

Testing Sparse Functions over the Reals

Vipul Arora 

School of Computing, National University of Singapore, Singapore

Arnab Bhattacharyya 

University of Warwick, UK

Philips George John 

CQT & CNRS@CREATE, National University of Singapore, Singapore

Sayantan Sen 

Centre for Quantum Technologies, National University of Singapore, Singapore

Abstract

Over the last three decades, function testing has been extensively studied over Boolean, finite fields, and discrete settings. However, to encode the real-world applications more succinctly, function testing over the reals (where the domain and range, both are reals) is of prime importance. Recently, there have been some works in the direction of testing for algebraic representations of such functions: the work by Fleming and Yoshida (ITCS 20), Arora, Kelman, and Meir (SOSA 25) on *linearity testing* and the work of Arora, Bhattacharyya, Fleming, Kelman, and Yoshida (SODA 23) for testing *low-degree polynomials*. Our work follows the same avenue, wherein we study three well-studied sparse representations of functions, over the reals, namely (i) *k-linearity*, (ii) *k-sparse, low-degree polynomials*, and (iii) *k-juntas*.

In this setting, given approximate query access to some $f : \mathbb{R}^n \rightarrow \mathbb{R}$, we want to decide if the function satisfies some property of interest, or if it is *far* from all functions that satisfy the property. Here, the distance is measured in the ℓ_1 -metric, under the assumption that we are drawing samples from the Standard Gaussian distribution. We present efficient testers and $\Omega(k)$ lower bounds for testing each of these three properties.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Streaming, sublinear and near linear time algorithms

Keywords and phrases Property testing, sparsity, linearity, low-degree polynomials, juntas, computation over reals

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.14

Category Track A: Algorithms, Complexity and Games

Related Version *Full version (with complete proofs)*: <https://arxiv.org/abs/2603.28061> [6]

Funding *Vipul Arora*: This work was done in part while the author was visiting the Simons Institute for the Theory of Computing.

Arnab Bhattacharyya: Research supported by a start-up grant at the University of Warwick.

Philips George John: Research supported by the NRF, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. This work was also done in part while supported by the CQT++ Core Research Funding Grant from Prof. Divesh Aggarwal.

Sayantan Sen: Research supported by the NRF Investigatorship award (NRF-NRFI10-2024-0006) and CQT Young Researcher Career Development Grant (25-YRCDDG-SS).

1 Introduction

Property testing [24, 71, 48] is a rigorous framework for studying the global properties of large datasets by accessing only a few entries of it. In particular, given query access to an unknown “huge object”, the goal of property testing is to verify some property of the object



© Vipul Arora, Arnab Bhattacharyya, Philips George John, and Sayantan Sen; licensed under Creative Commons License CC-BY 4.0
53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 14; pp. 14:1–14:26



Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



by only inspecting a small portion of it. Formally, we can define property testing of functions, or *function testing*, by considering the “huge object” as a function f over an underlying domain, which we can *query* at a small number of points in order to verify a property of the function f in an approximate sense. For example, consider the problem of *linearity testing* of functions $f : \mathbb{F}^n \rightarrow \mathbb{F}$, where $\mathbb{F}$ is a finite field. We are given access to a query oracle for f ; i.e., on input $\mathbf{x}$, the oracle returns $f(\mathbf{x})$. The goal is to distinguish with high probability between two cases, viz. (i) f is a linear function, or (ii) f is “far” from all linear functions, while performing as few queries to the oracle as possible.

The field of property testing was initiated in the seminal work of [24, 25], who studied the problem of *self-testing* of programs, where the goal was to understand the correctness of a program by verifying its outputs on a set of correlated input data. A fundamental problem they studied is that of testing whether f is *linear* (or generally, a homomorphism of abelian groups) or ε -far from linearity, with the notion of ε -farness of $f : \mathbb{F}^n \rightarrow \mathbb{F}$ from a function class $\mathcal{P}$ defined as: $\text{dist}_{\text{Unif}(\mathbb{F}^n), \ell_0}(f, \mathcal{P}) = \inf_{g \in \mathcal{P}} \Pr_{\mathbf{x} \sim \text{Unif}(\mathbb{F}^n)}[f(\mathbf{x}) \neq g(\mathbf{x})] \geq \varepsilon$, i.e., for any function g that satisfies property $\mathcal{P}$, f disagrees with g on at least an ε -fraction of the inputs $\mathbf{x}$ (uniformly drawn from $\mathbb{F}^n$). In the context of linearity testing, $\mathcal{P}$ is the class of linear functions/homomorphisms (the ℓ_0 -distance definition given above can be easily generalized in terms of function domain and range, input distribution etc). They showed a constant query (independent of $|\mathbb{F}|$ and n) linearity tester, known as the *BLR tester*.

Over the last three decades, property testing has been extensively studied in many settings, such as when the unknown object is a function, a graph, or a probability distribution, with many natural connections to real-world problems, e.g., in the context of probabilistically checkable proofs [3, 4, 67, 34], PAC learning [48], program checking [70, 71], approximation algorithms [48, 38] and many more.

In particular, starting from the very first work that initiated the field of property testing [24], over the last three decades, function property testing has been extensively studied in several settings, culminating in a wide array of tools and techniques. This includes settings where the function is defined over finite fields [1, 58, 57, 45, 14, 13, 51, 73], over hyper-grids [35, 21, 28, 9], etc. One may see the books [50, 15] and the surveys [41, 69, 68] for detailed references.

Our problem setting. Suppose we are given query access to an unknown function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, and our goal is to distinguish whether (i) f satisfies some property $\mathcal{P}$, or (ii) f is ε -far from all functions that satisfy $\mathcal{P}$. We define the notion of ε -farness over the reals in the following way: we fix a reference distribution $\mathcal{D}$ on $\mathbb{R}^n$, and we say f is ε -far from $\mathcal{P}$ if the following holds¹: $\text{dist}_{\mathcal{D}, \ell_1}(f, \mathcal{P}) \triangleq \inf_{g \in \mathcal{P}} \{\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [|f(\mathbf{x}) - g(\mathbf{x})|]\} \geq \varepsilon$.

Studying function testing over the reals often requires new techniques compared to testing on finite domains. A common reference distribution on $\mathbb{R}^n$ is the standard n -dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}, I_n)$, which is approximately a uniform distribution on an ℓ_2 -sphere of radius $\sqrt{n}$. And the notion of exact queries (oracle giving $f(\mathbf{x})$) is not realistic over the reals, since one may require infinitely many bits to represent the exact function value. Although the problem of function testing over the reals has several practical motivations, there have been comparatively few works in this setting till now (compared to testing over finite domains) such as testing surface areas [64, 59], testing halfspaces [61, 62, 63], linear separators [8], high-dimensional convexity [29], linear k -juntas [32] etc.

¹ The notion of ℓ_1 -distance is more appropriate for real-valued functions, rather than the commonly used Hamming (ℓ_0) distance, especially in the *approximate query setting* which we define later. For example, consider the case if $g = f + \varepsilon$ for some $\varepsilon > 0$, then $\text{dist}_{\mathcal{D}, \ell_1}(f, g) = \varepsilon$, whereas $\text{dist}_{\mathcal{D}, \ell_0}(f, g) = 1$.

Interestingly, these works mostly focus on the setting when f is Boolean valued, i.e., $f : \mathbb{R}^n \rightarrow \{\pm 1\}$. The setting where the range of f is real has also been studied, e.g., in the work of [12], wherein they test properties of functions, defined over finite hyper-grids, with respect to L_p -distances. In [16], functions over the hypercube, i.e., $f : \{0, 1\}^n \rightarrow \mathbb{R}$ are studied for monotonicity. Recently, [39, 40] studied L^p testing of monotonicity of Lipschitz functions $f : [0, 1]^n \rightarrow \mathbb{R}$.

[44] studied the problem of linearity testing for real-valued real-domain functions in full generality, i.e., for functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Later, [5] studied the problem of testing low-degree polynomials in this regime, followed by the work of [7], which improved these results by achieving query-optimality with respect to the proximity parameter ε . In this work, we expand the landscape of testing various fundamental notions of sparsity to the general $f : \mathbb{R}^n \rightarrow \mathbb{R}$ regime.

A recent work of [10] studies a problem similar to sparsity testing of low degree polynomials, in a sample-based access model. However, it assumes a promise that the unknown function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a *multi-linear* polynomial to begin with, and needs a large gap for the sparsity parameter between the YES and NO cases.

When the function f to be tested is real valued, two different models can be considered with respect to the *query accuracy*. In the first model, known as the *arbitrary precision arithmetic* or the *exact testing* model, we assume that the oracle representing f can give the exact value of f at any point of query $\mathbf{x}$. However, this model is unrealistic from an implementation, or even from a classical complexity-theoretic, viewpoint. So, we instead consider a *finite precision arithmetic* or *approximate testing* model:

► **Definition 1.1** (*η -approximate query*). *The oracle, when queried for $f(\mathbf{x})$, outputs $\tilde{f}(\mathbf{x})$ such that $|\tilde{f}(\mathbf{x}) - f(\mathbf{x})| \leq \eta$, for some small parameter $\eta \in (0, 1)$, for every query point $\mathbf{x}$.*

It is clear that for any property $\mathcal{P}$, any tester for $\mathcal{P}$ in the approximate model will also work in the exact model. The notion of approximate testing was studied in earlier works [46, 2, 37]. In this context, η can be thought of as the resolution limit of the computational machine, i.e., if the machine offers some α bits of precision, then $\eta = 2^{-\alpha}$. η can also be thought of the noise reliability threshold of a channel communicating reals, i.e., if some information $a \in \mathbb{R}$ is transmitted on a channel with a reliability threshold of η , then the received observable $\tilde{a} \in \mathbb{R}$ satisfies $|a - \tilde{a}| \leq \eta$. When $\eta = 0$, this is the exact query model.

All our testers in this work are analyzed in the approximate query model for $\eta > 0$. Our results also require appropriate upper bounds on η , which will be stated in each context. But we use the exact query model for proving the lower bounds, since lower bounds for $\eta = 0$ also hold for any η -approximate query model for $\eta > 0$.

Similar to the context of access to the unknown functions, there are some variations in terms of the error profile of the testing algorithms. A tester is said to have *two-sided error* if it can err in both the cases: when $f \in \mathcal{P}$, and when f is ε -far from $\mathcal{P}$. This is in contrast with *one-sided error* testers, which always decide correctly when $f \in \mathcal{P}$, and can only err when f is ε -far from $\mathcal{P}$. Likewise, a tester is said to be *adaptive* if it performs queries based on the answers it obtained for the previous queries. On the other hand, a *non-adaptive* tester performs all its queries together in a single round. A tester over a finite domain is said to be *local* if the number of queries performed by it is independent of the domain size.

In this work, we design testers in the approximate query model for three sparse function representations: k -linear, k -sparse low-degree polynomials, and k -juntas. Our testers for k -linearity, and k -sparse low-degree polynomials have two-sided error, whereas our k -junta tester has one-sided error. Our testers for k -linear functions and k -juntas are adaptive, while

our tester for k -sparse low-degree polynomials is non-adaptive. Importantly, all our testers are *local*: the number of queries performed by our testers is independent of the domain dimension n , and depends only on the sparsity parameter k and the proximity parameter ε (as well as the total degree d in the k -sparse low-degree polynomial case).

These properties are well-studied in the Boolean and finite fields regime, and this work strives to do the same over continuous domains. We believe the new techniques we have developed to design these testers will be of independent interest.

Boundedness. We assume that the unknown functions are bounded inside an ℓ_2 ball B (centered at the origin $\mathbf{0}$) of suitable radius (typically $O(\sqrt{n})$), i.e., for every $\mathbf{x} \in B(\mathbf{0}, O(\sqrt{n}))$, $|f(\mathbf{x})| \leq C\|\mathbf{x}\|_2$ for some fixed constant C (so $|f(\mathbf{x})| \leq C\sqrt{n}$ in $B(\mathbf{0}, 2\sqrt{n})$)². Note that testing only via bounded queries is impossible without such an assumption. For example, consider an arbitrary “good” function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (k -linear, k -sparse polynomial, or a k -junta), and choose a suitably small region R at random (e.g. by choosing $\mathbf{y}$ uniformly from $B(\mathbf{0}, \sqrt{n})$ and setting R to be a small radius ball around $\mathbf{y}$), where $0 < \mu_n(R) \ll 1/n^c$ for any $c > 0$ under the $\mathcal{N}(\mathbf{0}, I_n)$ Gaussian measure on $\mathbb{R}^n$. Further, subdivide R equally into R_1 and R_2 , and for some $A \geq 2\varepsilon/\mu_n(R)$, define a function $f' : \mathbb{R}^n \rightarrow \mathbb{R}$ which is identical to f outside R ; while $f'(\mathbf{x}) = A$ for $\mathbf{x} \in R_1$, $f'(\mathbf{x}) = -A$ for $\mathbf{x} \in R_2$. As $A > 0$ can be arbitrarily large, f' can be moved arbitrarily far from the required property (in ℓ_1 distance over the Gaussian measure). However, since $\mu_n(R)$ is arbitrarily small, no algorithm using bounded queries can distinguish between f and f' .

