

A Mysterious Connection Between Tolerant Junta Testing and Agnostically Learning Conjunctions

Xi Chen*
Columbia University

Shyamal Patel†
Columbia University

Rocco A. Servedio‡
Columbia University

Abstract

The main conceptual contribution of this paper is identifying a previously unnoticed connection between two central problems in computational learning theory and property testing: *agnostically learning conjunctions* and *tolerantly testing juntas*. Inspired by this connection, the main technical contribution is a pair of improved algorithms for these two problems.

In more detail,

- We give a distribution-free algorithm for agnostically PAC learning conjunctions over $\{\pm 1\}^n$ that runs in time $2^{\tilde{O}(n^{1/3})}$, for constant excess error ε . This improves on the fastest previously published algorithm, which runs in time $2^{\tilde{O}(n^{1/2})}$ [KKMS08].
- Building on the ideas in our agnostic conjunction learner and using significant additional technical ingredients, we give an adaptive tolerant testing algorithm for k -juntas that makes $2^{\tilde{O}(k^{1/3})}$ queries, for constant “gap parameter” ε between the “near” and “far” cases. This improves on the best previous results, due to [ITW21, NP24], which make $2^{\tilde{O}(\sqrt{k})}$ queries. Since there is a known $2^{\tilde{\Omega}(\sqrt{k})}$ lower bound for *non-adaptive* tolerant junta testers, our result shows that adaptive tolerant junta testing algorithms provably outperform non-adaptive ones.

*Email: xichen@cs.columbia.edu

†Email: shyamalpatelb@gmail.com

‡Email: ras2105@columbia.edu

Contents

1	Introduction	1
1.1	Our results: Better agnostic conjunction learning algorithms and tolerant junta testing algorithms	3
1.2	Discussion	4
2	Technical Overview	5
2.1	Overview for agnostically learning conjunctions	5
2.2	Overview of quantum tolerant junta testing	6
2.3	Overview of classical tolerant junta testing	8
3	Preliminaries	11
3.1	Preliminaries for agnostically learning conjunctions	11
3.2	Fourier basics	11
3.3	Preliminaries for tolerantly testing juntas	11
3.4	Junta Correlation	12
3.5	Bernoulli noise	13
3.6	Randomized algorithms and estimating function values, Fourier coefficients, sums of squares of Fourier coefficients, etc.	13
3.7	Flat Polynomials	15
3.8	Normalized Influences	16
3.9	Local estimators	16
3.10	Coordinate oracles and approximate versus exact computation	16
4	Agnostically learning conjunctions	17
4.1	Setup for the learning algorithm	17
4.2	The algorithm	18
4.3	Analysis of the algorithm	18
4.4	Proof of Theorem 1	21
5	Warm-Up: A $2^{\tilde{O}(k^{1/3})}$-Query Quantum Tolerant Junta Tester	23
5.1	Drawing from the spectral sample	23
5.2	Smooth functions and local mean estimation	23
5.3	The Sharp Noise Operator	25
5.4	Parameters in the main algorithm	26
5.5	Local estimation of junta correlations	27
5.6	Putting It Together: A Quantum Tester and the Proof of Theorem 2	31
5.6.1	Efficiency	31
5.6.2	Correctness	32
6	A $2^{\tilde{O}(k^{1/3})}$ Query Classical Tolerant Junta Tester	35
6.1	Approximating Normalized Influences	35
6.2	Filtering Coordinates	40
6.2.1	The Idealized Setting	41
6.2.2	Handling the Approximations	47
6.3	Finding a Pure Set of Coordinates	51
6.4	Putting It All Together	55
6.4.1	Parameters for the Main Algorithm	55
6.4.2	The Testing Algorithm	55
6.4.3	Efficiency	57
6.4.4	Correctness	58
6.4.5	Proof of Theorem 3	61

1 Introduction

The American mystery deepens.

— Don DeLillo, White Noise

A broad theme that has emerged across modern data analysis problems in various subfields of theoretical computer science is the importance of dealing with *noise*. In both computational learning theory and property testing, the first generation of models that were considered in the early days of the field were “noise-free”: the original PAC learning model of Valiant [Val84] did not allow for the possibility of mislabeled examples, and the original property testing model for Boolean functions that was introduced by Blum, Luby and Rubinfeld [BLR93] was about testing whether a function exactly has a given property of interest. As these fields matured, though, the motivation for extending these original models became increasingly clear: since imperfect data is ubiquitous in real settings, it is natural to consider relaxed learning and testing models that allow for noise. In learning theory, this led to widespread interest in the model of *agnostic learning* [Hau92, KSS94], which may be viewed as a demanding version of PAC learning in the presence of noise that can affect both examples and labels. In property testing, the model of *tolerant testing* [PRR06] was introduced and intensively studied, in which the goal is to determine whether a function is ε_1 -close to or ε_2 -far from having a property of interest. Today, a large fraction of contemporary research in learning theory deals with agnostic learning, and likewise for research in Boolean function testing and the tolerant testing model.

Beyond the clear practical motivations for tackling noisy problems, from a theoretical perspective many fundamental problems that are relatively straightforward in the noiseless setting become significantly richer and more challenging when we make the more realistic assumption that the functions that we are working with may be corrupted by noise. For example, in computational learning theory, PAC learning an unknown *conjunction* when there is no noise is an easy exercise, whereas agnostically learning conjunctions is much harder (as we discuss in detail below). As a different example, in property testing the well-studied problem of testing whether an unknown Boolean function is exactly, versus far from, a *junta* was first solved more than two decades ago and is now well understood [FKR⁺04, CG04, Bla08, Bla09, Saĝ18, CST⁺18], whereas, as we discuss in detail below, the *tolerant junta testing* problem remains largely open, with significant gaps in our current understanding.

Our work. In this paper we identify and explore a previously unnoticed connection between the two problems mentioned above: efficient algorithms for *agnostically learning conjunctions* and efficient¹ algorithms for *tolerantly testing juntas*. (See Section 3.1 and Section 3.4 for formal definitions of these well-studied problems.) We remark that conjunctions are one of the most fundamental classes to consider in the distribution-free agnostic learning setting; indeed, already in a FOCS 2003 tutorial Avrim Blum highlighted the importance of developing agnostic learning algorithms for conjunctions [Blu03]. Additionally, it is well known that sufficiently improved algorithms for agnostically learning conjunctions would imply faster algorithms for learning DNF formulas, one of the touchstone problems in computational learning theory (we discuss this more after stating Theorem 1). Similarly, junta testing has been a touchstone problem in the field of Boolean function property testing for more than twenty years since the influential work of Fischer et al. [FKR⁺04,

¹Since “efficient” in the learning context refers to *running time*, whereas “efficient” in the testing context refers to *query complexity*, such a connection may seem curious indeed. However, we remark that since the known algorithms for agnostically learning conjunctions can be cast in the Statistical Query model of Kearns [Kea98], we can view these algorithmic results as having information-theoretic analogues, and thus aligning with the (information-theoretic) query complexity results for tolerant junta testing.

CG04, Bla08, Bla09, BGMdW13, Sağ18, STW15, CST⁺18, Bsh19], and the problem of tolerant junta testing has been a particular focus of research effort in recent years [BCE⁺19, DMN19, LW19, ITW21, PRW22, CP23, CDL⁺24, NP24].

The connection that we identify between agnostically learning conjunctions and tolerantly testing juntas enables us to give new algorithms for both problems, improving on the best prior results. Before describing our new results, we begin by highlighting some similarities between the prior state of the art for these two problems and the ideas and tools used to establish those results.

In terms of upper bounds, prior to the current work the fastest published algorithm for distribution-free agnostic learning of conjunctions over $\{\pm 1\}^n$ was the “ L_1 polynomial regression algorithm” of [KKMS08], which runs in time $2^{\tilde{O}(\sqrt{n})}$ for agnostically learning conjunctions to constant excess error $\varepsilon = \Theta(1)$ (i.e. obtaining a hypothesis with error at most $\text{opt} + \varepsilon$). We remark that $2^{\tilde{O}(\sqrt{n})}$ is also the best known (again, prior to the current work) query complexity for tolerantly testing k -juntas over $\{\pm 1\}^n$ when $k = \Theta(n)$ [ITW21, NP24]. Furthermore, in both [KKMS08] and [NP24] the fact that the n -variable AND function has approximate degree $O(\sqrt{n})$ plays an important role in obtaining the square-root savings. Turning to lower bounds, we also find that there is an alignment between the current state of the art for these two problems. At the level of results, in [Fel12], Feldman gave a $n^{\Omega(\log(1/\varepsilon))}$ lower bound for agnostically learning conjunctions over $\{\pm 1\}^n$ to excess error ε under the uniform distribution in the Statistical Query (SQ) model, while Chen and Patel [CP23] gave a $k^{\Omega(\log(1/\varepsilon))}$ -query lower bound for tolerantly testing juntas with an ε -width gap between the “near” and “far” cases. In terms of techniques underlying these results, the proof in [Fel12] crucially relies on the fact that a k -variable conjunction disagrees with the parity function over the same set of variables on $1/2 - 2^{-k}$ fraction of all inputs, while the argument of [CP23] uses the related notion of k -wise independence.

At this point, perhaps the reader is wondering whether the quantitative correspondences between results that are highlighted above, as well as the similarities in the technical ingredients, may be merely a coincidence. To support our claim that there is a meaningful connection between these two problems, we discuss some further points of correspondence below. This discussion is presented in the form of a brief dialogue, taking place over some wine, in which a BELIEVER tries to increase the confidence of a SKEPTIC in the claimed connection:

SKEPTIC Agnostic learning and tolerant testing seem very different, for the following reason: When we are doing agnostic learning, there is an arbitrary distribution \mathcal{D} over the example space $\{\pm 1\}^n$. In contrast, in property testing (tolerant or otherwise), we measure the distance between two functions f and g with respect to the uniform distribution over $\{0, 1\}^n$, which is very different from an arbitrary distribution. So these frameworks do not seem to align with each other. *“I honestly tell you, I cannot believe it.”* [Mel57]

BELIEVER *“I told you, you must have confidence”* [Mel57] The alignment exists because of the specific learning problem (conjunctions) and testing problem (juntas) being considered. The relevant distribution, for the tolerant junta testing problem, is not the uniform distribution over $\{\pm 1\}^n$ but rather the *spectral sample* distribution \mathcal{P}_f , over all subsets of $[n]$, induced by the Fourier coefficients of $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$ (see [Definition 33](#)); the distribution \mathcal{P}_f , not the uniform distribution, is the appropriate analogue of the arbitrary distribution \mathcal{D} in the agnostic learning problem. To see this, observe that for the tolerant junta testing problem, the goal can (roughly speaking) be reframed as finding the “right” subset of variables (subject to being of size k) that “cover” the largest possible amount of the spectral sample distribution (see the discussion of “junta correlation” in [Section 3.2](#)). Similarly, for the agnostic conjunction problem, the goal can be reframed (again roughly speaking) as finding the “right” subset of

variables that “cover” the maximum possible amount of the distribution of the positive examples (subject to not covering too much of the distribution of negative examples). “*Confidence restored?... Sit down, sir, I beg, and take some of this wine.*” [Mel57]

SKEPTIC “*Ah, wine is good, and confidence is good; but can wine or confidence percolate down through all the stony strata of hard considerations, and drop warmly and ruddily into the cold cave of truth?*” [Mel57] I still have my doubts. The junta problem involves testing whether an n -variable function is a k -junta: there are inherently two parameters, n and k . But in agnostic learning of conjunctions there is no analogue of the “ k ” parameter in junta testing; the conjunction can be of arbitrary size relative to n . How can your claimed connection overcome this? “*But think of the obstacles!*” [Mel57]

BELIEVER “*I have confidence to remove obstacles, though mountains.*” [Mel57] The papers of [DMN19] and [ITW21] have developed powerful techniques, namely highly efficient simulations of “coordinate oracles” (see Section 3.10), which can be used in the tolerant junta testing problem to effectively reduce the number of coordinates from n down to $k' = O_\varepsilon(k)$. So to answer your question, we do not introduce a “ k ” parameter into the conjunction setting; rather, in the tolerant junta setting, we can effectively get rid of the “ n ” parameter and replace it by $k' = O_\varepsilon(k)$. A subset of $k \leq O_\varepsilon(k)$ relevant variables for the junta in the tolerant testing scenario is not so different from an arbitrary subset of relevant variables for the conjunction in the agnostic learning scenario. “*But again I say, you must have confidence.*” [Mel57]

Leaving the SKEPTIC and the BELIEVER to their dialogue and their wine, we remark that the connection between these two problems will be fleshed out in significantly more detail later, when we give an overview of our agnostic learning algorithm and our tolerant junta testing algorithm and describe the common underlying ideas and similarities between the two algorithms; see Section 2.2. We turn to a description of our technical results.

1.1 Our results: Better agnostic conjunction learning algorithms and tolerant junta testing algorithms.

In this work we extend the connection between these two problems, by (i) giving a faster algorithm for agnostically learning conjunctions, and (ii) using the ideas in that algorithm to obtain an improved tolerant junta testing algorithm with an analogous query complexity. Regarding (i), we prove the following:

Theorem 1. Algorithm 1 agnostically learns conjunctions over $\{\pm 1\}^n$ to excess error ε in time $2^{n^{1/3} \cdot \text{polylog}(n, 1/\varepsilon)}$.

