Distribution Free Fourier Sparsity Testing

Arijit Ghosh
arijitiitkgpster@gmail.com
Indian Statistical Institute, Kolkata, India

Manmatha Roy
reach.manmatha@gmail.com
Indian Statistical Institute, Kolkata, India

Abstract

Learning functions $f : \mathbb{F}_2^n \to \mathbb{R}$ with sparse Fourier representations plays a central role in computational learning theory, as such functions capture many natural concept classes in machine learning. In recent years, Fourier sparsity has also been studied from the related perspective of property testing, where the goal is to determine, given query access to a function, whether it is s-Fourier-sparse or far from every such function under an appropriate distance measure. Such testers can serve as a preprocessing step for model selection in agnostic learning of Fourier-sparse functions, where validating the structural assumption is crucial before applying computationally expensive learning algorithms. However, all prior work on testing Fourier sparsity considers distance with respect to the uniform distribution over the hypercube.

In this work, we initiate the study of Fourier sparsity testing in the distribution-free setting, the property-testing analogue of the PAC+MQ learning framework. In this model, the tester is given oracle access to an unknown function and sample access to an arbitrary and unknown input distribution, and must distinguish s-Fourier-sparse functions from those that are δ-far from every such function with respect to the underlying distribution. We design a nonadaptive randomized tester that succeeds with constant probability using $\widetilde{O}((s/\delta)^4)$ oracle queries and $\widetilde{O}(1/\delta)$ samples.

1 Introduction

Sparsity is one of the most powerful ideas connecting modern machine learning and theoretical computer science. It captures the intuition that, even in high-dimensional settings, many natural functions or signals depend only on a small number of relevant components. This assumption underlies algorithms that are both sample- and time-efficient, forming the basis of techniques such as sparse linear regression, decision tree learning, and compressed sensing. Across these frameworks, the common principle is simple yet profound: if a function admits a sparse representation in a suitable basis, such as Fourier, wavelet, polynomial, or a learned dictionary, then learning and inference can be made dramatically more efficient. A particularly elegant setting where sparsity plays a central role is that of real-valued functions over the Boolean hypercube $\mathbb{F}_2^n$. Fourier analysis [O'Donnell, 2014, Stein and Shakarchi, 2003] provides a natural orthonormal basis for such functions. Every function $f : \mathbb{F}_2^n \to \mathbb{R}$ can be expressed as

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

where $\widehat{f}(\alpha)$ denotes the Fourier coefficient at frequency α and $\langle \cdot, \cdot \rangle$ denotes the standard inner product over $\mathbb{F}_2^n$. The Fourier sparsity of f, defined as the number of nonzero coefficients in this expansion, quantifies how succinctly the function can be represented in the Fourier basis.

In learning theory, functions with sparse Fourier spectra arise naturally and have been studied extensively. Prominent examples include graph and hypergraph cut functions, as well as decision

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

trees of bounded depth, whose Fourier spectra are concentrated on low-degree coefficients [Amrollahi et al., 2019, Stobbe and Krause, 2012, Linial et al., 1993]. For instance, the cut function of a graph can be represented as a degree-2 multilinear polynomial in the Fourier basis, while the cut function induced by a degree-d hypergraph admits a similar representation of degree at most d. More generally, the Fourier spectrum of a Boolean decision tree of depth d is entirely supported on coefficients of degree at most d. Learning such functions has recently found applications in neural network hyper-parameter optimization [Hazan et al., 2018]. From an algorithmic perspective, efficiently learning Fourier-sparse Boolean functions from uniform random examples has long been a central open problem in computational learning theory [Blum, 2003]. Over the years, many approaches have been explored, often under relaxed models or with stronger access [Kushilevitz and Mansour, 1993, Negahban and Shah, 2012, Hassanieh et al., 2012, Vershynin, 2018, Haviv and Regev, 2017, Amrollahi et al., 2019, Indyk et al., 2014]. However, almost all these algorithms assume prior knowledge of the target function's Fourier sparsity level. This fact naturally motivates the need for *testing for Fourier Sparsity*, which aims to validate the Fourier sparsity assumption, without fully learning the function. A tester of this kind is a natural preprocessing step before an exact or proper learning algorithm: such algorithms assume that the target is exactly s-Fourier-sparse, so certifying this structural assumption first, at a cost far below that of learning, ensures the assumption holds before the more expensive learner is run.

Property testing was introduced in the seminal work of Blum, Luby, and Rubinfeld Blum et al. [1993] and has since developed into a mature branch of theoretical computer science, particularly well suited for questions of this type. It studies algorithms that, given query access to an unknown function, decide whether the function satisfies a specified property or is far from every function having that property. The objective is to make this decision using as few queries as possible, typically sublinear in the size of the domain. Distance is measured with respect to an underlying input distribution, most commonly the uniform distribution. We refer the interested reader to Goldreich [2017] for a comprehensive survey of classical results in this area. The problem of testing Fourier sparsity in the property-testing framework has been studied in a sequence of works [Gopalan et al., 2011, Yaroslavtsev and Zhou, 2020, Ghosh and Roy, 2025]. In Gopalan et al. [2011], the underlying distance notion was the Hamming distance, whereas later works considered the Euclidean distance, which we believe is a more natural measure in machine-learning settings. In particular, Yaroslavtsev and Zhou [2020] showed that using $\tilde{O}(s/\varepsilon^4)$ queries one can determine whether an unknown function is s-Fourier-sparse or ε-far from every such function in Euclidean distance. On the lower-bound side, they showed that any tester solving this problem must make at least $\Omega(\sqrt{s})$ queries. Very recently, Ghosh and Roy [2025] proposed a simpler tester, with query complexity $\tilde{O}(s/\varepsilon^2)$ and established a matching lower bound of $\Omega(s)$, thereby proving optimality up to logarithmic factors and nearly closing the gap between upper and lower bounds. Another important aspect of the upper-bound techniques of both Ghosh and Roy [2025] and Yaroslavtsev and Zhou [2020] is that the corresponding testers are nonadaptive, i.e., the set of queries is fixed before any oracle interaction. By contrast, the lower-bound arguments of Ghosh and Roy [2025] apply even to adaptive testers, which are allowed to select queries based on previous oracle responses.

However, all existing work on testing Fourier sparsity has focused on the uniform-distribution setting, where distance is measured with respect to the uniform measure on the Boolean cube. While this assumption is mathematically convenient, it is often unrealistic in learning scenarios, where the input distribution may be structured, biased, or entirely unknown [Valiant, 1984]. This observation motivates the study of the *distribution-free* setting, originally introduced in [Halevy and Kushilevitz, 2007]. In this model, distance between functions is measured with respect to the unknown underlying distribution, and the tester is required to distinguish functions that satisfy the property from those that are far from it under this distribution. We follow the standard convention of this model: the property being tested, Fourier sparsity in the canonical basis, is fixed, while only the distance is measured under the unknown distribution D. A complementary line of work instead adapts the Fourier representation itself to a non-uniform measure, for example to learn classes such as DNF through generalized Fourier representations [, see note above]; sparsity in such a distribution-dependent basis is a different, and largely open, structural notion, which we discuss further in Section 5. The tester must succeed for every fixed but unknown distribution over $\mathbb{F}_2^n$ and is given only sample access to the distribution. More precisely, distribution-free property testing is the analogue of the PAC+MQ model of learning (which has traditionally been studied via the equivalent EQ+MQ framework) [Kearns and Vazirani, 1994], whereas standard property testing corresponds to learning under the uniform distribution with membership queries. Designing efficient testers in the distribution-free model is significantly more

challenging, as many uniform-distribution testers rely on the fact that functions far from the property contain many small local witnesses, which can be found via random sampling. In contrast, in the distribution-free setting, a function may be very close to the property under the uniform distribution, hence containing few such witnesses, while simultaneously being far from the property with respect to some arbitrary input distribution. In fact, it is known that there is no generic transformation that can convert a standard (uniform-distribution) tester into a distribution-free tester for an arbitrary nontrivial property [Goldreich et al., 1998]. Indeed, designing distribution-free testers is particularly subtle: moving from the uniform-distribution setting to the distribution-free setting can, in general, lead to a dramatic increase in complexity. A concrete illustration of this phenomenon is provided by the class of *junta functions*. In the uniform-distribution setting, Blais [2008] gave a nonadaptive tester for k-juntas with query complexity $\widetilde{O}(k^{3/2})$. However, Liu et al. [2019] later showed that in the distribution-free model, any nonadaptive algorithm for testing k-juntas over $\{0,1\}^n$ must make $2^{\Omega(k)}$ queries, even for $\varepsilon = 1/3$.

