IMPLICIT SENSING FOR FOURIER SPARSE BOOLEAN FUNCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Boolean functions constitute a fundamental object of study in machine learning and, more broadly, in theoretical computer science. Among their various complexity measures, Fourier sparsity, defined as the number of nonzero Fourier coefficients in a Boolean function's Fourier expansion, serves as a key indicator of structural simplicity. For over three decades, learning Boolean functions with sparse Fourier representations has been a central focus of computational learning theory. A notable achievement in this line of work is the development of learning algorithms whose complexity primarily depends on the Fourier sparsity parameter. However, these approaches generally assume prior knowledge of this parameter.

In this work, we address this gap in the literature on learning Fourier-sparse Boolean functions. Specifically, we study the problem of Fourier sparsity testing: given query access to a Boolean function $f : \mathbb{F}_2^n \to \{-1, +1\}$, decide whether it is s-Fourier sparse or far (under Hamming distance) from every such function.

Our contributions are twofold. On the algorithmic side, we design a new tester with query complexity $\widetilde{O}(s^4)$, independent of the ambient dimension. On the lower bound side, we prove that any tester requires at least $\Omega(s)$ queries. Both bounds improve upon the best known results of Gopalan et al. (SICOMP 2011), who presented a tester with query complexity $\widetilde{O}(s^{14})$ and a lower bound of $\Omega(\sqrt{s})$. For our upper bound, we introduce a refined notion of a sampler from the junta testing framework of Chakraborty et al. (ICALP 2011) and combine it with ℓ_1-minimization-based compressed sensing techniques to construct our tester. In the process, we develop a novel method for sampling the leaves of parity decision trees associated with Fourier-sparse Boolean functions. The lower bound is obtained via a reduction from communication complexity, crucially leveraging structural properties of the Fourier coefficients of a specific class of cryptographically hard functions.

1 INTRODUCTION

Boolean functions are fundamental in machine learning and theoretical computer science, as they naturally model decision rules in binary classification, logical circuits, and related computational processes. They have been extensively studied in learning theory (Kearns & Vazirani (1994)), complexity theory (O'Donnell (2014)), and cryptography (Carlet (2020)). Their structural simplicity has enabled researchers to design efficient algorithms across many domains of computer science. Among various structural measures, one that is particularly important in the literature on learning theory is Fourier sparsity, which counts the number of nonzero coefficients in the Fourier expansion over $\mathbb{F}_2^n$. Formally, for $f : \mathbb{F}_2^n \to \{-1, +1\}$,

$$f(x) = \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}(\alpha)\, \chi_\alpha(x), \qquad \chi_\alpha(x) = (-1)^{\langle x, \alpha \rangle},$$

with Fourier support $\mathrm{supp}(\widehat{f}) = \{\alpha : \widehat{f}(\alpha) \neq 0\}$. A function is s-Fourier sparse if $|\mathrm{supp}(\widehat{f})| \leq s$.

Because many natural classes of Boolean functions are Fourier sparse, learning such functions has long been a central theme in computational learning theory. Notable examples include hypergraph cut functions (Stobbe & Krause (2012)) and bounded-depth decision trees (Mansour (1994)). The

study of learning Fourier sparse Boolean functions dates back to the pioneering work of Kushilevitz & Mansour (1993), itself inspired by the Goldreich–Levin algorithm from Goldreich & Levin (1989). Over the past two decades, there has been renewed interest in exact reconstruction of Fourier sparse Boolean functions with complexity primarily dependent on the sparsity parameter. Two main algorithmic paradigms for this task are based on the Sparse Hadamard Transform e.g. Indyk et al. (2014); Hassanieh et al. (2012), and compressed sensing techniques e.g. Haviv & Regev (2017). A common limitation of both approaches, however, is the assumption that the sparsity level is known in advance. This creates a critical gap: before applying these methods, one must first ensure that the target function is indeed Fourier sparse.

Our work addresses this gap by developing efficient procedures for testing Fourier sparsity, thereby providing a natural preprocessing step for learning algorithms on Boolean functions. Specifically, we study the problem of testing Fourier sparsity of Boolean functions $f : \mathbb{F}_2^n \to \{-1, +1\}$ under black-box query access, where one can evaluate the function on inputs of choice without knowledge of its internal structure.

> **Problem Statement:** Given a parameter $\epsilon > 0$ and black-box query access to a Boolean function $f : \mathbb{F}_2^n \to \{-1, +1\}$, decide whether f is s-Fourier sparse or ϵ-far from every s-Fourier sparse Boolean function, where distance is measured under the Hamming sense:
>
> $$\mathrm{dist}_0(g, h) = \Pr_{x \in \mathbb{F}_2^n}[g(x) \neq h(x)].$$

The efficiency of a Fourier sparsity tester is measured by its query complexity, and such testers can serve as a crucial preprocessing step for learning algorithms on Fourier-sparse Boolean functions, such as fast sparse Hadamard transforms Scheibler et al. (2015); Hassanieh et al. (2012).

Related Work. A related version has been studied for real-valued functions $f : \mathbb{F}_2^n \to \mathbb{R}$, where distance is measured via the Euclidean norm,

$$\mathrm{dist}_2(f, g) = (\mathbb{E}_x[(f(x) - g(x))^2])^{1/2}.$$

Yaroslavtsev & Zhou (2020) showed that testing closeness to s-Fourier sparse functions under this metric can be done with $\tilde{O}(s)$ queries and established a matching $\Omega(\sqrt{s})$ lower bound.

However, one should note that while closeness in the Hamming sense implies closeness in the Euclidean sense, the converse may not be true. To illustrate, consider two scenarios. In the first, an exactly k-Fourier sparse Boolean function with precisely k nonzero Fourier coefficients; by Gopalan et al. (2011), all of these must be large in magnitude. In the second, a Boolean function with k large Fourier coefficients but also a tail of many small, nonzero coefficients with small ℓ_2 norm. A k-Fourier sparsity tester under the Euclidean distance would accept both types of functions, whereas a tester under the Hamming distance should accept only the first type. This makes testing Fourier sparsity in the Hamming sense significantly harder: one must certify not only the presence of k large Fourier coefficients but also the absence of a Fourier tail.

The problem of testing Fourier sparsity under Hamming distance was first studied by Gopalan et al. (2011), who gave a non-adaptive tester with query complexity $O(s^6 \log s/\epsilon^2 + s^{14} \log s)$ and proved a lower bound of $\Omega(\sqrt{s})$. Furthermore, as Fourier sparsity is an affine-invariant property, i.e., it remains unchanged under affine transformations of the domain, it can in principle be tested using regularity-based frameworks e.g. Kaufman & Sudan (2008); Bhattacharyya et al. (2015; 2013); Hatami & Lovett (2013); however, such approaches incur impractical tower-type query complexities.

Our Contributions. We close the gap between existing upper and lower bounds for Fourier sparsity testing. Our main results are:

- **Upper bound:** A non-adaptive algorithm with query complexity $\widetilde{O}(s^4)$, improving over the previous $\widetilde{O}(s^{14})$ bound from Gopalan et al. (2011).
- **Lower bound:** A new, quadratically stronger lower bound of $\Omega(s)$, improving over the previous best $\Omega(\sqrt{s})$ bound from Gopalan et al. (2011).

More formally, we prove the following theorems.

Theorem 1.1. ***(Upper bound)*** *Let $s > 0$, $\epsilon > 0$, and let $f : \mathbb{F}_2^n \to \{-1, +1\}$ be an unknown Boolean function accessible via queries to its truth-table. There exists a non-adaptive property testing algorithm that decides whether f is s-Fourier sparse or ϵ-far (under Hamming distance) from any such function with success probability at least $2/3$, using at most $\widetilde{O}\big(\max\{s^2, 1/\epsilon\} \cdot s^2\big)$ queries*[1].

Theorem 1.2. ***(Lower bound)*** *Any adaptive property testing algorithm that decides whether a Boolean function is s-Fourier sparse or $(1/4)$-far (under Hamming distance) from every s-Fourier sparse function, with success probability at least $2/3$, must make at least $\Omega(s)$ queries.*

For the upper bound (Theorem 1.1), we design a tester that refines the notion of a sampler from the junta testing framework (Chakraborty et al. (2011)) and combines it with ℓ_1-minimization based compressed sensing framework. A key ingredient of our algorithm is a new sampling method for parity decision trees, which arises naturally in the analysis of Boolean functions.

For the lower bound (Theorem 1.2), we achieve a quadratic improvement by reducing Fourier sparsity testing to a certain linear-algebraic problem in communication complexity. Exploiting structural properties of the Maiorana–McFarland family (McFarland (1973)), we show that any tester, adaptive or non-adaptive, must make at least $\Omega(s)$ queries.

