# *Robust Testing of Low Dimensional Functions*

# Robust Testing of Low Dimensional Functions

Anindya De
University of Pennsylvania
Philadelphia, Pennsylvania, USA
anindyad@cis.upenn.edu

Elchanan Mossel
MIT
Cambridge, Massachusetts, USA
elmos@mit.edu

Joe Neeman
UT Austin
Austin, Texas, USA
jneeman@math.utexas.edu

## ABSTRACT

A natural problem in high-dimensional inference is to decide if a classifier $f : \mathbb{R}^n \to \{-1, 1\}$ depends on a small number of linear directions of its input data. Call a function $g : \mathbb{R}^n \to \{-1, 1\}$, a linear $k$-junta if it is completely determined by some $k$-dimensional subspace of the input space. A recent work of the authors showed that linear $k$-juntas are testable. Thus there exists an algorithm to distinguish between:
(1) $f : \mathbb{R}^n \to \{-1, 1\}$ which is a linear $k$-junta with surface area $s$.
(2) $f$ is $\epsilon$-far from any linear $k$-junta with surface area $(1 + \epsilon)s$.
The query complexity of the algorithm is independent of the ambient dimension $n$.

Following the surge of interest in noise-tolerant property testing, in this paper we prove a noise-tolerant (or robust) version of this result. Namely, we give an algorithm which given any $c > 0$, $\epsilon > 0$, distinguishes between:
(1) $f : \mathbb{R}^n \to \{-1, 1\}$ has correlation at least $c$ with some linear $k$-junta with surface area $s$.
(2) $f$ has correlation at most $c - \epsilon$ with any linear $k$-junta with surface area at most $s$.
The query complexity of our tester is $k^{\mathrm{poly}(s/\epsilon)}$. Using our techniques, we also obtain a fully noise tolerant tester with the same query complexity for any class $C$ of linear $k$-juntas with surface area bounded by $s$. As a consequence, we obtain a fully noise tolerant tester with query complexity $k^{O(\mathrm{poly}(\log k/\epsilon))}$ for the class of intersection of $k$-halfspaces (for constant $k$) over the Gaussian space. Our query complexity is independent of the ambient dimension $n$. Previously, no non-trivial noise tolerant testers were known even for a single halfspace.

## CCS CONCEPTS

• **Mathematics of computing** → **Probabilistic algorithms**; **Dimensionality reduction**; • **Computing methodologies** → *Feature selection*; *Spectral methods.*

## KEYWORDS

Property testing, Junta testing, Gaussian analysis.


*STOC '21, June 21–25, 2021, Virtual, Italy*

ACM ISBN 978-1-4503-8053-9/21/06…$15.00
https://doi.org/10.1145/3406325.3451115

**ACM Reference Format:**
Anindya De, Elchanan Mossel, and Joe Neeman. 2021. Robust Testing of Low Dimensional Functions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3406325.3451115

## 1 INTRODUCTION

To motivate our setting, consider the classical notion of a *Boolean* junta: a function $f : \{-1, 1\}^n \to \{-1, 1\}$ is said to be a $k$-junta if there are some $k$ coordinates $i_1, \ldots, i_k \in [n]$ such that $f(x)$ only depends on $x_{i_1}, \ldots, x_{i_k}$. The fundamental results for testing juntas were obtained more than a decade ago; more recently, spurred by motivation from several directions, several variants have appeared. Most importantly for this work are the notions of *tolerant testing*, in which we estimate the distance to the class of juntas (as opposed to the usual testing, where we are simply testing membership); and *linear juntas*, a natural continuum generalization of Boolean juntas. In the current work, we combine these two perspectives and show that linear juntas are noise-tolerantly testable.

### 1.1 Tolerant Junta Testing

Recall that a property testing algorithm for a class of functions $C$ is an algorithm which, given oracle access to an $f : \{-1, 1\}^n \to \{-1, 1\}$ and a distance parameter $\epsilon > 0$, satisfies

(1) If $f \in C$, then the algorithm accepts with probability at least $2/3$;
(2) If $\mathrm{dist}(f, g) \geq \epsilon$ for every $g \in C$, then the algorithm rejects with probability at least $2/3$. Here, we define $\mathrm{dist}(f, g) = \Pr_{\mathbf{x} \in \{-1,1\}^n}[f(\mathbf{x}) \neq g(\mathbf{x})]$.

The principal measure of the efficiency of the algorithm is its *query complexity*. Also, the precise value of the confidence parameter is irrelevant and $2/3$ can be replaced by any constant $1/2 < c < 1$.

Fischer *et al.*[22] were the first to study the problem of testing $k$-juntas and showed that $k$-juntas can be tested with query complexity $\tilde{O}(k^2/\epsilon)$. The crucial feature of their algorithm is that the query complexity is independent of the ambient dimension $n$. Since then, there has been a long line of work on testing juntas [4, 5, 14, 15, 44] and it continues to be of interest. The flagship result is that $k$-juntas can be tested with $\tilde{O}(k/\epsilon)$ queries and this is tight [5, 15]. While the initial motivation to study this problem came from long-code testing [3, 38] (related to PCPs and inapproximability), another strong motivation comes from the *feature selection* problem in machine learning (see, e.g. [7, 9]).

*Tolerant testing.* The definition of property testing above requires the algorithm to accept if and only if $f \in C$. However, for many applications, it is important consider a *noise-tolerant* definition

of property testing. In particular, Parnas, Ron and Rubinfeld [37] introduced the following definition of noise tolerant testers.

**Definition 1.1.** *For constants $1/2 > c_u > c_\ell \geq 0$ and a function class $C$, a $(c_u, c_\ell)$-noise tolerant tester for $C$ is an algorithm which given oracle access to a function $f : \{-1,1\}^n \to \{-1,1\}$*

(1) *accepts with probability at least $2/3$ if $\min_{g \in C} \mathrm{dist}(f,g) \leq c_\ell$.*
(2) *rejects with probability at least $2/3$ if $\min_{g \in C} \mathrm{dist}(f,g) \geq c_u$.*

*Further, a tester which is noise tolerant for any (given) $c_u > c_\ell \geq 0$ is said to be a "*fully noise tolerant*" tester.*

The restriction $c_u, c_\ell < 1/2$ comes from the fact that most natural classes $C$ are closed under complementation – i.e., if $g \in C$, then $-g \in C$. For such a class $C$ and for any $f$, $\min_{g \in C} \mathrm{dist}(f,g) \leq 1/2$. Further, note that the standard notion of property testing corresponds to a $(\epsilon, 0)$-noise tolerant tester.

The problem of testing juntas becomes quite challenging in the presence of noise. Parnas *et al.* [37] observed that any tester whose (individual) queries are uniformly distributed are inherently noise tolerant in a very weak sense. In particular, [21] used this observation to show that the junta tester of [22] is in fact a $(\epsilon, \mathrm{poly}(\epsilon/k))$-noise tolerant tester for $k$-juntas – note that $c_\ell$ is quite small, namely $\mathrm{poly}(\epsilon/k)$. Later, Chakraborty *et al.* [12] showed that the tester of Blais [5] yields a $(C\epsilon, \epsilon)$ tester (for some large but fixed $C > 1$) with query complexity $\exp(k/\epsilon)$. Recently, there has been a surge of interest in tolerant junta testing. On one hand, Levi and Waingarten showed that there are constants $1/2 > \epsilon_1 > \epsilon_2 > 0$ such that any non-adaptive $(\epsilon_1, \epsilon_2)$ tester requires $\tilde{\Omega}(k^2)$ non-adaptive queries. Contrast this with the result of Blais [5] who showed that there is a non-adaptive tester for $k$-juntas with $O(k^{3/2})$ queries when there is no noise. In particular, this shows a gap between testing in the noisy and noiseless case.

In the opposite (i.e., algorithmic) direction a sequence of recent works improved on the results of [12]. First, Blais *et al.* [6] improved on the results of [12] by obtaining a small and explicit value of $C$. Finally, De, Mossel and Neeman [18] gave a fully noise tolerant tester for $k$-juntas on the Boolean cube with query complexity $O(2^k \cdot \mathrm{poly}(k/\epsilon))$.

## 1.2 Linear Junta Testing

In a recent work, De, Mossel and Neeman [17] initiated the study of property testing of *linear juntas*. A function $f : \mathbb{R}^n \to [-1,1]$ is said to be a *linear $k$-junta* if there are $k$ unit vectors $u_1, \ldots, u_k \in \mathbb{R}^n$ and $g : \mathbb{R}^k \to [-1,1]$ such that $f(x) = g(\langle u_1, x\rangle, \ldots, \langle u_k, x\rangle)$. In other words, $f$ is a linear $k$-junta if there is a subspace $E := \mathrm{span}(u_1, \ldots, u_k)$ of $\mathbb{R}^n$ such that $f(x)$ depends only on the projection of $x$ on the subspace $E$. The class of linear $k$-juntas is the $\mathbb{R}^n$-analogue of the class of $k$-juntas on the Boolean cube

We note that the family of linear $k$-juntas includes important classes of functions that have been studied in the learning and testing literature. Notably it includes:

- Boolean juntas: If $h : \{-1,1\}^n \to \{-1,1\}$ is a Boolean junta, then the function $f(x) : \mathbb{R}^n \to \{-1,1\}$ defined as $f(x) = h(\mathrm{sgn}(x_1), \ldots, \mathrm{sgn}(x_n))$ is a linear $k$-junta.
- Functions of halfspaces: Linear $k$-juntas include as a special case both halfspaces and intersections of $k$-halfspaces. The testability of halfspaces was studied in [33, 41].

The focus of the paper is on property testing of linear $k$-juntas. Observe that to formally define a testing algorithm, we need to define a notion of distance between functions $f$ and $g$ on $\mathbb{R}^n$. In this work, we will use the $L^2(\gamma)$ metric, where $\gamma$ is the standard Gaussian measure. That is, the distance between $f$ and $g$ is $(\mathbf{E}_{\mathbf{x}\sim\gamma}[(f(\mathbf{x}) - g(\mathbf{x}))^2])^{1/2}$. Note that this reduces to $2\Pr_{\mathbf{x}\sim\gamma}[f(\mathbf{x}) \neq g(\mathbf{x})]$ when $f$ and $g$ are Boolean functions. The choice of the standard Gaussian measure is well-established in the areas of learning and testing [2, 13, 20, 26, 28, 29, 33, 36, 46]. It is particularly natural in our setup since the Gaussian measure is invariant under many linear transformations, e.g., all rotations.

De, Mossel and Neeman [17] obtained an algorithm for testing linear-$k$-juntas: given query access to $f : \mathbb{R}^n \to \{-1,1\}$, it makes $\mathrm{poly}(k \cdot s/\epsilon)$ queries and distinguishes between

(1) $f$ is a linear-$k$-junta with surface area at most $s$ versus
(2) $f$ is $\epsilon$-far from any linear $k$-junta with surface area at most $s(1+\epsilon)$.

Here surface area of $f$ refers to the Gaussian surface area [30] of the set $f^{-1}(1)$ [30] – see Definition 2.7 for the precise definition. Further, [17] showed that a polynomial dependence on $s$ is necessary for any non-adaptive tester and consequently, an $\Omega(\log s)$ dependence is necessary for any tester[1]. Informally, without any smoothness assumption a linear junta (even a linear 1-junta on $\mathbb{R}^2$) can look arbitrarily random to any finite number of queries. Crucially, [17] achieves a query complexity which is independent of the ambient dimension $n$ – thus, qualitatively matching the guarantee for junta testing on the Boolean cube.

## 1.3 Our Results: Noise Tolerant Testing of Linear-Juntas

In this paper, our focus is on the problem of noise tolerant testing of linear juntas. The original motivation of [17] was for dimension reduction in statistical and ML models involving real valued data. Modern ML models are often overparametrized, but are nevertheless suspected to output a predictor that is low-dimensional in some sense. The classical notion of juntas is not appropriate for measuring dimensionality here, because there is no natural choice of basis in many statistical models including PCA, ICA, kernel learning, or deep learning. This motivates the notion of a linear junta. The problem of testing linear-juntas is thus closely related to the problem of *model compression* in machine learning, whose goal is to take a complex predictor/classifier function and to output a simpler predictor/classifier (see e.g. [11]). Model compression is extensively studied in the context of deep nets, see e.g., [1], and follow up work, where the models are often rotationally invariant (with the caveat that the regularization often used in optimization might not be). Thus as a motivating example, [17] asked if given a complex deep net classifier, is there a classifier that has essentially the same performance and depends only on $k$ of the features? Observe that this is essentially the same question as asking whether the deep net classifier is a linear $k$-junta.

The main shortcoming of the motivation in [17] is that it is unrealistic to expect that in any of the statistical and ML models considered, the function constructed will be *exactly identical* to

[1] Recall that in a non-adaptive tester, the query points are chosen independently of the target $f$.

a function of a few linear direction. Rather, we only expect that the function will be *correlated* with a function of a few directions; this is the tolerant testing problem, and – as evidenced by the long history of tolerant testing in the Boolean case – it is much more challenging.

The main result of this paper is a fully noise tolerant tester for $k$-linear juntas over the Gaussian space whose query complexity is independent of the ambient dimension $n$. In particular, we prove the following:

**Theorem 1.2.** *There is an algorithm Robust-linear-junta-Boolean which given parameters $1/2 > c_u > c_\ell > 0$, junta arity $k$ and surface area parameter $s$ and oracle access to $f : \mathbb{R}^n \to \{-1, 1\}$ distinguishes between the following cases:*

1. *There is a linear-$k$-junta $g$ with surface area at most $s$ such that $\mathrm{dist}(f, g) \le c_\ell$.*
2. *For all linear-$k$-juntas $g$ with surface area at most $s$, $\mathrm{dist}(f, g) \ge c_u$.*

*The query complexity of the tester is $k^{\mathrm{poly}(s/\epsilon)}$ where $\epsilon = c_u - c_\ell$, and the tester makes non-adaptive queries.*

Note that qualitatively this result implies the main result of [17] – thus a dependence on $s$ is necessary, although we have no reason to believe that an exponential dependence on $s$ is sharp. In fact, the result here is *qualitatively stronger* than [17] as our "soundness guarantee" does not require relaxing the surface area to $s(1 + \epsilon)$. On the other hand, the query complexity here as an exponential dependence on $s$ vis-a-vis [17] which has a polynomial query complexity in all the parameters.

It is not hard to see that tolerant testing is essentially equivalent to estimating the maximum correlation between a function and a class. In particular, Theorem 1.2 follows from the following result about estimating correlation. Here (and in most of this work), it is more convenient to consider functions with values in $[-1, 1]$. For these functions, we need a more general notion of smoothness: we will define the notion of *$s$-smooth functions* later (in Definition 2.6); for now, we just note that it includes both Lipschitz functions and Boolean functions with bounded surface area.