We remark that Diakonikolas, Kane, and Ren have very recently achieved a similar result to Theorem 1 [DKR25].

It is well known that any algorithm for distribution-free agnostically learning conjunctions to excess error ε can be used, taking $\varepsilon = 1/(10s)$, in tandem with a boosting algorithm (e.g. [Sch90] or [Fre95]) to learn s -term DNF, or even the richer class of total-integer-weight- s linear threshold functions over conjunctions. This is simply because thanks to the “discriminator lemma” [HMP⁺93], given any weight- s -LTF-over-conjunctions target function, for any distribution some conjunction must have error at most $1/2 - 1/s$ in predicting the target function; so using an agnostic learner, we recover a weak hypothesis with error $1/2 - 1/(2s)$, which can be used for boosting. Plugging in Theorem 1 as the agnostic conjunction learner, we get a DNF learning algorithm running in

time $2^{n^{1/3} \cdot \text{polylog}(n,s)}$, which essentially matches the state of the art algorithm for this well-studied problem [KS04].

Inspired by [Algorithm 1](#) and its analysis, we give the following warm-up result for our main tolerant testing algorithm (in the theorem below, \mathcal{J}_k is the class of all Boolean-valued k -juntas):

Theorem 2. [Algorithm 2](#) $\pm\varepsilon$ -accurately estimates $\text{dist}(f, \mathcal{J}_k)$, for $f : \{\pm 1\}^{k'} \rightarrow \{\pm 1\}$, where $k' = \text{poly}(k, 1/\varepsilon)$, and makes $2^{k^{1/3} \cdot \text{polylog}(k, 1/\varepsilon)}$ quantum queries to f .

[Theorem 2](#) is stated in the language of “distance estimation,” i.e. estimating the distance $\text{dist}(f, \mathcal{J}_k) := \min_{g \in \mathcal{J}_k} \Pr_{\mathbf{x} \sim \{\pm 1\}^{k'}} [f(\mathbf{x}) \neq g(\mathbf{x})]$ between f and the closest k -junta. It is well known (see e.g. [PRR06]) that the problem of $\pm\varepsilon$ -accurate distance estimation for a class of functions is, up to small factors, equivalent to the problem of $(\varepsilon_1, \varepsilon_2 = \varepsilon_1 + \varepsilon)$ -tolerant testing for that class (see [Section 3.4](#)). Thus an equivalent statement of [Theorem 2](#) in the language of tolerant testing is the following: for $k' = \text{poly}(k, 1/\varepsilon)$, there is a quantum query algorithm that makes $2^{k^{1/3} \cdot \text{polylog}(k, 1/\varepsilon)}$ quantum queries to $f : \{\pm 1\}^{k'} \rightarrow \{\pm 1\}$ and $(\varepsilon_1, \varepsilon_2)$ -tolerantly tests f for the property of being a k -junta, where $\varepsilon = \varepsilon_2 - \varepsilon_1$.

Building on the ideas and ingredients in [Theorem 2](#), our main tolerant testing result is the following:

Theorem 3. For any n , [Algorithm 7](#) $\pm\varepsilon$ -accurately estimates $\text{dist}(f, \mathcal{J}_k)$, for $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$, and makes $2^{k^{1/3} \cdot \text{polylog}(k, 1/\varepsilon)}$ (classical) queries to f .

As with [Theorem 2](#), an equivalent statement of [Theorem 3](#) in the language of tolerant testing is that there is a (classical) algorithm that makes $2^{k^{1/3} \cdot \text{polylog}(k, 1/\varepsilon)}$ black-box queries to $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$ and $(\varepsilon_1, \varepsilon_2)$ -tolerantly tests f for the property of being a k -junta, where $\varepsilon = \varepsilon_2 - \varepsilon_1$.

Building on [CDL⁺24], in [NP24] Nadimpalli and Patel showed that for constant ε , any $\pm\varepsilon$ -accurate algorithm for estimating $\text{dist}(f, \mathcal{J}_k)$ must make $2^{\tilde{\Omega}(\sqrt{k})}$ many queries to f . In contrast, our adaptive algorithm requires only $2^{\tilde{O}(k^{1/3})}$ queries, and so we can conclude that adaptive algorithms provably outperform non-adaptive ones for tolerant junta testing. To the best of our knowledge, this gives the first super-polynomial separation between adaptive and non-adaptive algorithms for a natural tolerant Boolean function property testing problem.

1.2 Discussion

We provide some discussion of our results below.

Remark 4 (On the quantum nature of [Algorithm 2](#)). The only reason that [Algorithm 2](#) requires a quantum black-box oracle to f is because it makes draws from the “spectral sample” of f (see [Definition 33](#)) in lines 2 and 3(b); it is easily verified that given these draws, the rest of the algorithm is entirely classical. It is well known (see e.g. Chapter 5 of [NC16]) that a quantum algorithm with black-box quantum oracle access to a $\{\pm 1\}$ -valued function f can make a draw from the spectral sample of f by performing a single quantum oracle call to f ; this is why we refer to [Algorithm 2](#) as a quantum query algorithm. (In contrast, an argument similar to the classical lower bound for Simon’s problem [Sim97] shows that no classical algorithm can even approximately make draws from the spectral sample of an unknown and arbitrary Boolean function f .) [Theorem 2](#) could alternatively be formulated as a statement about purely classical algorithms which, in addition to having black-box oracle access to f , are equipped with the ability to make draws from the spectral sample.

Remark 5 (Why [Theorem 2](#) given [Theorem 3](#)?). [Theorem 3](#) is strictly stronger than [Theorem 2](#) in two senses: first, it only requires classical oracle access to f rather than quantum query access, and second, it holds for general n -variable functions f for any n whereas [Theorem 2](#) assumes that $k' = \text{poly}(k, 1/\varepsilon)$. Thus the reader may be wondering why we have mentioned [Theorem 2](#) at all; this is done for several reasons. First, the proof of [Theorem 3](#) builds on a number of ideas and ingredients that are introduced in the simpler context of [Theorem 2](#); [Theorem 2](#) should be viewed as an intermediate result on the path to [Theorem 3](#). Second, the connection between agnostic conjunction learning and tolerant junta testing is clearest and most evident in the technically simpler setting of [Theorem 2](#), cf. the discussion and comparison of [Algorithm 1](#) versus [Algorithm 2](#) in [Section 2.2](#). (Intuitively, this is because, as suggested earlier by the BELIEVER, making a draw from the spectral sample distribution of $f : \{\pm 1\}^{k'} \rightarrow \{\pm 1\}$ — which is possible in the quantum but not the classical setting — corresponds to drawing an n -bit string from the unknown and arbitrary distribution over $\{\pm 1\}^n$ in the agnostic learning scenario. In the classical tolerant junta testing setting, we have to do additional work to compensate for not being able to make draws from the spectral sample.)

Remark 6 (On the assumption that the number of variables in [Theorem 2](#) is $\text{poly}(k, 1/\varepsilon)$). The assumption that the number of variables for f is $k' = \text{poly}(k, 1/\varepsilon)$ rather than n is made for the sake of simplicity. As the BELIEVER mentioned, dealing with the general n -variable scenario requires the use of “coordinate oracles” to effectively reduce the number of variables down to $\text{poly}(k, 1/\varepsilon)$ (see [Section 3.10](#)), and integrating these coordinate oracles with quantum queries to efficiently make draws from the spectral sample seems to introduce non-trivial complications. We bypass these complications by simply assuming that $k' = \text{poly}(k, 1/\varepsilon)$. Again, we stress that our main (classical) tolerant property testing result, [Theorem 3](#), is not limited in any way by this assumption, which holds only for the warm-up quantum result given in [Theorem 2](#).

2 Technical Overview

2.1 Overview for agnostically learning conjunctions

Before sketching our approach, we remark that prior work has identified limitations of the [\[KKMS08\]](#) approach to agnostic learning. In fact, more than a decade before [\[KKMS08\]](#), Paturi [\[Pat92\]](#) showed that real polynomials require degree $\Omega(\sqrt{n})$ to pointwise approximate the conjunction $x_1 \wedge \cdots \wedge x_n$ over $\{\pm 1\}^n$, which implies that a direct application of [Theorem 10](#) to agnostically learn conjunctions must use degree $d = \Omega(\sqrt{n})$ and have running time $n^{\Omega(\sqrt{n})}$. A more substantial barrier was identified by Klivans and Sherstov [\[KS07\]](#), who showed that for *any fixed set* of real-valued functions (not just low-degree monomials), if all n -variable conjunctions can be pointwise approximated as a linear combination of those features then the set must have $2^{\Omega(\sqrt{n})}$ elements. This result is interpreted in [\[KS07\]](#) as “show[ing] the fundamental limitations of this approximation-based paradigm”. More recently, Gollakota et al. [\[GKK20\]](#) showed that any *Correlational Statistical Query (CSQ)* algorithm for agnostically learning conjunctions must have complexity $2^{\Omega(\sqrt{n})}$.

We circumvent the limitations defined by [\[KS07\]](#) by using an approximation that is not a pointwise good estimator of the conjunction.² Indeed as we’ll see shortly, we will define a “ball distribution” on which we want a pointwise approximation of the target conjunction, but outside of the support of this distribution, our approximation can (and provably must) make errors.

²We further remark that our algorithm evades the [\[GKK20\]](#) lower bound because it is not a CSQ algorithm. Recall that a CSQ algorithm can only obtain estimates of $\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}}[\mathbf{h}(\mathbf{x}) \neq \mathbf{y}]$; the way that our algorithm uses the event $E_{\bar{a}}$, which is defined using a fixed sample of examples and does not involve the labels at all, does not fit into this framework.

At a high level, our approach proceeds as follows.³ We (i) draw a small number $m = n^{1/3}$ of positively labeled examples $\mathbf{a}^1, \dots, \mathbf{a}^m$ from the distribution \mathcal{D} over $\{\pm 1\}^n \rightarrow \{\pm 1\}$, and use them to adaptively design a new distribution \mathcal{D}' , which we call a “ball distribution” and is supported on a carefully chosen subset of $\{\pm 1\}^n$. The distribution \mathcal{D}' is crafted in such a way that there exists a low-degree polynomial which is a small-squared-error approximator of c^* under \mathcal{D}' , where (ii) c^* is the conjunction which is optimal under \mathcal{D} . **Theorem 10** can then be used to find a high-accuracy hypothesis h' with respect to \mathcal{D}' , and this h' is then used to give a high-accuracy hypothesis h under the original distribution \mathcal{D} .

Remark 7. The above sketch is an oversimplification: in particular, the draw of a small number of examples in the first step alluded to above (iii) only succeeds in obtaining a “useful” set of examples with some non-negligible probability (which is roughly $\varepsilon^{n^{1/3}}$, see **Lemma 29**). The actual algorithm (iv) performs many independent repetitions of the approach sketched in the previous paragraph to obtain multiple hypotheses, and does a final stage of hypothesis testing to select one final hypothesis. In the rest of this overview we ignore this “outer loop,” the analysis of which is standard and straightforward, and give a more detailed overview of the approach sketched in the previous paragraph.

We describe our approach in a bit more detail. Let c^* be the (unknown) “target conjunction” which (v) satisfies $\text{err}_{\mathcal{D}}(c^*) = \text{opt}_{\mathcal{D}}$. (vi) The distribution \mathcal{D}' alluded to above is $\mathcal{D}' := \mathcal{D}|E$ where E is an event over $X = \{\pm 1\}^n$. The event $E = E_{\vec{\mathbf{a}}}$ is based on the initial draw of examples $\vec{\mathbf{a}} = (\mathbf{a}^1, \dots, \mathbf{a}^m)$ from \mathcal{D} alluded to above; we remark that E is “easily recognizable” in the sense that given an input $x \in X$ it is easy to determine whether or not x satisfies E . The distribution $\mathcal{D}' = \mathcal{D}|E$ is referred to as a “ball distribution” because (vii) it is supported on points which (in a certain set of coordinates based on $\mathbf{a}^1, \dots, \mathbf{a}^m$) lie in a certain Hamming ball of radius $n^{2/3}$. Assuming that the initial draw of examples $\vec{\mathbf{a}}$ is such that the event $E = E_{\vec{\mathbf{a}}}$ is “useful” (see **Definition 28**; ensuring that this happens is the purpose of the outer loop), we show that the distribution \mathcal{D}' has the following crucial property:

- The target conjunction c^* has approximate degree at most $O(n^{1/3} \cdot \log(1/\varepsilon))$ over all inputs in the support of \mathcal{D}' (see **Lemma 31**).

(We emphasize that the approximate degree bound given above is crucial to our algorithm’s achieving an overall $2^{n^{1/3} \cdot \text{polylog}(n, 1/\varepsilon)}$ running time.) As alluded to earlier, we use **Theorem 10** on examples drawn from \mathcal{D}' to obtain a hypothesis h' which has $\text{err}_{\mathcal{D}'}(h) \leq \text{err}_{\mathcal{D}'}(c^*) + \varepsilon$. We take the hypothesis h to be

$$h(x) = \begin{cases} h'(x) & \text{if } E(x) \text{ holds} \\ -1 \text{ (i.e. false)} & \text{if } \neg E(x) \text{ holds} \end{cases} \quad (1)$$

(note that h is easy to evaluate since E is “easily recognizable”). It is not too difficult to show that $\text{opt}_{\mathcal{D}}(h) \leq \text{opt}_{\mathcal{D}} + 2\varepsilon$, as desired; this is done in **Lemma 32**.