Our techniques are of independent interest and have potential applications in learning theory, property testing, and other algorithmic questions related to harmonic analysis.

2 PRELIMINARIES

We use the following notations and background results in the rest of the paper.

- By Boolean function we mean functions of the form $f : \mathbb{F}_2^n \to \{-1, +1\}$.
- Given $f : \mathbb{F}_2^n \to \mathbb{R}$, the expected value $\mathbb{E}_x[f]$ is defined as $\mathbb{E}_x[f] := 2^{-n} \sum_{x \in \mathbb{F}_2^n} f(x)$.
- For $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$ in $\mathbb{F}_2^n$, their inner product is $\langle \alpha, \beta \rangle := \sum_{i=1}^n \alpha_i \beta_i$.
- For $f, g : \mathbb{F}_2^n \to \mathbb{R}$, their inner product is $\langle f, g \rangle = 2^{-n} \sum_{x \in \mathbb{F}_2^n} f(x) g(x)$.
- Given $\alpha \in \mathbb{F}_2^n$, character function $\chi_\alpha : \mathbb{F}_2^n \to \{-1, +1\}$ corresponding to α is defined as $\chi_\alpha(x) := (-1)^{\langle \alpha, x \rangle}$. Note that the character functions $\{\chi_\alpha : \alpha \in \mathbb{F}_2^n\}$ are orthogonal, that is,
$$\langle \chi_\alpha, \chi_\beta \rangle = \begin{cases} 0 & \text{if } \alpha \neq \beta \\ 1 & \text{if } \alpha = \beta, \end{cases}$$
and character functions also forms a basis for all real-valued functions on $\mathbb{F}_2^n$.
- Fourier transformation of a function $f : \mathbb{F}_2^n \to \mathbb{R}$ is defined as $\widehat{f}(\alpha) := \langle f, \chi_\alpha \rangle = \mathbb{E}_x[f(x)\chi_\alpha(x)]$, for all $\alpha \in \mathbb{F}_2^n$. By, Fourier inversion formula we have $f(x) = \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}(\alpha)\chi_\alpha(x)$, for all $x \in \mathbb{F}_2^n$.
- For a function $f : \mathbb{F}_2^n \to \mathbb{R}$, Parseval's identity says that $\langle f, f \rangle = \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}(\alpha)^2$. Additionally, if f is also a Boolean functions then $\langle f, f \rangle = 1$.
- For any two functions $g, h : \mathbb{F}_2^n \to \mathbb{R}$, from Plancherel theorem we have $\langle g, h \rangle = \sum_{\alpha \in \mathbb{F}_2^n} \widehat{g}(\alpha)\widehat{h}(\alpha)$.
- Given Boolean functions $f, g : \mathbb{F}_2^n \to \{-1, +1\}$, the Hamming distance ($\ell_0$-distance) is defined as $\delta(f, g)$ is $\delta(f, g) := \frac{d(f,g)}{2^n}$. where $d(f, g) := |\{x \in \mathbb{F}_2^n : f(x) \neq g(x)\}|$.
- For $A \in \mathbb{F}_2^{m \times n}$, $\mathrm{rank}(A)$ denotes its rank over $\mathbb{F}_2$.

[1]Here, $\widetilde{O}(\cdot)$ hides factors polynomial in $\log s$.

3 IMPROVED UPPER BOUND FOR TESTING FOURIER SPARSITY

3.1 PROOF OUTLINE OF THEOREM 1.1

The approach of employing learning algorithms for property testing, introduced by Goldreich et al. (1998), is founded on the principle that any proper learning algorithm for a function class $\mathcal{C}$ can be converted into a property tester for $\mathcal{C}$. Nevertheless, while proper learning for most Boolean function classes requires at least $\Omega(\log n)$ queries, property testing typically targets sublogarithmic or even constant query complexity, independent of the ambient dimension n. A significant advancement in this context was made by Diakonikolas et al. (2007), who introduced the paradigm of testing via implicit learning. The key insight underlying this framework is that many natural classes of Boolean functions, such as monotone DNFs, decision lists, decision trees, branching programs, Boolean formulas, sparse polynomials over $\mathbb{F}_2$, and Boolean circuits admit succinct representations and are well-approximated by junta functions. For instance, every s-term DNF is ϵ-close to another s-term DNF depending on only $O(\log s + \log(1/\epsilon))$ variables. The testing via implicit learning framework leverages this structural approximation. Building on this idea, Chakraborty et al. (2011) proposed a query-efficient sample extractor that yields improved and in certain cases, optimal bounds for testing membership in many of the subclasses of junta functions.

Returning to the problem of testing Fourier sparsity, we first recall that Gopalan et al. (2011) designed their tester for fourier sparsity testing by exploiting certain locally testable structural features of Fourier-sparse functions, namely, the granularity of their nonzero Fourier coefficients, and subsequently identifying these coefficients via a hashing-based technique. In contrast, our approach is based on the paradigm of testing via implicit learning. However, a key challenge for us is that existing techniques for testing membership in subclasses of junta functions are insufficient for handling Fourier-sparse functions. This is because the class of Fourier-sparse functions is strictly more general than the class of junta functions. For example, a linear function that depends on $O(n)$ variables is 1-Fourier sparse, yet it is not a junta function, since its output depends on a linear number of variables. Thus, testing Fourier sparsity following the principle of testing via implicit learning, necessitates the development of new techniques that go beyond the existing junta testing framework.

At the core of our technique lies an exact learning procedure for s-Fourier sparse Boolean functions (see Algorithm 2), which leverages ℓ_1-minimization techniques from compressed sensing (see Chapter 4 of Moitra (2018)). This approach critically relies on a recent result by Haviv & Regev (2017), which establishes that subsampled Fourier-Hadamard matrices satisfy the Restricted Isometry Property (RIP). A key implication of this result is that, given $O(s \cdot \log^2 s \cdot n)$ uniformly random samples from an s-Fourier sparse Boolean function, it is possible to exactly recover the function via an appropriate ℓ_1-minimization procedure. However, a direct application of this method to our setting proves inadequate, as the above-mentioned sample complexity continues to depend on the ambient dimension n. To address this limitation, we leverage a structural result by Sanyal (2019), which establishes that the dimension of the linear span of the nonzero Fourier coefficients of an s-Fourier sparse function is at most $O(\sqrt{s})$. This implies that any s-Fourier sparse function $f : \mathbb{F}_2^n \to \{-1, +1\}$ can be expressed as a function

$$f^* : \mathbb{F}_2^r \to \{-1, +1\} = f \circ L,$$

where $L : \mathbb{F}_2^n \to \mathbb{F}_2^r$ is certain linear transformation and $r = O(\sqrt{s})$. It follows from the RIP result of Haviv & Regev (2017) that the above mentioned reconstruction technique would require only $O(s^{3/2} \cdot \log^2 s)$ random samples of $f^* : \mathbb{F}_2^r \to \{-1, +1\}$ to recover it exactly.

Motivated by this insight, our approach focuses on reconstructing the composition $f \circ U \circ R$, where $U : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is an unknown linear transformation, and $R : \mathbb{F}_2^n \to \mathbb{F}_2^r$ is a known linear map. The key machinery for this part of our proof is Algorithm 3, a subroutine that generates random samples from $f \circ U \circ R$ using only a small number of queries. Our readers may note that this can be viewed as a generalization of the sampler designed by Chakraborty et al. (2011) within the junta subclass testing framework discussed earlier. At the core of this algorithm is a local list-correction procedure for the Hadamard code, inspired by the tester of Gopalan et al. (2011) for testing induced subclasses of low dimensional functions. Though our adaptation differs substantially in both construction and analysis, yielding a significant improvement in query complexity, specifically, a polynomial improvement corresponding to a factor of six in the exponent.

3.2 PROOF OF THEOREM 1.1

We begin by introducing the notion of coset sampling, which plays a key role in generating samples for the exact learning machinery, albeit in a low-dimensional space. Let $\mathcal{O}_H^f$ denote the sampling procedure with respect to a subspace H, given query access to the function f, as described in Algorithm 1. Throughout this work, we refer to the samples produced by this procedure as coset samples of f with respect to the subspace H.

Algorithm 1: SubspaceExplicitCosetSampler

Input: Subspace H with basis $B := \{\beta_1, \beta_2, \cdots \beta_r\}$, and query access to f
Output: An uniform random sample of the function f restricted to a random coset of $H^\perp$

1. Sample b uniformly at random from $\mathbb{F}_2^r$ and define the coset $C(b)$
$$C(b) = \{\alpha \in \mathbb{F}_2^n \mid \langle \alpha, \beta_i \rangle = b_i\}.$$
2. Select a uniformly random element p from $C(b)$.
3. Return the pair $(b, f(p))$.