**Theorem 1.3.** *There is an algorithm Correlation-smooth-junta which, given parameters $\epsilon > 0$, junta arity $k$ and smoothness parameter $s$ and oracle access to $f : \mathbb{R}^n \to [-1, 1]$, outputs an estimate $\hat{\rho}_{\mathbb{R}^n,k,s}(f)$ such that with high probability,*

$$\left|\hat{\rho}_{\mathbb{R}^n,k,s}(f) - \rho_{\mathbb{R}^n,k,s}(f)\right| \le \epsilon.$$

*Here $\rho_{\mathbb{R}^n,k,s}(f)$ is the maximum correlation of $f$ with any $s$-smooth $k$-linear junta. The query complexity of the algorithm is $k^{\mathrm{poly}(s/\epsilon)}$.*

In particular, Theorem 1.2 follows as a simple corollary of Theorem 1.3.

## 1.4 List Decoding the Linear-Invariant Structure.

Given the previous theorem it is natural to ask for more, i.e., not just test if the function is a linear-junta but also find a junta in number of queries that depends only on $k$ and $s$ (but not on $n$) that has almost maximal correlation with $f$. In other words, the goal is to find, with query complexity independent of $n$, a function $g : \mathbb{R}^k \to \{-1, 1\}$ such that there exists a projection matrix $A : \mathbb{R}^n \to \mathbb{R}^k$ and such that the correlation between $f$ and $g(Ax)$ is at least $\rho_{\mathbb{R}^n,k,s}(f) - \epsilon$.

In the case where $f$ is a linear $k$-Junta with bounded surface area, i.e., $\rho_{\mathbb{R}^n,k,s}(f) = 1$, [17] provided such an algorithm with query complexity that is exponential in $k$. In the noisy case, we could have multiple different Juntas that have optimal or close to optimal correlation with $f$. Ideally we would like to find all those functions, which can be thought of as "list decoding" the Juntas that are hidden in $f$.

There is some subtlety in the meaning of "all" here; for example, if $f$ is a linear 1-Junta with some added noise and we set $k = 2$, then there can be a huge number (i.e. growing quickly with $n$) of linear 2-Juntas that are highly correlated with $f$, just because there is a lot of flexibility in choosing the second direction and defining the function in that direction. For this reason, rather than identifying all highly-correlated linear Juntas, we only identify their averages on a set of interesting directions; for a subspace $E$ of $\mathbb{R}^n$ and a function $g : \mathbb{R}^n \to \mathbb{R}$, let $\mathcal{A}_E g$ be obtained from $g$ by averaging over the directions orthogonal to $E$ (see Definition 2.1 for a full definition).

**Theorem 1.4.** *There is an algorithm Learn-all-invariant-structures which, given parameters $\rho, \epsilon > 0$, junta arity $k$, smoothness parameter $s$ and oracle access to $f : \mathbb{R}^n \to [-1, 1]$, outputs a set $\mathcal{G}$ of functions $\mathbb{R}^k \to [-1, 1]$ so that the following hold:*

- *for every $\hat{g} \in \mathcal{G}$ there exists an orthonormal set of vectors $w_1, \ldots, w_k \in \mathbb{R}^n$ such that*

$$\left|\mathbf{E}[f(x)\hat{g}(\langle w_1, x\rangle, \ldots, \langle w_k, x\rangle)] - \rho\right| = O(\epsilon),$$

  *and*
- *for every linear $k$-Junta $g : \mathbb{R}^n \to [-1, 1]$ with $\left|\mathbf{E}[f(x)g(x)] - \rho\right| \le \epsilon$, there exists a function $\hat{g} \in \mathcal{G}$ and an orthonormal set of vectors $w_1, \ldots, w_k \in \mathbb{R}^n$ such that, with $E = \mathrm{span}\{w_1, \ldots, w_k\}$, we have*

$$\mathbf{E}[\left((\mathcal{A}_E g)(x) - \hat{g}(\langle w_1, x\rangle, \ldots, \langle w_k, x\rangle)\right)^2] \le O(\epsilon).$$

  *Additionally,*

$$\mathbf{E}[f(x)g(x)] \approx_{O(\epsilon)} \mathbf{E}[f(x)(\mathcal{A}_E g)(x)].$$

*The query complexity of the algorithm is $k^{\mathrm{poly}(s/\epsilon)}$.*

Informally, the theorem states that it is possible to find the "linear-invariant" structures (i.e., the structure up to unitary transformation) of all Juntas that are almost optimally correlated with $f$ in number of queries that depends on $s$ and $k$. We note that one cannot hope to output the relevant directions $w_1, \ldots, w_k$ explicitly as even describing these directions will require $\Omega(\log n)$ bits of information and thus, at least those many queries.

The significance of Theorem 1.4 is related to one of the main difficulties in tolerant testing: there can be a large number of linear Juntas having almost optimal correlation with $f$. This in in contrast with the usual testing problem, because if $f$ is in fact a linear $k$-Junta then there is (obviously) only one linear $k$-Junta that is equal to $f$.

Even in the noiseless case, Theorem 1.4 improves on the the results of [17] which provided an algorithm for learning the linear structure with query complexity that is exponential in $k$. We

note that in [17] it was incorrectly stated (without proof) that the exponential dependence on $k$ is necessary.

Thanks to Theorem 1.4, we are also able to tolerantly test certain *subclasses* of linear Juntas; this is significant because in general the testability of a class does not imply the testability of a subclass.

**Definition 1.5.** *Let $C$ be any collection of functions mapping $\mathbb{R}^k$ to $\{-1,1\}$. For any $n \in \mathbf{N}$, define the* induced class of $C$ *by*

$$\mathsf{Ind}(C)_n = \{f : \exists g \in C \text{ and orthonormal vectors } w_1,\dots,w_k \text{ such that } f(x) = g(\langle w_1,x\rangle,\dots,\langle w_k,x\rangle)\}.$$

Note that every $f \in \mathsf{Ind}(C)_n$ is a linear $k$-Junta. As an example, if $C$ is the class of intersections of $k$-halfspaces over $\mathbb{R}^k$, then $\mathsf{Ind}(C)_n$ is the class of intersections of $k$-halfspaces over $\mathbb{R}^n$.

**Theorem 1.6.** *Let $C$ be a collection of functions mapping $\mathbb{R}^k$ to $[-1,1]$ such that each $f \in C$ is $s$-smooth. There is an algorithm Robust-$C$-test which given parameters $1/2 > c_u > c_\ell > 0$, junta arity $k$, surface area parameter $s$, and oracle access to $f : \mathbb{R}^n \to \{-1,1\}$, distinguishes between the following cases:*

1. *There is a linear-$k$-junta $g \in \mathsf{Ind}(C)_n$ with surface area at most $s$ such that $\mathsf{dist}(f,g) \le c_\ell$.*
2. *For all linear-$k$-juntas $g \in \mathsf{Ind}(C)_n$, $\mathsf{dist}(f,g) \ge c_u$.*

*The query complexity of the tester is $k^{\mathsf{poly}(s/\epsilon)}$ where $\epsilon = c_u - c_\ell$, and the tester makes non-adaptive queries.*

As an immediate corollary, this implies that there is a fully noise tolerant tester for intersections of $k$-halfspaces with query complexity $k^{\mathsf{poly}(\log k/\epsilon)}$. Previously, no noise tolerant tester was known for even a single halfspace [33].

## 1.5 Techniques

For ease of exposition, here we just explain the technique for proving Theorem 1.3. The high level proof technique for the other results is essentially the same albeit sometimes with added technical complications. The techniques of the current paper build on those of [17]. We briefly recap the main ideas of [17], restricted for now to the non-tolerant setting:

I. If we sample $T = \mathsf{poly}(k/\epsilon)$ random points $\mathbf{x}_1,\dots,\mathbf{x}_T$ from the standard Gaussian measure $\gamma_n$ and consider the subspace $E = \mathsf{span}(\nabla f(\mathbf{x}_1),\dots,\nabla f(\mathbf{x}_T))$, then if $f$ is a linear $k$-Junta then with high probability, $f$ has correlation $1-\epsilon$ with some linear $k$-junta defined on the space $E$.

II. For each $x, z$, it is possible to accurately estimate, in number of samples polynomial in $k$, quantities such as $\langle z, \nabla f(x)\rangle$ and $\langle \nabla f(x_i), \nabla f(x_j)\rangle$. Thus, for a randomly chosen $\mathbf{z} \sim \gamma_n$, we can (implicitly, in a sense to be made precise later) compute the orthogonal projection of $\mathbf{z}$ on $E$. (Note that a naive estimation of $\nabla f(\mathbf{x})$, or even $\langle \nabla f(\mathbf{x}_1), \nabla f(\mathbf{x}_2)\rangle$, requires a number of samples that depends on $n$.)

Observe that the implicit projection allows [17] to effectively reduce the dimension of the ambient space to $T = \mathsf{poly}(k/\epsilon)$, which is independent of $n$. We then take an $\epsilon$-net of linear $k$-juntas over $E$ with surface area $s$. The size of this net depends only on $s$, $k$ and $\epsilon$. For each function in the net, one can estimate its distance to $f$; by iterating over all functions in the net, one can check if $f$ is close to a linear $k$-junta. This last step is different from the one in [17], and in fact it is slightly worse. By following the ideas in the current paper, one can show that it gives a tester for with query complexity of $k^{O(s^2/\epsilon^2)}$. The advantage of the modification, however, is that it yields a method that is more robust to noise.

*Adding tolerance.* In adapting the outline above to the setting of tolerant testing, the main challenge is to imitate step I above. Our main structural result roughly shows that if $f$ has correlation $c$ with some linear $k$-junta of surface area at most $s$, then with high probability $f$ is at least $c-\epsilon$ correlated with an $s$-smooth linear $k$-junta defined on $E$. In fact, we need to define $E$ more carefully than what is outlined above, and a good error analysis is crucial. If we were to combine our new structural result with a naive error analysis, it would give a query complexity that is exponential in $\mathrm{poly}(k)$.

The proof of our structural result is non-trivial. At the intuitive level it is related to the idea of using SVD for PCA. In our case, we have a function, rather than a collection of data, and the right geometric information is encoded by gradients (of a smoothed version of this function). The procedure of using SVD to extract informative directions from the data can be thought of as "*Gradient Based* PCA". The proof that this procedure actually extracts the relevant dimensions requires combining linear algebraic and Poincare style geometric estimates in just the right way. Another challenge comes from the fact that gradients are only approximate due to sampling effect. We use results from random matrix theory to control the effect of sampling.

The methodology of Gradient Based PCA also allows us to improve on the results of [17] in the noiseless case for finding an approximation of the Junta. This has to do with the fact that the results of [17] used a more naive Gram–Schmidt based process to extract the linear structure which resulted in exponential query complexity, compared with the polynomial query complexity we achieve in the current work.

Another key new ingredient of this current work is the net argument outlined above. We show that the class of $s$-smooth linear $k$ juntas has an $\epsilon$-net of size $\exp\exp((s^2\log k)/\epsilon^2)$. For each function in the net, we can use the implicit projection algorithm to compute the correlation between this function and $f$ up to error $\epsilon$. The maximum of these correlations gives a good estimate of the best correlation between $f$ and any linear $k$-junta with surface area $s$. This concludes the proof sketch of Theorem 1.3.

We note that there is a high-level similarity between the current proof and the proof that Boolean juntas are tolerant testable [18]. Both strategies are based on oracle access to influential "directions" followed by a search for juntas depending only on those influential directions. In the Boolean case, the "directions" are influential variables, while here the directions are given by gradients of the function $f$. Note, however, that the Boolean case is easier, since the coordinates on the Boolean cube are automatically orthogonal, while in the continuous setup, "relevant directions" as sampled from data are often not orthogonal and indeed can be close to parallel. This is one of the major reasons we needed to introduce and analyze the methodology of gradient based PCA.

*Related work:* Besides being related to the long line of work (including [4–6, 10, 24, 43]) on junta testing, the current work is

also connected to the rich area of learning and testing of threshold functions. In particular, an immediate corollary of Theorem 1.6 gives a fully noise tolerant tester for any function of $k$-halfspaces over the Gaussian space. Despite prior work on testing of halfspaces [33, 34, 41], until this work, no non-trivial noise tolerant tester was known even for a single halfspace.

Finally, we remark that the notion of noise in property testing (including this paper) is the so-called *adversarial label noise* [27]. This is stronger than many other noise models in literature such as the *random classification noise* [27] and *Massart noise* [32]. Both these models are important from the point of view of learning theory – in particular, halfspaces (and polynomial threshold functions) are known to be efficiently learnable [8, 19] in both these models of noise even when the background distribution is arbitrary. On the other hand, for arbitrary background distributions, halfspaces are hard to learn in the adversarial label noise model [16, 23]. In contrast, the tester in the current paper is in the adversarial label noise model but works only when the background distribution is the Gaussian. This discussion raises the intriguing possibility that halfspaces (and more generally linear juntas) can be tested in the *distribution free model* [25] with weaker models of noise such as the Massart noise.

The problem of *learning* linear $k$-juntas was originally introduced in [45], although of course there the sample complexity depends (polynomially) on $n$.

## 2 PRELIMINARIES

In this section, we list some useful definitions and technical preliminaries. We begin with some definitions and properties of projections and averages.

**Definition 2.1.** *For a subspace $E$ of $\mathbb{R}^n$, we denote by $\Pi_E : \mathbb{R}^n \to \mathbb{R}^n$ the orthogonal projection onto $E$. For a subspace $E$ and $f : \mathbb{R}^n \to \mathbb{R}$, we define the operator $\mathcal{A}_E$ as $\mathcal{A}_E f(x) = \mathbf{E}_{\mathbf{z}\sim\gamma_n}[f(\Pi_E x + \Pi_{E^\perp}\mathbf{z})]$, where $\gamma_n$ is the standard $n$-dimensional Gaussian measure.*

*Finally, for any subspace $E$, we define*

$$\mathcal{J}_E = \{f : \text{for all } x \text{ and } z, \text{ if } \Pi_E z = \Pi_E x \text{ then } f(x) = f(z)\}.$$

One way to understand the operator $\mathcal{A}_E$ is that it averages $f$ on the directions orthogonal to the subspace $E$. The next lemma lists some useful properties of the operators $\Pi_E$ and $\mathcal{A}_E$. Let $C_b^1(\mathbb{R}^n)$ be the class of differentiable functions $f$ such that $f(x)$ and $\nabla f(x)$ are bounded.