Choice of Reference Distribution. In this work (for k -linear, and k -sparse low-degree testers), we have used an anti-concentration result of Glazer and Mikulincer (Theorem 4.6) which uses the Carbery-Wright anti-concentration inequality (Theorem 4.5), but with a more usable expression for the variance (in terms of the polynomial coefficients). This form assumes that the distribution (on the variables) is log-concave, isotropic, and non-discrete. We have chosen $\mathcal{N}(\mathbf{0}, I_n)$ as representative of such a distribution, but we can also take $P^{\otimes n}$ for any continuous, log-concave distribution P on $\mathbb{R}$. A probabilistic upper bound for the Hankel matrix eigenvalues, which we use in the analysis of the k -sparsity tester, also assumes the isotropic Gaussian distribution (Theorem 2.12).

2 Our results

We focus on testing (i) k -linearity, (ii) k -sparse, low-degree polynomials, and (iii) k -juntas, and assume that our reference distribution $\mathcal{D}$ is $\mathcal{N}(\mathbf{0}, I_n)$, unless otherwise stated.

2.1 Testing k -linear functions

► **Definition 2.1** (k -linearity). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, and $k \in \mathbb{N}$. f is k -linear if there exists a set $S \subseteq [n] : |S| \leq k$, and there exist coefficients $\{c_i \in \mathbb{R}\}_{i \in S}$, such that, $f \equiv \sum_{i \in S} c_i x_i$.*

This problem has been extensively studied over finite domains with exact query access. [43] designed the first tester for k -linearity with query complexity $\tilde{O}(k^2)$ by studying the related problem of testing k -juntas. Later, [18] improved the bound for testing k -juntas to

² This boundedness notion corresponds to the bounds we get for low-degree polynomials over compact domains, e.g., if $f(x) = a_1 x + \dots + a_d x^d$ is a polynomial in x , then $|f(x)| \leq \sum_{i=1}^d |a_i| R^i$ for any $x \in [-R, R]$, for all $R \geq 0$.

$O(k \log k)$ queries. Using the BLR test [24], along with this result gives a tester for k -linearity with $O(k \log k)$ query complexity, as done by [26], who presented an optimal, two-sided error, non-adaptive algorithm for this problem over Boolean domains. The first lower bounds for this problem were presented by [43], proving $\Omega(\sqrt{k})$ non-adaptive, and $\Omega(\log k)$ adaptive queries are necessary for testing k -linearity. These were first improved by [49], to $\Omega(k)$ non-adaptive, and $\Omega(\sqrt{k})$ adaptive query lower bounds. This was further improved by [19, 20] who proved $\Omega(k)$ adaptive query lower bound. Interestingly, the $\Omega(k)$ adaptive query lower bound by [19] was proved by showing a novel connection to communication complexity, which we will also use later to prove our lower bounds.

► **Theorem 2.2** (Informal, see Theorem 5.1). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function bounded on the ball $B(\mathbf{0}, 2\sqrt{n})$, given via an η -approximate query oracle. Let $k \in \mathbb{N}$, and $\varepsilon, \eta \in (0, 2/3)$ be parameters such that $\eta < \min \left\{ \varepsilon, O \left(\min_{i \in [n]: f(\mathbf{e}_i) \neq 0} \frac{|f(\mathbf{e}_i)|}{(nk)^2} \right) \right\}$, where $\mathbf{e}_i$ denotes the i^{th} standard basis vector. There exists an $\tilde{O}(k \log k + 1/\varepsilon)$ -query tester (Algorithm 1) that distinguishes whether f is k -linear, or is ε -far from all k -linear functions, with probability at least $2/3$.*

► **Remark 2.3.** Note the necessity of sufficiently good machine precision (small η) to successfully test k -linearity. This is inevitable because if a linear function f has k “large” coefficients $a_1, \dots, a_k$ ($\geq \eta$), and a small coefficient a_{k+1} (say $< \eta/n^2$), we cannot distinguish (up to ℓ_1 distance $\varepsilon \approx \eta$) between f and the k -linear function $\sum_{i=1}^k a_i x_i$ using η -approximate queries and Algorithm 1. The tester (specifically, Algorithm 2) uses the absolute difference between function values, which will be at most $|a_{k+1}(x_{k+1})| \ll 2\eta$. Similar restrictions apply to testing k -sparse polynomials, and k -juntas.

In another sense, a qualitative dependence of η on the function structure is inevitable. Otherwise, if given an η -approximate oracle $\tilde{f}$ to f , $\tilde{f}' \triangleq \tilde{f}/2^n$ (which can be computed in $O(n)$ time given $\tilde{f}$, assuming a variable-length binary floating-point representation) would be an $\eta/2^n$ -approximate oracle to f . But such a scaling would not help with our results, since the coefficients of f , the non-zero influences, etc. would be similarly scaled-down.

2.2 Testing k -sparse low degree polynomials

► **Definition 2.4** (k -sparsity). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, and $k \in \mathbb{N}$ be a parameter. A monomial $\mathbf{x}^\alpha \triangleq \prod_{i=1}^n x_i^{\alpha_i}$, for any $\mathbf{x} \triangleq (x_1, \dots, x_n) \in \mathbb{R}^n$, and $\alpha \triangleq (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ has total degree $\|\alpha\|_1 \triangleq \sum_{i=1}^n \alpha_i$. A polynomial $f(x_1, \dots, x_n) = \sum_{i=1}^\ell a_i \mathbf{x}^{\mathbf{d}_i}$, with $a_i \neq 0$ and $\mathbf{d}_i \in \mathbb{N}^n$, $\forall i \in [\ell]$, of total degree $\max_{i \in [\ell]} \{\|\mathbf{d}_i\|_1\}$, is said to be a k -sparse polynomial if $\ell \leq k$.*

Grigorescu *et al.* [52] solved this problem for polynomials on finite fields, i.e., $f : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$, for large enough q , using the machinery of Hankel matrices associated with polynomials:

► **Definition 2.5** (Hankel Matrix for polynomials [52, 11]). *Consider any $\mathbf{u} \triangleq (u_1, \dots, u_n) \in \mathbb{R}^n$, and define $\mathbf{u}^i \triangleq (u_1^i, \dots, u_n^i) \in \mathbb{R}^n$, for every $i \in \mathbb{N}$. For a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $t \in \mathbb{N}$, define the t -dimensional Hankel matrix associated with f at $\mathbf{u}$ to be the following:*

$$H_t(f, \mathbf{u}) \triangleq \begin{pmatrix} f(\mathbf{u}^0) & f(\mathbf{u}^1) & \dots & f(\mathbf{u}^{t-1}) \\ f(\mathbf{u}^1) & f(\mathbf{u}^2) & \dots & f(\mathbf{u}^t) \\ \vdots & \vdots & \ddots & \vdots \\ f(\mathbf{u}^{t-1}) & f(\mathbf{u}^t) & \dots & f(\mathbf{u}^{2t-2}) \end{pmatrix} \in \mathbb{R}^{t \times t}.$$

We note an observation about Hankel matrices. It follows essentially the same argument as in [11, 52], since the decomposition they use over finite domains also works over the reals.

► **Observation 2.6** (Generalization of [11, Section 4], and [52, Lemma 4]). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be an exactly k -sparse polynomial over the reals, i.e., $f(\mathbf{x}) = \sum_{i=1}^k a_i M_i(\mathbf{x})$, where $a_1, \dots, a_k \in \mathbb{R} \setminus \{0\}$, and $M_1, \dots, M_k$ are the monomials of f . Then for all $\ell + 1 \leq k$,*

$$\det(H_{\ell+1}(f, \mathbf{x})) = \sum_{\substack{S \subseteq [k] \\ |S| = \ell+1}} \prod_{i \in S} a_i \prod_{\substack{i, j \in S \\ i < j}} (M_j(\mathbf{x}) - M_i(\mathbf{x}))^2,$$

is a non-zero polynomial of degree $\leq 2 \binom{\ell+1}{2} \deg(f)$, while $\forall \ell + 1 > k$, $\det(H_{\ell+1}(f, \mathbf{x})) \equiv 0$.

From Ben-Or and Tiwari [11]’s Observation 2.6, Grigorescu *et al.* [52] designed a tester with a query complexity of $O(k)$ (independent of d), assuming f to be an individual-degree- d polynomial (note that all functions $\mathbb{F}_q^n \rightarrow \mathbb{F}_q$ are polynomials of individual degree $\leq q$).

Note that if we have exact query access to f , the Hankel matrix $H_t(f, \mathbf{u})$ can be computed using only $2t - 1$ queries to f for any point $\mathbf{u} \in \mathbb{R}^n$. For the problem of testing a *polynomial* $f : \mathbb{R}^n \rightarrow \mathbb{R}$ for sparsity, given exact query access, the tester of [52] works as it is. In fact, it achieves perfect soundness and completeness (proved in Lemma 6.4). However, the assumption of an exact query oracle is not realistic. Moreover, for general functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the promise of polynomiality no longer holds, and therefore a preliminary step is needed to eliminate functions that are far from being low-degree polynomials. [5] designed a *local*, approximate query, low degree tester with a query complexity of $O(d^5)$, for polynomials over $\mathbb{R}^n$ with *total degree* d , which we will use in our work. Thus, we restrict our attention to polynomials of *total degree* at most d .

► **Theorem 2.7** (Informal, see Theorem 6.1). *Let $k, d \in \mathbb{N}$, $\varepsilon, \eta \in (0, 1)$ be parameters such that $\eta < \min\{\varepsilon, 1/2^{2^n}\}$, and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be bounded in $B(\mathbf{0}, 2d\sqrt{n})$, given via an η -approximate query access. Then there exists an $\tilde{O}(d^5 + \frac{d^2}{\varepsilon} + dk^3)$ -query tester (Algorithm 4), that distinguishes whether f is a k -sparse, degree- d polynomial, or is ε -far from all such polynomials, with probability $\geq 2/3$.*

► **Remark 2.8.** We note that TEST- k -SPARSE (Algorithm 4) also works for testing k -linear functions (by setting the degree $d = 1$). However, we make the following remarks:

- (i) The query complexity of TEST- k -SPARSE ($\tilde{O}(d^5 + \frac{d^2}{\varepsilon} + dk^3)$), invoked for k -linearity testing, would be $\tilde{O}(k^3 + \frac{1}{\varepsilon})$, which is worse than that of TEST- k -LINEAR.
- (ii) The restriction on the approximation parameter η is $1/2^{2^n}$ for TEST- k -SPARSE, as compared to $1/(nk)^2$ for TEST- k -LINEAR. Thus, for a wider range of parameters when η is not too small, invoking TEST- k -LINEAR is better, compared to TEST- k -SPARSE.
- (iii) TEST- k -LINEAR is an adaptive tester, while TEST- k -SPARSE is non-adaptive.

2.3 Testing k -juntas

► **Definition 2.9** (k -junta). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, and $k \in \mathbb{N}$ be a parameter. A variable x_i , $i \in [n]$, is influential with respect to f , if for some $\mathbf{x} \triangleq (x_1, \dots, x_n) \in \mathbb{R}^n$, changing x_i changes $f(\mathbf{x})$. f is a k -junta, if there are at most k influential variables with respect to f .*

Testing whether a Boolean function is a k -junta has been extensively studied in the exact query model. The first result in this context was by [66], which was followed by the work of [43], who designed a $\tilde{O}(k^2)$ -query tester. Later, [33] extended it to the finite range setting. [17] then gave an $\tilde{O}(k^{3/2})$ -query non-adaptive tester for this problem, while for adaptive testers, [18] showed $\tilde{O}(k \log k + k/\varepsilon)$ queries suffice. It is important to note that all these results use Fourier-analytic techniques. Notably, [22] designed a new algorithm for

testing k -juntas with similar optimal bounds in the context of testing partial isomorphism of functions. Interestingly, this work deviates from the common Fourier analytic approach and instead presents a combinatorial approach to this problem. This algorithm from [22] will be used in designing our tester for k -juntas. Recently [32] studied the linear k -junta testing problem, where the function f is defined as $f : \mathbb{R}^n \rightarrow \{+1, -1\}$ ³. This is different from our setting of real-valued functions. It is not clear to us if their techniques can be generalized to our setting. In terms of lower bounds, [43] showed a lower bound of $\Omega(\sqrt{k})$ queries for non-adaptive testers, which was improved to $\tilde{\Omega}(k^{3/2}/\varepsilon)$ by [30]. For adaptive testers, [31] showed an $\Omega(k)$ lower bound, which was then improved to $\Omega(k \log k)$ by [72].