Let $\mathcal{S}_f$ denote the subspace spanned by the vectors corresponding to the nonzero Fourier coefficients of f. Observe that the coset samples generated by $\mathcal{O}_{\mathcal{S}_f}^f$ can be interpreted as uniform samples drawn from the set of leaves of $\mathcal{T}_f$, the non-adaptive parity decision tree representation of f. For a parameter $\theta > 0$, the θ-restricted Fourier span $\mathcal{S}_f(\theta)$ is defined as the subspace spanned by those vectors $\alpha \in \mathbb{F}_2^n$ for which $|\widehat{f}(\alpha)| \geq \theta$. In this work, we build a query-efficient implementation of $\mathcal{O}_{\mathcal{S}_g(\theta)}^g$, given query access to f and the threshold $\theta > 0$, where $g = f \circ A$ for some unknown invertible linear transformation $A : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

Lemma 3.1. *There exists an algorithm **SubspaceImplicitCosetSampler** (Algorithm 3) that given query access to a Boolean function $f : \mathbb{F}_2^n \to \{-1, +1\}$, a threshold $\theta > 0$, and a parameter $\lambda \in \mathbb{N}$, generates a set of λ-many uniform coset samples ξ, with respect to the subspace $\mathcal{S}_{f\circ A}(\theta)$, for a function $f \circ A$ where A is some non-singular (and possibly unknown) linear transformation of $\mathbb{F}_2^n$. The query complexity of the algorithm is $\widetilde{O}\left(\frac{1}{\theta^4} + \max\left\{\frac{1}{\theta^2}, \lambda\right\} \cdot \frac{1}{\theta^2}\right)$, and the failure probability of the algorithm is at most $\frac{1}{10}$.*

While the proof of Lemma 3.1 is deferred to the latter part of this section, we assume its validity for now and proceed to demonstrate how it can be used to establish Theorem 1.1.

Proof of Theorem 1.1. In Algorithm 2, the subroutine **SubspaceImplicitCosetSampler** (Algorithm 3) is invoked with parameters θ and λ. By Lemma 3.1, this procedure produces λ uniformly random coset samples (as described in Algorithm 1) with respect to the subspace $\mathcal{S}_{f\circ A}(\theta)$. Using these coset samples, we define a lower-dimensional Boolean function $f^* : \mathbb{F}_2^r \to \{-1, 1\}$, where $f^* = f \circ T \circ A$, and $T : \mathbb{F}_2^n \to \mathbb{F}_2^r$ is a linear transformation mapping the standard basis vectors e_i^n of $\mathbb{F}_2^n$ to the corresponding standard basis vectors e_i^r of $\mathbb{F}_2^r$, for $i \in \{1, 2, \ldots, r\}$. The transformation T also maps each of the 2^r cosets of $\mathcal{S}_{f\circ A}(\theta)^\perp$ (the orthogonal subspace of $\mathcal{S}_{f\circ A}(\theta)^\perp$) to unique elements of $\mathbb{F}_2^r$.

The next step is to determine whether the function f^* is s-Fourier sparse. It is kind of folklore (see Chapter 4 of Moitra (2018)) that for any Boolean function $h : \mathbb{F}_2^r \to \{-1, +1\}$ with at most s nonzero Fourier coefficients, if there exists an $m \times 2^r$ subsampled Walsh–Hadamard matrix M satisfying the Restricted Isometry Property (RIP) of order k with constant $\delta_k < \frac{1}{3}$ (and $\delta_{2k} + \delta_{3k} < 1$), then the Fourier spectrum of h can be recovered exactly with high probability. This recovery is achieved by solving the following ℓ_1-minimization problem using m uniformly random labeled examples $(x, y = h(x))$:
$$\widehat{h} = \arg\min \|\widehat{h}\|_1 \quad \text{subject to} \quad M\widehat{h} = y.$$

A bound on m determines the number of required random examples and, consequently, the number of coset samples needed from $f \circ A$. To establish this bound, we recall the result of Sanyal (2019),

which states that for any Fourier sparse function f_A, the dimension of the subspace spanned by its nonzero Fourier coefficients is at most $O(\sqrt{s})$. Following result of Haviv & Regev (2017), says that $m = O\left(s^{3/2} \cdot \text{poly}(\log s)\right)$ random samples suffice for exact recovery of f^*.

Lemma 3.2 (Haviv & Regev (2017)). *Let $M \in \mathbb{C}^{N\times N}$ be a unitary matrix satisfying $\|M\|_\infty \leq O(1/\sqrt{N})$, and let $\delta > 0$ be sufficiently small. Construct a measurement matrix $B \in \mathbb{C}^{q\times N}$ by selecting q rows uniformly and independently from M, each scaled by $\sqrt{N/q}$. If*

$$q = O\left(\log^2(1/\delta) \cdot \delta^{-2} \cdot k \cdot \log^2(k/\delta) \cdot \log N\right),$$

then, with probability at least $1 - 2^{-\Omega(\log N \cdot \log(k/\delta))}$, the matrix B satisfies the Restricted Isometry Property of order k with isometry constant δ.

Algorithm 2: EffcientFourierSparsityTester

Input: Fourier sparsity s, and query access to f,
Output: Whether f has fourier sparsity s or ε far from every such function
Initialization: $\theta = \frac{1}{4s}$, $\lambda = \max\left\{\frac{1}{\varepsilon}, Cs^2 \cdot \text{poly}(\log s)\right\}$ where C is some large constant

1. $(\zeta, r) \leftarrow$ **SubspaceImplicitCosetSampler**(θ, λ)
2. Initialize Φ such that for each $(x, y) \in \zeta$ and for each $z \in \mathbb{F}_2^r$, set
$$\Phi[x][z] = \frac{2^{\frac{r}{2}}}{\lambda^{\frac{1}{2}}} \cdot (-1)^{x\cdot z}.$$
3. Solve the following optimization problem:
$$\min \sum \|\hat{h}\| \quad \text{subject to} \quad \langle \Phi[x][*] \cdot \hat{h}\rangle = y, \quad \text{for all } (x, y) \in \zeta.$$
4. If $\hat{h}$ corresponds to the Fourier spectrum of some s-Fourier sparse Boolean function, accept; otherwise, reject.

We now show that if the input function f is s-Fourier sparse, then the algorithm always accepts.

Lemma 3.3 (Completeness). *If the function f is s-Fourier sparse, then Algorithm 2 accepts.*

Proof. We begin by recalling a result from Gopalan et al. (2011), which states that if a function f is s-Fourier sparse, then all of its nonzero Fourier coefficients are integer multiples of $\frac{1}{2^{\lceil \log s\rceil}}$. Moreover, Fourier sparsity is invariant under nonsingular linear transformations. When the subroutine **SubspaceImplicitCosetSampler** (Algorithm 3) is invoked with the threshold parameter $\theta = \frac{1}{4s}$, Lemma 3.1 guarantees that it indeed discovers the Fourier span of $f \circ A$, for some unknown invertible linear transformation A. Additionally, the algorithm produces λ uniform coset samples, where

$$\lambda = \max\left\{\frac{1}{\varepsilon},\ O\left(s^2 \cdot \text{poly}(\log s)\right)\right\}.$$

By the guarantees of the sparse recovery algorithm discussed previously, these samples suffice to exactly reconstruct the Fourier spectrum of the projected function $f^* = f \circ T \circ A$. Hence, the algorithm accepts. □

Lemma 3.4 (Soundness). *If the function f is ε-far from every s-Fourier sparse Boolean function, then Algorithm 2 rejects.*

Proof. We argue by contraposition. Suppose Algorithm 2 accepts. Then there exists a function $\tilde{f}$, obtained via composition with appropriate linear transformations, such that it agrees with f on all λ uniform samples collected during Step 4 of Algorithm 3, and moreover $\tilde{f}$ is s-Fourier sparse. Let us

define the set **GOOD** as the set of inputs where $\tilde{f}$ and f agree, and **BAD** as the complement. The probability that all λ samples fall within the **GOOD** set is

$$\left(1 - \frac{|\mathbf{BAD}|}{2^n}\right)^{\lambda}.$$

If this probability is at least a constant, say $\Omega(1)$, then by a standard union bound and Markov's inequality, we must have

$$\frac{|\mathbf{BAD}|}{2^n} \leq \frac{1}{O(\lambda)},$$

which implies that f is $\frac{1}{O(\lambda)}$-close to some s-Fourier sparse function $\tilde{f}$. □

We now analyze the query complexity of Algorithm 2. The algorithm initializes with parameters $\theta = O(1/s)$ and

$$\lambda = \max\left\{\frac{1}{\varepsilon},\ \widetilde{O}(s^2)\right\}.$$

The only queries made are via calls to Algorithm 3. According to Lemma 3.1, the total number of queries made within that algorithm is bounded by

$$\widetilde{O}\left(\frac{1}{\theta^4} + \max\left\{\frac{1}{\theta^2},\ \lambda\right\} \cdot \frac{1}{\theta^2}\right)$$

Substituting $\theta = O(1/s)$, we obtain the final query complexity:

$$\widetilde{O}\left(s^4 + \max\left\{s^2,\ \frac{1}{\varepsilon}\right\} \cdot s^2\right).$$

One may verify that Algorithm 2 fails with probability at most $\frac{1}{10} + o(1) < \frac{1}{3}$. □

3.3 Proof of Lemma 3.1

We prove this theorem by showing that **SubspaceImplicitCosetSampler** (Algorithm 3) satisfies the theoretical guarantees of Lemma 3.1. Here, we provide only a high-level overview of the algorithm; the full proof is presented in the appendix.