**Lemma 2.2.** *For any $f \in C_b^1(\mathbb{R}^n)$, any subspaces $E \subset E' \subset \mathbb{R}^n$, and any $x \in \mathbb{R}^n$, the following hold:*

1. *If $\Pi_E z = \Pi_E x$ then $\mathcal{A}_E f(x) = \mathcal{A}_E f(z)$. In other words, $\mathcal{A}_E f \in \mathcal{J}_E$.*
2. $(\nabla \mathcal{A}_E f)(x) = \mathop{\mathbf{E}}_{\mathbf{z}}[\Pi_E \nabla f(\Pi_E x + \Pi_{E^\perp}\mathbf{z})]$
3. $(\mathcal{A}_E \mathcal{A}_{E'} f)(x) = (\mathcal{A}_E f)(x)$
4. *For all $g \in \mathcal{J}_E$, $\mathop{\mathbf{E}}_{\mathbf{x}}[g(\mathbf{x})\,\mathcal{A}_E f(\mathbf{x})] = \mathop{\mathbf{E}}_{\mathbf{x}}[g(\mathbf{x}) f(\mathbf{x})]$*
5. *For all $g \in L^2(\gamma)$, $\mathop{\mathbf{E}}_{\mathbf{x}}[(\mathcal{A}_E f)(\mathbf{x}) g(\mathbf{x})] = \mathop{\mathbf{E}}_{\mathbf{x}}[f(\mathbf{x})(\mathcal{A}_E g)(\mathbf{x})]$.*

Note that parts 1 and 4 can be interpreted as saying that $\mathcal{A}_E f$ is the orthogonal projection (in the $L^2(\gamma)$ sense) of $f$ onto $\mathcal{J}_E$.

Proof.

(1) Part 1 is immediate from the definition of $\mathcal{A}_E$.
(2) To prove part 2, fix $v \in \mathbb{R}^n$. Then

$$(\mathcal{A}_E f)(x) - (\mathcal{A}_E f)(x - v) = \mathop{\mathbf{E}}_{\mathbf{z}}[f(\Pi_E x + \Pi_{E^\perp}\mathbf{z}) - f(\Pi_E x - \Pi_E v + \Pi_{E^\perp}\mathbf{z})]$$

Replacing $v$ by $hv$ and sending $h \to 0$, we obtain (and there is no trouble exchanging the limit and the expectation, because $f$ is Lipschitz)

$$(\nabla_v \mathcal{A}_E f)(x) = \mathop{\mathbf{E}}_{\mathbf{z}}[\nabla_{\Pi_E v} f(\Pi_E x + \Pi_{E^\perp}\mathbf{z})].$$

This proves the second item.

(3) Part 3 follows from the fact that if $E \subset E'$ then $\Pi_E \Pi_{E'} z = \Pi_E z$ and $\Pi_E \Pi_{(E')^\perp} z = 0$ for every $z$. Indeed, if $\mathbf{z}$ and $\mathbf{z}'$ are independent standard Gaussian variables then

$$(\mathcal{A}_E \mathcal{A}_{E'} f)(x) = \mathop{\mathbf{E}}_{\mathbf{z},\mathbf{z}'}[f(\Pi_E(\Pi_{E'} x + \Pi_{(E')^\perp}\mathbf{z}') + \Pi_{E^\perp}\mathbf{z})] = \mathbf{E}[f(\Pi_E x + \Pi_{E^\perp}\mathbf{z})] = (\mathcal{A}_E f)(x).$$

(4) For Item 4, let $\mathbf{z}$ and $\mathbf{z}'$ be standard Gaussian variables. Since $g \in \mathcal{J}_E$, we have $g(z) = g(\Pi_E z + \Pi_{E^\perp} z')$. Hence,

$$\mathbf{E}[g\,\mathcal{A}_E f] = \mathop{\mathbf{E}}_{\mathbf{z},\mathbf{z}'}[g(\mathbf{z}) f(\Pi_E \mathbf{z} + \Pi_{E^\perp}\mathbf{z}')] = \mathbf{E}[g(\Pi_E \mathbf{z} + \Pi_{E^\perp}\mathbf{z}') f(\Pi_E \mathbf{z} + \Pi_{E^\perp}\mathbf{z}')].$$

Since $\Pi_E \mathbf{z} + \Pi_{E^\perp}\mathbf{z}'$ has the same distribution as $\mathbf{z}$, the claim follows.

(5) Item 5 follows from applying claim 4 twice:

$$\mathop{\mathbf{E}}_{\mathbf{x}}[(\mathcal{A}_E f)(\mathbf{x}) g(\mathbf{x})] = \mathbf{E}[(\mathcal{A}_E f)(\mathbf{x})(\mathcal{A}_E g)(\mathbf{x})] = \mathbf{E}[f(\mathbf{x})(\mathcal{A}_E g)(\mathbf{x})].$$

□

As an immediate consequence of the properties above, we have the following two basic properties of $\nabla \mathcal{A}_E f$:

Claim 2.3.
(1) $\mathbf{E}_{\mathbf{x}}[\nabla \mathcal{A}_E f(\mathbf{x})] = \mathbf{E}_{\mathbf{x}}[\Pi_E \nabla f(\mathbf{x})]$.
(2) $\mathbf{E}_{\mathbf{x}}[\|\nabla \mathcal{A}_E f(\mathbf{x})\|_2^2] \le \mathbf{E}_{\mathbf{x}}[\|\Pi_E \nabla f(\mathbf{x})\|_2^2]$.

Proof. Item 1 follows by averaging over Item 2 from Lemma 2.2. To get Item 2, first observe that by Jensen's inequality (applied on Item 2 from Lemma 2.2), we have

$$\|(\nabla \mathcal{A}_E f)(x)\|^2 \le \mathop{\mathbf{E}}_{\mathbf{z}}[\|\Pi_E \nabla f(\Pi_E x + \Pi_{E^\perp}\mathbf{z})\|^2].$$

Averaging over $\mathbf{x} \sim \gamma$ and observing that the distribution of $\Pi_E \mathbf{x} + \Pi_{E^\perp}\mathbf{z}$ is the same as that of $\mathbf{x}$, we have Item 2. □

### 2.1 Smoothness and Juntas

The notion of smoothness that we will use in this work depends on the notion of Gaussian noise. In particular, we use the following Gaussian noise operator:

**Definition 2.4.** *For $t \ge 0$ and $f \in L_2(\gamma_n)$, we define $P_t f : \mathbb{R}^n \to \mathbb{R}$ as*

$$P_t f(x) = \mathbf{E}_{\mathbf{y}}[f(e^{-t}x + \sqrt{1 - e^{-t}}\,\mathbf{y})].$$

*The operator $P_t$ forms a semigroup, i.e., $P_t P_{t'} f = P_{t+t'} f$. Further, for $t > 0$, $P_t f$ is infinitely differentiable.*

We recall here a basic property of this noise operator – namely, that $P_t$ makes any bounded function Lipschitz, a fact that can be derived, for example, from (2.3) in [31].

**Fact 2.5.** *For any $f : \mathbb{R}^n \to [-1,1]$ and any $t > 0$, $P_t f$ is $\frac{C}{\sqrt{t}}$-Lipschitz for an absolute constant $C$.*

Now we come to the notion of *s-smooth* functions:

**Definition 2.6.** *A function $f : \mathbb{R}^n \to [-1,1]$ is referred to as* $s$-smooth *if for all $t > 0$,*

$$\mathbf{E}[|f(\mathbf{x}) - P_t f(\mathbf{x})|] \le s\sqrt{t}.$$

*In this case, we say that $\mathrm{Sm}(f) \le s$.*

To help illustrate the definition, let us recall the notion of Gaussian surface area:

**Definition 2.7.** *For a Borel set $A \subseteq \mathbb{R}^n$, we define its Gaussian surface area, denoted by $\Gamma(A)$ to be*

$$\Gamma(A) = \liminf_{\delta \to 0} \frac{\mathrm{vol}(A_\delta \setminus A)}{\delta}.$$

*Here $A_\delta$ denotes the set of points which are at Euclidean distance at most $\delta$ the set $A$.*

The next proposition shows that the class of $s$-smooth functions functions of bounded surface area. Later, we will also show that the notion of $s$-smoothness is equivalent to a certain decay in the Hermite coefficients (which can also be used to show that $\mathrm{Sm}(f) \le C\,\mathbf{E}[\|\nabla f\|^2]$, so for example Lipschitz functions are $s$-smooth).

**Proposition 2.8.**

(1) *If $f : \mathbb{R}^n \to \{-1,1\}$ has surface area at most $\frac{s\cdot\sqrt{\pi}}{2}$, then $\mathrm{Sm}(f) \le s$.*
(2) *Let $E$ be any subspace of $\mathbb{R}^n$. If $f$ is $s$-smooth, then so is $\mathcal{A}_E f$.*

Proof. (1) Part 1 was proved by Pisier [40] and Ledoux [30].
(2) To prove Part 2, observe that the operators $\mathcal{A}_E$ and $P_t$ commute. Thus,

$$\mathbf{E}[|\mathcal{A}_E f(\mathbf{x}) - P_t \mathcal{A}_E f(\mathbf{x})|] = \mathbf{E}[|\mathcal{A}_E f(\mathbf{x}) - \mathcal{A}_E P_t f(\mathbf{x})|].$$

However, Jensen's inequality implies that for any $f$ and $g$, $\mathbf{E}_{\mathbf{x}}[|\mathcal{A}_E f(\mathbf{x}) - \mathcal{A}_E g(\mathbf{x})|] \le \mathbf{E}_{\mathbf{x}}[|f(\mathbf{x}) - g(\mathbf{x})|]$. This finishes the proof. □

We now define the class of $s$-smooth linear $k$-juntas.

**Definition 2.9.** *For a subspace $E$ of $\mathbb{R}^n$, parameter $s > 0$ and $k \in \mathbb{N}$, we say $f : \mathbb{R}^n \to [-1,1] \in \mathcal{J}_{E,k,s}$ if*

- *There is a subspace $E' \subseteq E$ of dimension $k$ such that $f \in \mathcal{J}_{E'}$.*
- *$f$ is $s$-smooth.*

**Definition 2.10.** *For a function $h : \mathbb{R}^n \to [-1,1]$, a subspace $E$ of $\mathbb{R}^n$, $k \in \mathbb{N}$ and $s > 0$, we define*

$$\rho_{E,k,s}(h) = \max_{\phi \in \mathcal{J}_{E,k,s}} \mathbf{E}_{\mathbf{x}}[\phi(\mathbf{x}) \cdot h(\mathbf{x})].$$

**Definition 2.11.** *For a function $h : \mathbb{R}^n \to [-1,1]$ and a class $C$ of functions mapping $\mathbb{R}^k \to [-1,1]$, we define*

$$\rho_{\mathbb{R}^n,C}(h) := \max_{\phi \in \mathsf{Ind}_n(C)} \mathbf{E}_{\mathbf{x}}[\phi(\mathbf{x}) \cdot h(\mathbf{x})].$$

*For a subspace $E$ of $\mathbb{R}^n$, we define $\mathsf{Ind}_{E,C}$ to be the set of all functions $\Phi$ which can be expressed in the form*

$$\Phi(x) = h(\langle v_1, x\rangle), \ldots, \langle v_k, x\rangle),$$

*where $h \in C$ and $v_1, \ldots, v_k$ are orthonormal vectors in $E$. Thus, $\mathsf{Ind}_E(C)$ lifts functions in $C$ to functions over $\mathbb{R}^n$ where the relevant subspace is $E$.*

*For such a class $C$, a subspace $E$ of $\mathbb{R}^n$ and a function $f : \mathbb{R}^n \to [-1,1]$, we define*

$$\rho_{E,C}(f) := \max_{\Phi \in \mathsf{Ind}_E(C)} \mathbf{E}_{\mathbf{x}}[f(\mathbf{x}) \cdot \Phi(\mathbf{x})].$$

## 2.2 Useful Results about Matrices

**Definition 2.12.** *Let $B \in \mathbb{R}^{m\times n}$ matrix. Then, the singular value decomposition (SVD) of $B$ corresponds to $B = U \cdot D \cdot V^T$ where (i) $D \in \mathbb{R}^{r\times r}$ is a diagonal matrix with nonzero entries and (ii) the columns of $U$ and $V$ are orthonormal. The columns of $U$ form an orthonormal basis for the column span of $B$. Similarly, the columns of $V$ form an orthonormal basis for the row span of $B$.*

We will also need the following random sampling result concerning rank one matrices due to Rudelson and Vershynin [42].

**Theorem 2.13.** *Let $\mathbf{Z}$ be a distribution over $\mathbb{R}^n$ such that with probability 1, for $Z \sim \mathbf{Z}$, we have $\|Z\|_2 \le M$. Assume that $\|\mathbf{E}[\mathbf{Z}\otimes\mathbf{Z}]\|_2 \le 1$. Let $\mathbf{Z}_1, \ldots, \mathbf{Z}_d$ be i.i.d. copies of $\mathbf{Z}$. Let $a$ be defined as*

$$a = C\sqrt{\frac{\log d}{d}}M,$$

*for an absolute constant $C > 0$. Then,*

$$\Pr\left[\left\|\frac{1}{d}\cdot\Big(\sum_{j=1}^{d}\mathbf{Z}_j\otimes\mathbf{Z}_j\Big) - \mathbf{E}[\mathbf{Z}\otimes\mathbf{Z}]\right\|_2 > t\right] \le 2e^{-t^2/a^2}.$$

Next, we recall the notion of pseudoinverse of a matrix [35, 39]. Our definition below is specialized to real square matrices though the definition can be generalized to complex rectangular matrices as well.

**Definition 2.14.** *For any square matrix $A \in \mathbb{R}^{n\times n}$, there is a unique matrix $B$ which satisfies the following conditions (known as the Moore-Penrose conditions):*

(1) *$ABA = A$ and $BAB = B$.*
(2) *$(AB)^t = AB$ and $(BA)^t = BA$.*

*$B$ is referred to as the* pseudoinverse *of $A$. We remark that when $A$ is invertible, then $B = A^{-1}$. We will thus overload this notation and in general, use $A^{-1}$ to denote the pseudoinverse of $A$.*

Claim 2.15. *Let $A \in \mathbb{R}^{m\times m}$ be a symmetric matrix whose nonzero eigenvalues are $\{\lambda_1, \ldots, \lambda_t\}$ and corresponding orthonormal vectors $\{v_1, \ldots, v_t\}$ (note that $t \le m$). Then,*

$$A = \sum_{i=1}^{t}\lambda_i v_i v_i^t \quad \text{and} \quad A^{-1} = \sum_{i=1}^{t}\frac{1}{\lambda_i}v_i v_i^t.$$

Proof. It is immediate to verify that the Moore-Penrose conditions from Definition 2.14 hold for $A^{-1}$ defined as above (uses the fact that $v_i$ are orthonormal). □

**Definition 2.16.** *For a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a parameter $\eta \in \mathbb{R}$, we define $A_{\ge\eta} \in \mathbb{R}^{n\times n}$ as projection of $A$ to the eigenspaces*

*with eigenvalue more than $\eta$. In other words, let the spectral decomposition of $A$ be*

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^t.$$

*Then,*

$$A_{\geq\eta} = \sum_{i:\lambda_i \geq \eta} \lambda_i v_i v_i^t.$$

*Further, for $\eta > 0$, we define $A^{-1}_{\geq\eta}$ by*

$$A^{-1}_{\geq\eta} = \sum_{i:\lambda_i \geq \eta} \frac{1}{\lambda_i} v_i v_i^t.$$

*Note that this is the same as the pseudoinverse of $A_{\geq\eta}$. Finally, for a symmetric matrix $A$ and parameter $\eta \in \mathbb{R}$, we let $\mathsf{E}_\eta(A)$ denote* $\mathrm{span}(\{v_i\}_{\lambda_i \geq \eta})$.