► **Theorem 2.10** (Informal, see Theorem 7.1). *Let $k \in \mathbb{N}$, and $\varepsilon, \eta \in (0, 1)$ be such that $\eta < \min\{O(\varepsilon/k^2), O(\frac{1}{k^2 \log^2 k})\}$, and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be bounded in $B(\mathbf{0}, 2\sqrt{n})$, given via η -approximate queries. There exists a one-sided error, $\tilde{O}\left(\frac{k \log k}{\varepsilon}\right)$ -query tester (Algorithm 6), that distinguishes if f is a k -junta, or is ε -far from all k -juntas, with probability at least $2/3$.*

2.4 Lower bounds

We briefly mention our lower bound results, which hold even for adaptive testers, and exact query access. For all three properties (k -linearity, k -sparse degree- d polynomials, and k -juntas), we prove lower bounds of $\Omega(\max\{k, \frac{1}{\varepsilon}\})$ queries. Additionally, for k -sparse, degree- d polynomials, we prove a lower bound of $\Omega(\max\{k, d, \frac{1}{\varepsilon}\})$ queries. All our lower bounds follow from the general reduction from communication complexity, introduced by [19], coupled with some folklore results. We use SET-DISJOINTNESS as the hard instance to prove our results.

► **Theorem 2.11.** *Given exact query access to $f : \mathbb{R}^n \rightarrow \mathbb{R}$, some $k, d \in \mathbb{N}$ and a distance parameter $\varepsilon \in (0, 1)$, $\Omega(\max\{k, \frac{1}{\varepsilon}\})$ queries are necessary for testing (i) k -linear functions, (ii) k -sparse degree- d polynomials, and (iii) k -juntas, with success probability at least $2/3$.*

The lower bound for testing k -sparse degree- d polynomials is improved to $\Omega(\max\{d, k, \frac{1}{\varepsilon}\})$.

2.5 New Technical Contributions

k -sparsity Tester. A critical ingredient in proving the k -sparse low degree tester (Theorem 2.7) is a new probabilistic upper bound on the maximum singular value of Hankel matrices associated with sparse, low-degree polynomials. This may be of independent interest.

► **Theorem 2.12** (Probabilistic Upper Bound on $\sigma_{\max}$). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $f(\mathbf{x}) = \sum_{i=1}^k a_i M_i(\mathbf{x})$ be a k -sparse, degree- d polynomial, where M_i 's are its non-zero monomials, and $\sigma_{\max}(H_t(f, \mathbf{u}))$ denotes the largest singular value of the t -dimensional Hankel matrix associated with f at a point $\mathbf{u} \in \mathbb{R}^n$, $H_t(f, \mathbf{u})$. Then, for any $\gamma \in (0, 1)$, with $\mathbf{a} \triangleq (a_1, \dots, a_k)^\top \in \mathbb{R}^k$,*

$$\Pr_{\mathbf{u} \sim \mathcal{N}(\mathbf{0}, I_n)} \left[\sigma_{\max}(H_t(f, \mathbf{u})) \geq \|\mathbf{a}\|_2^2 \left(2^{d/2} \lceil d/2 \rceil! + \sqrt{\frac{k}{\gamma}} 2^{d/2} \sqrt{d!} \right)^{2t} \right] \leq \gamma.$$

Proof Sketch. For any vector $\mathbf{v} \in \mathbb{R}^n$, let $|\mathbf{v}| \in \mathbb{R}_{\geq 0}^n$ denote the vector with the absolute values of the coordinates of $\mathbf{v}$. Then, for any $\mathbf{z}, \mathbf{u} \in \mathbb{R}^n$, using the triangle inequality, and a Vandermonde-like decomposition of $H_{\mathbf{u}} \triangleq H_t(|f|, |\mathbf{u}|)$, (from Observation 2.6), where

³ A function $f : \mathbb{R}^n \rightarrow \{-1, 1\}$ is said to be a *linear k -junta* if there are k unit vectors $u_1, \dots, u_k \in \mathbb{R}^n$ and $g : \mathbb{R}^k \rightarrow \{-1, 1\}$ such that $f(x) = g(\langle u_1, x \rangle, \dots, \langle u_k, x \rangle)$.

$|f|(\mathbf{x}) \triangleq \sum_{i \in [k]} |a_i| M_i(\mathbf{x})$, we can upper-bound $|\mathbf{z}^\top H_{\mathbf{u}} \mathbf{z}|$ by $|\mathbf{z}^\top V^\top D V \mathbf{z}| = \|D^{\frac{1}{2}} V \mathbf{z}\|_2^2$, where $D = \text{diag}(|\mathbf{a}|)$, V is the Vandermonde matrix $V_t(|M_1(\mathbf{u})|, \dots, |M_k(\mathbf{u})|) \in \mathbb{R}^{k \times t}$, and $M(|\mathbf{u}|) = |M(\mathbf{u})|$. By the Courant-Fischer characterization, we have

$$\sigma_{\max}(H_{\mathbf{u}}) \leq \max_{\|\mathbf{z}\|_2 \leq 1} \|D^{\frac{1}{2}} V \mathbf{z}\|_2^2 \leq \max_{\|\mathbf{z}\|_2 \leq 1} \|D^{\frac{1}{2}} V\|_{\text{op}}^2 \|\mathbf{z}\|_2^2 \leq \|D^{\frac{1}{2}} V\|_{\text{F}}^2.$$

Using some results from [36] for the moments of the folded normal distribution, along with the properties of the Gamma function and the fact that all the monomials M_i have total degree $\leq d$, we can upper bound

$$\mathbb{E}_{\mathbf{u} \sim \mathcal{N}(\mathbf{0}, I_n)} [|M_i(\mathbf{u})|] \leq 2^{d/2} \lceil d/2 \rceil!, \text{ and } \text{Var}[|M_i(\mathbf{u})|] \leq 2^d \cdot d!.$$

Chebyshev's inequality and a union bound over the k monomials completes the proof. $\blacktriangleleft$

► **Remark 2.13.** Theorem 2.12 seems to be extendable to more general mean-zero distributions of $\mathbf{u}$ with appropriate concentration (e.g. subgaussian, subexponential) since the crux of the proof, in addition to the use of properties of the Hankel matrix (which do not depend on the distribution of $\mathbf{u}$), is to bound all the degree d moments of the distribution.

k -linearity Tester. For k -linearity, our tester is similar to the algorithm proposed in [26] for testing k -linearity of functions $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$, with the BLR test replaced by the approximate-query linearity tester from [44, 5, 7]. The crucial difficulties in the analysis over the reals come from (i) the η -approximate query access, and (ii) the use of ℓ_1 -distance for farness. We first reject all functions which are far from linearity by means of the linearity test, and, conditioned on the fact that the tester does not reject with high probability, proceed to testing the self-corrected function instead (with closeness to linearity guaranteed). But we now have only approximate query access to the function, which we test for k -linearity by splitting the variables into $O(k^2)$ random buckets as in [26] and identify the influential ones. However, there is no analogous result for real-valued functions. Hence, we present a new analysis in Claim 5.11 for the FindInfBucket (Algorithm 2) that performs a binary search over the buckets, but with a different test for influential buckets that accounts for η -approximate queries. The analysis with approximate queries involves the appropriate use of anti-concentration results (Theorem 4.5) for real linear polynomials.

k -junta tester. Our tester is inspired by the junta testers used in [18] and [22], but our definition of influence (Definition 7.2) is different (ℓ_1 rather than Hamming), to work with approximate query oracle. Theorem 7.3, Lemma 7.4, and Lemma 7.5, in our analysis are similar to that of [22]. The significant difference in analysis for the ℓ_1 distance comes in our Claim 7.7, and the subsequent proof of Theorem 7.1.

2.6 Discussion

Our efficient algorithms for the problem of testing real-valued functions given via approximate queries, over continuous domains, for three properties: (i) k -linearity, (ii) k -sparse, low-degree polynomials, and (iii) k -juntas, summarized in Table 1, open several interesting questions:

- We note that our results have constraints on the approximate query parameter η . The first open question is whether these can be improved.
- Furthermore, our k -sparse degree- d polynomial tester performs $\tilde{O}(d^5 + \frac{d^2}{\varepsilon} + dk^3)$ queries, and our lower bound for this problem is $\Omega(\max\{d, k, \frac{1}{\varepsilon}\})$. The second open question is whether the gaps in these bounds (w.r.t. k , and d) can be improved, e.g., by assuming additional structure on the underlying function, like Lipschitzness, etc.

■ **Table 1** A comparison of the upper bounds, as well as the corresponding restrictions, for the three sparse representation testing problems. The upper bounds, and the restrictions in the three rows follow sequentially from Theorem 2.2, Theorem 2.7, and Theorem 2.10, respectively.

Problem	Upper Bound	Restriction
k -linearity	$\tilde{O}(k \log k + \frac{1}{\varepsilon})$	$\eta < \min \left\{ \varepsilon, O \left(\min_{i \in [n]: f(e_i) \neq 0} \frac{ f(e_i) }{(nk)^2} \right) \right\}$
k -sparsity	$\tilde{O}(d^5 + \frac{d^2}{\varepsilon} + dk^3)$	$\eta < \min \{ \varepsilon, 1/2^{2^n} \}$
k -junta	$\tilde{O} \left(\frac{k \log k}{\varepsilon} \right)$	$\eta < \min \{ O(\varepsilon/k^2), O(\frac{1}{k^2 \log^2 k}) \}$

- Another interesting direction is to design tolerant testers [65] for these properties. This is different from the approximate query testing notion, as in tolerant testing, the decision boundary is expanded to require that functions that are sufficiently close to the property are also accepted with high probability, whereas in the approximate query model, we accept functions $\tilde{f}$ such that $\text{dist}_{\mathcal{D}, \infty}(f, \tilde{f}) \leq \eta$ (pointwise η -close) which is a stronger constraint compared to the expected ℓ_1 -distance $\text{dist}_{\mathcal{D}, \ell_1}(f, g)$ that we use between functions.
- We have focused on optimizing the query complexity in terms of the sparsity parameter k and degree d (for low-degree polynomial testing). It is an interesting problem to optimize the dependence of η in our arguments.
- Only our k -junta tester has a one-sided error profile, while our k -linearity/sparsity testers have two-sided errors. Designing one-sided error testers for these problems is left open.
- Finally, we use $\mathcal{N}(\mathbf{0}, I_n)$ as the reference distribution. It would be interesting to see if our results can be extended to other concentrated distributions as well (see Remark 2.13).

Organization of the paper

In Section 3, we present an overview of our results and techniques, followed by a discussion of the preliminaries in Section 4. In Section 5, we present our k -linearity tester, followed by our tester for k -sparse low-degree polynomials in Section 6, and our k -junta tester in Section 7.

Due to space constraints here, we omit all proofs of intermediate technical lemmata and claims, an elaborate discussion of lower bounds, and all subroutines from prior works (APPROXIMATEADDITIVITYTESTER [6, Algorithm 7], TESTADDITIVITY [6, Algorithm 8], APPROXIMATE- g [6, Algorithm 8], APPROXLOWDEGREETESTER [6, Algorithm 9], APPROXCHARACTERIZATIONTEST [6, Algorithm 10], APPROXQUERY- g [6, Algorithm 10] and APPROXQUERY- g -INBALL [6, Algorithm 10]). They are included in the full version [6].

3 Technical Overview

3.1 Testing k -linearity

We build upon the *self-correct and test* approach of [24, 53]. Instead of directly testing if f is k -linear, we construct a function $g_{\text{self-correct}}$ such that if f is k -linear, then $g_{\text{self-correct}}$ will also be k -linear. Moreover, we simulate queries to $g_{\text{self-correct}}$ using queries to f , and test this newly constructed function $g_{\text{self-correct}}$.

We use the Gaussian distribution as the reference distribution, since there are no uniform distributions over continuous domains with infinite support. (This deviates from the self-correction approach of [24].) In particular, we evaluate f on a set of points sampled from $\mathcal{N}(\mathbf{0}, I_n)$ to construct the self-corrected function $g_{\text{self-correct}}$. To deal with the fact that different points from $\mathcal{N}(\mathbf{0}, I_n)$ have different probability masses, the idea is to radially project

14:10 Testing Sparse Functions over the Reals

the sampled points from $\mathcal{N}(\mathbf{0}, I_n)$ into a small Euclidean ball $B(\mathbf{0}, r)$ of a small (constant) radius ($r = 1/50$ suffices) such that the probability masses of all points sampled from that ball is roughly the same. Moreover, since we work with an approximate oracle, we use a median-based notion of the self-corrected function, with a contraction factor $\kappa_{\mathbf{p}} : \mathbb{R}^n \rightarrow \mathbb{R}$:

$$g(\mathbf{p}) \triangleq \kappa_{\mathbf{p}} \cdot \text{med}_{\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I_n)} \left[f \left(\frac{\mathbf{p}}{\kappa_{\mathbf{p}}} - \mathbf{x} \right) + f(\mathbf{x}) \right], \text{ and } \kappa_{\mathbf{p}} \triangleq \begin{cases} 1 & , \text{ if } \|\mathbf{p}\|_2 \leq r \\ \lceil \|\mathbf{p}\|_2 / r \rceil & , \text{ if } \|\mathbf{p}\|_2 > r \end{cases},$$

so that $\mathbf{p}/\kappa_{\mathbf{p}} \in B(\mathbf{0}, r)$. This definition of self-correction function was used in [44, 5, 7].