At a high level, **SubspaceImplicitCosetSampler** is inspired by the implicit learning framework introduced in Gopalan et al. (2011) for designing their algorithm for testing induced subclasses of low-dimensional functions. However, our method achieves a significant improvement in query complexity, owing to a refined analysis of the underlying coset hashing process. In particular, we revisit coset decomposition and establish new concentration bounds (Lemmas A.3 and A.5) for the ℓ_1- and ℓ_2-norms of the projected Fourier spectrum. These results show that, within any coset, the norm of the Fourier projection is dominated by the contribution of heavy coefficients. Our analysis is inspired by techniques for heavy-hitter detection in the streaming literature: ℓ_2 concentration plays a role analogous to the Count–Min Sketch, while ℓ_1 concentration corresponds to the Count Sketch. Below, we provide a brief sketch of the overall algorithm.

- **Detecting Heavy Fourier Coefficients.** We begin by projecting the Fourier spectrum of f onto cosets of a random subspace H of dimension $\log O(1/\theta^4)$. This induces a pairwise-independent hashing of the Fourier coefficients of f into $O(1/\theta^4)$ cosets of H. With high probability, all Fourier coefficients of magnitude at least θ fall into distinct cosets (Lemma A.1). A key ingredient is Lemma A.3, which shows that the ℓ_2-norm of the Fourier projection within each coset is concentrated around its dominant Fourier coefficient. Given sufficiently accurate ℓ_2 estimates for individual cosets, this allows us to identify all cosets containing a heavy Fourier coefficient.
- **Evaluating Heavy Fourier Coefficients.** Once the heavy cosets are identified, we proceed to evaluate the corresponding heavy Fourier characters, without knowing them explcitly. This step relies on Lemma A.5, which shows that the ℓ_1-norm of the projected Fourier spectrum in each coset is dominated by its heaviest coefficient. Consequently, for any

$x \in \mathbb{F}_2^n$, the projection of the Fourier coefficients can be well approximated by the evaluation of the heaviest coefficient at x. However, the probability that this approximation is sufficiently accurate for a fixed x is only constant. Since the algorithm requires many such evaluations, we employ an amplification step in the spirit of the original Goldreich–Levin algorithm Goldreich & Levin (1989).

- **Generating Coset Samples.** Given sufficiently many evaluations of the heavy Fourier characters for enough points x, one can recover a linear basis among the heavy Fourier coefficients, thereby identifying the coefficients up to a linear transformation. In our work, we show that this is sufficient for our purposes.

4 Improved lower bound for testing Fourier sparsity

We begin by defining a class of Boolean functions known as the Maiorana–McFarland functions. These functions have a long history in theoretical computer science, including applications in circuit lower bounds and studies of Boolean function structure, e.g., see Paul (1977); Blum (1984); Nisan & Szegedy (1992); Sanyal (2019). They are also widely used in symmetric-key cryptography, particularly in stream cipher design, where they provide desirable Fourier and autocorrelation properties, see Sarkar & Maitra (2000).

Definition 4.1. *Given positive integers n and r with $r \leq n$, the family of Maiorana–McFarland functions, denoted $\mathrm{MM}_{r,n}$ McFarland (1973), consists of functions $f : \mathbb{F}_2^n \to \{-1, +1\}$ of the form*

$$g(x, y) = (-1)^{\langle x, \varphi(y) \rangle}, \quad (x, y) \in \mathbb{F}_2^r \times \mathbb{F}_2^{n-r},$$

where $\varphi : \mathbb{F}_2^{n-r} \to \mathbb{F}_2^r$ is an arbitrary mapping.

A key property of these functions is that, when composed with linear transformations, their Fourier sparsity is governed by the rank of the transformation.

Lemma 4.2. *Let $n = r + \log r$, let $\varphi : \mathbb{F}_2^{n-r} \to \mathbb{F}_2^r$ have r linearly independent outputs, and let $L \in \mathbb{F}_2^{r \times r}$ be linear. Define*

$$g_L(x, y) = (-1)^{\langle Lx, \varphi(y) \rangle}.$$

Then the Fourier sparsity of g_L satisfies

$$|\mathrm{supp}(\widehat{g_L})| \leq \mathrm{rank}(L) \cdot r.$$

Proof. For $(u, v) \in \mathbb{F}_2^r \times \mathbb{F}_2^{n-r}$, the Fourier coefficient is

$$\widehat{g_L}(u, v) = \mathbb{E}_{x,y} \left[(-1)^{\langle Lx, \varphi(y) \rangle + \langle u, x \rangle + \langle v, y \rangle} \right] = \mathbb{E}_y \left[(-1)^{\langle v, y \rangle} \mathbb{E}_x (-1)^{\langle L^T \varphi(y) + u, x \rangle} \right].$$

The inner expectation is 1 if $u = L^T \varphi(y)$ and 0 otherwise. Hence, (u, v) can be in the Fourier support only if $u \in \mathrm{Im}(L^T \circ \varphi)$, which has at most $\mathrm{rank}(L)$ distinct values. For each u, there are r choices for v, yielding the claimed bound. □

Randomized communication complexity studies the minimum number of bits two or more parties must exchange to compute a function using shared or private randomness, while producing a correct output with high probability. The Approximate Matrix Rank Problem is defined as follows: Alice and Bob hold $A, B \in \mathbb{F}_2^{r \times r}$ and are promised that $\mathrm{rank}(A + B) \in \{r, r/4\}$. Their goal is to determine which case holds using minimal communication and public randomness. We denote the randomized communication complexity (with public coins) with error at most $1/3$ by $R^*_{1/3}(\mathrm{RANK}_{r,r/4})$.

Theorem 4.3 (Sherstov & Storozhenko (2024))**.**

$$R^*_{1/3}(\mathrm{RANK}_{r,r/4}) = \Omega(r^2).$$

Equipped with these definitions, we now establish a Fourier sparsity testing lower bound that is quadratically stronger than the previously known result Gopalan et al. (2011).

4.1 Proof of Theorem 1.2

We prove the theorem via a reduction from the Approximate Matrix Rank problem. Alice has A, Bob has B, and $C = A + B$ satisfies $\mathrm{rank}(C) \in \{r, r/4\}$. Alice constructs f_A, Bob constructs f_B, and together they define $f = f_C = f_{A+B}$. By Lemma 4.2, if $\mathrm{rank}(C) = r$, then $|\mathrm{supp}(\widehat{f})| = r^2$, and if $\mathrm{rank}(C) = r/4$, then $|\mathrm{supp}(\widehat{f})| \leq r^2/4$. The following lemma quantifies the distance between these two cases.