## 2.3 Algorithmic Ingredients

We will require some algorithmic ingredients from the paper [17]. The first is Lemma 10 in the full version from [17] which is stated below.

**Lemma 2.17.** *There is an algorithm Compute-inner-product which given oracle access to function $g : \mathbb{R}^n \to [-1, 1]$, noise parameter $t > 0$, error parameter $\epsilon > 0$, confidence parameter $\delta > 0$ and has the following guarantee:*

1. *It makes* $\mathrm{poly}(t, 1/\epsilon, \log(1/\delta))$ *queries to $g$.*
2. *With confidence $1 - \delta$, it outputs $\langle \nabla(P_t g)(y_1), \nabla(P_t g)(y_2)\rangle$ up to additive error $\pm\epsilon$.*

The second lemma we need appears as Lemma 12 in the full version of [17] and is stated below.

**Lemma 2.18.** *There is an algorithm Project-on-gradient which given oracle access to function $g : \mathbb{R}^n \to [-1, 1]$, noise parameter $t > 0$, error parameters $\eta, \nu > 0$, confidence parameter $\delta > 0$, and $x, y \in \mathbb{R}^n$. The algorithm Project-on-gradient makes* $\mathrm{poly}(1/t, 1/\eta, 1/\nu, \log(1/\delta))$ *queries to $g$ and outputs, with probability $1-\delta$, a $\pm\nu$-additive estimate of* $\mathrm{Est}(x, y)$*, where* $\mathrm{Est}(x, y)$ *is some function satisfying*

$$\Pr_{\mathbf{y}\sim\gamma_n}[|\mathrm{Est}(x, \mathbf{y}) - \langle \nabla P_t g(x), \mathbf{y}\rangle| > \lambda\eta] \leq \frac{1}{\lambda^2}$$

*for every $\lambda > 0$.*

# 3 PROJECTION ON LOW-DIMENSIONAL SPACE AND CORRELATION WITH LINEAR JUNTAS

The goal of this section is to prove the following theorem.

**Theorem 3.1.** *Let $\Phi : \mathbb{R}^n \to [-1, 1]$ be a (differentiable) $L$-Lipschitz function and $\eta, \delta > 0$. Let $\mathbf{x}_1, \ldots, \mathbf{x}_M \sim \gamma_n$ where $\frac{M}{\log M} \geq C\frac{L^2}{\eta^2}\log(1/\delta)$. Then, with probability $1 - \delta$, the matrix $A \in \mathbb{R}^{n\times n}$ defined as*

$$A = \frac{1}{M}\sum_{j=1}^{M} \nabla\Phi(\mathbf{x}_j) \cdot \nabla\Phi(\mathbf{x}_j)^t,$$

*satisfies the following: for every subspace $E$ containing $\mathsf{E}_{\eta/2}(A)$, for every $s \geq 0$, and for every $h \in \mathcal{J}_{\mathbb{R}^n,k,s}$, we have*

$$|\mathbf{E}_{\mathbf{x}}[\Phi(\mathbf{x}) \cdot (\mathcal{A}_E h)(\mathbf{x})] - \mathbf{E}_{\mathbf{x}}[\Phi(\mathbf{x}) \cdot h(\mathbf{x})]| \leq \sqrt{k \cdot \eta}.$$

At a high level, this theorem says that for any Lipschitz function $\Phi$, its correlation with the best linear $k$-junta essentially remains preserved if we restrict our attention to a subspace obtained by spectrally truncating the empirical covariance matrix of $\nabla\Phi$. It is the first step in realizing part I. from Section 1.5 (the other step is to handle the fact that we can only estimate $A$).

The proof of Theorem 3.1 follows from the following lemma.

**Lemma 3.2.** *Let $E$ be a subspace of $\mathbb{R}^n$ and let $f : \mathbb{R}^n \to \mathbb{R}$ be such that for every unit vector $v \in E^\perp$, $\mathbf{E}[\langle v, \nabla f(\mathbf{x})\rangle^2] \leq \delta$. Then for every $s \geq 0$, and for every $h \in \mathcal{J}_{\mathbb{R}^n,k,s}$, we have*

$$|\mathbf{E}_{\mathbf{x}}[f(\mathbf{x}) \cdot (\mathcal{A}_E h)(\mathbf{x})] - \mathbf{E}_{\mathbf{x}}[f(\mathbf{x}) \cdot h(\mathbf{x})]| \leq \sqrt{k\delta}. \tag{1}$$

**Proof of Theorem 3.1:** Define the matrix $A_{\mathrm{avg}}$ as

$$A_{\mathrm{avg}} = \mathbf{E}_{\mathbf{x}}[\nabla\Phi(\mathbf{x}) \cdot \nabla\Phi(\mathbf{x})^t].$$

Observe that by Theorem 2.13, with probability $1 - \delta$, we have that $\|A_{\mathrm{avg}} - A\| \leq \eta/2$. This implies that for any $E \supseteq \mathsf{E}_{\eta/2}(A)$ and unit vector $v \in E^\perp$, we have $v \in \mathsf{E}_{\eta/2}(A)^\perp$ and hence

$$\mathop{\mathbf{E}}_{\mathbf{x}}[\langle v, \nabla\Phi(\mathbf{x})\rangle^2] = v^T \cdot A_{\mathrm{avg}} \cdot v \leq v^T \cdot A \cdot v + \frac{\eta}{2} \leq \eta. \tag{2}$$

Then, applying Lemma 3.2 to the function $h$ and the subspace $E$, we have the proof. □

We now turn to proving Lemma 3.2.

**Proof of Lemma 3.2:** Let $h \in \mathcal{J}_F$ for some subspace $F$ with $\dim(F) \leq k$. Let $E' = \mathrm{span}(E \cup F)$ and define $g = \mathcal{A}_E h$. Observe that $g$ is $s$-smooth (by Item 2 of Proposition 2) and thus $g \in \mathcal{J}_{E,k,s}$. Also, observe that $h = \mathcal{A}_{E'} h$. We now have

$$\begin{aligned}
&\left|\mathop{\mathbf{E}}_{\mathbf{x}}[f(\mathbf{x}) \cdot g(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}}[f(\mathbf{x}) \cdot h(\mathbf{x})]\right| && (3)\\
&= \left|\mathop{\mathbf{E}}_{\mathbf{x}}[f(\mathbf{x}) \cdot \mathcal{A}_E h(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}}[f(\mathbf{x}) \cdot \mathcal{A}_{E'} h(\mathbf{x})]\right| \\
&= \left|\mathop{\mathbf{E}}_{\mathbf{x}}[\mathcal{A}_E f(\mathbf{x}) \cdot h(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}}[\mathcal{A}_{E'} f(\mathbf{x}) \cdot h(\mathbf{x})]\right| \text{ (Item 5 of Lemma 2.2)} \\
&\leq \left(\mathop{\mathbf{E}}_{\mathbf{x}}[(\mathcal{A}_E f(\mathbf{x}) - \mathcal{A}_{E'} f(\mathbf{x}))^2]\right)^{\frac{1}{2}} \text{ (by Cauchy-Schwarz).} && (4)
\end{aligned}$$

We now seek to bound the right hand side of (4). Towards this, let us split $\mathbb{R}^n = E' \oplus H$ and $E' = E \oplus J$. Here $H$ is the orthogonal complement of $E'$ and $J$ is the orthogonal complement of $E$ inside $E'$. For any $x \in \mathbb{R}^n$, we express it as $(x_H, x_J, x_E)$ ($x_J$ represents the component of $x$ along the subspace $J$ and likewise for $H$ and $E$). Observe that for $x = (x_H, x_J, x_E)$, we have

$$\begin{aligned}
\mathcal{A}_{E'} f(x) &= \mathbf{E}_{\mathbf{x}'_H}[f(\mathbf{x}'_H, x_J, x_E)] \\
\text{and } \mathcal{A}_E f(x) &= \mathbf{E}_{\mathbf{x}'_H, \mathbf{x}'_J}[f(\mathbf{x}'_H, \mathbf{x}'_J, x_E)]. && (5)
\end{aligned}$$

Thus, we now have the following:

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\mathbf{x}}[(\mathcal{A}_E f(\mathbf{x}) - \mathcal{A}_{E'} f(\mathbf{x}))^2] \\
&= \mathbf{E}_{\mathbf{x}_H, \mathbf{x}_J, \mathbf{x}_E}[(\mathbf{E}_{\mathbf{x}'_H}[f(\mathbf{x}'_H, \mathbf{x}_J, \mathbf{x}_E)] - \mathbf{E}_{\mathbf{x}'_H, \mathbf{x}'_J}[f(\mathbf{x}'_H, \mathbf{x}'_J, \mathbf{x}_E)])^2] \\
&= \mathbf{E}_{\mathbf{x}_J, \mathbf{x}_E}[(\mathbf{E}_{\mathbf{x}'_H}[f(\mathbf{x}'_H, \mathbf{x}_J, \mathbf{x}_E)] - \mathbf{E}_{\mathbf{x}'_H, \mathbf{x}'_J}[f(\mathbf{x}'_H, \mathbf{x}'_J, \mathbf{x}_E)])^2 \\
&\leq \mathbf{E}_{\mathbf{x}_H, \mathbf{x}_J, \mathbf{x}_E}[(f(\mathbf{x}_H, \mathbf{x}_J, \mathbf{x}_E) - \mathbf{E}_{\mathbf{x}'_J}[f(\mathbf{x}_H, \mathbf{x}'_J, \mathbf{x}_E)])^2]. && (6)
\end{aligned}$$

The last inequality follows from Jensen's inequality. Next, for any $x = (x_J, x_H, x_E)$, define the function $f_{x_H, x_E} : \mathbb{R}^J \to \mathbb{R}$ as

$$f_{x_H, x_E}(x_J) = f(x_H, x_J, x_E).$$

Then,

$$\Pi_J \nabla f(x) = \nabla f_{x_H, x_E}. \tag{7}$$

Now applying the definition of $f_{x_H,x_E}$ to (6) and subsequently applying the Gaussian Poincaré inequality, we get

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\mathbf{x}}[(\mathcal{A}_E f(\mathbf{x}) - \mathcal{A}_{E'} f(\mathbf{x}))^2] \\
&\le \mathbf{E}_{\mathbf{x}_H,\mathbf{x}_J,\mathbf{x}_E}[(f_{\mathbf{x}_H,\mathbf{x}_E}(\mathbf{x}_J) - \mathbf{E}_{\mathbf{x}'_J}[f_{\mathbf{x}_H,\mathbf{x}_E}(\mathbf{x}'_J)])^2]. \\
&\le \mathbf{E}_{\mathbf{x}_H,\mathbf{x}_J,\mathbf{x}_E}[\|\nabla f_{\mathbf{x}_H,\mathbf{x}_E}(\mathbf{x}_H,\mathbf{x}_J,\mathbf{x}_E)\|_2^2].
\end{aligned}$$

Finally, applying (7), we get

$$\mathop{\mathbf{E}}_{\mathbf{x}}[(\mathcal{A}_E f(\mathbf{x}) - \mathcal{A}_{E'} f(\mathbf{x}))^2] \le \mathbf{E}_{\mathbf{x}_H,\mathbf{x}_J,\mathbf{x}_E}[\|\Pi_J \nabla f(\mathbf{x}_H,\mathbf{x}_J,\mathbf{x}_E)\|_2^2].$$

Now, by our assumption, for any direction $v$ in $J$ (since it is orthogonal to $E$), $\mathbf{E}_{\mathbf{x}}[\|\Pi_v \nabla f(\mathbf{x})\|_2^2] \le \delta$. Since the dimension of $J$ is at most $k$, we get that

$$\mathop{\mathbf{E}}_{\mathbf{x}}[(\mathcal{A}_E f(\mathbf{x}) - \mathcal{A}_{E'} f(\mathbf{x}))^2] \le k\delta.$$

Combining with (4), we get the claim.

□

# 4 ROADMAP FOR PROVING THEOREM 1.2, THEOREM 1.3, THEOREM 1.4 AND THEOREM 1.6

In this section, we give a roadmap for our main results – namely, Theorem 1.2 , Theorem 1.3, Theorem 1.4 and Theorem 1.6. First of all, observe that instantiating Theorem 1.6 for the class of linear $k$-juntas with surface area at most $s$ (which are $O(s)$-smooth by Proposition 2.8) implies Theorem 1.2. As mentioned earlier, noise tolerant testing for a class is equivalent to computing the maximum correlation between a function and the same class. Thus, we will prove the following (equivalent) version of Theorem 1.6.

**Theorem 4.1.** *For any class $C$ of functions mapping $\mathbb{R}^k \to [-1,1]$ (each of which is $s$-smooth), there is an algorithm Robust-$C$-test which has the following guarantee: given error parameter $\epsilon > 0$ and oracle access to $f : \mathbb{R}^n \to [-1,1]$, it outputs an estimate $\hat{\rho}_{\mathbb{R}^n,C}(f)$ such that*

$$\left|\hat{\rho}_{\mathbb{R}^n,C}(f) - \rho_{\mathbb{R}^n,C}(f)\right| \le \epsilon.$$

*The query complexity is $k^{\mathrm{poly}(s/\epsilon)}$.*

Note that by instantiating Theorem 4.1 with the class of $s$-smooth functions, we get Theorem 1.3. Finally, we note that the proof of Theorem 4.1 can be easily modified to yield Theorem 1.4. This is explained in Section 6. Thus, we now focus on proving Theorem 4.1 (which is equivalent to Theorem 1.6).

To do this, our first step is to replace the function $f$ by a smoothed version:

**Lemma 4.2.** *For smoothness parameter $s$, error parameter $\kappa > 0$ and $f : \mathbb{R}^n \to [-1,1]$, the function $f_{\mathrm{sm}}$ defined by $f_{\mathrm{sm}} = P_{\kappa^2/s^2} f$ has the following guarantees:*

1. *$f \in C^\infty$ and $f$ is $L$-Lipschitz for $L = O(s^2/\kappa^2)$.*
2. *For any $x \in \mathbb{R}^n$, $f_{\mathrm{sm}}(x)$ can be computed to error $\eta/10$ with probability $1-\delta$ using $T(\eta,\delta) = \mathrm{poly}(1/\eta, \log(1/\delta))$ queries to the oracle for $f : \mathbb{R}^n \to [-1,1]$.*
3. *Let $g : \mathbb{R}^n \to [-1,1]$ be a $s$-smooth function. Then,*

$$\left|\mathbf{E}_{\mathbf{x}}[f_{\mathrm{sm}}(\mathbf{x})g(\mathbf{x})] - \mathbf{E}_{\mathbf{x}}[f(\mathbf{x})g(\mathbf{x})]\right| \le \frac{\kappa}{2}.$$

Proof. The first property follows from Fact 2.5 and the definition of the noise operator $P_t$. The second property follows easily from the definition of $P_t$: we simply have to take enough samples to estimate the expectation. Finally, suppose $g$ is a $s$-smooth function. Then, it follows that $\mathbf{E}[|P_{\kappa^2/s^2} g(\mathbf{x}) - g(\mathbf{x})|] = O(\kappa)$. It follows that

$$\begin{aligned}
&\left|\mathbf{E}_{\mathbf{x}}[f_{\mathrm{sm}}(\mathbf{x})g(\mathbf{x})] - \mathbf{E}_{\mathbf{x}}[f(\mathbf{x})g(\mathbf{x})]\right| \\
&= \left|\mathbf{E}_{\mathbf{x}}[P_{\kappa^2/s^2} f(\mathbf{x})g(\mathbf{x})] - \mathbf{E}_{\mathbf{x}}[f(\mathbf{x})g(\mathbf{x})]\right| \\
&= \left|\mathbf{E}_{\mathbf{x}}[(P_{\kappa^2/s^2} g(\mathbf{x}) - g(\mathbf{x})) \cdot f(\mathbf{x})]\right| \\
&\le O(\kappa).
\end{aligned}$$

□

Using Lemma 4.2, it suffices to prove Theorem 1.3 for Lipschitz functions. In particular, we shall prove the following version of Theorem 1.3 for Lipschitz functions.