In this work, we initiate a systematic study of Fourier sparsity testing in the distribution-free setting. Given oracle access to an unknown function and sample access to a fixed but unknown input distribution, we ask whether one can design a tester that distinguishes functions with at most s nonzero Fourier coefficients from those that are ε-far (in appropriate distance measure) from every such function, with respect to the underlying distribution. Motivated by the junta-function scenario, a natural question is whether such a goal can be achieved with only a modest blowup in query complexity compared to the best-known tester in the uniform-distribution setting. Our work answers this question in the affirmative. We now formally define the problem studied in this work.

1.1 (ϵ, δ)-Distribution-Free s-Fourier Sparsity Tester

Input: Oracle access to a function $f : \mathbb{F}_2^n \to \mathbb{R}$ satisfying $\mathbb{E}_{x\sim\mathbb{F}_2^n}\left[f(x)^2\right] = 1$, and sample access to an unknown distribution D over $\mathbb{F}_2^n$.

Goal: Design a randomized algorithm that makes as few oracle queries to f and as few samples from D as possible, and distinguishes, with probability at least $2/3$, between the following two cases:

- **(YES)** The function f has at most s nonzero Fourier coefficients.
- **(NO)** For every function $g : \mathbb{F}_2^n \to \mathbb{R}$ with at most s nonzero Fourier coefficients, we have

$$\Pr_{x\sim D}\left[|f(x) - g(x)| > \varepsilon\right] > \delta$$

Complexity measure: The complexity of the algorithm is measured in terms of the number of oracle queries to f and the number of samples drawn from the distribution D.

Remark 1. *We note that the assumption on the expected squared norm of the unknown function is without loss of generality. Indeed, any function can be appropriately rescaled so that its expected squared norm with respect to the underlying distribution equals* 1.

1.2 Our Contribution

We first note that the notion of distance used in this work (see Definition 3) is a stable variant of the ℓ_2 distance. In particular, in the uniform distribution setting, being ε-far under this distance notion implies being far in the standard ℓ_2 distance. Consequently, existing nonadaptive testers for Fourier sparsity under the ℓ_2 distance in the uniform distribution setting remain applicable in our framework. Finally, we observe that the lower bound of Ghosh and Roy [2025] also extends to this distance measure for constant parameters $\varepsilon = O(1)$ and $\delta = O(1)$, yielding a lower bound of $\Omega(s)$ for this problem. We now state our result.

Theorem 1 (Main Theorem)**.** *For any $\varepsilon, \delta \in (0,1)$ and $s \in \mathbb{N}$, there exists a nonadaptive randomized algorithm with success probability at least $2/3$ for the (ε, δ)-distribution-free s-Fourier sparsity testing problem that makes $\widetilde{O}\left(\frac{s^4}{\delta^4\varepsilon^8}\right)$ oracle queries to the unknown function and draws $\widetilde{O}(1/\delta)$ samples from the input distribution.*

Somewhat surprisingly, this result shows that moving from the uniform distribution model to the distribution free setting incurs only a quartic overhead in query complexity. This behavior stands in

sharp contrast to the junta testing scenario, where, in the absence of adaptivity, the query complexity in the distribution-free setting is known to be exponentially larger than in the uniform-distribution model.

Beyond this algorithmic result, we develop new Fourier-analytic techniques for analyzing Boolean functions with sparse Fourier spectra under arbitrary input distributions. These techniques may be of independent interest for distribution-free testing and learning problems more broadly.

1.3 Organization

The remainder of the paper is organized as follows. In Section 2, we review the necessary background on Fourier analysis. In Section 3, we provide a high-level overview proof sketch of our main result, highlighting the key intuitions. Section 4 presents our testing algorithm together with its analysis and correctness proof.

2 Background

Let $f : \mathbb{F}_2^n \to \mathbb{R}$ be a real-valued function over the Boolean cube, and let U denote the uniform distribution on $\mathbb{F}_2^n$. For $\alpha \in \mathbb{F}_2^n$, define the character $\chi_\alpha(x) := (-1)^{\langle \alpha, x \rangle}$, and the corresponding Fourier coefficient

$$\widehat{f}(\alpha) := \mathop{\mathbb{E}}_{x \sim \mathbb{F}_2^n} [f(x)\chi_\alpha(x)] .$$

With this notation, every function f admits the Fourier expansion

$$f(x) = \sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}(\alpha)\chi_\alpha(x).$$

Throughout this work, unless otherwise stated explicitly, we assume that all functions $f : \mathbb{F}_2^n \to \mathbb{R}$ are normalized, that is, $\mathbb{E}_{x \sim U} \left[f(x)^2 \right] = 1$. By Parseval's identity, this implies $\sum_{\alpha \in \mathbb{F}_2^n} \widehat{f}(\alpha)^2 = 1$.

Definition 2 (Fourier sparsity)**.** *A function $g : \mathbb{F}_2^n \to \mathbb{R}$ is said to be s-*Fourier-sparse *if*

$$\left| \{ \alpha \in \mathbb{F}_2^n : \widehat{g}(\alpha) \neq 0 \} \right| \leq s.$$

We denote by $\mathcal{P}_s$ the class of all s-Fourier-sparse functions.

To measure the distance between functions, we consider the following notion with respect to an arbitrary input distribution.

Definition 3 (ε-Error Distance)**.** *Let D be a distribution over $\mathbb{F}_2^n$ and let $f, g : \mathbb{F}_2^n \to \mathbb{R}$ be two functions. For any $\varepsilon > 0$, the* ε-Error distance *between f and g under D is defined as*

$$\mathrm{dist}_D^\varepsilon(f, g) \ := \ \Pr_{x \sim D} \left[\, |f(x) - g(x)| \geq \varepsilon \, \right].$$

Definition 4 (Distance to $\mathcal{P}_s$)**.** *For a function $f : \mathbb{F}_2^n \to \mathbb{R}$, ε-Error Distance from $\mathcal{P}_s$ with respect to D is defined as*

$$\mathrm{dist}_D^\varepsilon(f, \mathcal{P}_s) := \inf_{g \in \mathcal{P}_s} \mathrm{dist}_D^\varepsilon(f, g)$$

3 Proof Sketch

We now provide a high-level overview of our algorithm for testing Fourier sparsity in the distribution-free setting, deliberately suppressing many implementation details. To this end, we assume access to two algorithmic primitives: (i) a tester for Fourier sparsity under the uniform distribution, and (ii) a self-approximation procedure for the class of Fourier-sparse functions.

Assumption 5 (Uniform-Distribution Fourier Sparsity Tester). *Given $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$, let $f : \mathbb{F}_2^n \to \mathbb{R}$ be an unknown function accessible only via black-box query access. There exists a randomized tester T such that:*

- ***Completeness:** If $f \in \mathcal{P}_s$, then* $\Pr\big[T(\mathcal{O}_f, \varepsilon, \delta) = \text{YES}\big] \ \geq \ \frac{9}{10}$.

- ***Soundness:*** *If* $\mathrm{dist}_U^{\varepsilon}(f, \mathcal{P}_s) > \delta$*, then* $\Pr\big[T(\mathcal{O}_f, \varepsilon, \delta) = \text{NO}\big] \;\geq\; \frac{9}{10}$.

Moreover, the tester T *makes* $\mathrm{poly}(s, 1/\varepsilon, 1/\delta)$ *queries to the oracle for* f.

Remark 2. *Assumption 5 is satisfied by the uniform-distribution Fourier sparsity testers of Yaroslavtsev and Zhou [2020] and Ghosh and Roy [2025], both of which are nonadaptive and make* $\mathrm{poly}(s, 1/\varepsilon)$ *queries; see the discussion in Section 1.*

Assumption 6 (Self-Approximator for Fourier Sparse Function). *Given* $s \in \mathbb{N}$*,* $\tau \in (0, 1)$*, and* $\eta' \in (0, 1)$*, let* $f : \mathbb{F}_2^n \to \mathbb{R}$ *be an unknown function accessible only via black-box query access. Define*

$$g^\star \;:=\; \arg\min_{g \in \mathcal{P}_s} \|f - g\|_2^2 \;=\; \arg\min_{g \in \mathcal{P}_s} \; \mathop{\mathbb{E}}_{x \sim \mathbb{F}_2^n} \big[(f(x) - g(x))^2\big],$$

that is, $g^\star$ *is an* s*-Fourier-sparse function minimizing the squared* ℓ_2 *distance from* f *under the uniform measure over* $\mathbb{F}_2^n$*. Then there exists a randomized algorithm* $A(\mathcal{O}_f, \tau, \eta', \cdot)$ *such that for every* $x \in \mathbb{F}_2^n$*,*

$$\Pr\big[\big|A(\mathcal{O}_f, \tau, \eta', x) - g^\star(x)\big| \leq \tau\big] \;\geq\; 1 - \eta'$$

where the probability is taken over the internal randomness of the algorithm. Moreover, the algorithm A *makes* $\mathrm{poly}(s, 1/\tau, 1/\eta')$ *queries to the oracle for* f.