To test k -linearity (Algorithm 1), we first test if f is pointwise close to some additive (aka. linear) function using APPROXIMATEADDITIVITYTESTER. If it rejects f , we also reject f . However, if APPROXIMATEADDITIVITYTESTER does not reject, then the self-corrected function g is pointwise close to some linear function. As we only have approximate oracle access to f , we can't simulate g exactly. So, we use APPROXIMATE- g , the approximate query oracle for g . The work of [5, 7] proves: (i) g and APPROXIMATE- g are pointwise close in $B(\mathbf{0}, r)$, and (ii) f and APPROXIMATE- g are also pointwise close. So, f is pointwise close to APPROXIMATE- g .

We partition the n -variables $[n]$ into k^2 buckets uniformly at random. So if f is k -linear, the influential variables will be separated into different buckets w.h.p. We can then detect them via the subroutine FindInfBucket (Algorithm 2), which recursively isolates the variables in a bucket and returns only the influential ones. We then reject if the number of buckets with influential variables (variables whose values determine the value of f) is more than k .

3.2 Testing k -sparsity

Our k -sparse, degree- d polynomial tester (Algorithm 4) adopts a similar approach. We first test if f is a low-degree polynomial, using the APPROXLOWDEGREETESTER from [5, 7]. If it rejects f , we also reject f . However, if APPROXLOWDEGREETESTER does not reject f , then f is point-wise close to a low-degree polynomial.

As in k -linearity testing, we use a self-corrected function g from [5, 7]: For points $\mathbf{p} \in B(\mathbf{0}, r)$, $g(\mathbf{p})$ is the (weighted) median value of $g_{\mathbf{q}}(\mathbf{p}) \triangleq \sum_{i=1}^{d+1} (-1)^{i+1} \binom{d+1}{i} f(\mathbf{p} + i\mathbf{q})$, weighted according to the probability of $\mathbf{q} \sim \mathcal{N}(\mathbf{0}, I_n)$, i.e., $g(\mathbf{p}) \triangleq \text{med}_{\mathbf{q} \sim \mathcal{N}(\mathbf{0}, I_n)} [g_{\mathbf{q}}(\mathbf{p})]$. Intuitively, $g_{\mathbf{q}}(\mathbf{p})$ is the value that f should take if, when restricted to the line $L_{\mathbf{p}, \mathbf{q}} \triangleq \{\mathbf{p} + t\mathbf{q}, t \in \mathbb{R}\}$, f would be a degree- d univariate polynomial. Taking the weighted median over all directions $\mathbf{q} \sim \mathcal{N}(\mathbf{0}, I_n)$, ensures that the self-correction proportionately respects the values of f , in a local neighborhood of $\mathbf{p}$. For $\mathbf{p} \notin B(\mathbf{0}, r)$, g is defined via radial extrapolation from within $B(\mathbf{0}, r)$ along the radial line $L_{\mathbf{0}, \mathbf{p}}$.

Given exact query access to f , we can simulate query access to the self-corrected function g . As we only have approximate query access to f , we use APPROXQUERY- g (the approximate oracle to the self-corrected function associated with f), which was proved to be pointwise close to g . As a result, f will be pointwise close to APPROXQUERY- g as well.

Once we have that f is close to a low-degree polynomial, we use a Hankel matrix (Definition 2.5) based characterization for sparse polynomials. [52, 11] proved Observation 2.6: a polynomial f (over finite fields) is k -sparse, if and only if its associated Hankel matrix has a non-zero determinant. This can be efficiently tested with only $2k + 1$ queries to f . For polynomials over the reals with exact-queries, the idea from [52] can be extended to give a zero-error (perfect soundness and completeness), zero-gap (does not require ε -farness for $\varepsilon > 0$ in the No case) tester, making $2k + 1$ queries for deciding whether (i) f is k -sparse, or (ii) f is not k -sparse (proved in Lemma 6.4). This result is even stronger than what [52] gets for finite

fields, since the zero set of any polynomial over the reals has zero Gaussian (or Lebesgue) measure, whereas the corresponding probability needs to be bounded in terms of the degree of the polynomials over finite fields (e.g. by the Schwartz-Zippel lemma). Unfortunately, since we only have approximate query access to f , this technique no longer works, as determinants of sums of matrices do not behave nicely. So we take a probabilistic approach, showing that the *noisy* Hankel matrix constructed from the approximate query to f is not too far from the exact Hankel matrix (proved in Observation 6.5). Finally, we show in Theorem 6.6 using Weyl's inequality (Theorem 4.4), that if f is a k -sparse, low-degree polynomial, then the smallest eigenvalue of the *noisy* Hankel matrix associated with APPROXQUERY- g is not too large. Combining them all, our main result is proved in Section 6.2.

Since, we only have access to the Hankel matrix of $\tilde{f} = \text{APPROXQUERY-}g$ which is η' -close to f (on all the query points, whp), we can express $\tilde{H}_{\mathbf{u}} \triangleq H_t(\tilde{f}, \mathbf{u})$ (for $t = k + 1$) of such a $\tilde{f}$ as the sum of $H_{\mathbf{u}} \triangleq H_t(f, \mathbf{u})$ and a Hankel-structured error matrix with bounded spectral/operator norm (Observation 6.5). From this, and Weyl's inequality, we get completeness as long as we ACCEPT when the smallest singular value $\sigma_{\min}(\tilde{H}_{\mathbf{u}}) \leq \eta'(k + 1)$ in all $\Theta(d(k + 1)^2)$ rounds.

To prove soundness, we need to show that $\sigma_{\min}(\tilde{H}_{\mathbf{u}}) > \eta'(k + 1)$ in some round with probability $\geq 2/3$, as long as $Q(\mathbf{u}) \triangleq \det(H_{\mathbf{u}})$ is a non-zero polynomial. This follows (in Theorem 6.6) by invoking the probabilistic upper bound on $\sigma_{\max}(H_{\mathbf{u}})$, for $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, I_n)$.

3.3 Testing k -juntas

Finally, we discuss our algorithm for testing k -juntas (Algorithm 6). Our approach is to first randomly partition the n -variables into k^2 buckets. If f is a k -junta, the k influential variables will be separated into distinct buckets w.h.p (by birthday paradox).

Now we run $O(k/\varepsilon)$ iterations to find if there exists any influential variable in any bucket, using the subroutine FindInfBucket (Algorithm 2). After these iterations, if we find more than k influential variables in f , we reject it. Otherwise, we accept f . Our analysis follows a combinatorial style similar to [22]. We would like to note that although we use FindInfBucket for k -linearity testing as well, the analysis here significantly deviates from that of k -linearity testing and is presented in Claim 7.7. The main result is formally proved in Section 7.1.

4 Preliminaries

Notations. Throughout this work, we use boldface letters to represent vectors of length n and normal face letters for variables. Specifically, $\mathbf{e}_i \triangleq (0, \dots, 0, 1_i, 0, \dots, 0)$ denotes the i^{th} standard unit vector. For $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, \dots, n\}$. For a matrix A , let $\|A\|_{\infty}$, $\|A\|_{\text{op}}$, and $\|A\|_F$ denote the supremum, operator, and Frobenius norms of A , respectively. See [56] for the formal definitions. For concise expressions and readability, we use the asymptotic complexity notion of $\tilde{O}(\cdot)$, where we hide poly-logarithmic dependencies of the parameters. For any $f : \mathbb{R}^n \rightarrow \mathbb{R}$, let $\|f\|_{-\infty(/\infty), C}$ denote the infimum(resp. supremum) value of f over some $C \subseteq \mathbb{R}^n$. Let Π_n be the class of functions $\mathbb{R}^n \rightarrow \mathbb{R}$ satisfying some particular first-order property and let $\Pi = \cup_{n \geq 1} \Pi_n$.

► **Definition 4.1** (ℓ_1 -distance). Let $\mathcal{D} = \{\mathcal{D}_n\}_{n \geq 1}$ is family of distributions with $\mathcal{D}_n$ being a distribution on $\mathbb{R}^n$. For two arbitrary functions $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$, the ℓ_1 -distance between f and g is defined as: $\text{dist}_{\mathcal{D}, \ell_1}(f, g) \triangleq \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_n} [|f(\mathbf{x}) - g(\mathbf{x})|]$. We also define the ℓ_p -distance of f to the class Π_n , and hence the class Π , by $\text{dist}_{\mathcal{D}, \ell_p}(f, \Pi) \triangleq \text{dist}_{\mathcal{D}, \ell_p}(f, \Pi_n) \triangleq \inf_{g \in \Pi_n} \text{dist}_{\mathcal{D}, \ell_p}(f, g)$.

We have $\text{dist}_{\mathcal{D}, \ell_p}(f, \Pi) = 0$ if and only if there exists a function $g \in \Pi_n$ which agrees with f almost everywhere with respect to $\mathcal{D}_n$ (in the measure-theoretic sense). We will only concern ourselves with $p \in \{0, 1\}$ since we are dealing with scalar-valued functions (aka functionals) and hence $\|f(\mathbf{x}) - g(\mathbf{x})\|_p = |f(\mathbf{x}) - g(\mathbf{x})|$ for all $p > 0$.

Our definitions hold for general reference distributions $\mathcal{D}$ which are suitably concentrated. Later, we work with $\mathcal{N}(\mathbf{0}, I_n)$, which also has this desired concentration property.

► **Definition 4.2** (Concentrated distribution). *Let $\varepsilon \in (0, 1)$, $R \geq 0$, and $\mathbf{c} \in \mathbb{R}^n$. A distribution $\mathcal{D}$ supported on $\mathbb{R}^n$ is $(\varepsilon, R, \mathbf{c})$ -concentrated if most of its mass is contained in a ball of radius R centered at some point $\mathbf{c} \in \mathbb{R}^n$, i.e., $\Pr_{\mathbf{p} \sim \mathcal{D}}[\mathbf{p} \in B(\mathbf{c}, R)] \geq 1 - \varepsilon$.*

For brevity, we may write “ $\mathcal{D}$ is (ε, R) -concentrated” to mean “ $\mathcal{D}$ is $(\varepsilon, R, \mathbf{0})$ -concentrated”.

► **Fact 4.3** ([23, Theorem 2.9]). *$\mathcal{N}(\mathbf{0}, I_n)$ is $(0.01, 2\sqrt{n}, \mathbf{0})$ -concentrated.*

4.1 Results from Perturbation Theory, and Measure Theory

For analyzing our testers, we use the following result from perturbation theory.

► **Theorem 4.4** (Weyl's Inequality [55]). *Let $\hat{\Sigma} = \Sigma + E$, where $\Sigma, E \in \mathbb{R}^{n \times n}$ are both symmetric matrices. Let $\lambda_i(A)$ denote the i^{th} eigenvalue of (symmetric matrix) A , sorted in non-increasing order, and let $\|A\|_{\text{op}}$ denote its ℓ_2 -operator (spectral) norm. Then,*

$$\max_{i \in [n]} \{|\lambda_i(\hat{\Sigma}) - \lambda_i(\Sigma)|\} \leq \|E\|_{\text{op}} \leq \max\{|\lambda_d(E)|, |\lambda_1(E)|\}.$$

The following celebrated inequality of Carbery and Wright [27] provides anti-concentration bounds for polynomials of i.i.d Gaussian random variables (more generally, log-concave), which will be used in our proofs, particularly in the analysis of our k -linearity tester.

► **Theorem 4.5** ([27, Theorem 8]). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a polynomial of degree d , such that $\text{Var}_{\mathcal{N}(\mathbf{0}, I_n)}[f] = 1$. Then $\forall t \in \mathbb{R}$ and $\forall \varepsilon > 0$, $\Pr_{\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I_n)}[|f(\mathbf{x}) - t| \leq \varepsilon] \leq O(d)\varepsilon^{1/d}$.*

A recent result of Glazer and Mikulincer [47] provided a more suitable form of Carbery-Wright for polynomials, with the variance lower-bounded in terms of the coefficients.

