columns of $\mathcal{W}$). Define $c(\beta)$ to be the bucket containing β; so far we know that this function is injective. To see that its range is contained in $\mathcal{L}'$, note that for each $\beta \in B$ we have $|\hat{f}(\beta)| \geq 1/(2s)$; hence the bucket containing β has weight at least $1/(4s^2)$ and therefore it will be put into $\mathcal{L}'$ in Step 5 (using $\tau < 1/(8s^2)$). To show that c is an onto map we need to verify that any bucket in $\mathcal{L}'$ contains a vector from B. Since $\mathrm{wt}(S) \leq O(\epsilon_1) \leq 1/(16s^2)$, even if all vectors $\alpha \notin B$ landed in the same bucket, that bucket would still have weight less than $1/(8s^2) - \tau$ (using $\tau < 1/(16s^2)$) and thus would not be added into $\mathcal{L}'$. Next, 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 $\mathcal{L}'$ and the fact that coset-projection functions have disjoint Fourier support, we have

$$O(\epsilon_1) \geq \|f - F\|_2^2 = \sum_{\beta \in B} \|G_\beta\|_2^2 + \sum_{C \notin \mathcal{L}'} \|\mathrm{P}_C f\|_2^2 \geq \sum_{\beta \in B} \|G_\beta\|_2^2. \tag{7}$$

Say that a string $x \in \mathbb{F}_2^n$ is *bad* for $\beta \in B$ if $|G_\beta(x)| > 1/(2s)$. Clearly the fraction of strings bad for β is at most $(2s)^2 \|G_\beta\|_2^2$. Thus we conclude that the fraction of strings x which are bad for *any* $\beta \in B$ is at most $4s^2 \sum_{\beta \in B} \|G_\beta\|_2^2 \leq O(s^2\epsilon_1)$, using (7). Since $m \leq O(s^2)$, the probability that $\mathcal{M}$ contains any string which is bad for any $\beta \in B$ is at most $O(s^4\epsilon_1) \leq 1/100$. So we assume all strings in $\mathcal{M}$ are good for all $\beta \in B$, and overall we have accumulated failure probability at most $5/100$. It remains to show that assuming x is good for $\beta \in B$, the sign $\mathcal{W}_{x,c(\beta)}$ equals $\mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(x)$. This is straightforward. Since $\tilde{f}(\beta)$ is a nonzero ℓ-granular number, $|\tilde{f}(\beta)\chi_\beta(x)| \geq 1/s$. Thus if x is good for β we must have both that $|\mathrm{P}_{c(\beta)} f(x)| \geq 1/(2s)$ and that $\mathrm{sgn}(\mathrm{P}_{c(\beta)} f(x)) = \mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(x)$. Now the fact that the estimate for $\mathrm{P}_{c(\beta)} f(x)$ is

accurate to within $\pm 1/(4s)$ means that $\mathcal{W}_{x,c(\beta)}$ will have the same sign as $\mathrm{P}_{c(\beta)}f(x)$, as required. □

7.3 An alternate k-dimensionality test

We can now give an alternate test for k-dimensionality. Its query complexity is essentially that of the sparsity test (so worse than that of Section 5, though still polynomial in $2^k/\epsilon$), but it has the crucial advantage of determining exhaustive implicit truth tables.

TESTING k-DIMENSIONALITY WITH EXHAUSTIVE IMPLICIT LEARNING
Inputs: k, ϵ
Additional parameter settings: $s = 2^k$, $m = 200k2^k$

1. Run Algorithm 4.
2. Run Algorithm 7.2.
3. Reject if $\mathcal{W}$ has more than 2^k distinct rows.

Our theorem about Algorithm 7.3 is the following:

Theorem 16. *If f is k-dimensional then this test accepts and outputs an exhaustive implicit truth table 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 test produces an exhaustive implicit truth table for F.*

Proof. For the first statement, if f is k-dimensional it is s-sparse, so Algorithm 4 passes with probability at least $3/4$. Except with probability at most $5/100$, Algorithm 7.2 produces a partial implicit truth table for F of size $200k2^k$. Since $F = f$ is k-dimensional, any implicit truth table for F has at most 2^k distinct rows, by Lemma 12. Thus the test accepts and produces an exhaustive implicit truth table with probability at least $3/4 - 6/100 > 2/3$, as claimed. For the second statement, suppose f passes Algorithm 7.3 with probability exceeding $1/4$. Certainly, f passes Algorithm 4 with probability at least $1/4$, so F is well-defined. Further, F must be k-dimensional as claimed, for otherwise the combination of Lemmas 14 and 13 would imply that f is accepted with probability at most $6/100$. Thus these same two lemmas imply that the test produces an exhaustive implicit truth table for F except with probability at most $6/100$.
□