Remark 3. *Assumption 6 is stated here as a proof-sketch device to isolate the high-level structure of our approach. The self-approximator is* explicitly constructed *in Section 4 through three procedures:* FOURIER-SIEVE *(Algorithm 2, Lemma 1),* LIST-CORRECTION *(Algorithm 3, Lemma 2), and* ESTIMATE-HEAVY-FOURIER-COEFFS *(Algorithm 4, Lemma 3). Together they provide a worst-case pointwise approximation guarantee with query complexity independent of the ambient dimension* n*; full proofs appear in Section 4 and the appendix.*

Assuming these primitives, we now show how they can be combined to solve the distribution-free Fourier sparsity testing problem. In particular, Algorithm 1 presents a high-level sketch of how these components interact to yield our tester.

Algorithm 1: DISTRIBUTION-FREE FOURIER SPARSITY TESTER (SKETCH)

Input: Oracle access to $f : \mathbb{F}_2^n \to \mathbb{R}$; sample access to D; ε, δ
Run $T(\mathcal{O}_f, \varepsilon, \delta)$; **if** $T = \mathit{NO}$ **then**
 | **return** NO
end
$m \leftarrow \lceil c/\delta \rceil$; sample $x_1, \ldots, x_m \sim D$
for $i \leftarrow 1$ **to** m **do**
 | compute $A(\mathcal{O}_f, \varepsilon/4, 1/(20m), x_i)$ and query $f(x_i)$
end
if $\exists i \in [m] : \; |A(\mathcal{O}_f, \varepsilon/4, 1/(20m), x_i) - f(x_i)| > 3\varepsilon/4$ **then**
 | **return** NO
end
return YES

3.1 Analysis of Algorithm 1

We prove that Algorithm 1 is a correct (ε, δ)-distribution-free tester for Fourier sparsity by establishing completeness and soundness.

Proof. The proof of completeness and soundness is given below.

Completeness. Assume that $f \in \mathcal{P}_s$. We show that the tester accepts with high probability. By Assumption 5, the uniform-distribution Fourier sparsity tester $T(\mathcal{O}_f, \varepsilon, \delta)$ accepts with probability at least $9/10$. Hence, Algorithm 1 proceeds to the consistency-testing stage. Next, by Assumption 6, for every $x \in \mathbb{F}_2^n$,

$$\Pr\big[\big|A(\mathcal{O}_f, \varepsilon/4, x) - f(x)\big| \leq \varepsilon/4\big] \geq 1 - 1/20m,$$

where the probability is over the internal randomness of the estimator A. Applying a union bound over the m sampled points $x_1, \ldots, x_m$, the estimator succeeds on all samples except with probability at

most $1/20$. Therefore, the consistency test succeeds with probability at least $19/20$, and Algorithm 1 outputs YES with probability at least $1 - \left(\frac{1}{10} + \frac{1}{20}\right) > \frac{2}{3}$.

Soundness. Assume that $\mathrm{dist}_D^{\varepsilon}(f, \mathcal{P}_s) > \delta$. We show that the tester rejects with high probability. We distinguish two cases.

Case 1: $\mathrm{dist}_U^{\varepsilon}(f, \mathcal{P}_s) > \delta$. By Assumption 5, the uniform-distribution tester rejects with probability at least $9/10$, and the algorithm outputs NO.

Case 2: $\mathrm{dist}_U^{\varepsilon}(f, \mathcal{P}_s) \le \delta$. Let $g^\star := \arg\min_{g \in \mathcal{P}_s} \|f - g\|_2^2$. Since $g^\star \in \mathcal{P}_s$ and $\mathrm{dist}_D^{\varepsilon}(f, \mathcal{P}_s) > \delta$, it follows that

$$\Pr_{x \sim D}\left[\, |f(x) - g^\star(x)| > \varepsilon \,\right] > \delta.$$

By Assumption 6, the estimator $A^f(x) := A(\mathcal{O}_f, \varepsilon/4, x)$ satisfies $|A^f(x) - g^\star(x)| \le \varepsilon/4$ for all sampled x, except with probability at most $1/20$, by a union bound over the m samples. The tester can incorrectly output YES only if none of the m points $x_1, \ldots, x_m$ drawn from D satisfy $|f(x_i) - g^\star(x_i)| > \varepsilon$. This event occurs with probability at most

$$(1-\delta)^m \le e^{-\delta m} < \frac{1}{10},$$

for our choice of m. Combining the above with the estimator failure probability, the tester rejects with probability at least $1 - \left(\frac{1}{10} + \frac{1}{20}\right) > \frac{2}{3}$.

This completes the proof of correctness of Algorithm 1. □

What remains is to outline the design of the uniform-distribution Fourier sparsity tester and the accompanying self-approximator for the class of Fourier-sparse functions. The design of both the tester and the self-approximator is based on the paradigm of *sketching by hashing*, adapted to the Fourier-analytic setting Feldman et al. [2006], Gopalan et al. [2011], Datta et al. [2026]. We first randomly project the Fourier spectrum onto cosets of random subspaces. Under this projection, heavy Fourier coefficients are distributed across cosets in an almost pairwise independent manner, enabling the algorithm to identify heavy cosets likely containing large coefficients. Two sources of error arise: estimation error from sampling-based coefficient estimation, and hashing error due to collisions with non-heavy coefficients. Nevertheless, the analysis shows that one can still obtain additive approximations to the top s Fourier coefficients, which suffices to design a one-sided tester for Fourier sparsity under the uniform distribution.

For the self-approximator, we adapt ideas from self-correction of Hadamard codes. Once heavy cosets are identified, the corresponding characters can be evaluated through a self-correction procedure, allowing estimation of the magnitudes of the associated Fourier coefficients. Aggregating these estimates yields an approximation to the closest s-Fourier-sparse function. Although the identities of the heavy coefficients are not explicitly recovered, the approximator can still evaluate the resulting hypothesis accurately using the estimated magnitudes together with self-corrected character evaluations, which is sufficient for reliable ℓ_1-error estimation.

4 The Algorithm

We begin by presenting our technical tools—all helper algorithms and their theoretical guarantees—before proceeding to the proof of the main result.

Notation. Throughout this section, let $\mathscr{C}$ denote the collection of all cosets in the random partition produced by choosing a subspace of codimension $\lceil \log(2^{32}/\theta^8) \rceil$. For each coset $C \in \mathscr{C}$, define the *dominant Fourier coefficient* $\alpha_C := \arg\max_{\alpha \in C} |\widehat{f}(\alpha)|$ (ties broken arbitrarily), the *Fourier weight* $\mathsf{W}_C f := \sum_{\beta \in C} \widehat{f}(\beta)^2$, and the *coset projection* $\mathsf{P}_C f(z) := \sum_{\beta \in C} \widehat{f}(\beta)\, \chi_\beta(z)$.

First, we show that given only oracle access to f, it is possible to implicitly identify the locations of all heavy Fourier coefficients without explicitly reconstructing the entire Fourier spectrum. To this

end, we design a randomized filtering procedure, FOURIER-SIEVE (Algorithm 2), which partitions the Fourier domain into carefully chosen cosets and isolates those containing significant spectral mass. At a high level, the algorithm outputs a small collection of candidate cosets such that every Fourier coefficient of magnitude at least θ is guaranteed to lie in one of them, while each selected coset is dominated by a unique large Fourier coefficient and contains only negligible residual Fourier mass. The precise guarantee is formalised in Lemma 1.

Lemma 1. *Given oracle access to f and a threshold θ, the algorithm* FOURIER-SIEVE *(Algorithm 2) identifies, except with probability at most $1/100$, a set of cosets $\mathcal{C} \subseteq \mathscr{C}$ satisfying the following properties, using $\widetilde{O}(1/\theta^4)$ queries to f:*

- ***(Completeness.)*** *For any $C \in \mathscr{C}$ with $|\widehat{f}(\alpha_C)| \geq \theta$, we have $C \in \mathcal{C}$.*
- ***(Soundness.)*** *For every $C \in \mathcal{C}$, $|\widehat{f}(\alpha_C)| \geq \theta/2$.*
- ***(Concentration.)*** *For every $C \in \mathcal{C}$, $\sum_{\beta \in C \setminus \{\alpha_C\}} \widehat{f}(\beta)^2 < \theta^4/2^{12}$.*

Proof of Lemma 1 is deferred to the appendix.