2.2 Overview of quantum tolerant junta testing

We turn to give an overview of our quantum tolerant k -junta testing algorithm, **Algorithm 2**, highlighting the points of correspondence with **Algorithm 1**. As mentioned earlier, we consider the quantum k -junta $\pm\varepsilon$ -distance-estimation problem over $\{\pm 1\}^{k'}$ where $k' = \text{poly}(k, 1/\varepsilon)$. For simplicity, we mostly ignore the ε -dependence in the following discussion.

³The tags (i), (ii), etc. in what follows are for later reference in **Section 2.2**, when we are explaining points of correspondence between our quantum tolerant junta testing algorithm and our agnostic conjunction learner.

We begin by highlighting some correspondences between various mathematical objects in [Algorithm 1](#) and its analysis, and their counterparts in [Algorithm 2](#) and its analysis. First, we remind the reader that k' (the number of variables in the testing setting) aligns with n , the dimension of the agnostic learning problem. Next, we recall the analogy between the arbitrary distribution \mathcal{D} over labeled examples in $\{\pm 1\}^n \times \{\pm 1\}$ in the agnostic learning setting (or more precisely its marginal over $\{\pm 1\}^n$, the domain of the examples) and the “spectral sample” distribution \mathcal{P}_f over $\{\pm 1\}^{k'}$ that is induced by the Fourier spectrum of $f : \{\pm 1\}^{k'} \rightarrow \{\pm 1\}$. The set of coordinates corresponding to the optimal “target conjunction” c^* in the agnostic setting (recall (ii) in the previous subsection) corresponds to the set of k relevant variables, which we call $U^* \subset [k']$, for the k -junta that is closest to f . Recalling (v) in the previous subsection, the value of $\text{opt}_{\mathcal{D}}$ in the agnostic setting, roughly speaking, corresponds to the *junta correlation* of f , which is closely tied to the amount of Fourier weight that f puts on sets containing only coordinates in U^* (see [Equation \(4\)](#)).

Corresponding to (iv) in the previous section, our quantum junta distance estimation algorithm performs many independent repetitions of a main loop, because each iteration of the loop only has a small but non-negligible success probability of $\varepsilon^{k^{1/3}}$, see [Claim 47](#) (this is analogous to the $\varepsilon^{n^{1/3}}$ success probability of each repetition described in (iv) in the previous subsection). In the discussion below, similar to the previous subsection, we focus on the case of a successful iteration of the loop.

Within each repetition of the loop, the quantum tester makes $k^{1/3}$ draws (the sets $\mathbf{A}_1, \dots, \mathbf{A}_m$ in line 2(a) of [Algorithm 2](#)) from the spectral sample distribution \mathcal{P}_f over $\{\pm 1\}^{k'}$, analogous to the draw of $m = n^{1/3}$ examples $\mathbf{a}^1, \dots, \mathbf{a}^m$ from the distribution \mathcal{D} alluded to in (i) of the previous subsection. (Recall that as mentioned in [Remark 4](#), drawing from the spectral sample distribution is the only quantum aspect of our tester.) A successful iteration of the loop corresponds to having the set $\mathbf{C} := \mathbf{A}_1 \cup \dots \cup \mathbf{A}_{k^{1/3}}$ be such that the spectral sample distribution, restricted to subsets of U^* , only puts a small amount of Fourier weight on sets that contain many (more than $k^{2/3}$) elements outside of \mathbf{C} . This notion of a successful iteration is closely analogous to (iii) the agnostic learning algorithm obtaining a “useful” set of examples, as is evident from the similarity in both the quantitative bounds and proofs of [Lemma 29](#) and [Claim 47](#). The spectral sample distribution \mathcal{P} restricted to subsets S of $[k']$ that have $|S \setminus \mathbf{C}| \leq k^{2/3}$ is analogous to (vii), the “ball distribution” $\mathcal{D}' = \mathcal{D}|_{E_{\bar{a}}}$ of the agnostic conjunction learner.

Remark 8. We quickly remark that in the context of tolerant junta testing, the above approach was essentially considered in [\[ITW21\]](#), albeit with different parameters. In particular, they perform the same procedure using \sqrt{k} draws from the spectral sample to get a $2^{\tilde{O}(\sqrt{k})}$ time quantum tolerant k -junta testing algorithm in this setting. (They also show that there is a classical tester with the same query complexity, which makes draws from a distribution that is related to the spectral sample, cf. the discussion of the “normalized influence distribution” in the next subsection.) However, as we’ll see shortly we will need to bring in several new tools in order to improve the query complexity to $2^{\tilde{O}(k^{1/3})}$.

Remark 9. We comment that while we first developed our agnostic learning algorithm and then ported over the underlying ideas to the tolerant junta testing problem, it seems equally natural, given the results and techniques of [\[ITW21\]](#), to go in the opposite direction and to give a $2^{\tilde{O}(\sqrt{n})}$ -time agnostic conjunction learning algorithm without [\[KKMS08\]](#) inspired by their approach. Alternatively, such a “[[KKMS08](#)]-free” agnostic learning algorithm could be constructed by changing the parameters of our ball distribution and using a brute-force approach over the $2^{\tilde{O}(\sqrt{n})}$ -size support of that distribution. Our $2^{\tilde{O}(n^{1/3})}$ -time agnostic learning algorithm can be viewed as the result of speeding up this brute-force search using [\[KKMS08\]](#).

We return to a discussion of our quantum tolerant tester. While we have established many points

of analogy between the agnostic conjunction learning setting and the tolerant junta testing setting in the discussion so far, from a technical perspective the two proofs now diverge. In particular, we would like to modify f so as to zero out all of its Fourier coefficients S with $|S \setminus \mathbf{C}| > k^{2/3}$ (in analogy (cf. (vi) and (vii)) with restricting our attention to \mathcal{D}' in the agnostic learning setting). While the standard approach to do this would be to suitably apply noise to the function, this will not decrease the weight enough. Indeed, we will require a very sharp attenuation of the Fourier weight for our algorithmic tools, described later, to work; in particular, there must be at most $2^{-\tilde{O}(k^{1/3})}$ Fourier weight on the coefficients S with $|S \setminus \mathbf{C}| \geq k^{2/3}$. To achieve this, we design a novel $\text{SHARPNOISE}_{\overline{\mathbf{C}}}$ operator, which we introduce and analyze in [Section 5.3](#), to define a function $f^{\overline{\mathbf{C}}}$ (see line 2(b) of [Algorithm 2](#)). This operator should be thought of as “zeroing out” all of the Fourier coefficients of f that have more than $k^{2/3}$ elements outside of \mathbf{C} and leaving all other coefficients roughly untouched. (In reality SHARPNOISE only approximately achieves this up to some error which we show is manageable and which we gloss over in the rest of this high-level discussion.) Thus the Fourier coefficients of $f^{\overline{\mathbf{C}}}$ are closely analogous to the “ball distribution” $\mathcal{D}' = \mathcal{D}|E_{\overline{\mathbf{a}}}$ that was mentioned in item (vi) of the previous subsection, and the Hamming ball of radius $n^{2/3}$ described in (vii) of the previous subsection is analogous to the condition that Fourier coefficients that survive into $f^{\overline{\mathbf{C}}}$ have no more than $k^{2/3}$ elements outside of $\overline{\mathbf{T}}$.

Given our attenuated function $f^{\overline{\mathbf{C}}}$, we now wish to estimate its junta correlation. This again marks a point of (now algorithmic) divergence from the agnostic learning problem (although conceptually the high-level mathematical structures remain in lockstep). As was already mentioned, for agnostic conjunction learning the main algorithmic workhorse (see line 2(c) of [Algorithm 1](#)) is the L_1 regression algorithm of [\[KKMS08\]](#). In contrast, for tolerant junta testing the chief algorithmic ingredient that we employ is the “local estimator” notion that was introduced in the recent work of Nadimpalli and Patel [\[NP24\]](#). Our quantum algorithm relies on a novel analysis of the [\[NP24\]](#) local estimator for the mean of a function that has small Fourier weight at high levels (see [Lemma 35](#)). We use this local mean estimator to efficiently estimate the correlation between f and the best junta over a subset U of variables, for various different sets U (see [Lemma 43](#)); crucially, using local estimators allows us to make relatively few (only $\approx \exp(k^{1/3})$) queries even though the number of sets for which we do this estimation may be as large as $\approx \exp(k)$ (see the discussion immediately preceding [Lemma 39](#)).

Finally, we remark there is some delicate and careful technical work required to combine the technology of local estimators with the SHARPNOISE operator; see [Section 5.2](#) through [Section 5.5](#).

2.3 Overview of classical tolerant junta testing

Finally, we discuss our classical tolerant tester. For simplicity, we will assume that we are working in a similar setting to the quantum algorithm. Namely, we will think of ε as a small constant, say 0.01, and mostly ignore the ε parameter in our discussions. Moreover we will assume that the function f is over $k' = 2k$ variables rather than n to avoid discussion of coordinate oracles. We also again denote the best set of k junta coordinates by U^* .

At a very high level, we would like to follow the same path as our quantum testing algorithm. Fortunately, many of the tools we developed there, such as SHARPNOISE and our local estimators, are already immediately implementable by a classical algorithm. That said, the central piece that will not carry over is our ability to generate a set $\mathbf{C} \subseteq U^*$ such that the spectral sample restricted to subsets of U^* only puts a small amount of Fourier weight on sets that contain more than $k^{2/3}$ many elements outside of \mathbf{C} . This brings us to our central technical challenge in this section: Assuming

that

$$\sum_{S \subseteq U^*: |S| \geq k^{2/3}} \widehat{f}^2(S) \text{ is large (say, at least } \Omega(1)),$$

we are tasked with finding a set of coordinates $C \subseteq U^*$ such that

$$\sum_{S \subseteq U^*: |S \setminus C| \geq k^{2/3}} \widehat{f}^2(S) \text{ is small (say, at most some } o(1)).$$

We start by noting that there is no hope of mimicking the quantum algorithm directly. Indeed, as alluded to earlier, any classical algorithm must make $2^{\Omega(k)}$ queries to f to make a draw from a distribution with small total variation distance from the spectral sample. Instead, we design an entirely new approach, which we describe below.

At a very high level, we give a procedure, called REFINE-COORDINATES (Algorithm 4), that takes as input a set of coordinates $C \subseteq [k']$ and outputs a collection of $2^{k^{1/3} \text{polylog}(k)}$ sets $C' \subseteq C$ such that for at least one set C^* that we output,

- (a) C^* has half as many irrelevant variables as C , i.e. $|C^* \cap \overline{U^*}| \leq \frac{1}{2}|C \cap \overline{U^*}|$; and moreover,
- (b) Only a very small amount of the Fourier mass of f is on terms $S \subseteq U^*$ with $|S \setminus C^*| \geq k^{2/3}$, i.e.

$$\sum_{S \subseteq U^*: |S \setminus C^*| \geq k^{2/3}} \widehat{f}^2(S) \leq \frac{1}{k}. \quad (2)$$

REFINE-COORDINATES is useful for us for the following reason: Given such a procedure, we can run it on $C := [k']$ to construct a family of sets C' . For each set C' in this family, we can then run REFINE-COORDINATES(f, C') to get a new family. After doing this $\log(k')$ times, we will eventually end up with a family of $2^{k^{1/3} \text{polylog}(k)}$ sets, and one of the sets (call it C^{**}) will satisfy

$$\sum_{S \subseteq U^*: |S \setminus C^{**}| \geq k^{2/3}} \widehat{f}^2(S) \leq o(1)$$

and $C^{**} \subseteq U^*$ as desired. Roughly speaking, we can then conclude by applying the SHARPNOISE operator and our local estimators as in the quantum tester.

With this performance goal for REFINE-COORDINATES in mind, we turn to a discussion of how to achieve it. We do this by using *normalized influences*, akin to [ITW21]. These are defined in detail in Section 3.8. For us, the crucial property, as we will see later, is that the normalized influence of a set U of size κ , denoted $\mathbf{NInf}_U[f]$, is defined in such a way that drawing a set \mathbf{T} of size κ at random so that $\Pr[\mathbf{T} = U]$ is proportional to $\mathbf{NInf}_U[f]$, corresponds to the following process:

- (\star) First draw a set \mathbf{S} from the spectral sample of f , conditioned on $|\mathbf{S}| \geq \kappa$, and then choose \mathbf{T} as a uniform random subset of \mathbf{S} of size κ .

To actually get access to these normalized influences, we give a new algorithm, Algorithm 3, that approximates $\mathbf{NInf}_U[f]$ to arbitrary accuracy. Using this algorithm, we can sample from a distribution with total variation distance at most η to the true distribution of level- κ normalized influences in time $\text{poly}(\exp(\tilde{O}(\kappa)), \eta^{-1})$.⁴ Notably, this improves on a prior result of [ITW21], which

⁴We believe that this is near optimal. In particular, we believe that drawing from the level- κ normalized influence distribution requires $2^{\Omega(\kappa)}$ black-box queries to f , as draws from this distribution seem useful for solving Simon's problem for a random function on 2κ variables.

could only compute $\mathbf{NInf}_U[f]$ up to a constant multiplicative factor. Given the above algorithmic result, we will assume throughout the remainder of our discussion that we can in fact draw samples from the level- $k^{1/3}\text{polylog}(k)$ normalized influence distribution.