Lemma 4.4. *If $\mathrm{rank}(C) = r$, then f is at least $(1/4)$-far from any function with Fourier sparsity at most $r^2/4$.*

Proof. Let $h : \mathbb{F}_2^n \to \{-1, +1\}$ be a $\frac{r^2}{4}$-Fourier sparse function. Then,

$$\begin{aligned}\Pr_x[h(x) \neq f(x)] &= \Pr_x[h(x) \neq f(x)] \qquad (1)\\ &= \frac{1}{2} + \frac{1}{2}\mathbb{E}_x[h(x)f(x)]\\ &= \frac{1}{2} + \frac{1}{2}\sum_{\alpha \in \mathbb{F}_2^n} \hat{h}(\alpha)\hat{f}(\alpha)\\ &= \frac{1}{2} + \frac{1}{2}\sum_{\alpha \in \mathrm{supp}(h)} \hat{h}(\alpha)\hat{f}(\alpha). \qquad (2)\end{aligned}$$

Recall that for Boolean function $f : \mathbb{F}_2^n \to \{-1, +1\}$, $\mathrm{supp}(\widehat{f}) := \left\{\alpha \in \mathbb{F}_2^n : \widehat{f}(\alpha) \neq 0\right\}$. Now applying Cauchy-Schwarz inequality, we get

$$\begin{aligned}\left|\sum_{\alpha \in \mathrm{supp}(h)} \hat{h}(\alpha)\hat{f}(\alpha)\right| &\leq \sqrt{\sum_{\alpha \in \mathrm{supp}(\widehat{h})} \hat{h}^2(\alpha) \cdot \sum_{\alpha \in \mathrm{supp}(\widehat{h})} \hat{f}^2(\alpha)}\\ &= \sqrt{\sum_{\alpha \in \mathrm{supp}(\widehat{h})} \hat{f}^2(\alpha)} \qquad (3)\end{aligned}$$

Note that h is a $\frac{r^2}{4}$-Fourier sparse Boolean function, that is, $|\mathrm{supp}(\widehat{h})| \leq \frac{r^2}{4}$. Observe that, by the construction of function f (see Lemma 4.2), the absolute values of any two non-zero Fourier coefficients are equal and the Fourier support $\mathrm{supp}(\widehat{f}) = r^2$. Using the fact that $\sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}(\alpha)^2 = 1$ (Parseval's identity), we get

$$\sqrt{\sum_{\alpha \in \mathrm{supp}(\widehat{h})} \hat{f}^2(\alpha)} \leq \frac{1}{2} \qquad (4)$$

Finally, plugging the bound from equation 4 into equation 1, we conclude that

$$\Pr_x[h(x) \neq f(x)] \geq \frac{1}{2} + \frac{1}{2} \cdot \left(-\frac{1}{2}\right) = \frac{1}{4}.$$

□

Suppose there exists a tester $\mathbb{T}$ for Fourier sparsity with parameter s and query complexity $q(s, 1/4)$. Here, $s = r^2/4$. To simulate a query x, Alice computes $f_A(x)$, Bob computes $f_B(x)$, and they exchange one bit each. Since $f_C(x) = f_A(x)f_B(x)$, each query costs two bits of communication. By Theorem 4.3, solving the rank problem requires $\Omega(r^2)$ bits. Therefore, $q(s, 1/4) = \Omega(r^2) = \Omega(s)$, establishing the claimed lower bound.

Disclaimer on LLM Usage

A large language model (LLM) was used solely for polishing the writing. No part of the technical contributions, ideation, or literature survey was generated using the LLM. All technical content and research results are entirely the authors' own.

REFERENCES

Arnab Bhattacharyya, Eldar Fischer, Hamed Hatami, Pooya Hatami, and Shachar Lovett. Every locally characterized affine-invariant property is testable. In *Proceedings of the 45th ACM Symposium on Theory of Computing, STOC*, pp. 429 – 436, 2013.

Arnab Bhattacharyya, Elena Grigorescu, and Asaf Shapira. A Unified Framework for Testing Linear-Invariant Properties. *Random Struct. Algorithms*, 46(2):232–260, 2015.

N. Blum. A Boolean function requiring $3n$ network size. *Theoretical Computer Science*, 28:337–345, 1984.

Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, 2020.

Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Efficient sample extractors for juntas with applications. In Luca Aceto, Monika Henzinger, and Jiří Sgall (eds.), *Automata, Languages and Programming*, pp. 545–556, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

Ilias Diakonikolas, Homin Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco Servedio, and Andrew Wan. Testing for concise representations. In *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 549–558, 2007.

Oded Goldreich and Leonid A Levin. A Hard-Core Predicate for all One-Way Functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pp. 25 – 32, 1989.

Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property Testing and its Connection to Learning and Approximation. *Journal of the ACM*, 45:653–750, 1998.

Parikshit Gopalan, Ryan O'Donnell, Rocco A Servedio, Amir Shpilka, and Karl Wimmer. Testing Fourier Dimensionality and Sparsity. *SIAM Journal on Computing*, 40(4):1075 – 1100, 2011.

Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Nearly Optimal Sparse Fourier Transform. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 563–578, 2012.

Hamed Hatami and Shachar Lovett. Estimating the Distance from Testable Affine-Invariant Properties. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 237 – 242, 2013.

Ishay Haviv and Oded Regev. *The Restricted Isometry Property of Subsampled Fourier Matrices*, pp. 163–179. Springer International Publishing, Cham, 2017.

Piotr Indyk, Michael Kapralov, and Eric Price. (Nearly) Sample-Optimal Sparse Fourier Transform. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pp. 480–499, 2014.

Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2008.

Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

Eyal Kushilevitz and Yishay Mansour. Learning Decision Trees Using the Fourier Spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

Yishay Mansour. Learning Boolean Functions *via* the Fourier Transform. *Theoretical Advances in Neural Computation and Learning*, pp. 391–424, 1994.

R. L. McFarland. A family of noncyclic difference sets. *Journal of Combinatorial Theory, Series A*, 15:1–10, 1973.

Ankur Moitra. *Algorithmic Aspects of Machine Learning*. Cambridge University Press, 2018.

N. Nisan and M. Szegedy. On the degree of boolean functions as real polynomials. In *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing (STOC'92)*, pp. 462–467, 1992. [Online].

Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

W. J. Paul. A $2.5n$-Lower Bound on the Combinational Complexity of Boolean Functions. *SIAM Journal on Computing*, 6(3):427–443, 1977.

Swagato Sanyal. Fourier sparsity and dimension. *Theory of Computing*, 15(11):1–13, 2019.

Palash Sarkar and Subhamoy Maitra. Construction of nonlinear boolean functions with important cryptographic properties. In Bart Preneel (ed.), *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pp. 485–506. Springer, 2000.

Robin Scheibler, Saeid Haghighatshoar, and Martin Vetterli. A Fast Hadamard Transform for Signals With Sublinear Sparsity in the Transform Domain. *IEEE Transactions on Information Theory*, 61 (4):2115–2132, 2015.

Alexander A. Sherstov and Andrey A. Storozhenko. The communication complexity of approximating matrix rank. In *Proceedings of the IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 433 – 462, 2024.

Peter Stobbe and Andreas Krause. Learning Fourier Sparse Set Functions. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 22 of *JMLR Proceedings*, pp. 1125–1133. JMLR.org, 2012.

Grigory Yaroslavtsev and Samson Zhou. Fast Fourier Sparsity Testing. In *Proceedings of the 2020 Symposium on Simplicity in Algorithms (SOSA)*, pp. 57 – 68, 2020.

A Analysis of SubspaceImplicitCosetSampler (Proof of Lemma 3.1)

Proof. We first give the details of Algorithm 3, and subsequently give its theoretical guarantees. Table 1 gives the notations used in Algorithm 3 and its analysis.

Notation	Description
$\mathscr{C}$	A random coset decomposition of $\mathbb{F}_2^n$
C	Individual cosets of $\mathscr{C}$
$\mathcal{C}$	Set of cosets of $\mathscr{C}$ that contains a *heavy* Fourier coefficient of f
$\alpha^*(C)$	The Fourier coefficient in C with the highest absolute value
$\mathcal{W}_C$	$\sum_{\beta \in C} \widehat{f}^2(\beta)$, total Fourier weight of coset C
$\mathcal{W}_C^*$	$\widehat{f}^2(\alpha^*(C))$, weight of the heaviest Fourier coefficient in C
$\mathcal{P}_C(z)$	$\sum_{\beta \in C} \widehat{f}(\beta)\chi_\beta(z)$, projection of $f(z)$ into coset C
$\mathcal{P}_C^*(z)$	$\widehat{f}(\alpha^*(C))\chi_{\alpha^*(C)}(z)$, projection of $f(z)$ onto the heaviest Fourier coefficient of C
$\chi_C(x)$	Alternative notation for $\chi_{\alpha^*(C)}(x)$

Table 1: Notations for Algorithm 3

We now proceed with the details of Algorithm 3. As a first step, we project the Fourier spectrum of f onto $\mathscr{C}$, a randomly permuted coset structure of a randomly chosen subspace of codimension

$$t = \left\lceil \log \frac{100}{\eta^4} \right\rceil.$$

This coset structure is defined as follows. First we select t vectors $\beta_1, \ldots, \beta_t$ independently and uniformly at random from $\mathbb{F}_2^n$ and define the subspace

$$H = \operatorname{span}\{\beta_1, \ldots, \beta_t\}^\perp.$$

This subspace has codimension t and consists of all vectors in $\mathbb{F}_2^n$ that are orthogonal to $\beta_1, \ldots, \beta_t$. For each $b \in \mathbb{F}_2^t$, we define the coset

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

Further, to introduce additional randomness, we select a random shift $z \in \mathbb{F}_2^t$ and relabel each coset $D(b)$ as $D(b+z)$, thereby obtaining a randomly permuted coset structure. Gopalan et al. Gopalan et al. (2011) showed that such a coset decomposition process behaves like a pairwise independent hashing scheme over elements of $\mathbb{F}_2^n$ when considering their placement in cosets. We present their results below, with slight changes in notation.