Next, we show that the candidate cosets identified by FOURIER-SIEVE can be exploited to obtain reliable pointwise evaluations of their associated dominant Fourier characters. To achieve this, we develop a specialised self-correction procedure, LIST-CORRECTION (Algorithm 3), which, given any query point $x \in \mathbb{F}_2^n$, simultaneously recovers the values of the characters $\chi_{\alpha_C}(x)$ for every $C \in \mathcal{C}$ with high probability. The precise performance guarantees are formalised in Lemma 2.

Algorithm 2: FOURIER-SIEVE

Input: Threshold θ, oracle access to $f : \mathbb{F}_2^n \to \mathbb{R}$
Choose random subspace $U \leq \mathbb{F}_2^n$ with codimension $\log(2^{32}/\theta^8)$;
let $\mathscr{C}$ be cosets of $U^\perp$
foreach $C \in \mathscr{C}$ **do**
  estimate $\mathsf{W}_C f$ within $\pm\theta^2/4$
end
return $\{C \in \mathscr{C} : \widetilde{\mathsf{W}}_C f > \frac{3}{4}\theta^2\}$

Lemma 2. *Let $\mathcal{C}$ be the collection of cosets guaranteed by Lemma 1. Then, for any $x \in \mathbb{F}_2^n$ and oracle access to f,* LIST-CORRECTION *(Algorithm 3) outputs values $s_C \in \{+1, -1\}$ for all $C \in \mathcal{C}$ such that $\Pr\big[\,\chi_{\alpha_C}(x) = s_C \text{ for all } C \in \mathcal{C}\,\big] \geq 1 - \delta'$. Moreover, the query complexity of the algorithm is $\widetilde{O}\big(\log(1/\delta')/\theta^2\big)$.*

The full proof of Lemma 2 is deferred to the appendix.

Having obtained implicit oracle access to the dominant Fourier characters through LIST-CORRECTION, we next show how to estimate the corresponding Fourier coefficients themselves with sufficient accuracy. To this end, we design the procedure ESTIMATE-HEAVY-FOURIER-COEFFS (Algorithm 4), which combines evaluations of f with the recovered character values to approximate the heavy Fourier coefficients to user-specified precision. The resulting approximation guarantee is formalised in Lemma 3.

Lemma 3. *For all $C \in \mathcal{C}$,* ESTIMATE-HEAVY-FOURIER-COEFFS *(Algorithm 4) outputs estimates $\widetilde{\widehat{f}}(\alpha_C)$ such that, with probability at least $1 - \delta'$, $\big|\widetilde{\widehat{f}}(\alpha_C) - \widehat{f}(\alpha_C)\big| \leq \varepsilon'$.*

Proof. Fix a coset $C \in \mathcal{C}$. Each sample generated by LIST-CORRECTION (Algorithm 3), used to estimate $\widehat{f}(\alpha_C)$, lies in the interval $[-1, 1]$. By Hoeffding's inequality, using $m = \Theta\big((1/(\varepsilon')^2)\log(|\mathcal{C}|/\delta')\big)$ independent samples ensures that the probability that $\big|\widetilde{\widehat{f}}(\alpha_C) - \widehat{f}(\alpha_C)\big| > \varepsilon'$ is at most $\delta'/|\mathcal{C}|$. Applying a union bound over all cosets $C \in \mathcal{C}$, we conclude that all Fourier coefficients are estimated within additive error ε' simultaneously with probability at least $1 - \delta'$. □

Algorithm 3: LIST-CORRECTION

Input: Cosets $\mathcal{C}$ (from FOURIER-SIEVE); oracle access to f; query x; ε', δ'
$\gamma \leftarrow \Theta\big(\log(|\mathcal{C}|/\delta')\big)$;;
sample $y_1, \dots, y_\gamma \sim \mathbb{F}_2^n$
foreach $C \in \mathcal{C}$ **do**
  for $j \leftarrow 1$ **to** γ **do**
    estimate $\mathsf{P}_C f(y_j)$ and $\mathsf{P}_C f(x + y_j)$ within $\pm\varepsilon'$
  end
end
foreach $C \in \mathcal{C}$ **do**
  $\widetilde{\chi}_{\alpha_C}(x) \leftarrow \mathrm{median}_{j\in[\gamma]}\big(\mathrm{sign}(\mathsf{P}_C f(y_j)) \cdot \mathrm{sign}(\mathsf{P}_C f(x + y_j))\big)$
end
return $\{\widetilde{\chi}_{\alpha_C}(x) : C \in \mathcal{C}\}$

Algorithm 4: ESTIMATE-HEAVY-FOURIER-COEFFS

Input: Cosets $\mathcal{C}$; accuracy ε'; confidence δ'
$\lambda \leftarrow \dfrac{100\log(1/(\delta'\theta))}{\varepsilon'^2}$;;
sample $x_1, \dots, x_\lambda \sim \mathbb{F}_2^n$
for $j \leftarrow 1$ **to** λ **do**
  run LIST-CORRECTION on $(\mathcal{C}, x_j, \varepsilon', 1/(20\lambda))$ to get $\{\chi_{\alpha_C}(x_j)\}_{C\in\mathcal{C}}$
end
foreach $C \in \mathcal{C}$ **do**
  $\widetilde{\widehat{f}}(\alpha_C) \leftarrow \frac{1}{\lambda}\sum_{j=1}^{\lambda} f(x_j)\chi_{\alpha_C}(x_j)$
end
return $\{\widetilde{\widehat{f}}(\alpha_C) : C \in \mathcal{C}\}$

4.1 Proof of Theorem 1

We begin by showing that if the heavy Fourier coefficients of a function are estimated with sufficient accuracy, then truncating to the top-s Fourier coefficients yields a good approximation in both ℓ_1 and ℓ_2^2 senses. For each $\alpha \in \mathbb{F}_2^n$, suppose we are given an estimate

$$\widetilde{\widehat{f}}(\alpha) = \widehat{f}(\alpha) + e_\alpha, \qquad \text{where } |e_\alpha| \le \eta.$$

Let $S^\star$ denote the set of indices corresponding to the s largest Fourier coefficients $|\widehat{f}(\alpha)|$, and let T denote the set of indices of the s largest estimated coefficients $|\widetilde{\widehat{f}}(\alpha)|$. Define

$$\Delta_1 := \left|\sum_{\alpha\in S^\star} |\widehat{f}(\alpha)| - \sum_{\alpha\in T} |\widetilde{\widehat{f}}(\alpha)|\right| \qquad \text{and} \qquad \Delta_2 := \left|\sum_{\alpha\in S^\star} \widehat{f}(\alpha)^2 - \sum_{\alpha\in T} \widetilde{\widehat{f}}(\alpha)^2\right|.$$

Claim 1. *We have $\Delta_1 \le 2\eta s$ and $\Delta_2 \le 4\eta\sqrt{s}$.*

Proof. By Parseval's identity, $\sum_{\alpha\in\mathbb{F}_2^n} \widehat{f}(\alpha)^2 = 1$. Let $\alpha_1, \alpha_2, \dots, \alpha_{2^n}$ denote the Fourier coefficients of f, ranked in decreasing order of magnitude, with ties broken arbitrarily. Then the s-th largest Fourier coefficient satisfies

$$|\widehat{f}(\alpha_s)| \;\le\; \frac{1}{\sqrt{s}}. \tag{1}$$

Let $A = S^\star \setminus T$ and $B = T \setminus S^\star$. Since both $S^\star$ and T have cardinality s, we have $|A| = |B| \le s$. For any $\alpha \in A$ and $\beta \in B$, the definition of T implies $|\widetilde{\widehat{f}}(\beta)| \ge |\widetilde{\widehat{f}}(\alpha)|$. Using the bound $|e_\alpha| \le \eta$, we obtain $|\widehat{f}(\beta)| + \eta \ge |\widehat{f}(\alpha)| - \eta$, and hence

$$|\widehat{f}(\alpha)| - |\widehat{f}(\beta)| \le 2\eta. \tag{2}$$

Algorithm 5: DISTRIBUTION-FREE FOURIER SPARSITY TESTER

Input: Oracle f; sample access to D; s, ε, δ
$\theta \leftarrow \varepsilon^2\delta/(20s)$;;
$\mathcal{C} \leftarrow$FOURIER-SIEVE(θ, f)
estimate $\widetilde{\widehat{f}}(\alpha_C)$ for all $C \in \mathcal{C}$ via ESTIMATE-HEAVY-FOURIER-COEFFS
$S \leftarrow$ top-s cosets by $|\widetilde{\widehat{f}}(\alpha_C)|$
if $\sum_{C\in S} \widetilde{\widehat{f}}(\alpha_C)^2 < 1 - \varepsilon^2\delta/2$ **then**
  return REJECT
end
$T \leftarrow 20/\delta$
for $i \leftarrow 1$ **to** T **do**
  sample $z_i \sim D$;;
  obtain $\chi_{\alpha_C}(z_i)$ via LIST-CORRECTION
  $\widetilde{f}(z_i) \leftarrow \sum_{C\in S} \widetilde{\widehat{f}}(\alpha_C)\chi_{\alpha_C}(z_i)$
  if $|f(z_i) - \widetilde{f}(z_i)| > 3\varepsilon/4$ **then**
    return REJECT
  end
end
return ACCEPT