► **Theorem 4.6** ([47, Corollary 4]). *If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a polynomial $f(\mathbf{x}) = \sum_{i=1}^k a_i \mathbf{x}^{\mathbf{M}_i}$ of degree d , then there exists an absolute constant $C > 0$ such that for any $t \in \mathbb{R}$ and $\varepsilon > 0$,*

$$\Pr_{\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I_n)}[|f(\mathbf{x}) - t| \leq \varepsilon] \leq Cd \left(\frac{\varepsilon}{\text{coeff}_d^2(f)} \right)^{1/d}, \text{ where } \text{coeff}_d^2(f) \triangleq \sum_{i \in [k]: \|\mathbf{M}_i\|_1 = d} a_i^2.$$

5 k -linearity Testing

In this section, we present and analyze our algorithm TEST- k -LINEAR (Algorithm 1).

► **Theorem 5.1** (Generalization of Theorem 2.2). *Let $k \in \mathbb{N}$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function bounded in the ball $B(\mathbf{0}, 2\sqrt{n})$, and $\varepsilon, \eta \in (0, 2/3)$ be such that $\eta < \min \left\{ \varepsilon, O \left(\min_{i \in [n]: f(\mathbf{e}_i) \neq 0} \frac{|f(\mathbf{e}_i)|}{(nk)^2} \right) \right\}$, where $\mathbf{e}_i$ denotes the i^{th} standard unit vector. Given η -approximate query access to f , there exists a tester TEST- k -LINEAR (Algorithm 1) that, in $\tilde{O}(k \log k + \frac{1}{\varepsilon})$ queries, guarantees:*

- **COMPLETENESS:** *If f is a k -linear function, Algorithm 1 ACCEPTS with probability $\geq 2/3$.*
- **SOUNDNESS:** *If f is ε -far from all k -linear functions, Algorithm 1 REJECTS w.p. $\geq 2/3$.*

Algorithm 1 TEST- k -LINEAR.

Inputs : η -approximate query oracle $\tilde{f}$ for $f : \mathbb{R}^n \rightarrow \mathbb{R}, k \in \mathbb{N}, \varepsilon \in (0, 1)$.
Output : Returns ACCEPT iff f is a k -linear function.

- 1 Run APPROXIMATEADDITIVITYTESTER($f, \mathcal{N}(\mathbf{0}, I_n), \eta, \varepsilon, 2\sqrt{n}$).
- 2 If APPROXIMATEADDITIVITYTESTER rejects, **return** REJECT.
- 3 Set $r = O(k^2), C \leftarrow \emptyset$; // APPROXIMATEADDITIVITYTESTER = [6, Algorithm 7]
- 4 Bucket each $i \in [n]$ into $B = \{B_1, \dots, B_r\}$, uniformly at random; // $[n] = \uplus_{i=1}^r B_i$.
- 5 $C = \text{FindInfBuckets}(\text{APPROXIMATE-g}, B, [r])$; // FindInfBuckets = Algorithm 3
- 6 if $|C| > k$ then **return** REJECT;
- 7 **return** ACCEPT; // APPROXIMATE-g in [6, Algorithm 8]

Algorithm 1 uses two subroutines FindInfBucket (Algorithm 2) and FindInfBuckets (Algorithm 3). We will first present and analyze them in Section 5.1, and then analyze Algorithm 1 in Section 5.2.

A closely related (and *efficiently testable*) notion of *additivity* may be noted here.

► **Definition 5.2** (Additive function). *A function $f : A \rightarrow B$ is additive, if for all $x, y \in A$, $f(x \oplus_A y) = f(x) \oplus_B f(y)$, where $\oplus_A$ and $\oplus_B$ denote the bitwise-xor operations in A and B , respectively.*

Over finite domains, additivity implies linearity. But over continuous domains, this isn't always the case. However, for continuous functions, testing additivity suffices:

► **Fact 5.3** ([60, Section 5.2]). *For continuous functions, additivity is equivalent to linearity.*

Algorithm 1 uses a query oracle to the self-corrected function, APPROXIMATE-g. We now note the following results from [5, 7] about their approximate additivity tester.

► **Theorem 5.4** ([5, Theorem D.1], and [7, Theorem 3.3]). *Let $\alpha, \varepsilon > 0$, for $L > 0$, suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a function that is bounded in the ball $B(0, L)$ and $\mathcal{D}$ be an unknown $(\varepsilon/4, R)$ -concentrated distribution. There exists a one-sided error, $O(\frac{1}{\varepsilon})$ -query tester (APPROXIMATEADDITIVITYTESTER) which with probability at least 99/100, distinguishes when f is pointwise α -close to some additive function and when, for every additive function h , $\Pr_{\mathbf{p} \sim \mathcal{D}}[|f(\mathbf{p}) - h(\mathbf{p})| > O(Rn^{1.5}\alpha)] > \varepsilon$.*

► **Lemma 5.5** ([5, Lemma D.3 and D.6]). *If TESTADDITIVITY($f, 3\alpha$) accepts with probability at least 1/3, then g is a 42α -additive function inside the small ball $B(\mathbf{0}, r)$, and furthermore, for every $\mathbf{p} \in B(\mathbf{0}, r)$ it holds that*

$$\Pr_{\mathbf{x} \sim \mathcal{N}(\mathbf{0}, I_n)}[|g(\mathbf{p}) - g_{\mathbf{x}}(\mathbf{p})| \geq 12\alpha] < 12/125.$$

► **Lemma 5.6** ([5, Lemma D.4 and D.5]). *If TESTADDITIVITY($f, 3\alpha$) accepts with probability at least 1/3, then for every $\mathbf{p}, \mathbf{q} \in B(\mathbf{0}, r)$ with $\|\mathbf{p} + \mathbf{q}\|_2 \leq r$, it holds that*

$$|g(\mathbf{p} + \mathbf{q}) - g(\mathbf{p}) - g(\mathbf{q})| \leq 42\alpha.$$

The following lemma gives us a way to scale the closeness for degree- d polynomials.

► **Lemma 5.7** ([5, Lemma 4.19]). *Let $R > r' > 0$ be any real numbers. If g is pointwise η -close to a degree- d polynomial in $B(\mathbf{0}, r')$, then g is pointwise $(12R/r')^d \eta$ -close to a degree- d polynomial on all points in $B(\mathbf{0}, R)$.*

Using Lemma 5.5, Lemma 5.7 and the structure of APPROXIMATE-g subroutine, they bound the distance between g and APPROXIMATE-g.

14:14 Testing Sparse Functions over the Reals

▷ **Claim 5.8.** If $\text{TESTADDITIVITY}(f, 3\alpha)$ accepts with probability at least $1/3$, then $\text{APPROXIMATE-}g$ is pointwise 6α -close to g , in the ball $B(\mathbf{0}, r)$ w.h.p.; i.e.,

$$\Pr_{\mathbf{p} \sim \mathcal{D}} [|g(\mathbf{p}) - \text{APPROXIMATE-}g(\mathbf{p})| \leq 6\alpha(12R/r) \mid \mathbf{p} \in B(\mathbf{0}, R)] \geq 1 - \varepsilon/4.$$

The following observation directly follows using the triangle inequality, along with the above claim and the notion of η -approximate queries.

► **Observation 5.9.** If $\text{APPROXIMATEADDITIVITYTESTER}$ does not reject f with probability at least $2/3$, then following the description of $\text{APPROXIMATEADDITIVITYTESTER}$, we have

$$\Pr_{\mathbf{p} \sim \mathcal{D}} [|f(\mathbf{p}) - \text{APPROXIMATE-}g(\mathbf{p})| \leq 750Rn^{1.5}\alpha \mid \mathbf{p} \in B(\mathbf{0}, R)] \geq 1 - \frac{\varepsilon}{4}.$$

We will invoke them with $\mathcal{D} = \mathcal{N}(\mathbf{0}, I_n)$, implying $R = 2\sqrt{n}$ (from Fact 4.3).

5.1 Analyses of Subroutines

The subroutine FindInfBucket is presented in Algorithm 2. To analyze it, we need:

■ **Algorithm 2** $\text{FindInfBucket}(f, B, S)$.

Inputs : η -approximate query oracle $\tilde{f}$ for $f : \mathbb{R}^n \rightarrow \mathbb{R}$, Random Bucketing
 $B = \{B_1, \dots, B_r\}$ of $[n]$, $\emptyset \neq S \subseteq [r]$

Output : Either $\emptyset$, or an influential bucket B_j (with $j \in S$) of f .

```

1 Let  $V \triangleq \{i \in [n] : i \in B_j \text{ for some } j \in S\}$ ; //  $|V| \approx \frac{|S|n}{r}$ .
2 Sample two independent Gaussian vectors  $\mathbf{x}, \mathbf{y} \sim \mathcal{N}(\mathbf{0}, I_n)$ .
3 if  $|\tilde{f}(\mathbf{x}_V \mathbf{y}_{\overline{V}}) - \tilde{f}(\mathbf{y})| \leq 2\eta$  then return  $\emptyset$ ;
4 else
5   if  $S = \{j\}$  for some  $j$  then return  $B_j$ ;
6   else
7     Partition  $S$  into:  $S^{(L)}$  and  $S^{(R)}$ , with  $1 \leq |S^{(R)}| \leq |S^{(L)}| \leq |S^{(R)}| + 1$ .
8      $\text{ret}_L \leftarrow \text{FindInfBucket}(f, B, S^{(L)})$ .
9     if  $\text{ret}_L = \emptyset$  then return  $\text{FindInfBucket}(f, B, S^{(R)})$ ;
10    else return  $\text{ret}_L$ ;

```

► **Definition 5.10** (Influential bucket for linear functions). For a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, a bucket $B \subseteq [n]$ is said to be an influential bucket if there exists at least one variable $x_i, i \in B$, which is influential with respect to f (i.e., $f(\mathbf{x})$ changes with a change in x_i). For a linear f , with $f(\mathbf{x}) \equiv \sum_{i \in [n]} a_i x_i$, x_i is influential w.r.t f if and only if $a_i \neq 0$, and hence B is an influential bucket iff $a_i \neq 0$ for some $i \in B$.

▷ **Claim 5.11** (Correctness of FindInfBucket). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, with $f(\mathbf{x}) = \sum_{i \in [n]} a_i x_i$ be given via an η -approximate query $\tilde{f}$, where $\eta \leq \frac{1}{100k^2} \min_{k \in [n]: a_k \neq 0} |a_k|$, $B = \{B_1, \dots, B_r\}$ be a partition of $[n]$, and $\emptyset \neq S \subseteq [r]$. Then $\text{FindInfBucket}(f, B, S)$ (Algorithm 2) guarantees:

1. If none of the buckets $\{B_i, i \in S\}$ are influential, $\text{FindInfBucket}(f, B, S)$ always returns $\emptyset$ and performs exactly 2 queries to f .
2. Otherwise, with probability at least $1 - 8\lceil \log |S| \rceil^2 / 10k^2$, $\text{FindInfBucket}(f, B, S)$ returns B_j for some $j \in S$ which is an influential bucket, and performs $\leq 8\lceil \log(|S|) \rceil^2$ queries to f .

■ **Algorithm 3** FindInfBuckets($\tilde{f}, B, X$).

Inputs : η -approximate query $\tilde{f}$ for $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $B = \{B_1, \dots, B_r\}$, $\emptyset \neq X \subseteq [r]$.
Output : A set of influential buckets $\text{InfBuckets} \subseteq \{B_j : j \in X\}$ with respect to f .

```

1 InfBuckets  $\leftarrow \emptyset$ 
2 for  $j=1$  to  $8k$  do
3   RetVal = FindInfBucket( $f, B, X$ ) ;           // FindInfBucket is Algorithm 2
4   if RetVal =  $B_i$  for some  $i$  then InfBuckets  $\leftarrow$  InfBuckets  $\cup \{B_i\}$ ,  $X \leftarrow X \setminus \{i\}$ . ;
5 return InfBuckets.
```

Next we describe and analyze the subroutine FindInfBuckets (Algorithm 3).

▷ **Claim 5.12 (Correctness of FindInfBuckets).** Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, with $f(\mathbf{x}) = \sum_{i \in [n]} a_i x_i$, be given via the η -approximate query oracle $\tilde{f}$, where $\eta \leq \frac{1}{100k^2} \min_{k \in [n]: a_k \neq 0} |a_k|$, and $\emptyset \neq X \subseteq [n]$.

Then FindInfBuckets(f, B, X) (Algorithm 3), in at most $64k \lceil \log(|X|) \rceil^2$ queries, guarantees:

- (i) If f is ℓ -linear function for some $\ell > 8k$, then Algorithm 3 will return a set of $8k$ influential buckets in f with probability at least $1 - 64 \lceil \log |X| \rceil^2 / 10k$.
- (ii) If f is ℓ -linear function for some $\ell \leq 8k$, then Algorithm 3 will return the set of all influential buckets in f with probability at least $1 - 64 \lceil \log |X| \rceil^2 / 10k$.

5.2 Analysis of the k -linearity tester (Algorithm 1)

We are now ready to prove the main theorem of this section:

Proof of Theorem 5.1.