Lemma A.1 (Proposition 3, Gopalan et al. (2011))**.** *Let (H, C) be a random permuted t-dimensional coset structure, where $t \geq 2\log s + \log 100$. Then*

- *If α and α' are distinct elements of $\mathbb{F}_2^n$, the probability that they belong to the same bucket is 2^{-t}.*
- *For any set $S \subseteq \mathbb{F}_2^n$ with $|S| \leq s+1$, then, except with probability at most δ, all elements of S are assigned to distinct buckets.*

In **Step 2**, we estimate the Fourier weight of each coset in $\mathscr{C}$ and discard those whose weight estimates fall below a specified threshold. Given the parameter settings in the algorithm, we demonstrate that at the end of **Step 3** of Algorithm 3, we successfully identify a set of cosets $\mathcal{C} \subseteq \mathscr{C}$ that collectively contain all heavy Fourier coefficients while ensuring that its size remains bounded. One may observe a resemblance to the celebrated Goldreich-Levin theorem Goldreich & Levin (1989). This resemblance is not coincidental; in fact, the first part of this algorithm can be viewed as an implicit version of the Goldreich-Levin algorithm. We now formally state and prove the following lemma.

Lemma A.2. *Let $\mathscr{C}$ be a randomly permuted coset structure of codimension $\log \frac{100}{\eta^4}$, with $\eta = \frac{\theta}{8}$ (see Algorithm 3). Then, except with probability at most $\frac{1}{20}$, after **Step 3**, our algorithm identifies a set of cosets $\mathcal{C} \subseteq \mathscr{C}$, such that*

- *For any $C \in \mathscr{C}$, if $|\widehat{f}(\alpha^*(C))| \geq \theta$, then C belongs to $\mathcal{C}$*
- *For every $C \in \mathcal{C}$, the Fourier coefficient satisfies $|\widehat{f}(\alpha^*(C))| \geq \frac{7}{8}\theta$. .*

The proof of Lemma A.2 relies on the crucial observation that, for every $C \in \mathscr{C}$, the weight of the coset, $\mathcal{W}_C := \sum_{\beta \in C} \widehat{f}^2(\beta)$, is concentrated around $\mathcal{W}_C^* := |\widehat{f}(\alpha^*(C))|^2$, the weight of its dominant Fourier coefficient, with high probability. We formally state and prove this below.

Lemma A.3. *Let $f : \mathbb{F}_2^n \mapsto \{-1, +1\}$, and $\mathscr{C}$ be a randomly permuted coset structure of codimension $\log \frac{100}{\eta^4}$. Then, for all $C \in \mathscr{C}$ and for all $\tau > 0$, we have*

$$\Pr[|\mathcal{W}_C - \mathcal{W}_C^*| > \eta^2] \leq \frac{\eta^2}{100}$$

when Fourier coefficients of f is projected onto $\mathscr{C}$.

Proof. First we show that, for random permuted coset structure of sufficiently large codimension, $\mathbb{E}\left[\mathcal{W}_C - \mathcal{W}_C^*\right]$ is small. Let I_β be the indicator random variable indicating, whether $\alpha^*(C)$ and $\beta \in (\mathbb{F}_2^n \setminus \alpha^*(C))$ fall in the same bucket C or not. Then,

$$\mathbb{E}\left[|\mathcal{W}_C - \mathcal{W}_C^*|\right] = \mathbb{E}\left[\sum_\beta \widehat{f}^2(\beta) I_\beta\right] = \sum_\beta \mathbb{E}\left[\widehat{f}^2(\beta)\right] \mathbb{E}\left[I_\beta\right] = \sum_\beta \widehat{f}^2(\beta) \mathbb{E}\left[I_\beta\right]$$

By Lemma A.1, when a random permuted coset structure of codimension t is defined over $\mathbb{F}_2^n$, the probability that distinct $\alpha_1 \in \mathbb{F}_2^n$ and $\alpha_2 \in \mathbb{F}_2^n$ are placed in the same bucket is $\frac{1}{2^t}$. Using this observation, we get,

$$\mathbb{E}[|\mathcal{W}_C - \mathcal{W}_C^*|] = \sum_\beta \frac{\eta^4 \widehat{f}^2(\beta)}{100} \leq \frac{\eta^4}{100}.$$

In the final inequality, we used the fact that $\sum_\beta \widehat{f}^2(\beta) \leq 1$, since f is a Boolean function. Notably, we have $\mathcal{W}_C - \mathcal{W}_C^* \geq 0$. Applying Markov's inequality, we obtain,

$$\Pr[|\mathcal{W}_C - \mathcal{W}_C^*| > \eta^2] \leq \frac{\mathbb{E}[Y]}{\eta^2} = \frac{\eta^4}{100\eta^2} = \frac{\eta^2}{100}$$

□

Equipped with Lemma A.3, we now give the proof of Lemma A.2 below.

Proof of Lemma A.2. We claim that when the Fourier spectrum of f is projected onto a randomly permuted coset structure $\mathscr{C}$ of codimension $\log \frac{100}{\eta^4}$, all Fourier coefficients with absolute magnitude at least η are mapped to distinct cosets of $\mathscr{C}$. To see this, note that by Parseval's identity for Boolean functions, there can be at most $\frac{1}{\eta^2}$ Fourier coefficients with absolute magnitude at least η. By Lemma A.1, these $\frac{1}{\eta^2}$ large coefficients are mapped to distinct cosets of $\mathscr{C}$.

We also assume that in **Step 2** of the algorithm, all weight estimates are accurate within $\pm\eta^2$, with confidence at least $1 - 1/\mathrm{poly}(1/\theta)$. It remains to show that the threshold of $\frac{56}{64}\theta^2$ in **Step 3** ensures that all heavy cosets are retained and all light cosets are discarded.

To see this, consider a coset C that contains a Fourier coefficient with absolute value at least θ. The total weight of such a coset is at least θ^2, and its estimated weight is at least

$$\theta^2 - \eta^2 = \frac{63}{64}\theta^2 \geq \frac{56}{64}\theta^2,$$

ensuring that such a coset is not discarded at **Step 3**.

Conversely, consider a coset D that survives **Step 3** (i.e., it is not discarded). Its actual weight must be at least

$$\frac{56}{64}\theta^2 - \eta^2 = \frac{55}{64}\theta^2.$$

Assuming Lemma A.2 holds for this coset, the weight of its heaviest Fourier coefficient, denoted by $\mathcal{W}_C^*$, satisfies

$$\frac{55}{64}\theta^2 - \eta^2 = \frac{54}{64}\theta^2 > \left(\frac{7}{8}\theta\right)^2,$$

which implies that the absolute value of the heaviest Fourier coefficient in coset D is at least $\frac{7}{8}\theta$.

It remains to bound the probability of failure in Lemma A.2. Failure can occur in two cases:

- **Lemma A.1 does not hold:** Two distinct Fourier coefficients, each with absolute value at least η, fall into the same coset. By Lemma A.1, the probability of this event is at most $\frac{1}{100}$.
- **Lemma A.3 does not hold:** There exists a coset $C \in \mathcal{C}$ where Lemma A.3 fails. Since cosets with estimated weight below $\frac{56}{64}\theta^2$ are discarded, each surviving coset has actual weight at least $\frac{55}{64}\theta^2$. By Parseval's identity, $\sum_{\alpha\in\mathbb{F}_2^n} \widehat{f}(\alpha)^2 = 1$, so the total number of surviving cosets is at most $\frac{64}{55}\theta^2$. Applying Lemma A.3 and a union bound over these cosets, the failure probability is at most

$$\frac{\eta^2}{100} \cdot \frac{64}{55}\theta^2 < \frac{1}{50}.$$

Combining both failure probabilities, the overall probability of failure is at most

$$\frac{1}{100} + \frac{1}{50} < \frac{1}{20}.$$

□

The core step in **Step 4** involves computing $\chi_{\alpha^*(C)}(x)$ for a uniformly chosen $x \in \mathbb{F}_2^n$, for each coset $C \in \mathcal{C}$. It is important to note that the identity of $\alpha^*(C)$ is not known explicitly. To address this, we employ a self-correction procedure based on Hadamard codes. Formally, we establish the following lemma:

Lemma A.4. *Consider a coset C for which Lemma A.3 holds, ensuring that $|\mathcal{W}_C - \mathcal{W}_C^*| \le \eta^2$. Let $x, y \sim \mathbb{F}_2^n$ be independently and uniformly chosen. Then,*

$$\Pr\left[\operatorname{sign}(\mathcal{P}_C(x+y)) \cdot \operatorname{sign}(\mathcal{P}_C(y)) = \chi_C(x)\right] \ge \frac{7}{8}.$$

The proof of this Lemma A.4 relies on the observation that for every coset C, for uniformly sampled z, $\mathcal{P}_C(z) := \sum_{\beta\in C} \widehat{f}(\beta)\chi_\beta(z)$, the projection of $f(z)$ onto a coset C is highly concentrated around $\mathcal{P}_C^*(z) := \widehat{f}(\alpha^*(C))\chi_{\alpha^*(C)}(z)$. More formally, we establish the following concentration result.