To bound Δ_1, pair each $\alpha \in A$ with a distinct $\beta \in B$. Using (2),

$$\sum_{\alpha\in A} |\widehat{f}(\alpha)| - \sum_{\beta\in B} |\widehat{f}(\beta)| \le \sum_{\text{pairs}} 2\eta \le 2\eta s,$$

which proves the first inequality. For Δ_2, observe that for each paired (α, β),

$$\widehat{f}(\alpha)^2 - \widehat{f}(\beta)^2 = \big(|\widehat{f}(\alpha)| - |\widehat{f}(\beta)|\big)\big(|\widehat{f}(\alpha)| + |\widehat{f}(\beta)|\big).$$

By (1) and (2), $|\widehat{f}(\alpha)| + |\widehat{f}(\beta)| \le 2/\sqrt{s}$, and therefore $\widehat{f}(\alpha)^2 - \widehat{f}(\beta)^2 \le 4\eta/\sqrt{s}$. Summing over at most s such pairs yields $\Delta_2 \le 4\eta\sqrt{s}$. □

We now return to the proof of the main theorem. Throughout the analysis, we condition on the event that all subroutines invoked in Steps 1–4 of Algorithm 5 succeed. Each such event occurs with high probability, and the total failure probability is absorbed at the end via a union bound. By the guarantee of FOURIER-SIEVE (Algorithm 2), all cosets containing Fourier coefficients of magnitude at least θ are identified. Furthermore, by invoking ESTIMATE-HEAVY-FOURIER-COEFFS (Algorithm 4), each such coefficient $\widehat{f}(\alpha_C)$ is estimated to additive accuracy at most $\theta/8$. We set $\eta := \theta/8$. By Claim 1, this choice implies $\Delta_1 \le \varepsilon/4$ and $\Delta_2 \le \varepsilon^2\delta/2$.

Completeness. Assume that $f \in \mathcal{P}_s$, i.e., f is exactly s-Fourier-sparse. In this case, all Fourier mass of f is supported on its top s coefficients, and hence the sum of squares of the top s true Fourier coefficients equals 1. By the bound on Δ_2, the corresponding estimate produced by the algorithm satisfies $\sum_{C\in S} \widetilde{\widehat{f}}(\alpha_C)^2 \ge 1 - \varepsilon^2\delta/2$, and therefore the algorithm does not reject in Step 3.

Let $\widetilde{f}$ denote the hypothesis constructed by the algorithm. By the guarantees of the estimation and truncation procedures, for each sample $z_i \sim D$ we have $\Pr[\,|f(z_i) - \widetilde{f}(z_i)| > \varepsilon/4\,] \le \delta/200$. Since the algorithm draws $T = 20/\delta$ independent samples, a union bound implies that with probability at least $9/10$ all samples satisfy $|f(z_i) - \widetilde{f}(z_i)| \le \varepsilon/4$. Thus the algorithm accepts, establishing completeness.

Soundness. Assume that $\mathrm{dist}_D^{\varepsilon}(f, \mathcal{P}_s) \ge \delta$. We consider two cases.

Case 1: $\mathrm{dist}_U^{\varepsilon}(f, \mathcal{P}_s) > \delta$. Let $g^\star$ be the best s-Fourier-sparse approximation to f under the uniform distribution. Then $\Pr_{x\sim U}[|f(x) - g^\star(x)| > \varepsilon] > \delta$, which implies $\mathbb{E}_{x\sim U}[(f(x) -$

$g^\star(x))^2] \geq \varepsilon^2\delta$. Since $g^\star$ is obtained by truncating f to its top s Fourier coefficients, this means that the ℓ_2^2-mass of the Fourier coefficients outside the top s is at least $\varepsilon^2\delta$, or equivalently that the sum of squares of the top s Fourier coefficients is at most $1 - \varepsilon^2\delta$. Using the bound on Δ_2, the estimate in Step 3 of the algorithm violates the acceptance condition, and the tester rejects.

Case 2: $\mathrm{dist}_U^\varepsilon(f, \mathcal{P}_s) \leq \delta$. Let $g^\star$ again denote the best s-Fourier-sparse approximation to f under the uniform distribution. Since $\mathrm{dist}_D^\varepsilon(f, \mathcal{P}_s) \geq \delta$, we have $\Pr_{x\sim D}[|f(x) - g^\star(x)| > \varepsilon] \geq \delta$. By construction of $\widetilde{f}$ and the bound $\Delta_1 \leq \varepsilon/4$, we have $|\widetilde{f}(x) - g^\star(x)| \leq \varepsilon/4$ for all but a $\delta/200$ fraction of points $x \sim D$. For any point satisfying both conditions, the triangle inequality gives $|f(x) - \widetilde{f}(x)| > 3\varepsilon/4$, and hence each sample causes rejection with probability at least $\delta - \delta/200 \geq \delta/2$. Since the algorithm draws $T = 20/\delta$ samples, the probability that none of them triggers rejection is at most $(1 - \delta/2)^T < 1/10$. Therefore, the algorithm rejects with probability at least $9/10$.

The completeness and soundness guarantees together establish that Algorithm 5 is an (ε, δ)-distribution-free tester for s-Fourier sparsity. Finally, note that each step of the tester succeeds with probability at least $99/100$. By a union bound over all steps, the overall tester succeeds with probability at least $2/3$. The analysis of the query complexity is deferred to the appendix.

5 Conclusion

We gave the first tester for Fourier sparsity in the distribution-free setting: given oracle access to f and samples from an unknown D, it decides whether f is s-Fourier-sparse or far from sparse under D, with complexity polynomial in s, $1/\varepsilon$, $1/\delta$ and independent of the dimension. Unlike junta testing, there is no exponential adaptive-versus-nonadaptive gap here: a nonadaptive tester already suffices. Two things are worth drawing out of this result — the tool that makes it work, and the boundaries it runs into.

The self-corrector as a standalone primitive. The tool is a pointwise self-corrector, and we view it as our main technical contribution. It estimates $g^\star(x)$ at any point, where $g^\star$ is the closest s-sparse function, without ever reconstructing the Fourier representation of $g^\star$. This is exactly what the distribution-free reduction needs, since the tester only has to evaluate the hypothesis at sampled points. The same black-box access should also suit learning with erasures or adversarial label noise, and we hope it finds further use.

Tolerant testing. The first boundary is tolerance, and the obstacle is completeness. For exactly sparse f, Parseval guarantees the top-s coefficients carry all the Fourier mass, which is what our spectral check relies on. If f is merely close to sparse under D, closeness under D says nothing about the spectrum under the uniform distribution, and the check no longer separates yes- from no-instances. The question is also less pressing than it first appears: a tolerant tester would accept functions that known algorithms, which assume uniform or structured distributions, cannot learn, whereas every function our tester accepts is efficiently learnable. We therefore see tolerant testing as important but genuinely open.

Beyond the uniform distribution. The second boundary is the input distribution itself. Adapting the basis to D is natural, but such a basis loses the subspace and character structure our algorithms exploit. Structured distributions are the more immediate target. For product distributions the adapted basis keeps much of that structure, so our "localise, then self-correct" strategy looks adaptable; the work lies in self-correction, which currently uses shift-invariance of the uniform distribution. For bounded-density distributions with $D(x) \leq c\,2^{-n}$ the standard basis still suffices and the spectral estimates transfer with $O(c)$ distortion, so we expect the tester to carry over directly.

References

Andisheh Amrollahi, Amir Zandieh, Michael Kapralov, and Andreas Krause. Efficiently Learning Fourier Sparse Set Functions. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, pages 15094–15103, 2019.

Eric Blais. Improved Bounds for Testing Juntas. In *Proceedings of the International Conference on Randomization and Computation (RANDOM 2008)*, pages 317–330, 2008.

Avrim Blum. Learning a function of r relevant variables. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 731–733. Springer, Berlin, Heidelberg, 2003. ISBN 978-3-540-45167-9. doi: 10.1007/978-3-540-45167-9_54.

Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *J. Comput. Syst. Sci.*, 47(3), 1993.