Completeness. Since f is a k -linear function, following Theorem 5.4, we have: APPROXIMATEADDITIVITYTESTER accepts f , and hence by Lemma 5.6, g is pointwise 42η -close to linearity in $B(\mathbf{0}, r)$. Combined with Lemma 5.7, we get g is pointwise $42\eta(12R/r)$ -close to linearity in $B(\mathbf{0}, R)$. From Claim 5.8, we know that g and APPROXIMATE- g are pointwise $6\eta(12R/r)$ -close in $B(\mathbf{0}, R)$ with probability at least $1 - \varepsilon/4$. Using the triangle inequality, this implies that APPROXIMATE- g is pointwise $48\eta(12R/r)$ -close to some linear function with probability $\geq 1 - \varepsilon/4$. Moreover, from Observation 5.9, we get: APPROXIMATE- g is in fact pointwise $750Rn^{1.5}\eta$ -close to f , with probability at least $1 - \varepsilon/2$. With $R = 2\sqrt{n}$, following the guarantee of Claim 5.12, ensured by our assumption on η :

$$O(n^2\eta) \leq \frac{1}{100k^2} \min_{k \in [n]: a_k \neq 0} |a_k|, \text{ or equivalently } \eta \leq O\left(\min_{i \in [n]: f(e_i) \neq 0} \frac{|f(e_i)|}{(nk)^2}\right),$$

we get that FindInfBuckets(APPROXIMATE- $g, B, [r]$) will return at most k -influential variables of f with probability at least $1 - 128 \lceil \log k \rceil^2 / 5k$. Thus with probability at least $1 - \varepsilon/2 - 128 \lceil \log k \rceil^2 / 5k$, TEST- k -LINEAR will ACCEPT.

Soundness. Let f be ε -far from k -linearity. We will prove the contrapositive. We will show that if TEST- k -LINEAR does not reject f with probability at least $1 - \delta (\geq 2/3)$, then f is pointwise close to some k -linear function, with non-zero probability.

Note that if APPROXIMATEADDITIVITYTESTER rejects f with probability $\geq 1 - \delta$, we are done. So, let us consider the case when APPROXIMATEADDITIVITYTESTER accepts with probability $\geq \delta$. As in the completeness proof, from Lemma 5.5, and Lemma 5.7, we know

14:16 Testing Sparse Functions over the Reals

that g is pointwise $42\eta(12R/r)$ -close to linearity in $B(\mathbf{0}, R)$ with probability at least $1 - 12/125$, and from Claim 5.8, we have that g and APPROXIMATE- g are pointwise $6\eta(12R/r)$ -close in $B(\mathbf{0}, R)$ with probability at least $1 - \varepsilon/4$. This implies that APPROXIMATE- g is pointwise $48\eta(12R/r)$ -close to some linear function with probability at least $1 - \varepsilon/4 - 12/125$. Again, from Observation 5.9, we get: APPROXIMATE- g is pointwise $750Rn^{1.5}\eta$ -close to f , with probability at least $1 - \varepsilon/4$, implying now f must be pointwise $(48(12R/r) + 750Rn^{1.5})\eta$ -close to linearity, with probability at least $1 - \varepsilon/2 - 12/125$. Note, here $R = 2\sqrt{n}$, and $r = 1/50$.

With our assumption on η again ensuring the conditions for Claim 5.12 are met, i.e.,

$$(48(12R/r) + 750Rn^{1.5})\eta \leq \frac{1}{100k^2} \min_{k \in [n]: a_k \neq 0} |a_k|,$$

we get FindInfBuckets(APPROXIMATE- g , $B, [r]$) will return at most $8k$ -influential variables of f with probability at least $1 - 128\lceil \log k \rceil^2/5k$.

Since TEST- k -LINEAR accepts f with probability $\geq \delta$, this implies that the total number of influential variables returned by FindInfBuckets is at most k , with probability $\geq \delta$. Combining the above, we conclude that f is pointwise $(48(12R/r) + 750Rn^{1.5})\eta$ -close to a k -linear function, with probability at least $\delta - 12/125 - \varepsilon/2 - 128\lceil \log k \rceil^2/5k$. This concludes the proof.

Query complexity. From Theorem 5.4, we have: APPROXIMATEADDITIVITYTESTER performs $O(\frac{1}{\varepsilon})$ queries. From Claim 5.12, we have: FindInfBuckets performs $\tilde{O}(k \log k)$ queries. Combining them, we have: TEST- k -LINEAR performs $\tilde{O}(k \log k + \frac{1}{\varepsilon})$ queries in total. ◀

6 k -Sparse Low Degree Testing

In this section, we present and analyze our sparse low degree tester (Algorithm 4).

► **Theorem 6.1** (Generalization of Theorem 2.7). *Let $\eta < \min\{\varepsilon, 1/2^{2^n}\}$. Given η -approximate query access to $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that is bounded in $B(\mathbf{0}, 2d\sqrt{n})$, there exists a tester TEST- k -SPARSE (Algorithm 4) that performs $\tilde{O}(d^5 + \frac{d^2}{\varepsilon} + dk^3)$ queries and guarantees:*

- **COMPLETENESS:** *If f is a k -sparse, degree- d polynomial, then Algorithm 4 ACCEPTS with probability at least $1 - \varepsilon/4$.*
- **SOUNDNESS:** *If f is ε -far from all k -sparse, degree- d polynomials, then Algorithm 4 REJECTS with probability at least $2/3$.*

■ **Algorithm 4** TEST- k -SPARSE.

Inputs : η -approximate query oracle $\tilde{f}$ for $f : \mathbb{R}^n \rightarrow \mathbb{R}$, bounded in $B(\mathbf{0}, R)$, sparsity parameter $k \in \mathbb{N}$, proximity parameter $\varepsilon \in (0, 1)$, degree parameter $d \in \mathbb{N}$.

Output : ACCEPT iff f is k -sparse function.

- 1 Run APPROXLOWDEGREETESTER($f, d, \mathcal{N}(\mathbf{0}, I_n), \eta, \varepsilon, R, R$); // [6, Algorithm 9]
- 2 if APPROXLOWDEGREETESTER rejects then return REJECT ;
- 3 Call APPROX-POLY-SPARSITY-TEST(APPROXQUERY- g) (Algorithm 5).
- 4 if APPROX-POLY-SPARSITY-TEST(APPROXQUERY- g) accepts then
- 5 return ACCEPT; // APPROXQUERY- g in [6, Algorithm 10]
- 6 else return REJECT;

We will prove Theorem 6.1 in Section 6.2, after developing the necessary machinery. Algorithm 4 first invokes APPROXLOWDEGREETESTER to reject functions that are far from any low-degree polynomial. We record some useful claims about APPROXLOWDEGREETESTER, and its subroutines (APPROXCHARACTERIZATIONTEST, APPROXQUERY- g and APPROXQUERY- g -INBALL) from [5] and (its improvement in) [7]:

► **Theorem 6.2** ([7, Theorem 3.6]). *Let $d \in \mathbb{N}$, for $L > 0$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be bounded in the ball $B(\mathbf{0}, L)$, and for $\varepsilon \in (0, 1)$, $R > 0$, let $\mathcal{D}$ be an $(\varepsilon/4, R)$ -concentrated distribution. For $\alpha > 0$, $\beta \geq 2^{(2n)^{O(d)}}(R/L)^d \alpha$, given α -approximate query access to f , and sampling access to $\mathcal{D}$, there is an one-sided error, $O(d^5 + \frac{d^2}{\varepsilon})$ -query APPROXLOWDEGREETESTER which, distinguishes between the case when f is pointwise α -close to some degree- d polynomial and the case when, for every degree- d polynomial $h : \mathbb{R}^n \rightarrow \mathbb{R}$, $\Pr_{\mathbf{p} \sim \mathcal{D}}[|f(\mathbf{p}) - h(\mathbf{p})| > \beta] > \varepsilon$.*

► **Lemma 6.3** ([5, Lemma 4.4]). *Let $r = (4d)^{-6}$, $\delta = 2^{d+1}\alpha$, and $R > r$. If APPROXCHARACTERIZATIONTEST fails with probability at most $2/3$, then g is pointwise $2^{(2n)^{45d}}(R/L)^d \delta$ -close to a degree- d polynomial in $B(\mathbf{0}, 2dR\sqrt{n}/L)$. Furthermore, $\forall \mathbf{p} \in B(\mathbf{0}, 2dR\sqrt{n}/L)$, APPROXQUERY- $g(\mathbf{p})$ well approximates $g(\mathbf{p})$ w.h.p., i.e.,*

$$\Pr_{\mathbf{p} \sim \mathcal{D}} \left[|g(\mathbf{p}) - \text{APPROXQUERY-}g(\mathbf{p})| \leq \left(\frac{24dR\sqrt{n}}{Lr} \right)^d 2^{d+4}\delta \right] \geq 1 - \frac{\varepsilon}{4}.$$

To invoke these results, we assume: (i) f is bounded in $B(\mathbf{0}, R)$, i.e., we set $L = R$, and (ii) $\alpha = \eta$. Additionally, since we work over standard Gaussians, we set $R = 2d\sqrt{n}$.

Testing sparsity of polynomials given exact query access. As a preliminary, we design an algorithm for testing sparsity of *polynomial functions* $f : \mathbb{R}^n \rightarrow \mathbb{R}$, by extending the machinery of Hankel matrices developed in [52, 11] to the real numbers. Moreover, Observation 2.6 can be used to exactly determine whether a polynomial f is $(\leq k)$ -sparse.

► **Lemma 6.4.** *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a polynomial function (of any degree). Given exact query access to f , there is an algorithm that makes $2k + 1$ queries to f , and exactly tests whether $\|f\|_0 \leq k$ (returns ACCEPT), or $\|f\|_0 > k$ (returns REJECT) with error probability 0.*

6.1 Testing sparsity of polynomials given approximate query access

Now, instead of exact query access, we have η -approximate query $\tilde{f}$ to a polynomial f , which guarantees $|\tilde{f}(\mathbf{z}) - f(\mathbf{z})| \leq \eta$ for all points $\mathbf{z} \in \mathbb{R}^n$, giving us:

► **Observation 6.5.** *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and let $\tilde{f}$ be an η -approximate query oracle to f . Then for any $t \geq 1$ and $\mathbf{u} \in \mathbb{R}^n$, we can express $H_t(\tilde{f}, \mathbf{u}) = H_t(f, \mathbf{u}) + E_t(\mathbf{u})$, where $E_t(\mathbf{u})$ is a Hankel-structured noise matrix, s.t. $\|E_t(\mathbf{u})\|_\infty \leq \eta$, and $\|E_t(\mathbf{u})\|_{\text{op}} \leq \eta t$.*

We now present and analyze the correctness of Algorithm 5, using Theorem 2.12:

► **Theorem 6.6** (Sparsity testing with approximate queries). *Given η -approximate query access $\tilde{f}$ to a polynomial $f : \mathbb{R}^n \rightarrow \mathbb{R}$ of total degree $\leq d$, assuming $\eta \leq \frac{(\text{coeff}_{d_Q}(Q))^{\frac{1}{2}}}{(2(k+1))^{2\Theta(k^3d)}(\sigma_{\max}(H_{\mathbf{u}}))^k}$, wherein $H_{\mathbf{u}} = H_{k+1}(f, \mathbf{u})$, $Q_{\mathbf{u}} = (\det(H_{\mathbf{u}}))^2$, $d_Q = \deg(Q)$, and $\text{coeff}_{d_Q}^2(Q)$ is as in Theorem 4.6, APPROX-POLY-SPARSITY-TEST (Algorithm 5), in $O(dk^3)$ queries, guarantees:*

- (i) *If f has sparsity at most k , the algorithm always ACCEPTS, and*
- (ii) *If f has sparsity $> k$, the algorithm REJECTS with probability at least $\frac{2}{3}$.*

■ **Algorithm 5** APPROX-POLY-SPARSITY-TEST: Approx-query sparsity test for polynomials.

Inputs : η -approximate query oracle $\tilde{f}$ to a polynomial $f : \mathbb{R}^n \rightarrow \mathbb{R}$ of total degree $\leq d$, sparsity parameter $k \in \mathbb{N}$.

Output : ACCEPT iff f is a k -sparse polynomial.