With this tool in hand, let us discuss how we can give a procedure `REFINE-COORDINATES` with the desired properties. As a starting point for the algorithm, we guess a random set $\mathbf{I} \subseteq C$ of $k^{1/3}\text{polylog}(k)$ coordinates. The point of doing this is to collect a set of $k^{1/3}\text{polylog}(k)$ coordinates which are hopefully irrelevant, i.e. lie outside of U^* , such that the function resulting from averaging out these coordinates, which we denote $f_{\text{ave}}^{\mathbf{I}}$, satisfies

$$\sum_{S \subseteq [k'] : |S \setminus U^*| \geq k^{2/3}/\text{polylog}(k)} \widehat{f_{\text{ave}}^{\mathbf{I}}}^2(S) \leq \frac{1}{k^{10}}. \quad (3)$$

In particular, note that the above Fourier attenuation is precisely what we would expect to get if \mathbf{I} were drawn uniformly at random from $C \cap \overline{U^*}$. Because $\mathbf{I} \subseteq C \cap \overline{U^*}$ with probability roughly $2^{-k^{1/3}\text{polylog}(k)}$, we expect to sample a “good” \mathbf{I} after randomly sampling $2^{k^{1/3}\text{polylog}(k)}$ sets \mathbf{I} from C uniformly at random.

Assuming that we sample an \mathbf{I} that satisfies [Equation \(3\)](#), we now turn to describe how to use this to construct a set C' . (Note that we will construct one set C' per set \mathbf{I} , resulting in a total of $2^{k^{1/3}\text{polylog}(k)}$ sets C' .) Given a candidate \mathbf{I} , we build C' iteratively, starting from $C' = \emptyset$, roughly as follows: (1) If $f_{\text{ave}}^{\mathbf{I}}$ has at most $\frac{1}{\text{poly}(k)}$ Fourier mass on terms S with $|S \setminus C'| \geq k^{2/3}$, output C' . Otherwise, (2) draw a random set \mathbf{T} from the normalized influences of $f_{\text{ave}}^{\mathbf{I}}$ conditioned on the set \mathbf{S} from (\star) satisfying $|\mathbf{S} \setminus C'| \geq k^{2/3}$. (These two steps can roughly be performed by measuring the variance and sampling from the normalized influences of a suitable application of `SHARPNOISE` to $f_{\text{ave}}^{\mathbf{I}}$.) Afterwards, update $C' \leftarrow C' \cup (\mathbf{T} \cap C)$ and repeat.

We note that condition (1) of the algorithm sketched above will roughly ensure that [Equation \(2\)](#) holds, giving (b). Moreover, it turns out that through a careful martingale-style analysis of the above process, we can show that we will not add too many irrelevant variables to C' , as required by (a). To give a flavor of why one should expect this, we'll give some intuition for why C' ought to contain more elements from $C \cap U^*$ than from $C \cap \overline{U^*}$. (This is a significantly weaker statement than what we will truly wish to show, but it illustrates the intuition well.) Observe that because \mathbf{I} satisfies [Equation \(3\)](#), we expect \mathbf{S} , the spectral sample draw from which \mathbf{T} is drawn in the (\star) view of the normalized-influence distribution, to contain at most $k^{2/3}/\text{polylog}(k)$ irrelevant coordinates. On the other hand, by construction we have that the set \mathbf{S} corresponding to the spectral sample has at least $k^{2/3}$ new elements outside of C' . So we can conclude $|(\mathbf{S} \setminus C') \cap U^*|$ is much larger than $|(\mathbf{S} \setminus C') \cap \overline{U^*}|$. Since \mathbf{T} is a uniform random subset of \mathbf{S} , we then expect that in each iteration, $\mathbf{T} \setminus C'$ also contains more variables from U^* than from $\overline{U^*}$. By a concentration argument, it should then follow that in every round we add more relevant variables than irrelevant ones. (We briefly remark that this final concentration step can be shown to break when we need to actually prove the stronger statement that $|C' \cap U^*| \leq \frac{1}{2}|C \cap U^*|$, requiring us to give a more careful analysis.)

This concludes our sketch of the ideas behind our `REFINE-COORDINATES` procedure, and thus our classical tolerant junta tester.

3 Preliminaries

3.1 Preliminaries for agnostically learning conjunctions

Recall that in the problem of agnostic learning of conjunctions, there is an unknown and arbitrary distribution \mathcal{D} over pairs $\{\pm 1\}^n \times \{\pm 1\}$. For a function $g : \{\pm 1\}^n \rightarrow \{\pm 1\}$ we define

$$\text{err}_{\mathcal{D}}(g) := \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [g(\mathbf{x}) \neq \mathbf{y}] \quad \text{and} \quad \text{opt}_{\mathcal{D}} := \min_{c \in \mathcal{C}} \text{err}_{\mathcal{D}}(c),$$

where \mathcal{C} denotes the class of all conjunctions (where +1 corresponds to True and -1 corresponds to False). An agnostic learning algorithm is an algorithm which, with probability at least $2/3$, constructs a hypothesis $h : \{\pm 1\}^n \rightarrow \{\pm 1\}$ that has $\text{err}_{\mathcal{D}}(h) \leq \text{opt}_{\mathcal{D}} + \varepsilon$.

Notation. Throughout the conjunction learning part (Section 4) we write X to denote the domain $\{\pm 1\}^n$ for convenience and \mathcal{C} to denote the class of all conjunctions over $\{\pm 1\}^n$. We write (a, b) to denote a tuple of labeled examples $((a^1, b^1), \dots, (a^m, b^m)) \in (X \times \{\pm 1\})^m$, and write \vec{a} to denote the corresponding tuple $(a^1, \dots, a^m) \in X^m$.

A generic algorithm for agnostic learning. We make essential use of the following result from [KKMS08]. It says that if every $c \in \mathcal{C}$ can be approximated by a low-degree polynomial with small squared error, then the L_1 polynomial regression algorithm is an efficient agnostic learner for \mathcal{C} :

Theorem 10 (Theorem 5 of [KKMS08]). Fix a distribution \mathcal{D} over $X \times \{\pm 1\}$. Suppose that for each $c \in \mathcal{C}$, there exists a degree-at-most- d polynomial p such that

$$\mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [(p(\mathbf{x}) - c(\mathbf{x}))^2] \leq \varepsilon^2.$$

Then the “degree- d L_1 polynomial regression” algorithm runs in time $\text{poly}(n^d, 1/\varepsilon, \log(1/\delta))$ using examples drawn i.i.d. from \mathcal{D} , and with probability at least $1 - \delta$ outputs a hypothesis h that has $\text{err}_{\mathcal{D}}(h) \leq \text{opt}_{\mathcal{D}} + \varepsilon$.

3.2 Fourier basics

Given two functions $f, g : \{\pm 1\}^n \rightarrow \mathbb{R}$, we write

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{\pm 1\}^n} f(x)g(x) = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^n} [f(\mathbf{x})g(\mathbf{x})] \quad \text{and} \quad \|f\|_2 = \sqrt{\langle f, f \rangle}.$$

Given $S \subseteq [n]$, let $\chi_S(x) : \{\pm 1\}^n \rightarrow \{\pm 1\}$ be the function $\chi_S(x) = \prod_{i \in S} x_i$, and let $\widehat{f}(S) := \langle f, \chi_S \rangle$ be the Fourier coefficients of f . Then we have

$$f = \sum_{S \subseteq [n]} \widehat{f}(S) \chi_S \quad \text{and} \quad \|f\|_2^2 = \sum_{S \subseteq [n]} \widehat{f}(S)^2 \quad (\text{by Parseval's identity}).$$

For $f : \{\pm 1\}^n \rightarrow \mathbb{R}$ and an integer $L \geq 0$, we write $\mathbf{W}^{\geq L}[f]$ to denote $\sum_{|S| \geq L} \widehat{f}(S)^2$, the Fourier weight of f at levels L and above.

3.3 Preliminaries for tolerantly testing juntas

We write \mathcal{J}_k to denote the class of all Boolean-valued k -juntas, i.e. all functions $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$ such that there exists a function $g : \{\pm 1\}^k \rightarrow \{\pm 1\}$ and indices $i_1 < \dots < i_k$ such that $f(x_1, \dots, x_n) = g(x_{i_1}, \dots, x_{i_k})$ for all $x \in \{\pm 1\}^n$.

3.4 Junta Correlation

We will primarily be working with junta correlation in this note, as our functions will not be boolean valued in general. Recall that the k -junta correlation of f is defined as

$$\text{corr}(f, \mathcal{J}_k) := \max_{T \in \binom{[n]}{k}} \text{corr}(f, \mathcal{J}_T),$$

where for any $T \subseteq [n]$, we have

$$\text{corr}(f, \mathcal{J}_T) := \max_{g \in \mathcal{J}_T} \mathbf{E}[f(\mathbf{x})g(\mathbf{x})].$$

We recall from [ITW21] (Section 2.2, specifically Claim 2.7) that

$$\text{corr}(f, \mathcal{J}_T) = \mathbf{E}[\text{sign}(f_{\text{ave}}^{\overline{T}}(\mathbf{x}))f(\mathbf{x})], \text{ where } f_{\text{ave}}^{\overline{T}}(x) = \mathbf{E}_{\mathbf{y} \sim \{\pm 1\}^n} [f(\mathbf{y}) \mid \mathbf{y}_T = x] = \sum_{S: S \subseteq T} \hat{f}(S) \chi_S(x). \quad (4)$$

We further recall from [ITW21] that for any subset $T \subseteq [n]$, we have

$$\text{corr}(f, \mathcal{J}_T) = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^n} \left[\left| f_{\text{ave}}^{\overline{T}}(x) \right| \right] = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^n} \left[\left| \sum_{A \subseteq T} \hat{f}(A) \chi_A(\mathbf{x}) \right| \right]. \quad (5)$$

Recall that for ± 1 -valued functions f, g we have that

$$\mathbf{E}[f(\mathbf{x})g(\mathbf{x})] = 1 - 2 \Pr[f(\mathbf{x}) \neq g(\mathbf{x})].$$

So it follows that estimating junta correlation to accuracy ε will also give us an estimate for $\text{dist}(f, \mathcal{J}_k)$ if f is boolean valued, where

$$\text{dist}(f, \mathcal{J}_k) := \min_{g \in \mathcal{J}_k} \Pr[f(\mathbf{x}) \neq g(\mathbf{x})]$$

is the distance from f to the nearest k -junta.

We also record the fact that changing a function by a small amount (in terms of expected squared error, i.e. changing it in the Fourier basis) does not significantly change its junta correlation:

Lemma 11. Suppose that $f, h : \{\pm 1\}^n \rightarrow \mathbb{R}$ are such that $\|f - h\|_2 \leq \varepsilon$. Then

$$\left| \max_{g \in \mathcal{J}_k} \mathbf{E}[f(\mathbf{x})g(\mathbf{x})] - \max_{g \in \mathcal{J}_k} \mathbf{E}[h(\mathbf{x})g(\mathbf{x})] \right| \leq \varepsilon.$$

Proof. For any $g \in \mathcal{J}_k$ we have that

$$\left| \mathbf{E} \left[(f(\mathbf{x}) - h(\mathbf{x})) g(\mathbf{x}) \right] \right| \leq \|f - h\|_2 \cdot \|g\|_2 \leq \varepsilon$$

by Cauchy-Schwartz. □

Tolerant testing For $0 < \varepsilon_1 < \varepsilon_2$, an $(\varepsilon_1, \varepsilon_2)$ -tolerant k -junta tester is an algorithm which makes black-box queries to an unknown and arbitrary $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$ and distinguishes between the two cases that $\text{dist}(f, \mathcal{J}_k) \leq \varepsilon_1$ versus $\text{dist}(f, \mathcal{J}_k) \geq \varepsilon_2$. As mentioned earlier, the problem of estimating $\text{dist}(f, \mathcal{J}_k)$ up to additive error $\pm \varepsilon$ equivalent to the problem of $(\varepsilon_1, \varepsilon_2 = \varepsilon_1 + \varepsilon)$ -tolerant testing the class of k -juntas.

3.5 Bernoulli noise

Given a set $V \subseteq [k']$, we recall that the *Bernoulli noise operator at noise rate ρ over coordinates in V* , denoted T_ρ^V , acts on $f : \{\pm 1\}^{k'} \rightarrow \mathbb{R}$ via

$$T_\rho^V f(x) = \mathbf{E}_{\mathbf{y} \sim N_\rho^V(x)} [f(\mathbf{y})]. \quad (6)$$

Here $N_\rho^V(x)$ is the distribution over $\mathbf{y} \sim \{\pm 1\}^{k'}$ such that each coordinate \mathbf{y}_i of \mathbf{y} is independently distributed as follows:

- If $i \notin V$ then $\mathbf{y}_i = x_i$;
- If $i \in V$ then \mathbf{y}_i is set to x_i with probability ρ and is set to be uniform random over $\{\pm 1\}$ with the remaining $1 - \rho$ probability.