Swarnalipa Datta, Arijit Ghosh, Chandrima Kayal, Manaswi Paraashar, and Manmatha Roy. Spectral Norm, Economical Sieve, and Linear Invariance Testing of Boolean Functions, 2026. URL `https://arxiv.org/abs/2308.02662`.

V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. New Results for Learning Noisy Parities and Halfspaces. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 563–574, 2006. doi: 10.1109/FOCS.2006.51.

Arijit Ghosh and Manmatha Roy. Price of Parsimony: Complexity of Fourier Sparsity Testing. *https://openreview.net/forum?id=7bCPXHq8xV*, 2025.

O. Goldreich, S. Goldwasser, and D. Ron. Property Testing and Its Connection to Learning and Approximation. *Journal of the ACM*, 45:653–750, 1998.

Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

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.

S. Halevy and E. Kushilevitz. Distribution-Free Property Testing. *SIAM Journal on Computing*, 37 (4):1107–1138, 2007.

Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Nearly optimal sparse fourier transform. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 563–578. Association for Computing Machinery (ACM), 2012. doi: 10.1145/2213977.2213983. URL `https://doi.org/10.1145/2213977.2213983`.

Ishay Haviv and Oded Regev. *The Restricted Isometry Property of Subsampled Fourier Matrices*, pages 163–179. Springer International Publishing, Cham, 2017. ISBN 978-3-319-45282-1. doi: 10.1007/978-3-319-45282-1_11. URL `https://doi.org/10.1007/978-3-319-45282-1_11`.

Elad Hazan, Adam R. Klivans, and Yang Yuan. Hyperparameter Optimization: A Spectral Approach. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=H1zriGeCZ`.

Piotr Indyk, Michael Kapralov, and Eric Price. (Nearly) Sample-Optimal Sparse Fourier Transform. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 36–55, 2014.

M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

Eyal Kushilevitz and Yishay Mansour. Learning Decision Trees Using the Fourier Spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993. An extended abstract has appeared in the proceedings of the 23rd STOC, 1991.

Nathan Linial, Yishay Mansour, and Noam Nisan. Constant Depth Circuits, Fourier Transform, and Learnability. *Journal of the ACM*, 40(3):607–620, 1993. A preliminary version of this paper appeared as an extended abstract in the proceedings of FOCS 1989.

Zhengyang Liu, Xi Chen, Rocco A. Servedio, Ying Sheng, and Jinyu Xie. Distribution-Free Junta Testing. *ACM Transactions on Algorithms*, 15(1):1–23, 2019.

Sahand Negahban and Devavrat Shah. Learning Sparse Boolean Polynomials. In *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 2032–2036, 2012.

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

AUTHORS TO VERIFY (see note above). Learning DNF through generalized fourier representations. In *Proceedings of the 38th Conference on Learning Theory (COLT)*, 2025. Bibliographic details to be verified from the COLT 2025 proceedings.

Elias M. Stein and Rami Shakarchi. *Fourier Analysis: An Introduction*. Princeton University Press, 2003.

Peter Stobbe and Andreas Krause. Learning Fourier Sparse Set Functions. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, pages 1125–1133, 2012.

L. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Grigory Yaroslavtsev and Samson Zhou. Fast Fourier Sparsity Testing. In *Proceedings of the 3rd Symposium on Simplicity in Algorithms (SOSA)*, pages 57–68, 2020.

Appendix

Proof of Lemma 1

Proof. We first establish a concentration result for the coset-hashing process. We show that when the Fourier spectrum of f is projected onto a coset structure of sufficiently large codimension, all heavy Fourier coefficients are mapped to distinct cosets with high probability, while the remaining small-magnitude coefficients are distributed nearly uniformly across the cosets. We believe this concentration result (Claim 2) may be of independent interest in the context of testing and learning Boolean functions.

Notation. Throughout this proof, let $t := \lceil \log(2^{32}/\theta^8) \rceil$ denote the codimension of the random subspace used by FOURIER-SIEVE. For a coset C in the resulting partition $\mathscr{C}$, define the *dominant Fourier coefficient* of C as

$$\alpha_C \;:=\; \arg\max_{\beta \in C} \big|\widehat{f}(\beta)\big|,$$

with ties broken arbitrarily. Define the *heavy set*

$$S \;:=\; \Big\{ \alpha \in \mathbb{F}_2^n \;:\; \big|\widehat{f}(\alpha)\big| \geq \tfrac{\theta^2}{64} \Big\}.$$

By Parseval's identity $\sum_\alpha \widehat{f}(\alpha)^2 = 1$ (our standing normalisation assumption), we have $|S| \leq 2^{12}/\theta^4$.

Claim 2. *Let $\mathscr{C}$ be a randomly permuted coset structure of codimension $t = \lceil \log(2^{32}/\theta^8) \rceil$. Then, except with probability at most $1/100$, the following three properties hold simultaneously.*

(i) ***(Isolation.)*** *Every pair of distinct elements of S is mapped to distinct cosets. Consequently, for every coset $C \in \mathscr{C}$, $\max_{\beta \in C \setminus \{\alpha_C\}} \big|\widehat{f}(\beta)\big| < \theta^2/64$.*

(ii) ***(Light cosets.)*** *For every coset C whose dominant coefficient satisfies $|\widehat{f}(\alpha_C)| < \theta^2/64$, we have $\mathsf{W}_C f < \theta^2/16$.*

(iii) ***(Heavy cosets.)*** *For every coset C whose dominant coefficient satisfies $|\widehat{f}(\alpha_C)| \geq \theta^2/64$, we have $\mathsf{W}_C f \;\leq\; \widehat{f}(\alpha_C)^2 + \theta^4/2^{12}$.*

Proof. We prove each part in turn.

Part (i): Isolation of heavy coefficients. The random partition of codimension t is generated by drawing t independent uniformly random vectors $r_1, \ldots, r_t \in \mathbb{F}_2^n$ and assigning to each frequency $\alpha \in \mathbb{F}_2^n$ the *hash value*

$$h(\alpha) \;:=\; \big(\langle r_1, \alpha\rangle, \ldots, \langle r_t, \alpha\rangle\big) \;\in\; \mathbb{F}_2^t.$$

Two frequencies lie in the same coset precisely when their hash values coincide; that is, the cosets of $\mathscr{C}$ are the level sets $h^{-1}(b)$, $b \in \mathbb{F}_2^t$. For any fixed nonzero $\gamma \in \mathbb{F}_2^n$, each coordinate $\langle r_i, \gamma\rangle$ is an independent uniform bit, so $\Pr[h(\gamma) = \mathbf{0}] = 2^{-t}$. Consequently, two distinct frequencies $\alpha \neq \alpha'$ collide, i.e. $h(\alpha) = h(\alpha')$, which is equivalent to $h(\alpha - \alpha') = \mathbf{0}$, with probability exactly 2^{-t}. A union bound over all $\binom{|S|}{2}$ pairs of elements of S yields a total collision probability of at most

$$\binom{|S|}{2} 2^{-t} \;\leq\; |S|^2\, 2^{-t} \;\leq\; \left(\frac{2^{12}}{\theta^4}\right)^2 \cdot \frac{\theta^8}{2^{32}} \;=\; \frac{2^{24}}{\theta^8} \cdot \frac{\theta^8}{2^{32}} \;=\; \frac{1}{2^8} \;<\; \frac{1}{100}\,.$$

Hence, with probability at least 0.99, every pair of elements in S is isolated into distinct cosets, which immediately gives the stated bound on the second-heaviest coefficient in each coset.

Part (ii): Light cosets. Let C be a coset with $|\widehat{f}(\beta)| < \theta^2/64$ for every $\beta \in C$. For each β, let I_β denote the indicator that β hashes to C. Then

$$\mathbb{E}_{H,b}[\mathsf{W}_C f] \;=\; \sum_\beta \widehat{f}(\beta)^2\, \mathbb{E}_{H,b}[I_\beta] \;=\; 2^{-t} \sum_\beta \widehat{f}(\beta)^2 \;\leq\; 2^{-t} \;\leq\; \frac{\theta^8}{2^{32}}\,.$$

Using pairwise independence of the hash,

$$\mathrm{Var}_{H,b}(\mathsf{W}_C f) \;=\; \sum_{\beta} \widehat{f}(\beta)^4 \,\mathrm{Var}_{H,b}(I_\beta) \;\le\; \sum_{\beta} \widehat{f}(\beta)^4 \,\mathbb{E}_{H,b}[I_\beta] \;=\; 2^{-t} \sum_{\beta} \widehat{f}(\beta)^4 .$$

Since every $\beta \in C$ satisfies $|\widehat{f}(\beta)| < \theta^2/64$, we have $\widehat{f}(\beta)^4 \le (\theta^4/2^{12})\,\widehat{f}(\beta)^2$, and therefore $\sum_\beta \widehat{f}(\beta)^4 \le (\theta^4/2^{12}) \sum_\beta \widehat{f}(\beta)^2 \le \theta^4/2^{12}$. Thus $\mathrm{Var}_{H,b}(\mathsf{W}_C f) \le 2^{-t} \cdot \theta^4/2^{12} \le \theta^{12}/2^{44}$. By Chebyshev's inequality,

$$\Pr_{H,b}\Big[\mathsf{W}_C f - \mathbb{E}_{H,b}[\mathsf{W}_C f] \;\ge\; \tfrac{\theta^2}{16}\Big] \;\le\; \frac{\theta^{12}/2^{44}}{\theta^4/16^2} \;=\; \frac{\theta^8}{2^{36}} .$$

There are at most $2^t \le 2^{32}/\theta^8$ cosets, so a union bound gives a total failure probability of at most $1/16$.