**Theorem 4.3.** *For any class $C$, there is an algorithm Correlation-smooth-junta-$C$ with the following guarantee: Let $f_{\mathrm{sm}} : \mathbb{R}^n \to [-1,1]$ be an infinitely differentiable $L$-Lipschitz function such that $f_{\mathrm{sm}} = P_u f$ for a parameter $u > 0$ (where $f : \mathbb{R}^n \to [-1,1]$). The algorithm is given oracle access to the functions $f_{\mathrm{sm}}$ and $f$. It also gets as inputs, error parameter $\epsilon > 0$, junta arity parameter $k$ and outputs an estimate $\hat{\rho}_{\mathbb{R}^n,C}(f_{\mathrm{sm}})$ (with probability at least $2/3$) with the following guarantee:*

$$|\hat{\rho}_{\mathbb{R}^n,C}(f_{\mathrm{sm}}) - \rho_{\mathbb{R}^n,C}(f_{\mathrm{sm}})| \le \epsilon.$$

*Here $\rho_{\mathbb{R}^n,C}(f_{\mathrm{sm}})$ is the maximum correlation of $f_{\mathrm{sm}}$ with any $s$-smooth $k$-linear junta. The query complexity of the algorithm is $\mathrm{poly}(L/u) \cdot k^{O(s^2/\epsilon^2)}$. Further, the algorithm also works even when we have a noisy oracle to $f_{\mathrm{sm}}$ – in particular, the above guarantee holds even when each evaluation of $f_{\mathrm{sm}}(\cdot)$ at $x$ returns $\pm\eta$ additive error estimate for $\eta = \mathrm{poly}(u/L) \cdot k^{O(-s^2/\epsilon^2)}$.*

To obtain Theorem 4.1, we let $\kappa = \epsilon/4$, $u = \kappa/s$. Define $f_{\mathrm{sm}} = P_u f$. We now invoke Theorem 4.3 on $f_{\mathrm{sm}}$ with error parameter $\epsilon/2$ – observe that the output $\hat{\rho}_{\mathbb{R}^n,C}(f_{\mathrm{sm}})$ satisfies

$$|\hat{\rho}_{\mathbb{R}^n,C}(f_{\mathrm{sm}}) - \rho_{\mathbb{R}^n,C}(f_{\mathrm{sm}})| < \epsilon.$$

Finally, observe that while we do not have oracle access to $f_{\mathrm{sm}}$, Theorem 4.3 only requires to evaluate $f_{\mathrm{sm}}(\cdot)$ with an additive error of $\pm\eta = \mathrm{poly}(u/L) \cdot k^{O(-s^2/\epsilon^2)}$. Observe that the number of queries made by Theorem 4.3 is $\mathrm{Q} = \mathrm{poly}(L/u) \cdot k^{\Theta(s^2/\epsilon^2)}$. Set $\delta = 1/(10\mathrm{Q})$. Using Lemma 4.2, we can evaluate $f_{\mathrm{sm}}(x)$ by making $\eta^{-2}\log(1/\delta)$ to the oracle for $f$. For our choice of $\delta$, this means that with probability $9/10$, all our evaluations of $f_{\mathrm{sm}}(\cdot)$ are $\pm\eta$ accurate. This means that we can simulate our queries to $f_{\mathrm{sm}}$ by using the oracle for $f$ with a multiplicative ovehead of $\eta^{-2}\log(1/\delta)$. Plugging in the values of $\eta$ and $\delta$, we get the final claim.

# 5 PROOF OF THEOREM 4.3

We now turn to the proof of Theorem 4.3. For the moment, we will just assume that we can evaluate $f_{\mathrm{sm}}$ at any point $x$ exactly. From the description of our algorithm, it would be clear that the guarantee of algorithm continues to hold even if each evaluation of $f_{\mathrm{sm}}(x)$

Inputs

$f$ := Oracle access to function $f : \mathbb{R}^n \to [-1, 1]$
$f_{\mathsf{sm}}$ := Oracle access to function $f_{\mathsf{sm}} : \mathbb{R}^n \to [-1, 1]$ where $f_{\mathsf{sm}} = P_u f$.
$L$ := Lipschitz parameter
$\nu$ := accuracy parameter
$k$ := junta arity parameter

Parameters

$\delta := \frac{1}{20}$
$M := \frac{L^2}{\eta^2} \log(L\delta/\eta)$
$\eta := \frac{\nu^2}{100k}$.
$\epsilon' := \frac{\eta^5 \nu^2}{L^8 C_0^2 M^6}$ (where $C_0$ is a large absolute constant - $10^6$ suffices)

Implicit projection algorithm

(1) Sample $M$ random points $\mathbf{x}_1, \ldots, \mathbf{x}_M \sim \gamma_n$.
(2) For each $1 \le i, j \le M$, with confidence parameter $\delta/M^2$ and error parameter $\epsilon'$, we compute $\langle \nabla f_{\mathsf{sm}}(\mathbf{x}_i), \nabla f_{\mathsf{sm}}(\mathbf{x}_j) \rangle = \langle \nabla P_u f(\mathbf{x}_i), \nabla P_u f(\mathbf{x}_j) \rangle$ using algorithm Compute-inner-product from Lemma 2.17. Denote this by $\widehat{A}_{i,j}$ and let $\widehat{A} \in \mathbb{R}^{M \times M}$ as the corresponding symmetric matrix.
(3) Let $\widehat{N}$ be the closest psd matrix to $\widehat{A}$ in Frobenius norm (can be computed using convex programming).
(4) Let $\widehat{V}\widehat{D}^2\widehat{V}^T$ be the spectral decomposition of $\widehat{N}$.
(5) Output the points $(\mathbf{x}_1, \ldots, \mathbf{x}_M)$ and the matrix $\widehat{W} = \widehat{D}^{-1}_{\ge \sqrt{\eta}/2} \cdot \widehat{V}^T$.

**Figure 1: Description of the testing algorithm Implicit projection**

has an additive error of $\pm\eta = \mathrm{poly}(u) \cdot k^{O(-L^2/\epsilon^2)}$. We will bring this to attention of the reader at the relevant points. The algorithm Correlation-smooth-junta invokes two crucial subroutines. The first is the routine Implicit projection described in Figure 1.

## 5.1 Implicit Projection Algorithm

**Lemma 5.1.** *The algorithm Implicit projection takes as input oracle access to $f : \mathbb{R}^n \to [-1, 1]$ and $f_{\mathsf{sm}} : \mathbb{R}^n \to [-1, 1]$, parameters $u, L > 0$, error parameter $\nu > 0$ and junta arity parameter $k$. Suppose $f_{\mathsf{sm}} = P_u f$. The algorithm makes $\mathrm{poly}(k, 1/u, 1/\nu, L)$ queries to $f$ and $f_{\mathsf{sm}}$ and with probability $9/10$, has the following guarantee: For $M = \mathrm{poly}(k/\nu)$, it outputs $M$ points $x_1, \ldots, x_M$ and a matrix $\hat{W} \in \mathbb{R}^{M \times M}$. Let $B^T \in \mathbb{R}^{M \times n}$ be the matrix whose $j^{th}$ row is $\nabla f_{\mathsf{sm}}(x_j)$ and $\hat{E}$ be the span of the rows of $\hat{W}B^T$. There exists a $k$-dimensional subspace $\tilde{E}$ of $\hat{E}$ with the following property. Let $h \in \mathcal{J}_{\mathbb{R}^n,k,s}$. Then, for $g = \mathcal{A}_{\tilde{E}} h$,*

$$|\mathbf{E}_{\mathbf{x}}[f_{\mathsf{sm}}(\mathbf{x})g(\mathbf{x})] - \mathbf{E}_{\mathbf{x}}[f_{\mathsf{sm}}(\mathbf{x})h(\mathbf{x})]| \le \frac{\nu}{2}. \tag{8}$$

*Further, the matrix $\hat{W}$ satisfies*

$$\|\Pi_{\hat{E}} - B\hat{W}^T\hat{W}B^T\|_F = \|\hat{I} - \hat{W}B^T\hat{W}^T\|_F \le \nu/2, \tag{9}$$

*where $\hat{I}$ denotes the identity matrix in $M$ dimensions. Finally, the matrix $\hat{W}$ satisfies $\|\hat{W}\|_2 \le \frac{20k}{\nu}$.*

The high level idea of the lemma is the following: Let $E$ denote the subspace spanned by the rows of $B^T$. Let us define $N = B^T B$ and $\widehat{\Pi} = B\widehat{W}^T\widehat{W}B^T$. To understand the high level idea behind the algorithm Implicit projection, observe that if in Step 2, we could compute $\langle \nabla f_{\mathsf{sm}}(\mathbf{x}_i), \nabla f_{\mathsf{sm}}(\mathbf{x}_j) \rangle$ exactly, then $\hat{N} = N$. Consequently, if $\eta > 0$ is sufficiently small, then it is easy to see that the rows of $\hat{W}B^T$ form an orthonormal basis of $E$ and consequently, $\hat{\Pi}$ is a projection matrix into $E$. Unfortunately for us, we will not have access to $B$ explicitly and thus are only able to compute an approximation to $N$, namely $\hat{N}$. The goal here is two-fold: (a) Understand why the rows of $\hat{W}B^T$ are essentially orthonormal; (b) show that for $g = \mathcal{A}_{\hat{E}} h$, $\mathbf{E}_{\mathbf{x}}[f_{\mathsf{sm}}(\mathbf{x})g(\mathbf{x})]$ is nearly as large as $\mathbf{E}_{\mathbf{x}}[f_{\mathsf{sm}}(\mathbf{x})h(\mathbf{x})]$.

The next claim quantifies the sense in which the rows of $\hat{W}B^T$ are almost orthonormal.

**Lemma 5.2.** *For matrices $\hat{D}$, $\hat{W}$, $B$, $\hat{N}$ and $\eta > 0$ (as described in the algorithm Implicit projection), let $\hat{I} = \hat{D}^{-1}_{\ge\sqrt{\eta}/2}\hat{D}$. (That is, $\hat{I}$ has a 1 corresponding to large eigenvalues of $\hat{N}$.) Let $\hat{E}$ be the span of the rows of $\hat{W}B^T$. Then*

$$\|\Pi_{\hat{E}} - B\hat{W}^T\hat{W}B^T\|_F = \|\hat{I} - \hat{W}B^T \cdot B\hat{W}^T\|_F \le \frac{4}{\eta}\|\hat{N} - B^T B\|_F.$$

Proof. Since $\hat{N} = \hat{V}\hat{D}^2\hat{V}^T$, we can write

$$\hat{I} = (\hat{D}_{\ge\sqrt{\eta}/2})^{-1}\hat{V}^T\hat{N}\hat{V}(\hat{D}_{\ge\sqrt{\eta}/2})^{-1}.$$

Then

$$\hat{I} - \hat{W}B^T B\hat{W}^T = (\hat{D}_{\ge\sqrt{\eta}/2})^{-1}\hat{V}^T(\hat{N} - B^T B)\hat{V}(\hat{D}_{\ge\sqrt{\eta}/2})^{-1}.$$

Finally, note that $\|(\hat{D}_{\ge\sqrt{\eta}/2})^{-1}\| \le \frac{2}{\sqrt{\eta}}$ and $\|AB\|_F \le \|A\|_F\|B\|$ for any matrices $A$ and $B$. This proves the claimed inequality. To see the equality, note that $B\hat{W}^T\hat{W}B^T$ and $\hat{W}B^T B\hat{W}^T$ have the same eigenvalues, and both expressions can be expressed as $(\sum(\lambda_i - 1)^2)^{1/2}$, where the sum ranges over non-zero eigenvalues. □

Having shown that the rows of $\hat{W}B^T$ are close to being orthonormal, we next show that the rows of $\hat{W}B^T$ essentially span $\mathsf{E}_{\ge\eta}(BB^T)$ – more precisely, we show that $\Pi_{\mathsf{E}_{\ge\eta}(BB^T)}(I - B\hat{W}^T\hat{W}B^T)$ is small.

**Lemma 5.3.** *For matrices $\hat{D}$, $\hat{W}$, $B$, $\hat{N}$ and $\eta > 0$ (as described in the algorithm Implicit projection),*

$$\|\Pi_{\mathsf{E}_{\ge\eta}(BB^T)}(I - B\hat{W}^T\hat{W}B^T)\| \le \frac{20\|B^T B\|_F \cdot \|B^T B\|}{\eta^{\frac{5}{2}}}\sqrt{\|\hat{N} - B^T B\|}.$$

Proof. Recall that $B = UDV^T$ is a singular value decomposition of $B$. Let $U_{\ge\sqrt{\eta}}$ consist of the rows of $U$ whose singular values are at least $\sqrt{\eta}$, so that

$$\Pi_{\mathsf{E}_{>\eta}(BB^T)} = U_{\ge\sqrt{\eta}}U^T_{\ge\sqrt{\eta}} = UI_{\ge\sqrt{\eta}}U^T. \tag{10}$$

$$\begin{aligned}
&\|\Pi_{E_{\geq\eta}(BB^T)}(I - B\hat{W}^T\hat{W}B^T)\| \\
&= \|\Pi_{E_{\geq\eta}(BB^T)}(I - B(\hat{N}_{\geq\eta/4})^{-1}B^T)\| \\
&= \|UI_{\geq\sqrt{\eta}}U^T - UI_{\geq\sqrt{\eta}}DV^T(\hat{N}_{\geq\eta/4})^{-1}VDU^T)\| \\
&= \|VI_{\geq\sqrt{\eta}}V^T - VI_{\geq\sqrt{\eta}}DV^T(\hat{N}_{\geq\eta/4})^{-1}VDV^T)\| \\
&= \|VI_{\geq\sqrt{\eta}}V^T - (N_{\geq\eta})^{1/2}(\hat{N}_{\geq\eta/4})^{-1}N^{1/2}\|.
\end{aligned}$$

where in the last line we set $N = B^TB$. The first equality uses $\hat{W}^T\hat{W} = (\hat{N}_{\geq\eta/4})^{-1}$. The second and third equality uses that $\|A\| = \|\Lambda A\Lambda^T\|$ for unitary matrix $\Lambda$ and the last equality sets $N = VDV^T$. Now, observe that $VI_{\geq\sqrt{\eta}}V^T = (N_{\geq\eta})^{1/2}(N_{\geq\eta})^{-1}N^{1/2}$, we have

$$\begin{aligned}
&\|\Pi_{\mathrm{E}_{\geq\eta}(BB^T)}(I - B\hat{W}^T\hat{W}B^T)\| \\
&= \|(N_{\geq\eta})^{1/2}\Big((N_{\geq\eta})^{-1} - (\hat{N}_{\geq\eta/4})^{-1}\Big)N^{1/2}\| \\
&\leq \|N\| \cdot \|\Pi_{\mathrm{E}_{\geq\eta}(N)}\Big((N_{\geq\eta})^{-1} - (\hat{N}_{\geq\eta/4})^{-1}\Big)\| \qquad (11)
\end{aligned}$$

Finally, we will use thee following lemma concerning stability of the pseudoinverse. The proof is deferred to the full version.