Equivalently, we have $T_\rho^V \chi_S(x) = \rho^{|S \cap V|} \chi_S(x)$. When V is the entire set $[k']$ of all variables, we omit V from the notation and simply write T_ρ and N_ρ .

3.6 Randomized algorithms and estimating function values, Fourier coefficients, sums of squares of Fourier coefficients, etc.

We will frequently work with functions that have real-valued outputs. Towards that end, we will use the following definition of a randomized algorithm computing a bounded function:

Definition 12 (*M -Bounded Randomized Algorithm for an $[-M, M]$ -Bounded Function*). Let $g : \{\pm 1\}^n \rightarrow [-M, M]$. We say that algorithm \mathcal{A} is a *M -bounded randomized algorithm for g* if on any fixed input x , algorithm \mathcal{A} outputs a random value $\mathbf{y} \in [-M, M]$ with $\mathbf{E}[\mathbf{y}] = g(x)$.

We can easily obtain a high-accuracy, high-confidence estimate of the evaluation of a $[-M, M]$ -bounded function or of its Fourier coefficient:

Claim 13 (*High-Accuracy, High-Confidence Estimation of an $[-M, M]$ -Bounded Function*). Let $g : \{\pm 1\}^n \rightarrow [-M, M]$ and let \mathcal{A} be an M -bounded randomized algorithm for g . Then there is an algorithm $\mathcal{A}_{\varepsilon, \delta}$ which, on input $x \in \{\pm 1\}^n$, makes $O((M/\varepsilon)^2 \cdot \log(1/\delta))$ calls to \mathcal{A} on input x , estimates $g(x)$ up to additive error $\pm \varepsilon$ with probability at least $1 - \delta$, and always outputs a value in $[-M, M]$.

Proof. The algorithm simply calls \mathcal{A} repeatedly on x and takes the empirical average of those calls. The analysis follows from a standard Hoeffding bound. \square

We will also need the following result which says that we can use a M -bounded randomized algorithm for g to estimate $\mathbf{E}_{\mathbf{x}}[g^2(\mathbf{x})]$:

Lemma 14 (*Estimating the L_2 Norm*). Let $\varepsilon, \delta \in [0, 1/4]$ and suppose that there exists a M -bounded randomized algorithm \mathcal{A} for computing $g : \{\pm 1\}^n \rightarrow [-M, M]$. It then follows that there is an algorithm that estimates the value of $\mathbf{E}[g(\mathbf{x})^2]$ to additive error $\pm \varepsilon$ with probability at least $1 - \delta$ and makes $O\left(\frac{M^8}{\varepsilon^4} \log^2(1/\delta)\right)$ queries to \mathcal{A} . Moreover, the value of the estimate always lies in $[0, M^2]$.

Proof. Let $N := 1000 \frac{M^4}{\varepsilon^2} \log(1/\delta)$ and fix some $x \in \{\pm 1\}^n$. Let $\mathbf{z}_1(x), \mathbf{z}_2(x), \dots, \mathbf{z}_N(x)$ denote the output of N independent runs of $\mathcal{A}(x)$. Let $\mathbf{d}_i(x) = \mathbf{z}_i(x) - g(x)$ and observe $\mathbf{E}_{\mathcal{A}}[\mathbf{d}_i(x)] = 0$. Now consider the random variables

$$\mathbf{Z}(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{z}_i(x)$$

and

$$\mathbf{D}(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{d}_i(x).$$

Since the \mathbf{d}_i 's are independent, have mean 0, and lie in $[-2M, 2M]$, we have that

$$\mathbf{E}_{\mathcal{A}} \left[(\mathbf{D}(x))^2 \right] = \frac{1}{N^2} \sum_{i=1}^N \mathbf{E} \left[(\mathbf{d}_i(x))^2 \right] \leq \frac{4M^2}{N} \leq \frac{\varepsilon}{2}.$$

On the other hand, we have

$$\mathbf{E}_{\mathcal{A}} \left[(\mathbf{Z}(x))^2 \right] = \mathbf{E}_{\mathcal{A}} \left[(g(x) + \mathbf{D}(x))^2 \right] = g(x)^2 + \mathbf{E}_{\mathcal{A}} \left[(\mathbf{D}(x))^2 \right],$$

which implies

$$g(x)^2 \leq \mathbf{E}_{\mathcal{A}} \left[(\mathbf{Z}(x))^2 \right] \leq g(x)^2 + \frac{\varepsilon}{2}.$$

Given the above, our estimator works as follows: it samples N independent uniform values $\mathbf{x}^1, \dots, \mathbf{x}^N \sim \{\pm 1\}^n$ and outputs

$$\frac{1}{N} \sum_{i=1}^N (\mathbf{Z}(\mathbf{x}^i))^2$$

as the estimate for $\mathbf{E}[g^2(\mathbf{x})]$. Note that estimate always lies in $[0, M^2]$. By the previous computation, we have that

$$\left| \mathbf{E}_{\mathbf{x}^i, \mathcal{A}} \left[\frac{1}{N} \sum_{i=1}^N (\mathbf{Z}(\mathbf{x}^i))^2 \right] - \mathbf{E}_{\mathbf{x}} \left[(g(\mathbf{x}))^2 \right] \right| \leq \frac{\varepsilon}{2}.$$

Since $\mathbf{Z}(x)^2$ always lies in $[0, M^2]$, by a Hoeffding bound we then have that

$$\Pr \left[\left| \frac{1}{N} \sum_{i=1}^N (\mathbf{Z}_i(\mathbf{x}^i))^2 - \mathbf{E}_{\mathbf{x}} \left[(g(\mathbf{x}))^2 \right] \right| \geq \varepsilon \right] \leq \exp \left(\frac{-N\varepsilon^2}{2 \cdot M^4} \right) \leq \delta$$

as desired. We can easily see that computing $\frac{1}{N} \sum_{i=1}^N (\mathbf{Z}(\mathbf{x}^i))^2$ requires at most N^2 queries to \mathcal{A} , yielding the claimed query bound. \square

We will also frequently need to apply noise operators, so it will be convenient to have the following lemma.

Lemma 15. Suppose that \mathcal{A} is an M -bounded randomized algorithm for $g : \{\pm 1\}^n \rightarrow [-M, M]$. Then for any $\rho \in [0, 1]$ and $V \subseteq [n]$, there exists a M -bounded randomized algorithm \mathcal{A}' for $\mathsf{T}_{\rho}^V g$ making a single query to \mathcal{A} .

Proof. Given $x \in \{\pm 1\}^n$, the algorithm \mathcal{A}' simply samples a string $\mathbf{y} \sim N_\rho^V(x)$ and then outputs $\mathcal{A}(\mathbf{y})$. This is clearly M -bounded and satisfies

$$\mathbf{E}_{\mathcal{A}'}[\mathcal{A}'(x)] = \mathbf{E}_{\mathbf{y}, \mathcal{A}}[\mathcal{A}(\mathbf{y})] = \mathbf{E}_{\mathbf{y}}[g(\mathbf{y})] = T_\rho^V g(x). \quad \square$$

It will also be convenient to have a lemma about randomized algorithms for functions obtained by averaging out over a subset of coordinates:

Lemma 16. Suppose that \mathcal{A} is a M -bounded randomized algorithm for $g : \{\pm 1\}^n \rightarrow [-M, M]$. Then for any $V \subseteq [n]$, there exists a M -bounded randomized algorithm, making a single call to \mathcal{A} , for the function $h(x) = \mathbf{E}_{\mathbf{y} \sim \{\pm 1\}^n} \left[g(\mathbf{y}) \middle| y|_{\overline{V}} = x \right]$.

Proof. Consider the algorithm \mathcal{A}' that given x samples a random $\mathbf{z} \in \{\pm 1\}^V$ and computes $\mathcal{A}(x|_{\overline{V}} \sqcup \mathbf{z})$. We can then observe

$$\mathbf{E}_{\mathcal{A}'}[\mathcal{A}'(x)] = \mathbf{E}_{\mathbf{z}, \mathcal{A}}[\mathcal{A}(x|_{\overline{V}} \sqcup \mathbf{z})] = \mathbf{E}_{\mathbf{z}}[g(x|_{\overline{V}} \sqcup \mathbf{z})] = \mathbf{E}_{\mathbf{y} \sim \{\pm 1\}^n} \left[g(\mathbf{y}) \middle| y|_{\overline{V}} = x \right]$$

as desired. Clearly, \mathcal{A}' is M bounded and makes a single query to \mathcal{A} . \square

3.7 Flat Polynomials

Underlying our results are constructions of “flat” polynomials, originally developed by Linial and Nisan [LN90] and Kahn, Linial, and Samorodnitsky [KLS96] to prove *approximate inclusion-exclusion* bounds. We first start with the following construction due to Kahn et al. [KLS96]:

Lemma 17 (Theorem 2.1 of [KLS96]). Fix integers r, N with $2\sqrt{N} \leq r \leq N$. Then there exists a polynomial $p : \mathbb{R} \rightarrow \mathbb{R}$ of degree at most r with the following properties:

$$(i) \ p(0) = 0 \quad \text{and} \quad (ii) \ \max_{i \in [N]} |p(i) - 1| \leq 2 \exp \left(-\Omega \left(\frac{r^2}{N \log N} \right) \right).$$

We will also need the construction of Linial and Nisan [LN90] to obtain improved bounds when $\varepsilon_2 \gg \varepsilon_1$.

Lemma 18 (Theorem 1 of [LN90]). Fix integers r, N with $r \leq 2\sqrt{N}$. Then there exists a polynomial $p : \mathbb{R} \rightarrow \mathbb{R}$ of degree at most r with the following properties:

$$(i) \ p(0) = 0 \quad \text{and} \quad (ii) \ \max_{i \in [N]} |p(i) - 1| \leq \left(1 + \Theta \left(\frac{r^2}{N} \right) \right)^{-1}.$$

It will be important for us that these polynomials do not blow up too much for values greater than N , and also that they do not have very large coefficients. Towards this, we will use the following results:

Lemma 19 (Lemma 11 of [NP24]). Let r, N be integers and suppose that $p : \mathbb{R} \rightarrow \mathbb{R}$ is a polynomial of degree r such that $|p(i)| \leq 2$ for all $i = 1, \dots, N$ with $p(0) = 0$. For any $\ell \geq N$, $p(\ell) \leq 4\ell^r$.

Lemma 20 (Lemma 12 of [NP24]). Let $r \leq N$ be integers and suppose that $p : \mathbb{R} \rightarrow \mathbb{R}$ is a polynomial of degree r such that $|p(i)| \leq 2$ for all $i = 1, \dots, N$ with $p(0) = 0$. Moreover, set $\alpha_i^{r, N}$ such that

$$p(x) = \sum_{i=1}^r \alpha_i^{r, N} \binom{x}{i} \quad \text{where} \quad \binom{x}{i} := \frac{x(x-1)\dots(x-i+1)}{i!}.$$

Then we have $|\alpha_i^{r, N}| \leq 2r^r$.

3.8 Normalized Influences

Definition 21. For a given subset $U \subseteq [n]$, we define its *normalized influence* as follows:

$$\mathbf{NInf}_U[f] := \sum_{S: U \subseteq S} \frac{\widehat{f}(S)^2}{\binom{|S|}{|U|}}.$$

Note that the sum $\sum_{|U|=m} \mathbf{NInf}_U[f]$ of all normalized influences of size m is precisely $W^{\geq m}[f]$. We remark that normalized influences correspond to drawing a set S from the spectral sample and sampling $|U|$ elements. (In more detail, given a particular value of m , if we were to draw a random set U of size m by choosing each $|U| = m$ with probability $\frac{\mathbf{NInf}_U[f]}{\sum_{|U|=m} \mathbf{NInf}_U[f]}$, we could alternately describe this distribution of U as follows: Repeatedly sample a set S from the spectral sample until a set of size $\geq m$ is obtained, and then sample a uniform random subset of m of its elements and take that subset to be U .)

3.9 Local estimators

We recall the notion of a *local estimator* from [NP24].

Definition 22 (Statistic). A *statistic* is a function S that maps a function $f : \{\pm 1\}^n \rightarrow \mathbb{R}$ to a real number in \mathbb{R} .

We recall that the Hamming ball $B(x, r)$ of radius r around a point $x \in \{\pm 1\}^n$ is the set of all $y \in \{\pm 1\}^n$ such that $\text{dist}(x, y) \leq r$, where $\text{dist}(x, y)$ is the Hamming distance between x and y .

Definition 23 (r -local estimator). Given $f : \{\pm 1\}^n \rightarrow \mathbb{R}$, $x \in \{\pm 1\}^n$, and a positive integer $r \leq n$, an r -local estimator \mathcal{E} takes as input the values of f restricted to the Hamming ball $B(x, r)$ (we denote this by $f|_{B(x, r)}$) and outputs a real number.

For $\tau \geq 0$, an r -local estimator \mathcal{E} is said to τ -approximate a statistic S if

$$\left| \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^n} \left[\mathcal{E}(f|_{B(\mathbf{x}, r)}) \right] - S(f) \right| \leq \tau.$$

Finally, we say that the estimator \mathcal{E} is κ -bounded if its range is $[-\kappa, \kappa]$.