- 1 Let $T \leftarrow 4 \min\{C, 1\} \cdot d(k+1)^2$, where C is the absolute constant in Theorem 4.6.
- 2 **for** $t \in [T]$ **do**
- 3 Sample $\mathbf{u}_t = (u_{t,1}, \dots, u_{t,n}) \sim \mathcal{N}(\mathbf{0}, I_n)$.
- 4 Compute $\mathbf{u}_t^i \triangleq (u_{t,1}^i, \dots, u_{t,n}^i)$, and $\tilde{f}(\mathbf{u}_t^i)$, for every $i \in \{0, 1, \dots, 2k\}$.
- 5 Compute the noisy Hankel matrix $H_{k+1}(\tilde{f}, \mathbf{u}_t)$ as in Definition 2.5.
- 6 Compute the smallest singular value $\sigma_{\min}^{(t)}$ of $H_{k+1}(\tilde{f}, \mathbf{u}_t)$.
- 7 **if** $\sigma_{\min}^{(t)} \leq \eta(k+1)$ **for all** $t \in [T]$ **then return** ACCEPT ($(\leq k)$ -sparse) ;
- 8 **else return** REJECT ($(> k)$ -sparse) ;

6.2 Analysis of k -Sparsity Low Degree Tester

We are now ready to analyze our sparse low degree tester (Algorithm 4).

Proof of Theorem 6.1. Let us start with completeness. **Completeness:** Since f is a k -sparse, degree- d polynomial, from Theorem 6.2, we have: APPROX-LOW-DEGREE-TESTER always accepts f , and hence, from Lemma 6.3, we have that g is pointwise $2^{(2n)^{45d}} R^d 2^{d+1} \eta$ -close to a degree- d polynomial, say h , in $B(\mathbf{0}, R)$. Moreover,

$$\Pr_{\mathbf{p} \sim \mathcal{D}} \left[|g(\mathbf{p}) - \text{APPROXQUERY-}g(\mathbf{p})| \leq \left(\frac{24d\sqrt{n}}{r} \right)^d 2^{2d+5} \eta \mid \mathbf{p} \in B(\mathbf{0}, 2d\sqrt{n}) \right] \geq 1 - \frac{\varepsilon}{2}.$$

$$\implies \Pr_{\mathbf{p} \sim \mathcal{D}} \left[|h(\mathbf{p}) - \text{APPROXQUERY-}g(\mathbf{p})| \leq \left(\left(\frac{24d\sqrt{n}}{r} \right)^d 2^{2d+5} + 2^{(2n)^{45d}} R^d 2^{d+1} \right) \eta \right] \geq 1 - \frac{\varepsilon}{4}. \quad (1)$$

By setting $\eta = 1/2^{2^n}$, the conditions of Theorem 6.6 are met, giving us that APPROX-POLY-SPARSITY-TEST(APPROXQUERY- g) (Algorithm 5) will ACCEPT with probability $\geq 1 - \varepsilon/4$. So with probability at least $1 - \varepsilon/4$, TEST- k -SPARSE (Algorithm 4) will accept f .

Soundness. Let f be ε -far from all k -sparse, degree- d polynomials. We will show that if Algorithm 4 accepts with probability at least $1/3$, then f must be ε -close to some k -sparse, degree- d polynomial. From the premise, APPROX-LOW-DEGREE-TESTER also accepts f with probability at least $1/3$. Then, as in completeness, from Lemma 6.3, we have: g is pointwise $2^{(2n)^{45d}} R^d 2^{d+1} \eta$ -close to some degree- d polynomial, say $h(\mathbf{x}) = \sum_{\|\mathbf{M}_i\|_1 \leq d} a_i \mathbf{M}_i(\mathbf{x})$, i.e., for all $\mathbf{p} \in B(\mathbf{0}, R)$, $|g(\mathbf{p}) - h(\mathbf{p})| \leq 2^{(2n)^{45d}} R^d 2^{d+1} \eta$, and (1) still holds.

From the premise, APPROX-POLY-SPARSITY-TEST(APPROXQUERY- g) (Algorithm 5) does not reject with probability $\geq 1/3$. In this case, as long as the closeness of APPROXQUERY- g and h satisfies the assumption in Theorem 6.6, h will be k -sparse. The assumption is:

$$(3 \cdot 2^{(2n)^{45d}} + 1) \eta \leq \frac{(\text{coeff}_{d_Q}(Q))^{\frac{1}{2}}}{(2(k+1))2^{\Theta(k^3d)} (\sigma_{\max}(H_{\mathbf{u}}))^k}, \quad (2)$$

We first prove a structural result. Let $\mathcal{J}_k$ denote the set of all k -juntas on n variables.

► **Theorem 7.3.** *If $\text{dist}(f, \mathcal{J}_k) \geq \varepsilon$, then $\forall S \subseteq [n] : |S| \leq k$, with $\bar{S} \triangleq [n] \setminus S$, $\text{Infl}_f(\bar{S}) \geq \varepsilon$.*

Next we prove a lemma that connects the influence of the union of two sets of variables with the influences of individual sets.

► **Lemma 7.4** (Sub-additivity of Influence). *For every $f : \mathbb{R}^n \rightarrow \mathbb{R}$, and any $S, T \subseteq [n]$,*

$$\max\{\text{Infl}_f(S), \text{Infl}_f(T)\} \leq \text{Infl}_f(S \cup T) \leq \text{Infl}_f(S) + \text{Infl}_f(T).$$

Next we show, if f is far from being a k -junta, the influence of $\bar{S}$ in Algorithm 6 (i.e. all buckets not yet identified as influential) is high, unless we identify more than k parts.

► **Lemma 7.5.** *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $\mathcal{B} = \{B_1, \dots, B_r\}$ be a random partition of $[n]$, for $r = \Theta(k^2)$. If $\text{dist}(f, \mathcal{J}_k) \geq \varepsilon$, with probability $\geq 99/100$ over the randomness of the partition, we have $\text{Infl}_f(\bar{S}) \geq \varepsilon/4$ for any $S \subseteq [n]$ which is a union of at most k parts of $\mathcal{B}$.*

7.1 Proof of correctness of TEST- k -JUNTA

As Algorithm 6 invokes Algorithm 2, we first prove its correctness for juntas.

► **Definition 7.6** (Influential variables and buckets). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$, and $k \in \mathbb{N}$. A variable $x_i, i \in [n]$ is influential with respect to f , if for some $\mathbf{x} \triangleq (x_1, \dots, x_n) \in \mathbb{R}^n$, only changing x_i changes the value of $f(\mathbf{x})$. A bucket $B \subseteq [n]$ is said to be influential if it contains the index of at least one influential variable, and non-influential otherwise.*

Let $\mathcal{E}$ denote the event that for $\mathbf{x}, \mathbf{y} \sim \mathcal{N}(\mathbf{0}, I_n)$, $\mathbf{x}, \mathbf{y} \in B(\mathbf{0}, 2\sqrt{n})$. By Fact 4.3, we have $\Pr_{\mathbf{x}, \mathbf{y} \sim \mathcal{N}(\mathbf{0}, I_n)}[\bar{\mathcal{E}}] = \Pr_{\mathbf{x}, \mathbf{y} \sim \mathcal{N}(\mathbf{0}, I_n)}[\mathbf{x}, \mathbf{y} \notin B(\mathbf{0}, 2\sqrt{n})] \leq 0.01$. Let

$$\kappa \triangleq \min_{\substack{V \subseteq [n] \\ V \text{ is influential}}} \left\{ \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \mathcal{N}(\mathbf{0}, I_n)} [|f(\mathbf{x}_V \mathbf{y}_{\bar{V}}) - f(\mathbf{y})| \mid \mathcal{E}] \right\}.$$

We use FindInfBucket (Algorithm 2) for testing k -linearity as well. However, the proof of correctness of Algorithm 2 in Section 5, presented in Claim 5.11, holds only for linear functions, which may not be the case in general. So, we prove the same for juntas too:

► **Claim 7.7** (Correctness of Algorithm 2 for juntas). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be given via the η -approximate query $\tilde{f}$, where $\eta \leq \min\{\kappa/4, 1/(1000k^2 \log^2 k)\}$, $B = \{B_1, \dots, B_r\}$ be a partition of $[n]$, and $\emptyset \neq S \subseteq [r]$. With $\kappa \geq \max\{2\|f\|_{\infty, B(\mathbf{0}, 2\sqrt{n})}, \varepsilon/(4k^2)\}$, we have:*

1. If none of the B_i 's are influential, FindInfBucket(f, B, S) always returns $\emptyset$ and performs exactly 2 queries to f .
2. Otherwise, with probability at least $1 - 16\eta \lceil \log |S| \rceil^2 / \kappa$, FindInfBucket(f, B, S) returns B_j for some $j \in S$ which is κ -influential and performs $\leq 8 \lceil \log(|S|) \rceil^2$ queries to f .

We are now ready to prove the main theorem of this section:

Proof of Theorem 7.1. Let us start with the completeness proof.

Completeness. The argument of completeness of Algorithm 6 holds easily, as when f is a k -junta, at most k of the buckets in the random partition $\mathcal{B}$ will be influential. From Claim 7.7, we get that each execution of the For loop increments I by: either the index of an influential bucket (if it finds one), or nothing, while also removing the bucket from future loop executions. Hence after all its executions, $|I|$ is incremented at most k times, falsifying the check in line 6, making f ACCEPT.

Soundness. Here our goal is to prove if f is ε -far from all k -juntas, then Algorithm 6 rejects it with probability $\geq 2/3$. In particular, if f is ε -far from being k -junta, and S is the union of at most k -parts of $\mathcal{I}$, then Line 7 of Algorithm 6 will be satisfied w.h.p.

By Lemma 7.5, we have: if $|I| \leq k$, $\text{Infl}_f(S) \geq \varepsilon/4$ ($S = [n] \setminus I$ in the execution of Algorithm 6). During the execution, when exactly k influential buckets have been identified, $\text{Infl}_f(S) \geq \varepsilon/4$, $|S| = O(k^2) - k = O(k^2)$, and thus, by Lemma 7.4, there must exist a bucket $B_i, i \in S$ such that $\text{Infl}_f(B_i) \geq \varepsilon/(4k^2)$. This bucket will then be output by $\text{FindInflBucket}(f, B, S)$. So, until $k+1$ influential buckets have been identified, each invocation of $\text{FindInflBucket}(f, B, S)$ in an execution of its For loop, returns an influential bucket with probability $\geq 1 - 64\eta \lceil \log k \rceil^2 / \kappa$. Treating the outcome of each invocation as a geometric random variable, we conclude: to recover at least $k+1$ such buckets with probability $\geq 2/3$,

$$k / (1 - 64\eta \lceil \log k \rceil^2 / \kappa) = k + k \sum_{i=1}^{\infty} (64\eta \lceil \log k \rceil^2 / \kappa)^i = O(k/\varepsilon)$$

iterations of For loop suffice. When $|I| > k$, in line 6, algorithm rejects with same probability.

Query Complexity. From Claim 7.7, we know: each invocation of $\text{FindInflBucket}(f, B, S)$ performs $\leq 4 \log |S|$ queries to f . With $O(k/\varepsilon)$ such invocations in total, and $|S| = r = O(k^2)$, the overall query complexity thus is $O((k \log k)/\varepsilon)$. ◀

8 Lower Bounds

► **Theorem 2.11.** *Given exact query access to $f : \mathbb{R}^n \rightarrow \mathbb{R}$, some $k, d \in \mathbb{N}$ and a distance parameter $\varepsilon \in (0, 1)$, $\Omega(\max\{k, \frac{1}{\varepsilon}\})$ queries are necessary for testing (i) k -linear functions, (ii) k -sparse degree- d polynomials, and (iii) k -juntas, with success probability at least $2/3$.*

The lower bound for testing k -sparse degree- d polynomials is improved to $\Omega(\max\{d, k, \frac{1}{\varepsilon}\})$.

For proving the lower bounds in the exact query model (which covers approximate queries), we follow the approach of [19], i.e., we show the hardness of testing by reducing from SET-DISJOINTNESS problem, a canonical hard problem in the field of communication complexity. For brevity, we only present the lower bound argument for k -linearity.

$\Omega(\frac{1}{\varepsilon})$ queries for linearity. For any $\varepsilon \in (0, 1)$, it is a folklore result that distinguishing if a function f is linear, or ε -far from being linear in ℓ_1 -distance, requires $\Omega(1/\varepsilon)$ queries [42].

$\Omega(d)$ queries for degree- d polynomial testing. It is another folklore result that given a proximity parameter $\varepsilon \in (0, 1)$, $\Omega(d)$ queries are necessary to distinguish if an unknown function f is a degree- d polynomial, or a degree- $(d+1)$ (or $(d-1)$) polynomial (and hence, is ε -far from all degree- d polynomials) with probability at least $2/3$.

To prove the $\Omega(k)$ query lower bound for k -linearity, we use a reduction from SET-DISJOINTNESS which is known to require $\Omega(k)$ communication complexity [54].

Connection between Communication and Query complexity. Consider two functions f and g , a property $\mathcal{P}$, $h \triangleq f - g$, and a communication problem $C_{h, \mathcal{P}}$: Alice and Bob receive f and g , respectively, and they want to decide if $h \in \mathcal{P}$, or f is ε -far from $\mathcal{P}$. [19] (Lemma 2.2) proved that $\mathcal{R}(C_{h, \mathcal{P}})$, the randomized communication complexity of $C_{h, \mathcal{P}} \leq 2Q(\mathcal{P})$, the query complexity of deciding the property $\mathcal{P}$.