Part (iii): Heavy cosets. Fix a coset C containing a unique element $\alpha_C \in S$ (guaranteed by Part (i) with probability at least 0.99). Write

$$\mathsf{W}_C f \;=\; \widehat{f}(\alpha_C)^2 \;+\; \sum_{\beta \neq \alpha_C} \widehat{f}(\beta)^2 \, I_\beta ,$$

where I_β indicates that β hashes to C. Since $\mathbb{E}_{H,b}[I_\beta] = 2^{-t}$ for each $\beta \neq \alpha_C$,

$$\mathbb{E}_{H,b}\big[\mathsf{W}_C f - \widehat{f}(\alpha_C)^2\big] \;=\; 2^{-t} \sum_{\beta \neq \alpha_C} \widehat{f}(\beta)^2 \;\le\; 2^{-t} \;\le\; \frac{\theta^8}{2^{32}} .$$

By Markov's inequality,

$$\Pr_{H,b}\Big[\mathsf{W}_C f - \widehat{f}(\alpha_C)^2 \;\ge\; \tfrac{\theta^4}{2^{12}}\Big] \;\le\; \frac{\theta^8/2^{32}}{\theta^4/2^{12}} \;=\; \frac{\theta^4}{2^{20}} .$$

Since there are at most $|S| \le 2^{12}/\theta^4$ such cosets, a union bound gives an overall failure probability of at most $2^{12}/\theta^4 \cdot \theta^4/2^{20} = 1/2^8 < 1/100$. □

We now complete the proof of Lemma 1, establishing the guarantees of FOURIER-SIEVE (Algorithm 2). We condition on the event $\mathcal{E}$ that all Fourier-weight estimates produced by the algorithm are accurate to within $\pm\theta^2/4$; standard concentration arguments show that $\mathcal{E}$ holds with probability at least $1 - o(1)$.

(i) All cosets containing heavy coefficients are selected. Let C be a coset whose dominant Fourier coefficient satisfies $|\widehat{f}(\alpha_C)| \ge \theta$. Then $\mathsf{W}_C f \ge \widehat{f}(\alpha_C)^2 \ge \theta^2$. By Claim 2 (iii), $\mathsf{W}_C f \le \widehat{f}(\alpha_C)^2 + \theta^4/2^{12}$, and on the event $\mathcal{E}$ the estimate satisfies $\widetilde{\mathsf{W}}_C f \;\ge\; \mathsf{W}_C f - \theta^2/4 \;\ge\; 3\theta^2/4$, so the coset is retained in the output set $\mathcal{C}$.

(ii) No coset with a small dominant coefficient is selected. Let C be a coset with $|\widehat{f}(\alpha_C)| \le \theta/2$. We distinguish two cases.

Case 1: $|\widehat{f}(\alpha_C)| < \theta^2/64$. By Claim 2 (ii), $\mathsf{W}_C f < \theta^2/16$. On the event $\mathcal{E}$, $\widetilde{\mathsf{W}}_C f \le \mathsf{W}_C f + \theta^2/4 < 3\theta^2/4$, so C is discarded.

Case 2: $\theta^2/64 \le |\widehat{f}(\alpha_C)| \le \theta/2$. By Claim 2 (iii), $\mathsf{W}_C f \le \widehat{f}(\alpha_C)^2 + \theta^4/2^{12} < 5\theta^2/16$. On the event $\mathcal{E}$, $\widetilde{\mathsf{W}}_C f \le \mathsf{W}_C f + \theta^2/4 < 3\theta^2/4$, so C is not selected.

(iii) Concentration of coset weight around the dominant coefficient. This follows directly from Claim 2 (iii).

Finally, estimating the weight of each coset to within $\theta^2/4$ accuracy and with confidence $1 - \mathrm{poly}(\theta)$ requires $\widetilde{O}(1/\theta^4)$ queries to the black-box oracle. □

Proof of Lemma 2

Proof. The proof relies on the fact that a noisy variant of the standard self-correction procedure succeeds simultaneously for all cosets returned by FOURIER-SIEVE. We formalise this in the following claim.

Claim 3. *Let $\mathcal{C}$ be the set of cosets returned by Algorithm 2. Then, for every fixed $x \in \mathbb{F}_2^n$,*

$$\Pr_{y\sim\mathbb{F}_2^n}\Big[\operatorname{sign}\big(\mathsf{P}_C f(x+y)\big)\cdot\operatorname{sign}\big(\mathsf{P}_C f(y)\big) = \chi_{\alpha_C}(x) \ \textit{for all } C\in\mathcal{C}\Big] \ \geq\ \frac{7}{8}\,.$$

Proof. We first establish a concentration lemma showing that the coset projection $\mathsf{P}_C f(z) := \sum_{\beta\in C}\widehat{f}(\beta)\,\chi_\beta(z)$ is close to its dominant term $\mathsf{P}^*_C f(z) := \widehat{f}(\alpha_C)\,\chi_{\alpha_C}(z)$ for a uniformly random z.

Claim 4. *Let $f:\mathbb{F}_2^n\to\mathbb{R}$, and let $C\in\mathscr{C}$ be a coset with dominant Fourier coefficient α_C satisfying*

$$\text{(a)}\ \sum_{\beta\in C\setminus\{\alpha_C\}}\widehat{f}(\beta)^2 < \frac{\theta^4}{2^{12}},\qquad \text{(b)}\ |\widehat{f}(\alpha_C)| > \frac{\theta}{2},\qquad \text{(c)}\ \max_{\beta\in C\setminus\{\alpha_C\}}|\widehat{f}(\beta)| < \frac{\theta^2}{64}\,.$$

Then, for every $\tau>0$,

$$\Pr_{z\sim\mathbb{F}_2^n}\Big[\,\big|\mathsf{P}_C f(z)-\mathsf{P}^*_C f(z)\big| \ \geq\ \frac{\theta^2}{64}+\tau\Big] \ \leq\ \frac{\theta^4}{2^{12}\,\tau^2}\,.$$

Proof. Write $C' := C\setminus\{\alpha_C\}$, so that $\mathsf{P}_C f(z)-\mathsf{P}^*_C f(z) = \sum_{\beta\in C'}\widehat{f}(\beta)\,\chi_\beta(z)$.

Expectation. We distinguish two cases depending on whether $\mathbf{0}$ belongs to C'.

Case I ($\mathbf{0}\notin C'$, i.e. either $\mathbf{0}\notin C$ or $\mathbf{0}=\alpha_C$). Every $\beta\in C'$ is nonzero, so $\mathbb{E}_z[\chi_\beta(z)]=0$, giving

$$\mathbb{E}_z\big[\mathsf{P}_C f(z)-\mathsf{P}^*_C f(z)\big] \ =\ \sum_{\beta\in C'}\widehat{f}(\beta)\cdot 0 \ =\ 0\,.$$

Case II ($\mathbf{0}\in C'$, i.e. $\mathbf{0}\in C$ and $\mathbf{0}\neq\alpha_C$). Since $\chi_{\mathbf{0}}\equiv 1$ and all other characters in C' have zero expectation,

$$\mathbb{E}_z\big[\mathsf{P}_C f(z)-\mathsf{P}^*_C f(z)\big] \ =\ \widehat{f}(\mathbf{0})\,.$$

Because $\mathbf{0}\neq\alpha_C$ and $\mathbf{0}\in C'$, the element $\mathbf{0}$ is not the dominant coefficient of C, so condition (c) gives $|\widehat{f}(\mathbf{0})|<\theta^2/64$.

Variance. For a uniform z, the characters $\{\chi_\beta:\beta\in C'\}$ are pairwise independent random variables taking values in $\{-1,+1\}$. Hence

$$\operatorname{Var}_z\Big[\sum_{\beta\in C'}\widehat{f}(\beta)\,\chi_\beta(z)\Big] \ =\ \sum_{\beta\in C'}\widehat{f}(\beta)^2 \ \leq\ \frac{\theta^4}{2^{12}}\,,$$

where the inequality is condition (a).