3.10 Coordinate oracles and approximate versus exact computation

A central tool in the tolerant testing of k -juntas is the notion of approximate coordinate oracles developed by [DMN19]. Essentially, these let us reduce the number of coordinate from n down to $\text{poly}(k, 1/\varepsilon)$.

In particular, we will need the following Corollary 4.7 from [ITW21], which builds on [DMN19].

Theorem 24 (Corollary 4.7 of [ITW21]). With $\text{poly}(k, \varepsilon^{-1}, \log(\delta^{-1}))$ queries to f , we can gain access to a set of approximate oracles $\mathcal{O} = \{\mathcal{O}_1, \dots, \mathcal{O}_{k'}\}$ for a set \mathcal{S} of k' coordinates from $[n]$. Moreover, these coordinates satisfy the following properties:

1. For every coordinate, $i \in \mathcal{S}$, there exists a $g \in \mathcal{O}$ such that g is 0.1 close to $\text{Dict}_i(x) = x_i$ with probability at least $1 - \delta$. (Hence we refer to \mathcal{O} as a *set of approximate coordinate oracles*.)
2. $\min_{S \subseteq [n]: |S| \leq k} \text{dist}(f, \mathcal{J}_S) - \min_{S \subseteq \mathcal{S}: |S| \leq k} \text{dist}(f, \mathcal{J}_S) \leq \varepsilon$.
3. $k' = |\mathcal{S}| \leq \text{poly}(k, \varepsilon^{-1}, \log(1/\delta))$.

4. For an algorithm A that uses at most q queries to \mathcal{D} , we can assume that we have perfect oracle access to the dictator correspond to each coordinate oracle up to an additive loss of δ in the confidence and a multiplicative overhead of $\text{poly}(\log(q/\delta))$ in query complexity. (We refer to this set \mathcal{O}' of oracles for the exact dictator functions corresponding to the elements of \mathcal{O} as a *set of coordinate oracles*.)

Given point (4), we will assume throughout the paper that we indeed have exact access to the coordinate oracles i.e. each coordinate oracle \mathcal{O}_i exactly corresponds to the dictator function of its corresponding coordinate. Now it's important to note that while each \mathcal{O}_i corresponds to a single coordinate, we do not know which of the n coordinates this is. (Notably, that would information-theoretically require $\Omega(\log(n))$ queries, which is too large when k is small.) As such we only have *implicit* access to these coordinates (cf. [Ser10]). Nonetheless, [ITW21] are also able to show that we can query these coordinates to get a randomized algorithm for $f_{\text{ave}}^{\bar{S}}$:

Theorem 25 (Theorem 4.10 of [ITW21]). Let $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$ and $\mathcal{O}' = \{\mathcal{O}'_1, \dots, \mathcal{O}'_{k'}\}$ be a set of k' coordinate oracles corresponding to a set $\mathcal{S} \subset [n], |\mathcal{S}| = k'$. Let g be a function from $\{\pm 1\}^{k'} \rightarrow \{\pm 1\}$ defined by

$$g(y) := \mathbf{E}_{\mathbf{x}} \left[f(\mathbf{x}) \mid \mathcal{O}'_1(\mathbf{x}) = y_1, \mathcal{O}'_2(\mathbf{x}) = y_2, \dots, \mathcal{O}'_{k'}(\mathbf{x}) = y_{k'} \right] = f_{\text{ave}}^{\bar{S}}.$$

Then there exists a 1-bounded randomized algorithm for g that makes $\text{poly}(k')$ queries to f in expectation.

4 Agnostically learning conjunctions

In this section we present our agnostic learning algorithm for conjunctions and prove [Theorem 1](#).

4.1 Setup for the learning algorithm

Before explaining the algorithm we establish some helpful notation and terminology. (We view $+1$ as corresponding to True and -1 as corresponding to False, and recall that $X = \{\pm 1\}^n$.)

Definition 26. Given a tuple of points $\vec{a} = (a^1, \dots, a^m) \in X^m$, we write $\text{NON-CONST}(\vec{a}) \subseteq [n]$ to denote the set of coordinates that are not constant across the strings in \vec{a} . Formally,

$$\text{NON-CONST}(\vec{a}) := \{i \in [n] : a_i^1 \neq a_i^j \text{ for some } j \in [m]\}$$

and $\text{CONST}(\vec{a}) = [n] \setminus \text{NON-CONST}(\vec{a})$. So all strings in \vec{a} agree on coordinates in $\text{CONST}(\vec{a})$.

Given any $x \in \{\pm 1\}^n$ and $\vec{a} = (a^1, \dots, a^m)$, we use $\text{dist}(x, \vec{a})$ to denote the Hamming distance between x and a^1 over coordinates in $\text{CONST}(\vec{a})$, i.e., the number of $i \in \text{CONST}(\vec{a})$ with $x_i \neq a_i^1$.

Definition 27. Given a tuple of points $\vec{a} = (a^1, \dots, a^m) \in X^m$, the event $E_{\vec{a}} \subseteq X$ is defined to be the set of all $x \in X$ such that $\text{dist}(x, \vec{a}) \leq n^{2/3}$. Formally,

$$E_{\vec{a}} := \left\{ x \in X : \text{number of } i \in \text{CONST}(\vec{a}) \text{ with } x_i \neq a_i^1 \leq n^{2/3} \right\}.$$

Given a distribution \mathcal{D} over $X \times \{\pm 1\}$, we write $\mathcal{D}|E_{\vec{a}}$ to denote the distribution of $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$ conditioned on \mathbf{x} belonging to $E_{\vec{a}}$, and we refer to $\mathcal{D}|E_{\vec{a}}$ as a “ball distribution” since its marginal over the coordinates in $\text{CONST}(\vec{a})$ is supported on strings contained in a Hamming ball.

4.2 The algorithm

Our algorithm for agnostically learning conjunctions is presented in [Algorithm 1](#). Let \mathcal{D} be the input distribution over $X \times \{\pm 1\}$, and let c^* be its (unknown) closest conjunction with $\text{err}_{\mathcal{D}}(c^*) = \text{opt}_{\mathcal{D}}$. Our algorithm will draw an m -tuple of independent labeled examples

$$\overrightarrow{(\mathbf{a}, \mathbf{b})} = ((\mathbf{a}^1, \mathbf{b}^1), \dots, (\mathbf{a}^m, \mathbf{b}^m))$$

from \mathcal{D} . Intuitively, we would like to work with the ball distribution $\mathcal{D}|E_{\vec{\mathbf{a}}}$ obtained from a tuple $\vec{\mathbf{a}} = (\mathbf{a}^1, \dots, \mathbf{a}^m)$ such that $c^*(\mathbf{a}^i) = 1$ for each i , but we do not have access to the value of $c^*(\mathbf{a}^i)$; rather, we only have access to the label bit \mathbf{b}^i . For the moment, though, let us hope or pretend that if $\mathbf{b}^1 = \dots = \mathbf{b}^m = 1$ then additionally indeed $c^*(\mathbf{a}^i) = 1$ for all i (this is roughly what it means to obtain a “useful” set of examples as alluded to in [Section 2.1](#)).⁵

Now if indeed $\vec{\mathbf{a}}$ satisfies that $c^*(\mathbf{a}^i) = 1$ for each i , the relevant variables of the conjunction c^* must all belong to $\text{CONST}(\vec{\mathbf{a}})$, the set of “previously unanimous” coordinates that always took the same value across all m examples $\mathbf{a}^1, \dots, \mathbf{a}^m$. Intuitively, the event $E_{\vec{\mathbf{a}}}$ is the event that a string $x \in X$ disagrees with \mathbf{a}^1 on “not too many” of these “previously unanimous” coordinates in $\text{CONST}(\vec{\mathbf{a}})$ (this is the event E that was alluded to in [Section 2.1](#)).

We briefly describe each step of [Algorithm 1](#). In each iteration i of the main loop of Step 2, the algorithm draws $\overrightarrow{(\mathbf{a}, \mathbf{b})}$ from \mathcal{D}^m (Step 2(a)) and checks that all examples $(\mathbf{a}^1, \mathbf{b}^1), \dots, (\mathbf{a}^m, \mathbf{b}^m)$ have $\mathbf{b}^i = 1$ (Step 2(b)), and continues only when this happens. As discussed above, the algorithm “hopes” that $c^*(\mathbf{a}^i) = 1$ for all i , and further checks that $\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}}[E_{\vec{\mathbf{a}}}(\mathbf{x})]$ is not too small (Step 2(c)). If this check passes, it runs the L_1 regression algorithm on examples from $\mathcal{D}|E_{\vec{\mathbf{a}}}$ (this is the \mathcal{D}' alluded to in [Section 2.1](#)) to obtain a hypothesis h_i . At the end of the main loop, it draws more examples from \mathcal{D} and output the best hypothesis h_i with the minimum disagreement.

4.3 Analysis of the algorithm

Let us write \mathcal{D}_{c^*} to denote the distribution of $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$ conditioned on having $c^*(\mathbf{x}) = \mathbf{y} = 1$. (Note that this is well defined if there are such pairs in the support of \mathcal{D} . We will always assume this is the case in this subsection. The case when the support of \mathcal{D} contains no such pairs is trivial and will be handled by the default h_0 in the algorithm. See [Section 4.4](#).)

To analyze a single iteration of the loop, we will use the following definition with $m = n^{1/3}$:

Definition 28. We say a tuple of pairs $\overrightarrow{(a, b)} = ((a^1, b^1), \dots, (a^m, b^m)) \in (X \times \{\pm 1\})^m$ is *useful* if $c^*(a^t) = b^t = 1$ for all $t \in [m]$ and $\vec{\mathbf{a}} = (a^1, \dots, a^m)$ satisfies the following condition:

$$\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_{c^*}} [E_{\vec{\mathbf{a}}}(\mathbf{x})] \geq 1 - \varepsilon. \quad (8)$$

The following lemma shows that the probability of $\overrightarrow{(\mathbf{a}, \mathbf{b})} \sim \mathcal{D}_{c^*}^m$ being useful is not too small:

Lemma 29. Let $\overrightarrow{(\mathbf{a}, \mathbf{b})} \sim (\mathcal{D}_{c^*})^m$ with $m = n^{1/3}$. The probability that $\overrightarrow{(\mathbf{a}, \mathbf{b})}$ is useful is at least ε^m .

⁵We remark that this approach of “hoping” that all examples satisfy c^* is similar in spirit to an idea used in the DNF learning context in [\[DDS14\]](#). In that setting the algorithm “hopes” that all members of some set of positive examples satisfy the same term of the DNF, and takes a bitwise-AND of examples to try to identify that term. Both [\[DDS14\]](#) and the current paper use multiple repeated trials, since in both settings the desired “hope” may only hold with fairly small probability.

Input: Samples drawn from an arbitrary and unknown distribution \mathcal{D} over $X \times \{\pm 1\}$.

Output: A hypothesis $h : X \rightarrow \{\pm 1\}$.

AGNOSTICALLY-LEARN-CONJUNCTION:

1. Set $m := n^{1/3}$, $L := (1/\varepsilon)^{O(m)}$, and $h_0 : X \rightarrow \{\pm 1\}$ to be $h_0 \equiv -1$.
2. Repeat for $i = 1, \dots, L$:
 - (a) Sample $\overrightarrow{(\mathbf{a}, \mathbf{b})} = ((\mathbf{a}^1, \mathbf{b}^1), \dots, (\mathbf{a}^m, \mathbf{b}^m)) \sim \mathcal{D}^m$.
 - (b) If any $\mathbf{b}^i = -1$, set $h_i \equiv -1$ and skip to the next iteration of the loop.
 - (c) Run the L_1 regression algorithm ([Theorem 10](#)) (with its failure probability parameter δ set to 0.01, its error parameter set to ε , and its degree parameter set to $d = O(n^{1/3} \log(1/\varepsilon))$), using examples drawn from $\mathcal{D}|E_{\vec{\mathbf{a}}}$, to obtain a hypothesis h'_i . Access to $\mathcal{D}|E_{\vec{\mathbf{a}}}$ is simulated by drawing from \mathcal{D} and checking if the sample satisfies $E_{\vec{\mathbf{a}}}$. Let N be the number of samples needed by the L_1 regression algorithm. Draw $O(1/\varepsilon) \cdot N$ samples from \mathcal{D} and feed the first N of them that belong to $E_{\vec{\mathbf{a}}}$ to the L_1 regression algorithm; set $h_i \equiv -1$ and skip to the next iteration if there are less than N samples in $E_{\vec{\mathbf{a}}}$.
 - (d) Finally, set h_i to be the following hypothesis: For each $x \in X$:

$$h_i(x) = \begin{cases} h'_i(x) & \text{if } E_{\vec{\mathbf{a}}}(x) \text{ holds} \\ -1 \text{ (i.e. False)} & \text{if } E_{\vec{\mathbf{a}}}(x) \text{ does not hold} \end{cases} \quad (7)$$

3. Draw a set of $\text{poly}(n/\varepsilon)$ samples from \mathcal{D} and choose the hypothesis from h_0, h_1, \dots, h_L that has the minimum disagreement with the samples drawn.

Algorithm 1: An algorithm to agnostically learn conjunctions.