Construction of the hard instances. Given $A \subseteq [n] : |A| = k$, Alice constructs a polynomial $f = \sum_{i \in A} x_i$. Similarly Bob, given $B \subseteq [n], |B| = k$, constructs a polynomial $g = \sum_{i \in B} x_i$. Note that $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$. Let $h \triangleq f - g$. If $|A \cap B| = 1$, h would be a $(2k - 2)$ linear function, whereas if $|A \cap B| = 0$, h would be a $2k$ -linear function.

► **Lemma 8.1.** *Let f_1 be a $(2k + 2)$ -linear function, and f_2 be a $2k$ -linear function with coefficients from $\{0, 1\}$. Under the standard Gaussian distribution $\mathcal{N}(\mathbf{0}, I_n)$, f_1 is $\Omega(1)$ -far from f_2 in ℓ_1 -distance with probability at least $2/3$.*

References

- 1 Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing reed-muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005. doi:10.1109/TIT.2005.856958.
- 2 Sigal Ar, Manuel Blum, Bruno Codenotti, and Peter Gemmell. Checking approximate computations over the reals. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of Computing*, pages 786–795, 1993. doi:10.1145/167088.167288.
- 3 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998. doi:10.1145/278298.278306.
- 4 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, January 1998. doi:10.1145/273865.273901.
- 5 Vipul Arora, Arnab Bhattacharyya, Noah Fleming, Esty Kelman, and Yuichi Yoshida. Low degree testing over the reals. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 738–792. SIAM, 2023. doi:10.1137/1.9781611977554.ch31.
- 6 Vipul Arora, Arnab Bhattacharyya, Philips George John, and Sayantan Sen. Testing sparse functions over the reals. *CoRR*, abs/2603.28061, 2026. doi:10.48550/arXiv.2603.28061.
- 7 Vipul Arora, Esty Kelman, and Uri Meir. On optimal testing of linearity. In *Proceedings of the 2025 Symposium on Simplicity in Algorithms (SOSA)*, pages 65–76. SIAM, 2025. doi:10.1137/1.9781611978315.5.
- 8 Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 21–30. IEEE, IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.64.
- 9 Roksana Baleshazar, Deeparnab Chakrabarty, Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and C. Seshadhri. Optimal unateness testers for real-valued functions: Adaptivity helps. *CoRR*, abs/1703.05199, 2017. doi:10.48550/arXiv.1703.05199.
- 10 Yiqiao Bao, Anindya De, Shivam Nadimpalli, Rocco A. Servedio, and Nathan White. Testing noisy low-degree polynomials for sparsity, 2025. doi:10.48550/arXiv.2511.07835.
- 11 Michael Ben-Or and Prasoona Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA, STOC '88*, pages 301–309, New York, NY, USA, 1988. ACM. doi:10.1145/62212.62241.
- 12 Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lp-testing. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, STOC '14, pages 164–173, New York, NY, USA, 2014. ACM. doi:10.1145/2591796.2591887.
- 13 Arnab Bhattacharyya, Eldar Fischer, Hamed Hatami, Pooya Hatami, and Shachar Lovett. Every locally characterized affine-invariant property is testable. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 429–436, 2013. doi:10.1145/2488608.2488662.

- 14 Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 488–497. IEEE, 2010. doi:10.1109/FOCS.2010.54.
- 15 Arnab Bhattacharyya and Yuichi Yoshida. *Property Testing - Problems and Techniques*. Springer, 2022. doi:10.1007/978-981-16-8622-1.
- 16 Hadley Black, Iden Kalemaj, and Sofya Raskhodnikova. Isoperimetric Inequalities for Real-Valued Functions with Applications to Monotonicity Testing. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2023.25.
- 17 Eric Blais. Improved bounds for testing juntas. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 317–330. Springer, 2008. doi:10.1007/978-3-540-85363-3_26.
- 18 Eric Blais. Testing juntas nearly optimally. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 151–158, 2009. doi:10.1145/1536414.1536437.
- 19 Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *computational complexity*, 21:311–358, 2012. doi:10.1007/s00037-012-0040-x.
- 20 Eric Blais and Daniel Kane. Tight bounds for testing k-linearity. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 435–446. Springer, 2012. doi:10.1007/978-3-642-32512-0_37.
- 21 Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavl'tsev. Lower bounds for testing properties of functions over hypergrid domains. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 309–320. IEEE, 2014. doi:10.1109/CCC.2014.38.
- 22 Eric Blais, Amit Weinstein, and Yuichi Yoshida. Partially symmetric functions are efficiently isomorphism testable. *SIAM Journal on Computing*, 44(2):411–432, 2015. doi:10.1137/140971877.
- 23 Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, 2020. doi:10.1017/9781108755528.
- 24 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 73–83. ACM, 1990. doi:10.1145/100216.100225.
- 25 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993. doi:10.1016/0022-0000(93)90044-W.
- 26 Nader H Bshouty. An optimal tester for k-linear. *Theoretical Computer Science*, 950:113759, 2023. doi:10.1016/j.tcs.2023.113759.
- 27 Anthony Carbery and James Wright. Distributional and L^q Norm Inequalities for Polynomials over Convex Bodies in $\mathbb{R}^n$. *Mathematical Research Letters*, 8:233–248, 2001.
- 28 Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and lipschitz testing over hypercubes and hypergrids. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 419–428. ACM, 2013. doi:10.1145/2488608.2488661.
- 29 Xi Chen, Adam Freilich, Rocco A. Servedio, and Timothy Sun. Sample-based high-dimensional convexity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 37:1–37:20, 2017. doi:10.4230/LIPIcs.APPROX-RANDOM.2017.37.
- 30 Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the query complexity of non-adaptive junta testing. *J. ACM*, 65(6):40:1–40:18, 2018. doi:10.1145/3213772.

- 31 Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, 90(6):301–305, 2004. doi:10.1016/j.ipl.2004.01.023.
- 32 Anindya De, Elchanan Mossel, and Joe Neeman. Is your function low dimensional? In *Conference on Learning Theory*, pages 979–993. PMLR, 2019. URL: <http://proceedings.mlr.press/v99/de19a.html>.
- 33 Ilias Diakonikolas, Homin K Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A Servedio, and Andrew Wan. Testing for concise representations. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 549–558. IEEE, 2007. doi:10.1109/FOCS.2007.32.
- 34 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, June 2007. doi:10.1145/1236457.1236459.
- 35 Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 97–108. Springer, 1999. doi:10.1007/978-3-540-48413-4_10.
- 36 Regina C. Elandt. The folded normal distribution: Two methods of estimating parameters from moments. *Technometrics*, 3(4):551–562, 1961. URL: <http://www.jstor.org/stable/1266561>.
- 37 Funda Ergün, S Ravi Kumar, and Ronitt Rubinfeld. Checking approximate computations of polynomials and functional equations. *SIAM Journal on Computing*, 31(2):550–576, 2001. doi:10.1137/S0097539798337613.
- 38 Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Approximating clique is almost np-complete (preliminary version). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, October 1-4, 1991*, pages 2–12. IEEE Computer Society, 1991. doi:10.1109/SFCS.1991.185341.
- 39 Renato Ferreira Pinto Jr. Directed Poincaré Inequalities and L^1 Monotonicity Testing of Lipschitz Functions. In Nicole Megow and Adam Smith, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023)*, volume 275 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:18, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.61.
- 40 Renato Ferreira Pinto Jr. Directed isoperimetry and monotonicity testing: A dynamical approach. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2295–2305. IEEE, 2024. doi:10.1109/FOCS61266.2024.00134.
- 41 Eldar Fischer. The art of uninformed decisions. *Bull. EATCS*, 75:97, 2001.
- 42 Eldar Fischer. A basic lower bound for property testing, 2024. doi:10.48550/arXiv.2403.04999.
- 43 Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753–787, 2004. doi:10.1016/j.jcss.2003.11.004.
- 44 Noah Fleming and Yuichi Yoshida. Distribution-free testing of linear functions on $\mathbb{R}^n$. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, Seattle, Washington, USA, January 12-14, 2020*, volume 151 of *LIPIcs*, pages 22:1–22:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.22.
- 45 Katalin Friedl and Madhu Sudan. Some improvements to total degree tests. In *Proceedings Third Israel Symposium on the Theory of Computing and Systems*, pages 190–198. IEEE, 1995. doi:10.1109/ISTCS.1995.377032.
- 46 Peter Gemmell, Richard Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *STOC*, volume 91, pages 32–42. Citeseer, 1991. doi:10.1145/103418.103429.

- 47 Itay Glazer and Dan Mikulincer. Anti-concentration of polynomials: Dimension-free covariance bounds and decay of fourier coefficients. *Journal of Functional Analysis*, 283(9):109639, 2022. doi:10.1016/j.jfa.2022.109639.
- 48 O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 339–348, 1996. doi:10.1109/SFCS.1996.548493.
- 49 Oded Goldreich. On testing computability by small width obdds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings*, pages 574–587. Springer, 2010. doi:10.1007/978-3-642-15369-3_43.
- 50 Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017. doi:10.1017/9781108135252.
- 51 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. doi:10.1137/100785429.
- 52 Elena Grigorescu, Kyomin Jung, and Ronitt Rubinfeld. A local decision test for sparse polynomials. *Inf. Process. Lett.*, 110(20):898–901, 2010. doi:10.1016/j.ip1.2010.07.012.
- 53 Shirley Halevy and Eyal Kushilevitz. Distribution-free property-testing. *SIAM Journal on Computing*, 37(4):1107–1138, 2007. doi:10.1137/050645804.
- 54 Johan Håstad and Avi Wigderson. The randomized communication complexity of set disjointness. *Theory of Computing*, 3(1):211–219, 2007. doi:10.4086/toc.2007.v003a011.
- 55 Roger A. Horn and Charles R. Johnson. *Topics in matrix analysis*. Cambridge University Press, 1991. doi:10.1017/CB09780511840371.
- 56 Roger A. Horn and Charles R. Johnson. *Matrix Analysis, 2nd Ed.* Cambridge University Press, 2012. doi:10.1017/CB09781139020411.
- 57 Charanjit S Jutla, Anindya C Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. *Random Structures & Algorithms*, 35(2):163–193, 2009. doi:10.1002/rsa.20262.
- 58 Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM Journal on Computing*, 36(3):779–802, 2006. doi:10.1137/S0097539704445615.
- 59 Praveesh Kothari, Amir Nayyeri, Ryan O'Donnell, and Chenggang Wu. Testing surface area. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1204–1214. SIAM, SIAM, 2014. doi:10.1137/1.9781611973402.89.
- 60 Marek Kuczma. *An Introduction to the Theory of Functional Equations and Inequalities*. Birkhäuser Basel, 2009.
- 61 Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A Servedio. Testing halfspaces. *SIAM Journal on Computing*, 39(5):2004–2047, 2010. doi:10.1137/070707890.
- 62 Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco Servedio. Testing (subclasses of) halfspaces. In *Property testing: current research and surveys*, pages 334–340. Springer, 2010. doi:10.1007/978-3-642-16367-8_27.
- 63 Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A Servedio. Testing ± 1 -weight halfspace. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 646–657. Springer, 2009. doi:10.1007/978-3-642-03685-9_48.
- 64 Joe Neeman. Testing surface area with arbitrary accuracy. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 393–397. ACM, 2014. doi:10.1145/2591796.2591807.
- 65 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006. doi:10.1016/j.jcss.2006.03.002.

- 66 Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM Journal on Discrete Mathematics*, 16(1):20–46, 2002. doi:10.1137/S0895480101407444.
- 67 Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcg characterization of np. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 475–484, 1997. doi:10.1145/258533.258641.
- 68 Dana Ron. Algorithmic and analysis techniques in property testing. *Found. Trends Theor. Comput. Sci.*, 2009. doi:10.1561/0400000029.
- 69 Dana Ron et al. Property testing: A learning theory perspective. *Foundations and Trends® in Machine Learning*, 1(3):307–402, 2008. doi:10.1561/2200000004.
- 70 Ronitt Rubinfeld and Madhu Sudan. Self-Testing Polynomial Functions Efficiently and Over Rational Domains. In *Proceedings of the 3rd Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms (SODA)*, pages 23–32, 1992. URL: <http://dl.acm.org/citation.cfm?id=139404.139410>.
- 71 Ronitt Rubinfeld and Madhu Sudan. Robust Characterizations of Polynomials with Applications to Program Testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.
- 72 Mert Sağlam. Near Log-Convexity of Measured Heat in (Discrete) Time and Consequences. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 967–978. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00095.
- 73 Alex Samorodnitsky. Low-degree tests at large distances. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 506–515, 2007. doi:10.1145/1250790.1250864.