7.4 Correcting the implicit truth table

Definition 12. *A* corrected implicit truth table *is an implicit truth table with the following additional properties:*

1. *$\mathcal{W}$ and $\mathcal{F}$ have exactly $2^{\dim(F)}$ distinct rows.*
2. *$\mathcal{W}$ has a column for all $\beta \in \mathrm{span}(B)$, not just all $\beta \in B$.*
3. *The $\mathcal{W}_{x,\beta}$ entry is equal to $\chi_\beta(x)$.*

Notice that a corrected implicit truth table has potentially many more columns than an exhaustive implicit truth table. Also, the $\mathcal{W}$ matrix for the corrected version drops the $\mathrm{sgn}(\tilde{f}(\beta))$ term from the exhaustive version. This kind of truth table will be help us do implicit learning. To obtain such a truth table, the main trick is to achieve property 3. Assuming we can do this for all $\beta \in B$, achieving properties 1 and 2 is easy. For 1, we simply eliminate all duplicate rows. For 2, it suffices to widen the matrix $\mathcal{W}$ so that it contains all $2^{\dim(F)}$ columns in its column space; it is easy to do this using Gaussian elimination to find a basis.

To achieve property 3 we need to slightly modify Algorithm 7.2 and the proof in Lemma 14, using the most basic form of linear self-correction. In Step 7, we first draw another list $\mathcal{M}'$ of m uniformly random strings. Then, instead of determining the matrix $\mathcal{W}$ associated to the list $\mathcal{M}$, we instead determine the matrix $\mathcal{W}'$ associated to the list $\mathcal{M}'$, and also the matrix $\mathcal{W}''$ associated to the list $\mathcal{M}'' := \mathcal{M} + \mathcal{M}'$. (By this we mean that the ith string in $\mathcal{M}''$ is the sum of the ith strings in $\mathcal{M}$ and $\mathcal{M}'$.) Finally, we set $\mathcal{W} = \mathcal{W}' \circ \mathcal{W}''$, where $\circ$ denotes the entrywise multiplication. (In the "$\mathbb{F}_2$-identified" versions of these matrices, we are simply doing $\mathcal{W} = \mathcal{W}' + \mathcal{W}''$.) Note that $\mathcal{M}'$ and $\mathcal{M}''$ are both uniformly random lists. By suitably adjusting constants (which ultimately only increases the query complexity by a constant factor), we can ensure that both $\mathcal{W}'$ and $\mathcal{W}''$ are completely correct tables except with probability at most $5/100$. By this we mean that $\mathcal{W}'_{x',c(\beta)} = \mathrm{sgn}(\tilde{f}(\beta))\chi_\beta(x')$ for each $x' \in \mathcal{M}'$ and $\beta \in B$, and similarly for $\mathcal{W}''$. Now by setting $\mathcal{W} = \mathcal{W}' \circ \mathcal{W}''$ we get that $\mathcal{W}_{x,c(\beta)} = \chi_\beta(x)$ for each $x \in \mathcal{M}$ and $\beta \in B$, as required.

Using this modified version of Algorithm 7.2 in Algorithm 7.3, our test is the following:

TESTING k-DIMENSIONALITY WITH CORRECTED IMPLICIT LEARNING
Inputs: Same as those for Algorithms 4 and 7.2.

1. Run Algorithm 4.
2. Run Algorithm 7.2 with self-correction as described above.
3. Reject if $\mathcal{W}$ has more than 2^k distinct rows.

Our arguments have established:

Theorem 17. *In Theorem 16, we can replace "exhaustive" with "corrected" if we use Algorithm 7.4 instead of Algorithm 7.3.*

7.5 Testing subclasses of k-dimensionality with implicit learning

As described in Section 1.2, let $\mathcal{C}'$ be a class of Boolean functions on up to k bits, and let $\mathcal{C}$ be the induced subclass of k-dimensional functions on $\mathbb{F}_2^n$.

Definition 13. *We define a* k-restricted truth table *of $\mathcal{W}$ and $\mathcal{F}$ to be the truth table gotten by taking only k columns of $\mathcal{W}$ while keeping the same $\mathcal{F}$.*

We note the identification of k-restricted truth tables with functions of k characters, since each column of $\mathcal{W}$ corresponds to χ_β for some $\beta \in \mathrm{span}(B)$. We say that a k'-restricted truth table (for $k' \leq k$) is consistent with a function $h \in \mathcal{C}'$ if it is the (normal) truth table of h. We now state our test for testing subclasses of k-dimensionality:

TESTING $\mathcal{C}$
Inputs: k, ϵ.