**Lemma 5.4.** *Let $A, \tilde{A} \in \mathbb{R}^{n\times n}$ be psd matrices. Let $\eta \geq 0$ and $V$ denote the subspace spanned by the eigenvalues of $A$ in $[\eta, \infty)$. Then,*

$$\|(A^{-1}_{\geq\eta} - \tilde{A}^{-1}_{\geq\eta/2}) \cdot \Pi_V\| \leq \frac{20\|A\|_F\sqrt{\|A - \tilde{A}\|_2}}{\eta^{5/2}}.$$

Applying Lemma 5.4 to get that

$$\|\Pi_{\mathrm{E}_{\geq\eta}(N)}\Big((N_{\geq\eta})^{-1} - (\hat{N}_{\geq\eta/4})^{-1}\Big)\| \leq \frac{20\|N\|_F\sqrt{\|N - \hat{N}\|}}{\eta^{5/2}}.$$

Combining this with (11), we get the result. □

**Lemma 5.5.** *Let $f_{\mathsf{sm}} : \mathbb{R}^n \to [-1, 1]$, $L$, $M$, $\eta$, $k$, $\delta$, $\hat{N}$, $\hat{W}$ and $B$ be as described in the Algorithm Implicit projection. Let $\hat{E}$ denote the span of the rows of $\hat{W}B^T$. If $f_{\mathsf{sm}}$ is $L$-Lipschitz, with probability $1 - \delta$, there is a subspace $\tilde{E}$ of $\hat{E}$ with the following property: For all $h : \mathbb{R}^n \to [-1, 1]$ such that $h \in \mathcal{J}_{\mathbb{R}^n,k,s}$,*

$$\begin{aligned}
&|\mathbf{E}_{\mathbf{x}}[f_{\mathsf{sm}}(\mathbf{x}) \cdot \mathcal{A}_{\tilde{E}}h(\mathbf{x})] - \mathbf{E}_{\mathbf{x}}[f_{\mathsf{sm}}(\mathbf{x}) \cdot h(\mathbf{x}) \\
&\quad\leq \sqrt{k \cdot \eta} - \frac{80L^4 \cdot M^{5/2}}{\eta^{5/2}}\sqrt{\|\hat{N} - B^TB\|} - \frac{16L}{\eta}\|\hat{N} - B^TB\|_F.
\end{aligned}$$

The proof of this lemma uses the ingredients established thus far along with a certain stability estimate for subspaces. The full proof is deferred to the full version.

**Proof of Lemma 5.1:** By our setting of parameters, observe that with probability $1 - \delta$, the matrix $\hat{A}$ satisfies $\|\hat{A} - B^TB\|_\infty \leq \epsilon'$. This in turn implies that $\|\hat{A} - B^TB\|_F \leq \epsilon' \cdot M$. Since $\hat{N}$ is the closest psd matrix to $\hat{A}$, this means $\|N - \hat{N}\|_F \leq 2\epsilon' \cdot M$.

Plugging the values of $\epsilon'$, $\eta$ and $M$ into Lemma 5.5 shows that (8) is satisfied with probability at least $1 - 2\delta = 9/10$. Similarly, (9) follows by plugging the values of $\epsilon'$, $\eta$ and $M$ into Claim 5.2. Finally, observe that the query complexity of the algorithm is dictated by Step 2 (i.e., the query complexity of the routine Compute-inner-product). By plugging in Lemma 2.17, we get that the query complexity is $\mathrm{poly}(M, 1/u, 1/\epsilon')$. Plugging in the values of these parameters (from the description of the algorithm Implicit projection), we get the claim.

Finally, to get an upper bound on $\|\hat{W}\|_2$, observe that $\hat{W} = \hat{D}_{\geq\sqrt{\eta}/2} \cdot \hat{V}$. This means that $\|\hat{W}\|_2 \leq 2/\sqrt{\eta}$. Plugging in the value of $\eta$ from the description of Implicit projection, we get the claim. □

## 5.2 The Averaged Class

We next describe a preprocessing step for our class of functions $C$. The point is that it is possible in principle for $f$ to be an $E$-Junta but be well-correlated with some function $g \in C$ that was embedded in $\mathbb{R}^n$ along a *different* subspace $\tilde{E}$. We handle this by adding to $C$ all possible projections of functions from $C$ that were embedded in different subspaces.

**Definition 5.6.** *For a class $C$ of functions $\mathbb{R}^k \to [-1, 1]$, define $C^*$ to be the set of all functions $\mathbb{R}^k \to [-1, 1]$ of the form*

$$x \mapsto \mathop{\mathbf{E}}_{\mathbf{z}\sim\gamma_k}[g(W^T\binom{x}{\mathbf{z}})],$$

*where $g$ ranges over $C$ and $W$ ranges over all $(2k) \times k$ matrices with orthonormal columns.*

In other words, we are taking functions from $C$, embedding them in $\mathbb{R}^{2k}$ along an arbitrary $k$-dimensional subspace, and then averaging them back down to $\mathbb{R}^k$. As a consequence of Proposition 2.8, if every function is $C$ is $s$-smooth, then so is every function in $C^*$. Also, $C^*$ contains $C$, as can be seen by taking the first $k$ rows of $A$ to be an orthonormal basis of $\mathbb{R}^k$, and the next $k$ rows to be zero. We next have the following claim.

Claim 5.7. *Let $C$ be a class of functions mapping $\mathbb{R}^k \to [-1, 1]$. Define the set $\mathcal{F}$ to be the functions of the form $\mathcal{A}_{\mathbb{R}^m}f$ where $f \in \mathsf{Ind}_n(C)$ (where $n \geq m + k$ and $m \geq k$). Then, $\mathcal{F} = \mathsf{Ind}_m(C^*)$.*

Proof. It is easy to see that $\mathsf{Ind}_m(C^*) \subseteq \mathcal{F}$ as long as $n \geq m + k$. So, we now argue that $\mathcal{F} \subseteq \mathsf{Ind}_m(C^*)$. Let $f \in \mathsf{Ind}_n(C)$. Let $E$ be the relevant subspace for $f$ and let $E = J \oplus J'$ where $J = \mathbb{R}^m \cap E$ and $J'$ is the orthogonal complement of $J$ inside $E$. It is obvious that the dimension of $J$ is at most $k$. It now easily follows that $g \in \mathsf{Ind}_m(C^*)$. □

We remark that although it might be challenging in general to characterize $C^*$ in terms of $C$, there are several classes of functions where this is easy:

- if $C$ is the class of all $s$-smooth functions then $C^* = C$;
- if $C$ is the class of all half-spaces then $C^*$ is the class of all functions of the form $x \mapsto \Phi(\langle a, x\rangle + b)$, where $\Phi$ is the Gaussian c.d.f.;
- more generally, if $C$ is closed under taking subspaces – in the sense that if $g \in C$, $E \subset \mathbb{R}^k$ is a subspace, and $z \in E^\perp$ then $x \mapsto f(\pi_E x + z)$ also belongs to $C$ – then $C^*$ is contained in the convex hull of $C$. In this situation, and because we will be interested in maximizing a linear function over $C$, we can essentially replace $C^*$ by $C$ in what follows.

## 5.3 Hypothesis Testing on Low-Dimensional Space

Our final technical task is to show that functions on a low-dimensional space can be adequately "pulled back" to $\mathbb{R}^n$ under an approximate projection. The first observation is that an approximate projection can be approximated by a projection:

**Lemma 5.8.** *For any $m \le n$ and any $m \times n$ matrix $X$ of rank $m$, there exists an $m \times n$ matrix $Y$ with orthogonal rows, such that*

$$\|X - Y\|_F \le \|XX^T - I\|_F.$$

Proof. Let $UD^2U^T = XX^T$ be a singular value decomposition of $XX^T$. Then $I = (D^{-1}U^TX)(D^{-1}U^TX)^T$, and it follows that $V^T := D^{-1}U^TX$ is an orthogonal matrix. Let $Y = UV^T$. Noting that $X = UDV^T$, we have $\|X - Y\|_F^2 = \|D - I\|_F^2$, and if $\sigma_1, \ldots, \sigma_m$ are the singular values of $X$ then

$$\|D - I\|_F^2 = \sum(\sigma_i - 1)^2 \le \sum(\sigma_i^2 - 1)^2 = \|XX^T - I\|_F^2.$$

□

*5.3.1 The existence of a small net.* We now prove the existence of a small net of Lipschitz functions for families of $s$-smooth $k$-linear Juntas in $\mathbb{R}^m$. The main result is Proposition 5.15.

We begin with a few preliminaries related to approximate by Lipschitz functions, namely, $s$-smooth functions can be approximated by Lipschitz functions and Lipschitz functions don't change much under composition by nearby linear maps.

**Lemma 5.9.** *For every $s$-smooth function $f : \mathbb{R}^n \to [-1, 1]$ and every $\epsilon > 0$, there is a $\frac{C \cdot s}{\epsilon}$-Lipschitz function $g : \mathbb{R}^n \to [-1, 1]$ such that $\|f - g\|_{L^2(\gamma)} \le \epsilon$. Here $C$ is the absolute constant appearing in Fact 2.5.*

Proof. Choose $t = \frac{\epsilon^2}{s^2}$ and set $g = P_t f$, so that the bound $\|f - g\|_{L^2(\gamma)} \le \epsilon$ follows from the fact that $f$ is $s$-smooth. The claim follows from Fact 2.5. □

**Lemma 5.10.** *Suppose that $g : \mathbb{R}^m \to \mathbb{R}$ is Lipschitz and let $X$ and $Y$ be two $m \times n$ matrices. Then $\|g \circ X - g \circ Y\|_{L^2(\gamma)} \le (\mathrm{Lip}\, g)\|X - Y\|_F$ (here $\mathrm{Lip}\, g$ denotes the Lipschitz constant of $g$).*

Proof. Let $\mathbf{x}$ be a standard normal random variable on $\mathbb{R}^n$. Then

$$\begin{aligned}\mathbf{E}[((g \circ X)(\mathbf{x}) - (g \circ Y)(\mathbf{x}))^2] &\le (\mathrm{Lip}\, g)^2\, \mathbf{E}[\|X\mathbf{x} - Y\mathbf{x}\|^2] \\ &= (\mathrm{Lip}\, g)^2 \|X - Y\|_F^2.\end{aligned}$$

□

Our procedure for producing a net for $s$-smooth $k$-juntas in $\mathbb{R}^m$ proceeds in three steps. First, we will construct a net for $s$-smooth functions on $\mathbb{R}^k$. Then we will find a net for $k$-dimensional subspaces of $\mathbb{R}^m$. Combining these two nets will give a net for $s$-smooth $k$-juntas in $\mathbb{R}^m$.

WWe next have the following lemma. The proof of this lemma relies on standard tools from Hermite analysis and properties of the noise operator (and is deferred to the full version).

**Lemma 5.11.** *For any $k \in \mathbf{N}$ and any $s, \epsilon > 0$, there exists a set* Net *of functions $\mathbb{R}^k \to [-1, 1]$ such that*

(1) *every function in* Net *is $\frac{C s}{\epsilon}$-Lipschitz (here $C$ is the absolute constant appearing in Fact 2.5),*
(2) Net *is an $\epsilon$-net for the set of $s$-smooth functions $\mathbb{R}^k \to [-1, 1]$,*
(3) $\log |\mathrm{Net}| \le k^{O(s^2/\epsilon^2)}$, *and*
(4) *Every function $f$ in* Net *is $s$-smooth.*

Next, we need to turn our net of functions on $\mathbb{R}^k$ into a net of $k$-linear-juntas on $\mathbb{R}^m$. We will do this by finding an appropriate net for $k$-dimensional subspaces of $\mathbb{R}^m$, and then using the net of Lemma 5.11 for each of these subspaces.

**Lemma 5.12.** *There is a set $\mathcal{E}$ of $k$-dimensional subspaces of $\mathbb{R}^m$ such that*

(1) *for every $k$-dimensional subspace $E$ of $\mathbb{R}^m$, there is some $E' \in \mathcal{E}$ with $\|\Pi_E - \Pi_{E'}\|_F \le \epsilon$; and*
(2) $|\mathcal{E}| \le \left(\frac{O(k)}{\epsilon}\right)^{mk}$.

Proof. We begin the proof by recalling the following simple fact.

**Fact 5.13.** *For the unit sphere in $\mathbb{R}^m$ (denoted by $\mathbb{S}^{m-1}$), there is a $\delta$-net (in Euclidean sphere) of size $(1/\delta)^{O(m)}$.*

Now, let $T$ be a $\delta$-net of $\mathbb{S}^{m-1}$ (the unit Euclidean sphere in $\mathbb{R}^m$) of cardinality at most $(1/\delta)^{O(m)}$ (as described in Fact 5.13). Let $\mathcal{E}$ be the set of all $k$-dimensional subspaces that are spanned by $k$ elements of $T$. The claimed bound on the cardinality of $\mathcal{E}$ follows, provided we choose $\delta$ so that $\epsilon \le C'k\delta$ (for an absolute constant $C'$).

Let $E$ be a $k$-dimensional subspace of $\mathbb{R}^m$, and let $x_1, \ldots, x_k$ be an orthonormal basis of $E$. Choose $y_1, \ldots, y_k \in T$ with $\|x_i - y_i\| \le \delta$ for all $i$; then the $y_i$ are unit vectors, and for $i \ne j$ we have

$$|\langle y_i, y_j\rangle| = |\langle y_i, y_j\rangle - \langle x_i, x_j\rangle| \le |\langle x_i, x_j - y_j\rangle| + |\langle y_j, x_i - y_i\rangle| \le 2\delta.$$

It follows that if $Y$ is the matrix with rows $y_i$, and if $E'$ is the span of $y_1, \ldots, y_k$, then $\|Y^TY - \Pi_{E'}\|_F^2 = \|YY^T - I\|_F^2 \le 4\delta^2 k$. Hence,

$$\|\Pi_E - \Pi_{E'}\|_F = \|X^TX - \Pi_{E'}\|_F \le \|X^TX - Y^TY\|_F + 2\delta\sqrt{k}.$$

It remains to bound $\|X^TX - Y^TY\|_F$, and it will suffice to show that $\|X^TX - Y^TY\|_F = O(k\delta)$.