Lemma A.5. *Let z is chosen uniformly at random from $\mathbb{F}_2^n$ and C be a coset of $\mathscr{C}$ such that $|\mathcal{W}_C - \mathcal{W}_C^*| \le \eta^2$. Then*

$$\Pr\left[|\mathcal{P}_C(z) - \mathcal{P}_C^*(z)| > \eta + \tau\right] \le \frac{\eta^2}{\tau^2}$$

Proof. To establish this, we first recall that Fourier characters act as pairwise independent hash functions $\chi : \mathbb{F}_2^n \to \{-1, +1\}$. Specifically, when z is sampled uniformly from $\mathbb{F}_2^n$, each coordinate z_i is independent and uniformly distributed over $\{0, 1\}$. For any non-empty subset $\alpha \subseteq [n]$, we have:

$$\mathbb{E}_z[\chi_\alpha(z)] = \mathbb{E}_z[(-1)^{\langle\alpha,z\rangle}] = \frac{1}{2}(1) + \frac{1}{2}(-1) = 0.$$

Moreover, for distinct subsets $\alpha_1, \alpha_2 \subseteq [n]$, the joint expectation satisfies:

$$\mathbb{E}_z[\chi_{\alpha_1}(z) \cdot \chi_{\alpha_2}(z)] = \mathbb{E}_z[(-1)^{\langle\alpha_1+\alpha_2,z\rangle}] = 0 = \mathbb{E}_z[\chi_{\alpha_1}(z)] \cdot \mathbb{E}_z[\chi_{\alpha_2}(z)].$$

Algorithm 3: SubspaceImpicitCosetSampler

Input: Threshold: θ, Number of samples : λ, Distance: ϵ
Output: Coset Samples of $f \circ A$, with respect to subspace $\mathcal{S}(\theta)$
Parameters: $\eta = \frac{\theta}{8}$, $\kappa = \max(400\theta^2, \lambda)$, $\gamma = \log 100\kappa$,

Step 1: Let $\mathscr{C}$ be a randomly permuted coset structure of a randomly chosen subspace of codimension $\log \frac{100}{\eta^4}$;
Step 2: foreach $C \in \mathscr{C}$ **do**
  Estimate weight of C within $\pm\eta^2$ accuracy;
end
Step 3: Discard any coset with estimated weight $\leq \frac{56}{64}\theta^2$; Let $\mathcal{C}$ be the set of surviving cosets;
Step 4: for $i \in \{1, 2, 3, \cdots, \kappa\}$ **do**
  Sample x_i uniformly at random from $\mathbb{F}_2^n$ and set $F[x_i] \leftarrow f(x_i)$;
  Sample $\{y_1, y_2, \ldots, y_\gamma\}$ each uniformly at random from $\mathbb{F}_2^n$;
  foreach y_j **do**
    foreach $C \in \mathcal{C}$ **do**
      Estimate $P_C f(y_j)$ and $P_C f(x_i + y_j)$, each within $\pm\frac{1}{8}\theta$ accuracy;
    end
  end
  Set $Q[x_i][C] \leftarrow \underset{j}{\text{median}}\{\text{sign}(\mathcal{P}_C[f(y_j)]) \cdot \text{sign}(\mathcal{P}_C[f(x_i + y_j)])\}$;
end
Step 5: Relabel the columns of Q as $\{B_1, \ldots, B_k\}$, where $B_1 = e_1^n, B_2 = e_2^n, \ldots, B_r = e_r^n$, such that for all $i \in \{r+1, \ldots, k\}$, the column B_i is a linear combination of $\{B_1, \ldots, B_r\}$. Here, $(e_i^n)_{i\in[r]}$ denote the standard basis vectors of $\mathbb{F}_2^n$ along coordinate direction i.
Step 6: foreach $x \in \{x_1, x_2, x_3, \cdots, x_\kappa\}$ **do**
  Let $b \in \mathbb{F}_2^r$ such that $b_j = \frac{1-Q[x][B_j]}{2}$ for all $j \in [r]$;
  $\zeta \leftarrow \{b, f(x)\} \cup \zeta$;
end
Step 7: Construct the matrix $H \in \mathbb{F}_2^{r\times n}$ whose rows are the vectors $B_1, B_2, \ldots, B_r$
Step 8: return $\{H, \zeta, r\}$

We now proceed to bound the expectation of $|\mathcal{P}_C(z) - \mathcal{P}^*_C(z)|$, analyzing two separate cases depending on whether $0 \in C'$ or not, where $C' = \{C - \alpha^*(C)\}$.

Case I: When C does not contain 0 or 0 is the leader of the coset C .

$$\mathbb{E}[|\mathcal{P}_C(z) - \mathcal{P}^*_C(z)|] = \mathbb{E}[\sum_{\beta \in C'} \widehat{f}(\beta)\chi_\beta(z)] = \sum_{\beta \in C'} \mathbb{E}[\widehat{f}(\beta)] \cdot \mathbb{E}[\chi_\beta(z)] = \sum_{\beta \in C'} \mathbb{E}[\widehat{f}(\beta)] \cdot 0 = 0.$$

Case II: When C contains 0 and 0 is not the leader of the coset C.

$$\mathbb{E}[|\mathcal{P}_C(z) - \mathcal{P}^*_C(z)|] = \mathbb{E}[\sum_{\beta \in C'} \widehat{f}(\beta)\chi_\beta(z)] + |\widehat{f}(0)| = 0 + |\widehat{f}(0)| = \eta.$$

For the final inequality, observe that after projecting the function onto a randomly permuted coset structure of codimension $\log \frac{100}{\eta^4}$, all Fourier coefficients with an absolute magnitude of at least η are mapped to distinct cosets. It follows that the second-largest Fourier coefficient in any coset has an absolute value of at most η. The second equality follows directly by substituting the analysis from **Case I** into the left operand of the sum.

Next, we show that the variance of Y is small by analyzing the sum over all $\beta \in C'$:

$$\begin{aligned}
\mathrm{Var}[\mathcal{P}_C(z) - \mathcal{P}^*_C(z)] &= \mathrm{Var}\left[\sum_{\beta \in C'} \widehat{f}(\beta)\chi_\beta(z)\right] \\
&= \sum_{\beta \in C'} \mathrm{Var}\left[\widehat{f}(\beta)\chi_\beta(z)\right] \\
&\leq \sum_{\beta \in C'} \mathbb{E}\left[\left(\widehat{f}(\beta)\chi_\beta(z)\right)^2\right] \\
&= \sum_{\beta \in C'} \left(\widehat{f}(\beta)\right)^2 \cdot \mathbb{E}\left[(\chi_\beta(z))^2\right] \\
&= \sum_{\beta \in C'} \left(\widehat{f}(\beta)\right)^2 \\
&= \mathcal{W}_C - \mathcal{W}^*_C \\
&\leq \eta^2.
\end{aligned}$$

Applying Chebyshev's inequality, we obtain:

$$\Pr\left[|\mathcal{P}_C(z) - \mathcal{P}^*_C(z)| > \eta + \tau\right] \leq \frac{\mathrm{Var}\left[\mathcal{P}_C(z) - \mathcal{P}^*_C(z)\right]}{\tau^2} \leq \frac{\eta^2}{\tau^2}.$$

□

Equipped with Lemma A.5, we now return to the proof of Lemma A.4.