1. Run Algorithm 7.4.
2. Accept if and only if there exists a function in $\mathcal{C}'$ that is consistent with some k'-restricted truth table of the corrected implicit truth table from Step 1, 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 18. *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 $\mathcal{C}$ the induced subclass of k-dimensional functions on $\mathbb{F}_2^n$. Then Algorithm 7.5 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 2.*

Proof. Both the completeness and soundness follow straightforwardly from Theorems 16 and 17. The main thing to note in the completeness is that if $f = h(\chi_{\alpha_1}, \ldots, \chi_{\alpha_{k'}})$, 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.) □

Regarding the running time for Step 2, we can give some naive upper bounds. Using brute force search for the right $k' \leq k$ columns, we have a running time of $O(2^{k^2})T$, where T is the time required to check if a given k'-bit truth table is in $\mathcal{C}'$. Further, 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.

8 Lower bounds

In this section we show that the query complexities of our k-dimensionality test and s-sparsity test are tight up to polynomial factors. In fact, our lower bound Theorem 19 is somewhat stronger. First, though, let us review some known lower bounds.

Buhrman et al. [11] 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 $k \leq n-1$) must make $\Omega(2^{k/2})$ queries. In earlier work, Alon et al. [2] 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 19. *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 $f : \mathbb{F}_2^{Ck} \to \{-1, 1\}$:*

- *$\mathcal{D}_{\text{yes}}$: 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.*
- *$\mathcal{D}_{\text{no}}$: Choose a completely random function on $\mathbb{F}_2^{Ck}$ conditioned on it being $(1/2 - \tau)$-far from having $\mathbb{F}_2$-degree k.*

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

Note that $\mathcal{D}_{\text{yes}}$ is supported on k-dimensional functions and $\mathcal{D}_{\text{no}}$ is supported on functions far from even having $\mathbb{F}_2$-degree k. Using (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 19 for *deterministic* adaptive query algorithms. This is the "easy direction" of Yao's Principle: if $\mathcal{A}$ is a randomized distinguisher, we have

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

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 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 19, the following suffices: Fix a decision tree $\mathcal{T}$ with depth

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

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

$$\|\mathcal{L}_{\text{yes}} - \mathcal{L}_{\text{no}}\|_{TV} \leq 1/3, \tag{8}$$

and this establishes Theorem 19.

We will prove (8) via two lemmas.

Lemma 15. *Let $\mathcal{D}_{\mathrm{unif}}$ denote the uniform distribution on functions $\mathbb{F}_2^{Ck} \to \{-1,1\}$. Under $\mathcal{D}_{\mathrm{unif}}$, 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 [2]; we fill in the details of the volume argument here. Fix any function $g : \mathbb{F}_2^{Ck} \to \{-1,1\}$; when $f \sim \mathcal{D}_{\mathrm{unif}}$, 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 $2^{\binom{Ck}{k}}$, gives an overall probability of at most

$$2^{\binom{Ck}{k}} \cdot \exp(-2\tau^2 2^{Ck}) \leq \exp(k\ln(Ck) - 2\tau^2 2^{Ck}).$$

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

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

Lemma 16. $\|\mathcal{L}_{\mathrm{yes}} - \mathcal{L}_{\mathrm{unif}}\|_{TV} \leq 1/99$

Proof. This proof is similar to the one in [11], although we believe we are correcting a gap in that argument. Consider a draw $f \sim \mathcal{D}_{\mathrm{yes}}$; 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 $k \geq 2\log q + \log(100)$; hence Proposition 3(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,

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

The lemma now follows from Proposition 6 below. □

Proposition 6. *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 $\|P - U\|_{TV} \leq \delta/(1-\delta)$.*

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

Finally, to complete the proof of (8) and thus Theorem 19, simply note that Lemma 15 implies $\|\mathcal{D}_{\mathrm{no}} - \mathcal{D}_{\mathrm{unif}}\|_{TV} \leq 1/100$, hence $\|\mathcal{L}_{\mathrm{no}} - \mathcal{L}_{\mathrm{unif}}\|_{TV} \leq 1/100$; then use Lemma 16 and the triangle inequality: $1/100 + 1/99 \leq 1/3$.