Now, if $x$ and $y$ are unit vectors with $\|x - y\| \le \delta$, then $\langle x, y\rangle \ge 1 - O(\delta^2)$. It follows that $\|xx^T - yy^T\|_F^2 = 2 - 2\langle x, y\rangle^2 \le O(\delta^2)$. Thus, by the triangle inequality,

$$\|X^TX - Y^TY\|_F \le \sum_{i=1}^k \|x_ix_i^T - y_iy_i^T\|_F = O(k\delta).$$

□

**Definition 5.14.** *Let $\hat{E}$ be a $m$-dimensional subspace of $\mathbb{R}^n$ and let $C$ be a subset of $s$-smooth linear $k$-juntas over $\mathbb{R}^k$. We define $\mathrm{Ind}_{\hat{E}}(C)$ to be the set of all functions $h : \mathbb{R}^n \to \mathbb{R}$ of the form*

$$\Phi(x) = h(\langle v_1, x\rangle, \ldots, \langle v_k, x\rangle),$$

*where $v_1, \ldots, v_k$ are orthonormal vectors in $\hat{E}$. In other words, $\mathrm{Ind}_{\hat{E}}(C)$ lifts the functions in $C$ to linear $k$-juntas over $\mathbb{R}^n$ where the relevant subspace is a $k$-dimensional subspace of $\hat{E}$. Note that $\mathrm{Ind}_{\mathbb{R}^n}(C) = \mathrm{Ind}_n(C)$ (see Definition 1.5).*

*Finally, for such a class $C$, subspace $\hat{E}$ and a function $f : \mathbb{R}^n \to [-1, 1]$,*

$$\rho_{\hat{E},C}(f) := \max_{\Phi \in \mathrm{Ind}_{\hat{E}}(C)} \mathbf{E}_{\mathbf{x}}[\Phi(\mathbf{x}) \cdot f(\mathbf{x})].$$

**Proposition 5.15.** *Let $C$ be a subset of $s$-smooth linear functions $\mathbb{R}^k \to [-1,1]$. Then, for any $m \geq k$, there is a set $\mathrm{Net}_{m,C}$ of functions mapping $\mathbb{R}^m$ to $[-1,1]$ which satisfies the following properties:*

1. *Any $g \in \mathrm{Net}_{m,C}$ is $Cs/\epsilon$-Lipschitz.*
2. *$\mathrm{Net}_{m,C}$ is an $\epsilon$-net for $\mathrm{Ind}_{\mathbb{R}^m}(C)$ – i.e., for every $h \in \mathrm{Ind}_{\mathbb{R}^m}(C)$, there is a $g \in \mathrm{Net}_{m,C}$ such that $\|h-g\|_{L^2(\gamma)} \leq \epsilon$.*
3. *$\log|\mathrm{Net}_{m,C}| \leq k^{O(s^2/\epsilon^2)} + O\big(mk \log \frac{ks}{\epsilon}\big)$, and*
4. *For every function in $g \in \mathrm{Net}_{m,C}$, there is $h \in \mathrm{Ind}_{\mathbb{R}^m}(C)$ such that $\|h-g\|_{L^2(\gamma)} \leq \epsilon$.*

Proof. It suffices to consider the case that $C$ is the set of *all* $s$-smooth functions $\mathbb{R}^k \to \mathbb{R}$. Indeed, once we have found a net (call it $\mathrm{Net}_0$) for this case, we can handle the case of general $C$ by simply discarding any $g \in \mathrm{Net}_0$ for which there is no $h \in \mathrm{Ind}_{\mathbb{R}^m}(C)$ such that $\|h-g\|_{L^2(\gamma)} \leq \epsilon$. In this way, we ensure that property 4 is satisfied, noting that properties 1, 2 and 3 remain unchanged if we remove functions $g$ from $\mathrm{Net}_0$. For the rest of this proof, we consider the case that $C$ is the set of all $s$-smooth functions.

Let $\widehat{\mathrm{Net}}$ be a net for $s$-smooth functions on $\mathbb{R}^k$, with the properties guaranteed by Lemma 5.11. Let $\mathcal{E}$ be a collection of $k$-dimensional subspaces of $\mathbb{R}^m$, with the properties guaranteed by Lemma 5.12 with accuracy $\epsilon' = \epsilon^2/s$. We define $\mathrm{Net}_0$ to be the set of functions of the form $x \mapsto f(\Pi_E x)$, where $f \in \widehat{\mathrm{Net}}$ and $E \in \mathcal{E}$. Clearly, $\mathrm{Net}_0$ satisfies Property 1. To see Property 3, note that $\log|\mathrm{Net}_0| = \log|\widehat{\mathrm{Net}}| + \log|\mathcal{E}|$. By using Lemma 5.11 and Lemma 5.12, the bound on $\log|\mathrm{Net}_0|$ follows. Thus, it remains to show Property 2.

To see Property 2, suppose that $f$ is an $s$-smooth $k$-Junta. Then there is some $k$-dimensional subspace $E$ and an $s$-smooth function $g$ on $\mathbb{R}^k$ such that $f = g \circ \Pi_E$. Choose $h \in \widehat{\mathrm{Net}}$ to be $\epsilon$-close to $g$ and choose $E' \in \mathcal{E}$ such that $\|\Pi_E - \Pi_{E'}\|_F \leq \epsilon^2/s$. Then $h \circ \Pi_{E'}$ belongs to Net, and satisfies

$$\|h\circ\Pi_{E'} - f\|_{L^2(\gamma)} \leq \|h\circ\Pi_{E'} - h\circ\Pi_E\|_{L^2\gamma} + \|h\circ\Pi_E - f\circ\Pi_E\|_{L^2(\gamma)}$$

The second term is at most $\epsilon$, and the first term can be bounded (using Lemma 5.10) by $(\mathrm{Lip}\, h)\|\Pi_{E'} - \Pi_E\| \leq \frac{Cs}{\epsilon}\cdot\epsilon^2/2 \leq C\epsilon$. This proves the claim (after we change $\epsilon$ by a constant factor). □

*5.3.2 Proof of the theorem.* Finally, we apply Proposition 5.15 to the to the analysis of our algorithm. The proof can be found in the full version of the paper.

**Lemma 5.16.** *Let $C$ be a subset of $s$-smooth functions $\mathbb{R}^k \to [-1,1]$ and let $\mathrm{Net}_{C,m}$ be an $\epsilon$-net as guaranteed by Proposition 5.15. Let $\hat{E}$ be a $m$-dimensional subspace of $\mathbb{R}^n$ and let $A \in \mathbb{R}^{m\times n}$ with the following two properties: (i) the rows of $A$ span $\hat{E}$ and (ii) $\|AA^T - I\|_F \leq \kappa$. Then, for any Lipschitz function $f_{\mathrm{sm}}$, we have that*

$$\Big|\rho_{\hat{E},C}(f_{\mathrm{sm}}) - \max_{h\in\mathrm{Net}_{m,C}} \mathop{\mathbf{E}}_{\mathbf{x}}[h(A\mathbf{x})\cdot f_{\mathrm{sm}}(\mathbf{x})]\Big| \leq \frac{C's\kappa}{\epsilon} + \epsilon,$$

*for an absolute constant $C'$.*

**Proof of Theorem 4.3:**

Let $C^*$ be the averaged class of $C$, as in Definition 5.6. Let $\mathrm{Net}_{m,C^*}$ be the set of functions guaranteed by Proposition 5.15 – with smoothness parameter $s$, error parameter $\epsilon/4$ and $m = M$ as instantiated in the algorithm Implicit Projection. Let us also set $\nu = \epsilon^2/(100C's)$ for the constant $C'$ appearing in Lemma 5.16. Let us now invoke algorithm Implicit projection with smoothness parameter $u = \nu/s$, Lipschitz parameter $L = O(s/\nu)$, the error parameter $\nu$ and junta arity parameter $k$.

Suppose $h_* \in \mathrm{Ind}_n(C)$ such that

$$h_* = \arg\max_{h\in\mathrm{Ind}_n(C)} \mathop{\mathbf{E}}_{\mathbf{x}}[f_{\mathrm{sm}}(\mathbf{x})h(\mathbf{x})].$$

Lemma 5.1 guarantees that with probability 9/10, we get a matrix $\hat{W}$ and points $\mathbf{x}_1,\ldots,\mathbf{x}_M$ such that the following conditions are satisfied: let $B^T$ be the matrix where the $j^{th}$ row is $\nabla f_{\mathrm{sm}}(\mathbf{x}_j)$. Let $\hat{E}$ be the row span of $B^T$.

1. $\|\hat{I} - \hat{W}B^T B\hat{W}^T\|_F \leq \nu/2$. Here $\hat{I}$ is the identity matrix in $m$ dimensions where $m = \dim(\hat{E})$.
2. For $g = \mathcal{A}_{\hat{E}}h_*$,

$$|\mathbf{E}[f_{\mathrm{sm}}(\mathbf{x})\cdot g(\mathbf{x})] - \mathbf{E}[f_{\mathrm{sm}}(\mathbf{x})\cdot h_*(\mathbf{x})]| \leq \frac{\nu}{2}. \tag{12}$$

Also, by Lemma 5.16, we have that

$$\Big|\rho_{\hat{E},C^*}(f_{\mathrm{sm}}) - \max_{h\in\mathrm{Net}_{m,C^*}} \mathop{\mathbf{E}}_{\mathbf{x}}[h(\hat{W}B^T\mathbf{x})\cdot f_{\mathrm{sm}}(\mathbf{x})]\Big| \leq \frac{C's}{\epsilon}\cdot\frac{\nu}{2} + \frac{\epsilon}{4} < \frac{51\cdot\epsilon}{200} \tag{13}$$

Next, since $g = \mathcal{A}_{\hat{E}}h_* \in \mathrm{Ind}_{\hat{E}}(C^*)$ (by Claim 5.7), we have that

$$\rho_{\hat{E},C^*}(f_{\mathrm{sm}}) \geq \mathbf{E}[f_{\mathrm{sm}}(\mathbf{x})\cdot g(\mathbf{x})] \geq \mathbf{E}[f_{\mathrm{sm}}(\mathbf{x})\cdot h_*(\mathbf{x})] - \frac{\nu}{2}, \tag{14}$$

where the second inequality follows from (12). On the other hand, if $\tilde{g} \in \mathrm{Ind}_{\hat{E}}(C^*)$ maximizes the correlation with $f_{\mathrm{sm}}$, there exists (again by Claim 5.7) $\tilde{h}_* \in \mathrm{Ind}_n(C)$ with $\mathcal{A}_{\hat{E}}\tilde{h}_* = \tilde{g}$, and hence (by Lemma 5.1)

$$\begin{aligned}\rho_{\hat{E},C^*}(f_{\mathrm{sm}}) &= \mathbf{E}[f_{\mathrm{sm}}(\mathbf{x})\cdot(\mathcal{A}_{\hat{E}}\tilde{h}_*)(\mathbf{x})]\\ &\leq \mathbf{E}[f_{\mathrm{sm}}(\mathbf{x})\cdot\tilde{h}_*(\mathbf{x})] + \frac{\nu}{2}\\ &\leq \mathbf{E}[f_{\mathrm{sm}}(\mathbf{x})\cdot h_*(\mathbf{x})] + \frac{\nu}{2}.\end{aligned} \tag{15}$$

Together with (14), we have

$$|\rho_{\hat{E},C^*}(f_{\mathrm{sm}}) - \mathbf{E}[f_{\mathrm{sm}}(\mathbf{x})\cdot h_*(\mathbf{x})]| \leq \frac{\nu}{2};$$

combined with (13) (and recalling that we chose $h_*$ to be a correlation-maximizer in $\mathrm{Ind}_n(C)$, we have

$$\Big|\rho_{\mathbb{R}^n,C} - \max_{h\in\mathrm{Net}_{m,C^*}} \mathop{\mathbf{E}}_{\mathbf{x}}[h(\hat{W}B^T\mathbf{x})\cdot f_{\mathrm{sm}}(\mathbf{x})]\Big| \leq \frac{51\epsilon}{100} + \frac{\nu}{2} = \frac{52\epsilon}{100}. \tag{16}$$

Thus, for our purposes, it suffices to (approximately) compute $\max_{h\in\mathrm{Net}_{m,C^*}} \mathbf{E}_{\mathbf{x}}[h(\hat{W}B^T\mathbf{x})\cdot f_{\mathrm{sm}}(\mathbf{x})]$. Towards this, consider any fixed $h \in$ Net. We set $T = O(\epsilon^{-2}\log(1/\zeta))$ where $\zeta = 1/(10\cdot|\mathrm{Net}_{m,C^*}|)$. Sample $T$ points from the standard Gaussian $\gamma_n$ – call these points $\mathbf{z}_1,\ldots,\mathbf{z}_T$. By applying the Chernoff bounds, observe that for any $h \in$ Net, with probability $1-\zeta$,

$$\left|\mathbf{E}_{\mathbf{x}}[h(\hat{W}B^T\mathbf{x})\cdot f_{\mathrm{sm}}(\mathbf{x}) - \frac{1}{T}\sum_{j=1}^{T} h(\hat{W}B^T\mathbf{z}_j)\cdot f_{\mathrm{sm}}(\mathbf{z}_j)\right| \leq \epsilon/4.$$

From a union bound, it follows that with probability 9/10,

$$\left| \max_{h \in \mathrm{Net}_{m,C^*}} \mathbf{E}_{\mathbf{x}}[h(\hat{W}B^T\mathbf{x}) \cdot f_{\mathrm{sm}}(\mathbf{x})] - \max_{h \in \mathrm{Net}_{m,C^*}} \frac{1}{T}\sum_{j=1}^{T} h(\hat{W}B^T\mathbf{z}_j) \cdot f_{\mathrm{sm}}(\mathbf{z}_j) \right| \leq \epsilon/4. \quad (17)$$

Combining (17) and (16), we get

$$\left| \rho_{\mathbb{R}^n,C}(f_{\mathrm{sm}}) - \max_{h \in \mathrm{Net}_{m,C^*}} \frac{1}{T}\sum_{j=1}^{T} h(\hat{W}B^T\mathbf{z}_j) \cdot f_{\mathrm{sm}}(\mathbf{z}_j) \right| < \frac{2\epsilon}{3}. \quad (18)$$