Proof of Lemma A.4. We now show that for a uniform random choice of $z \in \mathbb{F}_2^n$, the sign of $\mathcal{P}^*_C(z)$ matches the sign of $\mathcal{P}_C(z)$ with high probability. To see why, let us fix $\tau = \frac{1}{2}\theta$ and define the following event:

$$\mathscr{E}_z : |\mathcal{P}_C(z) - \mathcal{P}^*_C(z)| < \eta + \tau = \frac{1}{8}\theta + \frac{1}{2}\theta = \frac{5}{8}\theta.$$

Remember that by Lemma A.2, for every $C \in \mathcal{C}$

$$|\widehat{f}(\alpha^*(C))| \geq \frac{7}{8}\theta.$$

So, assuming $\mathscr{E}_z$ occurs, the deviation between $\mathcal{P}^*_C(z)$ and $\mathcal{P}_C(z)$ is at most

$$|\mathcal{P}^*_C(z) - \mathcal{P}_C(z)| \leq \frac{7}{8}\theta - \frac{5}{8}\theta = \frac{1}{4}\theta.$$

Since this deviation is small, it follows that the signs of $\mathcal{P}^*_C(z)$ and $\mathcal{P}_C(z)$ do match. Using this fact, we can apply a self-correction procedure to compute $\chi_C(x)$ as follows:

$$\text{sign}(\mathcal{P}_C(\mathbf{y})) \cdot \text{sign}(\mathcal{P}_C(x+\mathbf{y})) = \text{sign}(\mathcal{P}^*_C(\mathbf{y})) \cdot \text{sign}(\mathcal{P}^*_C(x+\mathbf{y})) = \chi_{\alpha^*(C)}(x)$$

It is easy to see that this self-correction fails with probability at most $\frac{1}{8}$. □

However, the algorithm does know $\mathcal{P}_C(z)$ exactly, though it can estimate $\mathcal{P}_C(z)$ with reasonable accuracy. In particular, if the estimated value $\tilde{\mathcal{P}}_C(z)$ lies within $\pm\frac{\theta}{8}$ of the true value, the preceding argument still remains valid. Furthermore, to proceed to the next phase of the algorithm, we need to compute $\chi_S(x)$ for a sufficiently large number of x and for all $C \in \mathcal{C}$. This requires improving the success probability of the self-correction process. To achieve this, we employ the *median trick*, a standard technique for error reduction. Specifically, for a fixed x, we perform multiple independent trials using different values of y and take the median of the observed outcomes. By applying a standard Chernoff bound, it follows that taking $O(\log \kappa)$ independent samples is sufficient to amplify the success probability to at least $1 - 1/\text{poly}(1/\theta)$.

Having access to evaluation of $\chi_{\alpha^*(C)}(x)$ for sufficiently many x, we can now identify the characters $\alpha^*(C)$ for all $C \in \mathcal{C}$, up to linear transformation, without knowing their explicit identity. To see how, let $\mathcal{S}$ be a subset of $\mathcal{C}$ such that

$$\sum_{C \in \mathcal{S}} \alpha^*(C) = \mathbf{0}^n.$$

Then, for every $x \in \mathbb{F}_2^n$, we have

$$\prod_{C \in \mathcal{S}} \chi_{\alpha^*(C)}(x) = \prod_{C \in \mathcal{S}} (-1)^{\langle \alpha^*(C), x \rangle} = (-1)^{\left\langle \sum_{C \in \mathcal{S}} \alpha^*(C), x \right\rangle} = 1.$$

Thus, for any such set $\mathcal{S}$, the product of the corresponding columns of Q, denoted as $\Pi_{\mathcal{S}}$, always equals $\mathbf{1}^\lambda$. On the other hand, consider any subset $\mathcal{B} \subseteq \mathcal{C}$ where the vectors $\{\alpha^*(C) : C \in \mathcal{B}\}$ are linearly independent. We aim to show that the probability of all entries of $\prod_{C \in \mathcal{B}} \chi_{\alpha^*(C)}(x)$ being equal to $\mathbf{1}$ is small.

Lemma A.6. *The probability that all entries of $\prod_{C \in \mathcal{B}} \chi_{\alpha^*(C)}(x_i)$ are equal to 1 is at most $2^{-\kappa}$. Moreover, for any given $\mathcal{B}$, the same event occurs for every subset of $\mathcal{B}$ with probability close to $1/100$.*

Proof. Since the vectors $\alpha^*(C)$ are linearly independent and the points $x_1, \ldots, x_\kappa$ are uniformly distributed in $\mathbb{F}_2^n$, we have

$$\Pr\left[\prod_{C \in \mathcal{B}} \chi_{\alpha^*(C)}(x_i) = 1\right] = \frac{1}{2}.$$

Thus, the probability that this holds for all κ independent trials is at most $2^{-\kappa}$. Furthermore, since the number of possible subsets of $\mathcal{B}$ is at most $2^{|\mathcal{C}|} = 2^{O(\frac{1}{\theta^2})}$ it follows that the event

$$\prod_{C \in \mathcal{D}} \chi_{\alpha^*(C)}(x_i) = 1$$

occurs for every $\mathcal{D} \subseteq \mathcal{B}$, except with probability at most

$$2^{-\kappa} \cdot 2^{O(\frac{1}{\theta^2})} < o(1).$$

□

We cannot explicitly determine $\alpha^*(C)$, but using the linear relationships established in the previous lemma, we can identify them up to a linear transformation. Specifically, in **step 5** of the algorithm, we extracted a basis of size r for the span of $\alpha^*(C)$ and relabeled them as the standard basis vectors e_i^n in $\mathbb{F}_2^n$. Let H be the subspace spanned by these basis vectors. Define the matrix $M \in \mathbb{F}_2^{r\times n}$ with rows given by the basis vectors,

$$M = \begin{bmatrix} e_1^n \\ e_2^n \\ \vdots \\ e_r^n \end{bmatrix}.$$

Then, for every $x \in \{x_1, x_2, \cdots, x_\kappa\}$, the coset of $H^\perp$ containing x is determined by the set of linear equations $Mx = b$, where b is defined in **Step 6** of the algorithm. Finally, we analyze the probability distribution of cosets of the subspace $H^\perp$ when an element $x \in \mathbb{F}_2^n$ is chosen uniformly at random and its corresponding coset is determined. We show that the distribution over the cosets of $H^\perp$, induced by the uniform random selection of an element $x \in \mathbb{F}_2^n$, is uniform.

Lemma A.7. *Given any $G \subseteq \mathbb{F}_2^n$, the distribution of cosets induced by uniform random selection of $x \in \mathbb{F}_2^n$ is uniform over the cosets of the subspace G.*

Proof. Let $G \subseteq \mathbb{F}_2^n$ be a subspace of dimension k. The cosets of G in $\mathbb{F}_2^n$ are of the form $C_z = z+G$ for $z \in \mathbb{F}_2^n$, and the total number of cosets is 2^{n-k}. We randomly select $x \in \mathbb{F}_2^n$ uniformly, and then determine which coset C_h the element x belongs to. Since the size of each coset C_z is 2^k and the total size of $\mathbb{F}_2^n$ is 2^n, the probability that x belongs to a particular coset C_z is:

$$P(x \in C_z) = \frac{|C_z|}{|\mathbb{F}_2^n|} = \frac{2^k}{2^n} = \frac{1}{2^{n-k}}.$$

Since this probability is the same for all cosets C_z, the distribution over the cosets of G is uniform. □

It remains to bound the failure probability of Lemma 3. Assuming that all estimates in **Step 2** and **Step 4** are accurate within the specified accuracy and confidence level of $1 - 1/\mathrm{poly}(1/\theta)$, the probability of failure in either of these steps is at most $o(1)$. Similarly, in **Step 6**, the linear relationship detection routine fails with probability at most $o(1)$. The only significant failure probability arises in **Step 3**, which, by Lemma A.2, is bounded above by $\frac{1}{20}$. Therefore, the total failure probability is at most $\frac{1}{20} + o(1) < \frac{1}{10}$.

Finally, we upper bound the total number of function queries made by the algorithm. In **Step 2**, estimating the weight

$$\mathcal{W}_{r+H} = \mathbb{E}_{x\in\mathbb{F}_2^n, z\in H^\perp}\left[\chi_r(z)\cdot f(x)\cdot f(x+z)\right]$$

for each coset within $\pm\eta^2$ accuracy and with confidence $1 - 1/\mathrm{poly}(1/\theta)$ requires $\tilde{O}(1/\theta^4)$ queries, by standard applications of the Chernoff bound. Observe that the same set of queries, namely $\{f(x), f(x+z)\}_{x\in H^\perp}$, can be used to compute $\mathcal{W}_{r+H}$ for all cosets $r + H$ of H simultaneously.

Similarly, in **Step 4**, estimating

$$\mathcal{P}_{r+H}(x) = \mathbb{E}_{y\in H^\perp}\left[\chi_r(y)\cdot f(x+y)\right]$$

within $\pm\frac{\theta}{8}$ accuracy and with confidence $1 - 1/\mathrm{poly}(1/\theta)$ requires $\tilde{O}(1/\theta^2)$ queries. Again, the same batch of samples can be reused for all cosets. Finally, recall that $\kappa = \max(1/\theta^2, \lambda)$ samples must be generated. Combining all the steps, the total query complexity of the algorithm is:

$$\tilde{O}\left(\frac{1}{\theta^4} + \max\left\{\frac{1}{\theta^2}, \lambda\right\}\cdot\frac{1}{\theta^2}\right)$$

□