Conclusion. Combining the expectation and variance bounds via Chebyshev's inequality,

$$\Pr_z\Big[\,\big|\mathsf{P}_C f(z)-\mathsf{P}^*_C f(z)\big| \geq \tfrac{\theta^2}{64}+\tau\Big] \ \leq\ \frac{\theta^4}{2^{12}\,\tau^2}\,. \qquad \square$$

We now return to the proof of Claim 3. Setting $\tau=\theta/8$ in Claim 4 gives

$$\Pr_{z\sim\mathbb{F}_2^n}\Big[\,\big|\mathsf{P}_C f(z)-\mathsf{P}^*_C f(z)\big| > \tfrac{\theta}{4}\Big] \ \leq\ \frac{\theta^2}{64}\,, \tag{1}$$

for each surviving coset $C\in\mathcal{C}$.

Next, we bound $|\mathcal{C}|$. Every surviving coset contains a dominant Fourier coefficient of magnitude at least $\theta/2$. By Parseval's identity ($\sum_\alpha \widehat{f}(\alpha)^2 = 1$), the number of Fourier coefficients of magnitude at least $\theta/2$ is at most $1/(\theta/2)^2 = 4/\theta^2$. Hence $|\mathcal{C}| \le 4/\theta^2$.

Applying a union bound over all $C \in \mathcal{C}$ to (1),

$$\Pr_{z\sim\mathbb{F}_2^n}\Big[\,\exists\, C \in \mathcal{C} :\ \big|\mathsf{P}_C f(z) - \mathsf{P}^*_C f(z)\big| \ge \tfrac{\theta}{4}\,\Big] \;\le\; \frac{4}{\theta^2}\cdot\frac{\theta^2}{64} \;=\; \frac{1}{16}\,.$$

That is, with probability at least $15/16$ over a uniform z, every surviving coset simultaneously satisfies $|\mathsf{P}_C f(z) - \mathsf{P}^*_C f(z)| < \theta/4$.

Now fix any $x \in \mathbb{F}_2^n$. Since y is uniform over $\mathbb{F}_2^n$, both y and $x+y$ are individually uniform. On the event that both y and $x+y$ yield small deviations for every $C \in \mathcal{C}$ (which holds with probability at least $1 - 2\cdot 1/16 = 7/8$), the signs of $\mathsf{P}_C f(y)$ and $\mathsf{P}_C f(x+y)$ are determined by their respective dominant terms. Since $|\widehat{f}(\alpha_C)| \ge \theta/2$ and the deviation is less than $\theta/4$, we have $\operatorname{sign}(\mathsf{P}_C f(y)) = \operatorname{sign}(\widehat{f}(\alpha_C)\,\chi_{\alpha_C}(y))$ and similarly for $x+y$. Hence

$$\begin{aligned}\operatorname{sign}\big(\mathsf{P}_C f(y)\big)\cdot\operatorname{sign}\big(\mathsf{P}_C f(x+y)\big) \;&=\; \operatorname{sign}\big(\widehat{f}(\alpha_C)\,\chi_{\alpha_C}(y)\big)\cdot\operatorname{sign}\big(\widehat{f}(\alpha_C)\,\chi_{\alpha_C}(x+y)\big)\\ &=\; \chi_{\alpha_C}(y)\cdot\chi_{\alpha_C}(x+y) \;=\; \chi_{\alpha_C}(x)\,,\end{aligned}$$

simultaneously for all $C \in \mathcal{C}$. This establishes the claim.

Remark. The algorithm does not have access to the exact projection $\mathsf{P}_C f(z)$; it can only estimate it. However, if the estimate $\widetilde{\mathsf{P}}_C f(z)$ lies within $\pm\theta/8$ of the true value, the gap between the dominant-term magnitude ($\ge \theta/2$) and the total error (deviation $< \theta/4$ plus estimation error $\le \theta/8 < \theta/4$) still leaves the sign determined correctly. □

We now complete the proof of Lemma 2. By Claim 3, each independent trial of the self-correction step succeeds (simultaneously for all $C \in \mathcal{C}$) with probability at least $7/8$. Applying the standard median-of-means amplification with $O(\log(1/\delta'))$ independent repetitions boosts the success probability to at least $1-\delta'$. Each repetition requires $\widetilde{O}(1/\theta^2)$ oracle queries (to estimate each coset projection to accuracy $\theta/8$), yielding an overall query complexity of $\widetilde{O}(\log(1/\delta')/\theta^2)$. □

Query and Time Complexity

Recall that the threshold is set to $\theta = \varepsilon^2\delta/(20s)$. We analyse each step of Algorithm 5.

Step 1 (FOURIER-SIEVE). By Lemma 1, this step uses $\widetilde{O}(1/\theta^4)$ oracle queries.

Step 2 (ESTIMATE-HEAVY-FOURIER-COEFFS). For each candidate coset the list-correction procedure is invoked with estimation accuracy $\theta/8$ and confidence $1-\mathrm{poly}(\theta)$. By Lemma 2, each invocation requires $\widetilde{O}(1/\theta^2)$ oracle queries. Since the procedure is applied to $\widetilde{O}(1/\theta^2)$ candidate cosets overall, the total query complexity of this step is $\widetilde{O}(1/\theta^4)$.

Step 4 (Distribution-free consistency check). The list-correction algorithm is invoked $O(1/\delta)$ times (once per sample from D), each invocation requiring $\widetilde{O}(\log(1/\delta)/\theta^2)$ oracle queries. The total cost of this step is $\widetilde{O}(1/(\delta\,\theta^2))$.

Overall. The dominant contribution comes from Steps 1 and 2. Substituting $\theta = \varepsilon^2\delta/(20s)$, we obtain

$$\widetilde{O}\!\left(\frac{1}{\theta^4}\right) \;=\; \widetilde{O}\!\left(\frac{(20s)^4}{\varepsilon^8\,\delta^4}\right) \;=\; \widetilde{O}\!\left(\frac{s^4}{\varepsilon^8\,\delta^4}\right),$$

which matches the bound stated in Theorem 1. The Step 4 contribution is $\widetilde{O}(s^2/(\varepsilon^4\,\delta^3))$, which is dominated by the above for $s \ge 1$.

Time complexity. The running time is dominated by the same two steps. FOURIER-SIEVE estimates the spectral weight of each of the $2^t \le 2^{32}/\theta^8$ cosets in the partition; each estimate is an average over

$\widetilde{O}(1/\theta^4)$ oracle evaluations, and every evaluation of the coset projection takes time linear in $t = \widetilde{O}(1)$, so this step runs in time $\widetilde{O}(1/\theta^4)$. LIST-CORRECTION and ESTIMATE-HEAVY-FOURIER-COEFFS maintain $|\mathcal{C}| = \widetilde{O}(1/\theta^2)$ candidate cosets and spend $\widetilde{O}(1/\theta^2)$ time per coset per evaluation, again totalling $\widetilde{O}(1/\theta^4)$. Hence the overall running time is

$$\widetilde{O}\left(\frac{1}{\theta^4}\right) \;=\; \widetilde{O}\left(\frac{s^4}{\varepsilon^8\,\delta^4}\right),$$

matching the query complexity up to logarithmic factors. In particular, the entire tester runs in time polynomial in s, $1/\varepsilon$, and $1/\delta$, and independent of the ambient dimension n beyond the $\widetilde{O}(1)$ cost of a single oracle evaluation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: It is theoretical and the work contains all the proof to substantiate the result.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed it the last section of the paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Yes, each theorem and lemma is accompanied by a complete set of assumptions and detailed, correct proofs

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: No experimental results were included in the paper; hence, this question is not applicable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: Our work does not involve data collection or code execution; hence, this question is not applicable

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification: No experimental setup was used in this work; therefore, this item is not applicable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: As no experiments were conducted, questions of statistical significance or error bars do not arise.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: This work does not involve any experiments requiring computational resources; thus, the question is not applicable

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our work conforms to it in all respects, wherever applicable.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is a theoretical paper, and to the best of our understanding, it does not have direct societal implications

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: As no data or models were released in this work, this question does not apply.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: Although no experimental assets were used, all prior work has been appropriately cited to the best of our ability.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: While our work does not introduce experimental assets, we have clearly defined the theoretical model proposed.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No experiments involving human subjects or crowdsourcing were conducted;thus, this question is not applicable

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects were involved in this study; therefore, IRB approval was not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We have used large language models solely for editorial purposes and not in any aspect of the methodology or technical contribution.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.