Proof. We view the samples $\mathbf{a}^1, \dots, \mathbf{a}^m$ as being drawn one by one. We always use $\vec{\mathbf{a}}$ to denote the current tuple of $\mathbf{a}^1, \dots, \mathbf{a}^t$ collected after t rounds, and we let $E_{\vec{\mathbf{a}}}$ be the event using the current $\vec{\mathbf{a}}$:

$$E_{\vec{\mathbf{a}}} := \left\{ x \in X : \text{number of } i \in \text{CONST}(\vec{\mathbf{a}}) \text{ with } x_i \neq \mathbf{a}_i^1 \leq n^{2/3} \right\}.$$

Note that since our draws are from \mathcal{D}_{c^*} we get $c^*(\mathbf{a}^t) = \mathbf{b}^t = 1$ for all t “for free,” and we need only worry about the condition given in [Equation \(8\)](#).

After drawing \mathbf{a}^1 , we have that $\text{CONST}(\vec{\mathbf{a}}) = [n]$, and $x \in X$ satisfies $E_{\vec{\mathbf{a}}}$ if and only if the number of $i \in [n]$ with $x_i \neq \mathbf{a}_i^1$ (i.e. the Hamming distance $\text{dist}(x, \mathbf{a}^1)$) is at most $n^{2/3}$. If the probability of $E_{\vec{\mathbf{a}}}(\mathbf{x})$ over $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_{c^*}$ at this point is already at least $1 - \varepsilon$ then we are done (this is because as we draw more samples, $\text{CONST}(\vec{\mathbf{a}})$ can only shrink and the probability of $E_{\vec{\mathbf{a}}}(\mathbf{x})$ can only increase). So assume that after the first round, the probability of $E_{\vec{\mathbf{a}}}(\mathbf{x})$ over $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_{c^*}$ is at most $1 - \varepsilon$; this this means that with probability at least ε over $(\mathbf{a}^2, \mathbf{b}^2) \sim \mathcal{D}_{c^*}$, the size of $\text{CONST}(\vec{\mathbf{a}})$ goes down by $n^{2/3}$. We pay this factor of ε in probability, and ask for such a second sample $(\mathbf{a}^2, \mathbf{b}^2)$ and add it to $\vec{\mathbf{a}}$; after this, the size of $\text{CONST}(\vec{\mathbf{a}})$ has shrunk by at least $n^{2/3}$.

We repeat the above argument. This can repeat no more than $n^{1/3}$ times since $\text{CONST}(\mathbf{a})$ can shrink by at least $n^{2/3}$ for no more than $n^{1/3}$ rounds. This means that within the $m = n^{1/3}$ rounds,

there must be a round after which the probability of $E_{\vec{a}}(\mathbf{x})$ over $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_{c^*}$ becomes at least $1 - \varepsilon$. The lemma follows given that the total probability we paid for this to happen is at most ε^m . \square

Before giving the next lemma we recall some basics about Chebyshev polynomials. We write T_k to denote the k th Chebyshev polynomial of the first kind. We will use the following well-known facts about Chebyshev polynomials:

Fact 30. (I) $T_k(1 + \varepsilon) \geq (1/2)e^{k\sqrt{\varepsilon}}$ for any $0 \leq \varepsilon \leq 0.4$. (II) $|T_k(x)| \leq 1$ for all $x \in [-1, 1]$.

Item (I) follows immediately from the definition

$$T_k(x) = \frac{1}{2} \left[\left(x + \sqrt{x^2 - 1} \right)^k + \left(x - \sqrt{x^2 - 1} \right)^k \right]$$

and that $1 + \sqrt{2\varepsilon} \geq e^{\sqrt{\varepsilon}}$ for $0 \leq \varepsilon \leq 0.4$. Item (II) follows from the fact that $T_k(\cos \theta) = \cos(k\theta)$.

Lemma 31. Suppose that $\vec{(a, b)} \in (X \times \{\pm 1\})^m$ is useful. Then there exists a polynomial p of degree $d = O(n^{1/3} \log(1/\varepsilon))$ such that $|p(x) - c^*(x)| \leq \varepsilon$ for every x in $E_{\vec{a}}$ and consequently,

$$\mathbf{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}|E_{\vec{a}}} \left[(p(\mathbf{x}) - c^*(\mathbf{x}))^2 \right] \leq \varepsilon^2.$$

Proof. For simplicity, we assume that c^* is a monotone conjunction of the form $x_1 \wedge \dots \wedge x_s$; the general case follows similarly. We essentially use the usual Chebyshev construction of approximate polynomials for AND, but we take advantage of the fact that by the nature of $E_{\vec{a}}$, we are promised that at least $s - n^{2/3}$ literals in the conjunction are always satisfied. To see this is the case, note that $c^*(a^i) = 1$ for all i so $[s] \subseteq \text{CONST}(\vec{a})$ and $a_1^1 = \dots = a_s^1 = 1$. Given that $x \in E_{\vec{a}}$ can only disagree with a^1 at $n^{2/3}$ coordinates in $\text{CONST}(\vec{a})$, it can only falsify $n^{2/3}$ of x_1, \dots, x_s .

Inspired by this observation, let

$$\Delta := \min(n^{2/3}, s), \quad d := \left\lceil 3\sqrt{\Delta} \log(1/\varepsilon) \right\rceil \quad \text{and} \quad q(t) = \frac{T_d\left(\frac{t}{\Delta}\right)}{T_d\left(\frac{\Delta+1}{\Delta}\right)},$$

where T_d is the d th Chebyshev polynomial of the first kind. It's clear that $q(\Delta + 1) = 1$. Moreover, by **Fact 30**, we have that for any $t \in [-\Delta, \Delta]$,

$$|q(t)| \leq 2e^{-d\sqrt{1/\Delta}} \leq 2e^{-3\log(1/\varepsilon)\sqrt{\Delta}\cdot\sqrt{1/\Delta}} \leq \varepsilon, \tag{9}$$

when ε is sufficiently small. Thus it suffices to set

$$p(x) = q\left(\left(\sum_{i \in [s]} x_i\right) - (s - \Delta) + 1\right).$$

Using **Equation (9)**, we have that $|p(x) - c^*(x)| \leq \varepsilon$ for any string x at distance at most Δ from satisfying c^* . Since $E_{\vec{a}}$ contains such points only, the lemma is proved. \square

4.4 Proof of [Theorem 1](#)

Efficiency. It is easy to verify that the overall running time of [Algorithm 1](#) is dominated by the L calls to the L_1 regression algorithm in Step 2(d). Using [Theorem 10](#), the overall runtime is

$$L \cdot \text{poly}\left(n^d, 1/\varepsilon\right) = \left(\frac{1}{\varepsilon}\right)^{\tilde{O}(n^{1/3})},$$

where $d = O(n^{1/3} \log(1/\varepsilon))$, which is as claimed in [Theorem 1](#).

Correctness. Our goal is to prove the following lemma:

Lemma 32. With probability at least 0.9, at least one of the functions h_0, h_1, \dots, h_L at the end of the main loop satisfies $\text{err}_{\mathcal{D}}(h_i) \leq \text{opt}_{\mathcal{D}} + 2\varepsilon$.

On the other hand, with probability at least 0.9, Step 3 can approximate $\text{err}_{\mathcal{D}}(h_i)$ for every i up to error ε . So by a union bound, with probability at least 0.8, the hypothesis h returned satisfies $\text{err}_{\mathcal{D}}(h) \leq \text{opt}_{\mathcal{D}} + 4\varepsilon$. We prove [Lemma 32](#) in the rest of this section.

We start by dealing with a trivial case:

$$\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [c^*(\mathbf{x}) = \mathbf{y} = 1] < \varepsilon. \quad (10)$$

(Note that this includes the case mentioned earlier about \mathcal{D}_{c^*} being not well defined.) We show that in this case $h_0 \equiv -1$ has a small $\text{err}_{\mathcal{D}}(h_0)$. To see that this is the case, we have

$$1 - \text{opt}_{\mathcal{D}} = \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [c^*(\mathbf{x}) = \mathbf{y} = 1] + \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [c^*(\mathbf{x}) = \mathbf{y} = -1] < \varepsilon + \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [c^*(\mathbf{x}) = \mathbf{y} = -1]$$

and thus,

$$1 - \text{err}_{\mathcal{D}}(h_0) \geq \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [c^*(\mathbf{x}) = \mathbf{y} = -1] > 1 - \text{opt}_{\mathcal{D}} - \varepsilon$$

so $\text{err}_{\mathcal{D}}(h_0) \leq \text{opt}_{\mathcal{D}} + \varepsilon$. In the rest of the proof we assume that [Equation \(10\)](#) does not hold.

We will show that the following are sufficient conditions for h_i from the i th loop to satisfy $\text{err}_{\mathcal{D}}(h_i) \leq \text{opt}_{\mathcal{D}} + 2\varepsilon$.

1. $\overrightarrow{(a, b)}$ drawn in this loop is useful.
2. The L_1 regression algorithm receives all of the samples it needs.
3. The L_1 regression algorithm returns a function h'_i that satisfies

$$\text{err}_{\mathcal{D}|E_{\vec{a}}}(h'_i) \leq \text{opt}_{\mathcal{D}|E_{\vec{a}}} + \varepsilon. \quad (11)$$

Before that we show that these conditions hold for some i with probability at least 0.9.

Using [Lemma 29](#) and the assumption that [Equation \(10\)](#) does not hold, we have that $\overrightarrow{(a, b)} \sim \mathcal{D}^m$ is useful with probability at least $\varepsilon^m \cdot \varepsilon^m = \varepsilon^{O(m)}$, where we pay the first ε^m to draw m samples from \mathcal{D}_{c^*} (using [Equation \(10\)](#)) and the second ε^m is from [Lemma 29](#) for the tuple to be useful. Given our choice of $L = (1/\varepsilon)^{O(m)}$, we have that with probability at least $1 - o_n(1)$, $\overrightarrow{(a, b)}$ satisfies the first item for at least one loop i . Let's fix such a useful pair $\overrightarrow{(a, b)}$.

Next, given that $\overrightarrow{(a, b)}$ is useful, we have

$$\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}_{c^*}} [E_{\vec{a}}(\mathbf{x})] \geq 1 - \varepsilon.$$

Combining this with the assumption that Equation (10) does not hold, we have

$$\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [E_{\bar{a}}(\mathbf{x})] \geq \varepsilon(1 - \varepsilon) \geq \varepsilon/2,$$

from which the second item occurs during that loop with probability at least 0.99. Assuming both the first and second items, it follows from Lemma 31 and Theorem 10 that the third item holds with probability at least 0.99. As a result, the probability of having at least one loop satisfying all three items is at least 0.9. It suffices to show that when all three items hold, h_i (obtained from h'_i) satisfies $\text{err}_{\mathcal{D}}(h_i) \leq \text{opt}_{\mathcal{D}} + 2\varepsilon$.

First, from Equation (11) we have (using $c^* \in \mathcal{C}$)

$$\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D} | E_{\bar{a}}} [h'_i(\mathbf{x}) \neq \mathbf{y}] \leq \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D} | E_{\bar{a}}} [c^*(\mathbf{x}) \neq \mathbf{y}] + \varepsilon. \quad (12)$$

To relate the error of h'_i under $\mathcal{D} | E_{\bar{a}}$ to the error of h_i under \mathcal{D} , we note that

$$\text{err}_{\mathcal{D}}(h_i) = \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [h_i(\mathbf{x}) \neq \mathbf{y}] = \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) \wedge (h_i(\mathbf{x}) \neq \mathbf{y})] + \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [E_{\bar{a}}(\mathbf{x}) \wedge (h_i(\mathbf{x}) \neq \mathbf{y})].$$

For the first term, recalling Equation (7) we have that if $\neg E_{\bar{a}}$ holds then $h_i = -1$, so

$$\begin{aligned} & \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) \wedge (h_i(\mathbf{x}) \neq \mathbf{y})] \\ &= \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) \wedge (h_i(\mathbf{x}) = -1, \mathbf{y} = 1)] \\ &\leq \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) \wedge (\mathbf{y} = 1)] \\ &= \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) \wedge (\mathbf{y} = 1, c^*(\mathbf{x}) = -1)] + \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) \wedge (\mathbf{y} = c^*(\mathbf{x}) = 1)] \\ &\leq \underbrace{\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) \wedge (c^*(\mathbf{x}) \neq \mathbf{y})]}_{=A} + \underbrace{\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) \wedge (\mathbf{y} = c^*(\mathbf{x}) = 1)]}_{=B}. \end{aligned}$$

For the second term, we have

$$\begin{aligned} \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [E_{\bar{a}}(\mathbf{x}) \wedge (h_i(\mathbf{x}) \neq \mathbf{y})] &= \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D} | E_{\bar{a}}} [h_i(\mathbf{x}) \neq \mathbf{y}] \cdot \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [E_{\bar{a}}(\mathbf{x})] \\ &\leq \underbrace{\left(\Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D} | E_{\bar{a}}} [c^*(\mathbf{x}) \neq \mathbf{y}] + \varepsilon \right)}_{=C} \cdot \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [E_{\bar{a}}(\mathbf{x})], \end{aligned}$$

where the inequality is by Equation (12). It remains to argue that $A + B + C \leq \text{opt}_{\mathcal{D}} + 2\varepsilon$. Since

$$C \leq \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [E_{\bar{a}} \wedge (c^*(\mathbf{x}) \neq \mathbf{y})] + \varepsilon,$$

by inspection we have that $A + C \leq \text{opt}_{\mathcal{D}} + \varepsilon$. Finally, we have

$$B \leq \Pr_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}} [\neg E_{\bar{a}}(\mathbf{x}) | c^*(\mathbf{x}) = \mathbf{y} = 1] \leq \varepsilon$$

using the assumption that $\overrightarrow{(a, b)}$ is useful. This finishes the proof of Lemma 32.