Thus, it suffices to compute the quantity

$$\mathrm{Corr} = \max_{h \in \mathrm{Net}_{m,C^*}} \frac{1}{T}\sum_{j=1}^{T} h(\hat{W}B^T\mathbf{z}_j) \cdot f_{\mathrm{sm}}(\mathbf{z}_j),$$

up to additive error $\pm\epsilon/3$ and upper bound the query complexity of computing this estimate. Observe that computing $\{f_{\mathrm{sm}}(\mathbf{z}_j)\}_{j=1}^{T}$ requires $T$ queries. Using Lemma 5.1, we have

$$\|\hat{W}\|_2 \leq \frac{20k}{\nu} := \Delta$$

Set $\theta = \frac{\epsilon^2}{200 \cdot C \cdot \Delta \cdot s \cdot \sqrt{m}}$. Here $C$ is the constant appearing in Fact 5.13. We now invoke algorithm Project-on-gradient from Lemma 2.18. Then, we get that for any $\mathbf{z}_j$ (for $1 \leq j \leq T$),

$$\Pr_{\mathbf{x}_i \sim \gamma_n}[|\mathrm{Est}(\mathbf{x}_i, \mathbf{z}_j) - \langle \nabla f_{\mathrm{sm}}(\mathbf{x}_i), \mathbf{z}_j \rangle| > \theta] \leq \frac{1}{200T \cdot m}.$$

Further, we can compute $\pm\theta$ estimate to $\mathrm{Est}(\mathbf{x}_i, z_j)$ (with confidence $1 - \frac{1}{200T \cdot m}$) where the query complexity is $\mathrm{poly}(T \cdot m, 1/\theta)$. This means that with probability 0.99, for each $1 \leq j \leq T$ and $1 \leq i \leq m$, we have $\pm 2\theta$ estimates (denoted by $\chi_{i,j}$) for each $\langle \nabla f_{\mathrm{sm}}(\mathbf{x}_i), \mathbf{z}_j \rangle$. In other words, for each $1 \leq j \leq T$, we get a vector $\Xi_j$ which satisfies

$$\|\Xi_j - B^T\mathbf{z}_j\| \leq 2\theta\sqrt{m}.$$

Since $\|\hat{W}\| \leq \Delta$, this means that for all $1 \leq j \leq T$,

$$\|\hat{W}\Xi_j - \hat{W}B^T\mathbf{z}_j\| \leq 2\theta\sqrt{m}\Delta = \frac{\epsilon^2}{100C \cdot s}.$$

Since $h \in \mathrm{Net}$ is $Cs/\epsilon$-Lipschitz, this implies that for each $1 \leq j \leq T$,

$$\left|h(\hat{W}\Xi_j) - h(\hat{W}B^T\mathbf{z}_j)\right| \leq \frac{\epsilon}{100}.$$

Consequently, this gives a $\pm\epsilon/100$ additive estimate of the quantity

$$\frac{1}{T}\sum_{j=1}^{T} h(\hat{W}B^T\mathbf{z}_j) \cdot f_{\mathrm{sm}}(\mathbf{z}_j).$$

Recalling (18), we have shown that the algorithm produces a $\pm\epsilon$-additive estimate of $\rho_{\mathbb{R}^n,C}(f_{\mathrm{sm}})$. It remains to bound the query complexity of the algorithm. The query complexity of the algorithm Implicit projection (from Lemma 5.1) is $\mathrm{poly}(k, 1/u, 1/v, L)$ where $v = \epsilon^2/(100C's)$ ($C'$ is the constant appearing in Lemma 5.16). Thus, the query complexity of this part is $\mathrm{poly}(k, s, L, 1/\epsilon)$.

For the hypothesis testing part, the query complexity can be bounded as follows:

(1) We make $T$ queries to $f_{\mathrm{sm}}$ where $T = O(\epsilon^{-2}\log|\mathrm{Net}|$.

(2) For each $1 \leq j \leq m$ and $1 \leq i \leq T$, we compute a $\pm\theta$ approximation to $\mathrm{Est}(\mathbf{x}_i, \mathbf{z}_j)$ – the query complexity of each is $\mathrm{poly}(T \cdot m, 1/\theta)$.

Thus, the total query complexity is bounded by $\mathrm{poly}(T, m, 1/\theta)$. Using the fact that $m \leq M$ (where $M$ is set in algorithm Implicit projection) and plugging in the value of the parameters, we get the final bound on the query complexity.

Finally, we remark that our analysis so far was based on assuming that we have exact oracle access to $f_{\mathrm{sm}}$. However, we only have oracle access to $f$ and approximate oracle to $f_{\mathrm{sm}}$ (via Lemma 4.2). To address this issue, we observe that the algorithm Implicit projection only uses the oracle to $f$ and not to $f_{\mathrm{sm}}$ (the only invocation of these oracles is when we call the routine Compute-inner-product). In the hypothesis testing part, (i) we only use the oracle to $f$ when we invoke the algorithm Project-on-gradient. (ii) we use the oracle for $f_{\mathrm{sm}}$ when we approximate Corr to error $\pm\epsilon/3$. However, it is easy to see that for this, it suffices to have an oracle for $f_{\mathrm{sm}}$ with (say) $O(\epsilon^{-1.5})$ additive accuracy. By Lemma 4.2, this can be simulated with an oracle for $f$ with $O(\epsilon^{-3})$ overhead given an oracle to $f$. This finishes our proof.

□

## 6 LEARNING THE LINEAR-INVARIANT STRUCTURE

The proof of Theorem 1.4 is essentially the same as the proof of Theorem 4.3; we construct the same net of functions and estimate the correlations of each of them. The only difference is that instead of outputting the maximum correlation value of a function in the net, we output the set of functions that have a large correlation.

**Proof of Theorem 1.4:** Let $\mathrm{Net}_{m,C^*}$ be as in the proof of Theorem 4.3. With the same $\delta$ as in that proof, with probability 9/10 we can simultaneously estimate $\mathbf{E}_x[h(\hat{W}B^T\mathbf{x}) \cdot f_{\mathrm{sm}}(\mathbf{x})]$ to error $\pm\epsilon/8$ for all $h \in \mathrm{Net}_{m,C^*}$.

Now consider the algorithm that returns all $h \in \mathrm{Net}_{m,C^*}$ for which our estimate of $\mathbf{E}_x[h(\hat{W}B^T\mathbf{x}) \cdot f_{\mathrm{sm}}(\mathbf{x})]$ is at least $\rho - 4\epsilon$; call the returned set $\mathcal{G}$. It follows that for every $\hat{g} \in \mathcal{G}$,

$$\mathbf{E}_x[\hat{g}(\hat{W}B^T\mathbf{x}) \cdot f_{\mathrm{sm}}(\mathbf{x})] \geq \rho - 5\epsilon,$$

and so the first claim of the theorem follows.

For the second claim, take any $g \in \mathrm{Ind}_n(C)$ and let $\hat{E}$ be the range of $\hat{W}B^T$. Since (by Claim 5.7) $\mathcal{A}_{\hat{E}}g \in \mathrm{Ind}_{\hat{E}}(C^*)$, there is some $\hat{g} \in \mathrm{Net}_{m,C^*}$ such that

$$\mathbf{E}_{\mathbf{x}}[(\hat{g}(\hat{W}B^T\mathbf{x}) - \mathcal{A}_{\hat{E}}g)^2] \leq \epsilon^2. \quad (19)$$

Now, if $g$'s correlation with $f$ is at least $\rho - \epsilon$, then by Lemma 5.1 $\mathcal{A}_{\hat{E}}g$ has correlation at least $\rho - 2\epsilon$ with $f$, and so by (19), $\hat{g} \circ (\hat{W}B^T)$ has correlation with $f$ at least $\rho - 3\epsilon$, and hence the definition of $\mathcal{G}$ ensures that $\hat{g} \in \mathcal{G}$. Going back to (19), the function $\hat{g}$ witnesses the second claim of the theorem. □

## ACKNOWLEDGMENTS

A. D. is supported by NSF grant CCF-1926872, CCF-1910534 and CCF-2045128 (CAREER). E. M. is Supported by Simons-NSF DMS-2031883, Vannevar Bush Faculty Fellowship ONR-N00014-20-1-2826, Simons Investigator award and NSF DMS-1737944. J. N. is

supported by by the Alfred P. Sloan Foundation and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2047/1 – 390685813. This work was done (in part) while the authors were participating in the program on "Probability, Geometry, and Computation in High Dimensions" at the Simons Institute for the Theory of Computing.

## REFERENCES

[1] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?. In *Advances in Neural Information Processing Systems*. 2654–2662.
[2] Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. 2012. Active property testing. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE, 21–30.
[3] Mihir Bellare, Oded Goldreich, and Madhu Sudan. 1998. Free bits, PCPs, and nonapproximability–towards tight results. *SIAM J. Comput.* 27, 3 (1998), 804–915.
[4] E. Blais. 2008. Improved bounds for testing juntas. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. Springer, 317–330.
[5] E. Blais. 2009. Testing juntas nearly optimally. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 151–158.
[6] E. Blais, C. Canonne, T. Eden, A. Levi, and D. Ron. 2018. Tolerant junta testing and the connection to submodular optimization and function isomorphism. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2113–2132.
[7] A. Blum. 1994. Relevant examples and relevant features: Thoughts from computational learning theory. (1994). in AAAI Fall Symposium on 'Relevance'.
[8] A. Blum, A. Frieze, R. Kannan, and S. Vempala. 1997. A polynomial time algorithm for learning noisy linear threshold functions. *Algorithmica* 22, 1/2 (1997), 35–52.
[9] A. Blum and P. Langley. 1997. Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence* 97, 1-2 (1997), 245–271.
[10] Nader H Bshouty. 2019. Almost Optimal Distribution-Free Junta Testing. In *34th Computational Complexity Conference (CCC 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
[11] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 535–541.
[12] S. Chakraborty, E. Fischer, D. García-Soriano, and A. Matsliah. 2012. Junto-symmetric functions, hypergraph isomorphism and crunching. In *27th Annual Conference on Computational Complexity (CCC)*. IEEE, 148–158.
[13] Xi Chen, Adam Freilich, Rocco A Servedio, and Timothy Sun. 2017. Sample-Based High-Dimensional Convexity Testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*.
[14] X. Chen, Z. Liu, Rocco A. Servedio, Y. Sheng, and J. Xie. 2018. Distribution free junta testing. In *Proceedings of the ACM STOC 2018*.
[15] Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. 2017. Settling the Query Complexity of Non-adaptive Junta Testing. In *Proceedings of the 32Nd Computational Complexity Conference*. 26:1–26:19.
[16] Amit Daniely. 2016. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 105–117.
[17] Anindya De, Elchanan Mossel, and Joe Neeman. 2019. Is your function low dimensional?. In *Conference on Learning Theory, COLT 2019 (Proceedings of Machine Learning Research)*, Vol. 99. 979–993. Full version at `https://arxiv.org/abs/1806.10057`.
[18] Anindya De, Elchanan Mossel, and Joe Neeman. 2019. Junta correlation is testable. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 1549–1563.
[19] Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. 2019. Distribution-independent PAC learning of halfspaces with Massart noise. In *Advances in Neural Information Processing Systems*. 4749–4760.
[20] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. 2018. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 1061–1073.
[21] I. Diakonikolas, H. Lee, K. Matulef, K. Onak, R. Rubinfeld, R. Servedio, and A. Wan. 2007. Testing for Concise Representations. In *Proc. 48th Ann. Symposium on Computer Science (FOCS)*. 549–558.
[22] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. 2004. Testing juntas. *J. Computer & System Sciences* 68, 4 (2004), 753–787.
[23] V. Guruswami and P. Raghavendra. 2006. Hardness of learning halfspaces with noise. In *Proc. 47th IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 543–552.
[24] S. Halevy and E. Kushilevitz. 2004. Distribution-Free Connectivity Testing for Sparse Graphs. *Algorithmica* 51, 1 (2004), 24–48.
[25] S. Halevy and E. Kushilevitz. 2007. Distribution-Free Property Testing. *SIAM J. Comput.* 37, 4 (2007), 1107–1138.
[26] Prahladh Harsha, Adam Klivans, and Raghu Meka. 2013. An invariance principle for polytopes. *Journal of the ACM (JACM)* 59, 6 (2013), 1–25.
[27] M. Kearns, R. Schapire, and L. Sellie. 1994. Toward Efficient Agnostic Learning. *Machine Learning* 17, 2/3 (1994), 115–141.
[28] A. Klivans, R. O'Donnell, and R. Servedio. 2008. Learning Geometric Concepts via Gaussian Surface Area. In *Proc. 49th IEEE Symposium on Foundations of Computer Science (FOCS)*. 541–550.
[29] P. Kothari, A. Nayyeri, R. O'Donnell, and C. Wu. 2014. Testing Surface Area. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*. 1204–1214.
[30] M. Ledoux. 1994. Semigroup proofs of the isoperimetric inequality in Euclidean and Gauss space. *Bull. Sci. Math.* 118 (1994), 485–510.
[31] Michel Ledoux. 2000. The geometry of Markov diffusion generators. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, Vol. 9. 305–366.
[32] Pascal Massart and Élodie Nédélec. 2006. Risk bounds for statistical learning. *The Annals of Statistics* 34, 5 (2006), 2326–2366.
[33] K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. 2010. Testing Halfspaces. *SIAM J. on Comput.* 39, 5 (2010), 2004–2047.
[34] Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. 2009. Testing ±1-weight halfspace. In *APPROX-RANDOM*. 646–657.
[35] E. H. Moore. 1920. On the Reciprocal of the General Algebraic Matrix. *Bull. Amer. Math. Soc.* 26 (1920), 394–395.
[36] J. Neeman. 2014. Testing surface area with arbitrary accuracy. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*. 393–397.
[37] M. Parnas, D. Ron, and R. Rubinfeld. 2006. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences* 72, 6 (2006), 1012–1042.
[38] M. Parnas, D. Ron, and A. Samorodnitsky. 2002. Testing Basic Boolean Formulae. *SIAM J. Disc. Math.* 16 (2002), 20–46. citeseer.ifi.unizh.ch/parnas02testing.html
[39] R. Penrose. 1955. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society* 51, 3 (1955), 406–413. https://doi.org/10.1017/S0305004100030401
[40] G. Pisier. 1986. Probabilistic methods in the geometry of Banach spaces. In *Lecture notes in Math.* Springer, 167–241.
[41] Dana Ron and Rocco A Servedio. 2015. Exponentially improved algorithms and lower bounds for testing signed majorities. *Algorithmica* 72, 2 (2015), 400–429.
[42] Mark Rudelson and Roman Vershynin. 2007. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)* 54, 4 (2007), 21–es.
[43] M. Sağlam. 2018. Near Log-Convexity of Measured Heat in (Discrete) Time and Consequences. In *59th IEEE Annual Symposium on Foundations of Computer Science*. 967–978.
[44] R. Servedio, L-Y. Tan, and J. Wright. 2015. Adaptivity helps for testing juntas. In *Proceedings of CCC*, Vol. 33. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
[45] Santosh Vempala and Ying Xiao. 2013. Complexity of learning subspace juntas and ICA. In *2013 Asilomar Conference on Signals, Systems and Computers*. IEEE, 320–324.
[46] Santosh S Vempala. 2010. Learning convex concepts from gaussian distributions with PCA. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, 124–130.