5 Warm-Up: A $2^{\tilde{O}(k^{1/3})}$ -Query Quantum Tolerant Junta Tester

In this section we prove [Theorem 2](#). While this result will be subsumed by [Theorem 3](#), which gives a classical tester that has a matching query complexity, many of the tools which we develop in a simpler setting here will be needed later for our classical tester. Throughout the section, we take k' to be a generic parameter and show that the algorithm for testing functions $f : \{\pm 1\}^{k'} \rightarrow \{\pm 1\}$ makes $\exp(k^{1/3} \cdot \text{polylog}(k'/\varepsilon))$ queries. [Theorem 2](#) follows by plugging in $k' = \text{poly}(k, \varepsilon^{-1})$.

5.1 Drawing from the spectral sample

We recall the definition of the spectral sample of a Boolean function:

Definition 33 (Spectral sample, Definition 1.18 of [\[O'D14\]](#)). Given a function $f : \{\pm 1\}^{k'} \rightarrow \{\pm 1\}$, the *spectral sample* of f , denoted \mathcal{P}_f , is the probability distribution on subsets of $[k']$ (equivalently, on elements of $\{\pm 1\}^{k'}$) in which the set S has probability $\hat{f}(S)^2$. (Recall from [Section 3.2](#) that $\sum_S \hat{f}(S)^2 = 1$, so this is indeed a valid probability distribution.)

As discussed in [Remark 4](#), the only quantum aspect of [Algorithm 2](#) occurs in lines 2 and 3(b) where the algorithm makes draws from the spectral sample \mathcal{P}_f of f . Thus the rest of our discussion in this section will not involve any quantum considerations.

5.2 Smooth functions and local mean estimation

We will *informally* refer to a function $f : \{\pm 1\}^{k'} \rightarrow \mathbb{R}$ as an L -smooth function for some positive integer $L \in [k']$ if $\mathbf{W}^{\geq L}[f]$ is *tiny*. As we will see, L -smooth functions are useful because, when $\mathbf{W}^{\geq L}[f]$ is sufficiently small, using the techniques of [\[NP24\]](#), the magnitude of their means can be estimated using evaluations of f that come from a random ball of radius roughly \sqrt{L} , i.e. they have *local estimators*. To establish this, we start with a simple lemma from [\[NP24\]](#).

Lemma 34. Let \mathbf{X} be a random variable with $\text{Var}[\mathbf{X}] = \sigma^2$. Then

$$0 \leq \mathbf{E}[|\mathbf{X}|] - |\mathbf{E}[\mathbf{X}]| \leq \sigma.$$

Proof. The first inequality is trivial since $\mathbf{E}[|\mathbf{X}|] \geq |\mathbf{E}[\mathbf{X}]|$ for every random variable \mathbf{X} .

For the second one, we have

$$\mathbf{E}[|\mathbf{X}|] - |\mathbf{E}[\mathbf{X}]| = \mathbf{E}[|\mathbf{X}| - |\mathbf{E}[\mathbf{X}]|] \leq \mathbf{E}[|\mathbf{X} - \mathbf{E}[\mathbf{X}]|] \leq \sqrt{\mathbf{E}[(\mathbf{X} - \mathbf{E}[\mathbf{X}])^2]} = \sigma,$$

which completes the proof. \square

Now we prove the existence of local estimators for L -smooth functions. The error of these local estimators depends on the variance of the L -smooth function as well as its Fourier weight on levels at least L .

Lemma 35 (Local Estimators for Smooth Functions). Let τ and L be two parameters such that $\tau \in (0, 1/2]$ and $L \leq k'$ is a positive integer. There exists an r -local estimator \mathcal{E} with

$$r = \Theta\left(\sqrt{L \log(L) \log(1/\tau)}\right)$$

that approximates $|\mathbf{E}[f]|$ for all functions $f : \{\pm 1\}^{k'} \rightarrow \mathbb{R}$ with error at most

$$\tau \sqrt{\text{Var}[f]} + 5(k')^r \sqrt{\mathbf{W}^{\geq L}[f]}.$$

Proof. We closely follow the proof of Lemma 17 of [NP24]. Let $r := \Theta(\sqrt{L \log(L) \log(1/\tau)})$ and let $p_r^L(x)$ be the polynomial from Lemma 17 (with L playing the role of N). We set the hidden constant in r large enough so that

$$\max_{i \in [L]} \left| p_r^L(i) - 1 \right| \leq 2 \exp \left(-\Omega \left(\frac{r^2}{L \log L} \right) \right) \leq \tau. \quad (13)$$

As in Lemma 20, let $\alpha_i^{r,L}$ be the coefficients with $|\alpha_i^{r,L}| \leq 2r^r$ such that

$$p_r^L(x) = \sum_{i \in [r]} \alpha_i^{r,L} \binom{x}{i}.$$

We take

$$g \left(f|_{B(x,r)} \right) := f(x) - \sum_{i \in [r]} \alpha_i^{r,L} \sum_{S \subseteq [k']; |S|=i} \frac{\partial f}{\partial x_S}(x) \chi_S(x). \quad (14)$$

Note that this can be computed by only querying f on $B(x, r)$, as

$$\frac{\partial f}{\partial x_S}(x) \chi_S(x) = \frac{1}{2^{|S|}} \cdot \left(\sum_{T \subseteq S} (-1)^{|T|} f(x^{\oplus T}) \right). \quad (15)$$

Writing $f(x^{\oplus T}) = \sum_{U \subseteq [k']} \hat{f}(U) \chi_U(x^{\oplus T})$, it is easy to check that

$$\begin{aligned} g \left(f|_{B(x,r)} \right) &= f(x) - \sum_{i \in [r]} \alpha_i^{r,L} \sum_{S \subseteq [k']; |S|=i} \sum_{U \supseteq S} \hat{f}(U) \chi_U(x) \\ &= f(x) - \sum_{U \neq \emptyset} p_r^L(|U|) \hat{f}(U) \chi_U(x) \\ &= \mathbf{E}[f] + \sum_{U \neq \emptyset} \left(1 - p_r^L(|U|) \right) \hat{f}(U) \chi_U(x), \end{aligned} \quad (16)$$

where we used the fact that $\hat{f}(\emptyset) = \mathbf{E}[f]$. It is immediate from the above that

$$\mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[g \left(f|_{B(\mathbf{x},r)} \right) \right] = \mathbf{E}[f], \quad (17)$$

where we used $\mathbf{E}[\chi_S(\mathbf{x})] = 0$ for $S \neq \emptyset$. Furthermore, we have

$$\mathbf{Var}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[g \left(f|_{B(\mathbf{x},r)} \right) \right] = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\left(g \left(f|_{B(\mathbf{x},r)} \right) - \mathbf{E}[f] \right)^2 \right] = \sum_{i \in [k']} \left(1 - p_r^L(i) \right)^2 \mathbf{W}^{=i}[f], \quad (18)$$

where Equation (18) follows from Equation (16) via Parseval's formula.

Splitting the sum into two parts, the RHS of Equation (18) becomes

$$\sum_{i \leq L} \left(1 - p_r^L(i) \right)^2 \mathbf{W}^{=i}[f] + \sum_{i > L} \left(1 - p_r^L(i) \right)^2 \mathbf{W}^{=i}[f] \leq \tau^2 \cdot \mathbf{Var}[f] + 25(k')^{2r} \cdot \mathbf{W}^{\geq L}[f]$$

where we used Item (ii) of Lemma 17 and Equation (13) to bound the first term, and we bounded the second term using Lemma 19. Taking the estimator \mathcal{E} to be

$$\mathcal{E} \left(f|_{B(\mathbf{x},r)} \right) := \left| g \left(f|_{B(\mathbf{x},r)} \right) \right|, \quad (19)$$

the lemma follows from Equation (17) and Lemma 34. \square

5.3 The Sharp Noise Operator

We now turn to designing a “sharp noise” operator on a set of variables V , which we denote by $\text{SHARPNOISE}_{\ell, \kappa, \Delta}^V$ with three parameters ℓ, κ and Δ . Intuitively, we would like this operator, when applied to a function $f : \{\pm 1\}^{k'} \rightarrow \mathbb{R}$, to zero out all Fourier coefficients $\widehat{f}(S)$ for which $|S \cap V| > \ell$, and to leave unchanged all other Fourier coefficients. We do not achieve this exactly; instead, all Fourier coefficients for which $|S \cap V| \geq \kappa \ell$ are made “very small” (as captured by the Δ parameter), and all Fourier coefficients for which $|S \cap V| \leq \ell$ are “approximately preserved.” More precisely, the SHARPNOISE operator has the following properties:

Lemma 36. Given any $V \subseteq [k']$ and three positive integers ℓ, κ, Δ such that $\ell \in [k']$ and $\kappa \geq 5$ (so that $e^{-\kappa/2} \kappa \leq 0.5$), there exists an operator $\text{SHARPNOISE}_{\ell, \kappa, \Delta}^V$ on $f : \{\pm 1\}^{k'} \rightarrow \mathbb{R}$ such that

$$\text{SHARPNOISE}_{\ell, \kappa, \Delta}^V f = \sum_{S \subseteq [k']} \lambda(S) \widehat{f}(S) \chi_S(x),$$

where each $\lambda(S) \in [0, 1]$ and satisfies

$$\lambda(S) = \begin{cases} \geq 1 - \Delta 2^{-\kappa} & \text{if } |S \cap V| \leq \ell \\ \leq 2^{-\Delta} & \text{if } |S \cap V| \geq \kappa \ell. \end{cases}$$

Moreover, this operator can be written as

$$\text{SHARPNOISE}_{\ell, \kappa, \Delta}^V = \sum_{i=0}^{\kappa \Delta} \alpha_i \mathbf{T}_{\rho^i}^V, \quad \text{where } \rho := 1 - \frac{1}{2\ell} \quad \text{and} \quad \sum_{i=0}^{\kappa \Delta} |\alpha_i| \leq 2^{2\kappa \Delta}. \quad (20)$$

For functions $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$, we always have

$$\mathbf{Var} \left[\text{SHARPNOISE}_{\ell, \kappa, \Delta}^V f \right] \leq 1.$$

Proof. We design $\text{SHARPNOISE}_{\ell, \kappa, \Delta}^V$ in stages using a few intermediate operators.

First we consider the operator U , which is defined as

$$Uf := f - \mathbf{T}_{\rho}^V f = \sum_{S \subseteq [k']} \left(1 - \rho^{|S \cap V|}\right) \widehat{f}(S) \chi_S(x),$$

with $\rho = 1 - 1/(2\ell)$. Note that by our choice of ρ , we have that

$$\left(1 - \rho^{|S \cap V|}\right) \text{ is } \begin{cases} \leq 1/2 & \text{if } |S \cap V| \leq \ell \\ \geq 1 - e^{-\kappa/2} & \text{if } |S \cap V| \geq \kappa \ell. \\ \in [0, 1] & \text{otherwise} \end{cases}$$

Next we consider the operator U' , which is defined as

$$U'f := f - U^\kappa f = \sum_{S \subseteq [k']} \left(1 - \left(1 - \rho^{|S \cap V|}\right)^\kappa\right) \widehat{f}(S) \chi_S(x),$$

which has brought us closer to our goal as

$$\left(1 - \left(1 - \rho^{|S \cap V|}\right)^\kappa\right) \text{ is } \begin{cases} \geq 1 - 2^{-\kappa} & \text{if } |S \cap V| \leq \ell \\ \leq 1/2 & \text{if } |S \cap V| \geq \kappa \ell, \\ \in [0, 1] & \text{otherwise} \end{cases}$$

where we used $e^{-\kappa/2} \leq 1/2$. Finally we define SHARPNOISE by applying U' repeatedly Δ times:

$$\text{SHARPNOISE}_{\ell, \kappa, \Delta}^V f := (U')^\Delta f = \sum_{S \subseteq [k']} \left(1 - \left(1 - \rho^{|S \cap V|}\right)^\kappa\right)^\Delta \hat{f}(S) \chi_S(x).$$

This has the desired properties, since

$$\left(1 - \left(1 - \rho^{|S \cap V|}\right)^\kappa\right)^\Delta \text{ is } \begin{cases} \geq 1 - \Delta 2^{-\kappa} & \text{if } |S \cap V| \leq \ell \\ \leq 2^{-\Delta} & \text{if } |S \cap V| \geq \kappa \ell \\ \in [0, 1] & \text{otherwise} \end{cases}$$

To obtain Equation (20) about writing $\text{SHARPNOISE}_{\ell, \kappa, \Delta}^V$ as a linear combination of Bernoulli noise operators over V with different noise rates, observe that

$$\left(1 - \left(1 - \rho^{|S \cap V|}\right)^\kappa\right)^\Delta$$

can be written as $p(\rho^{|S \cap V|})$ where

$$p(x) = \left(1 - (1 - x)^\kappa\right)^\Delta = \sum_{i=0}^{\kappa \Delta} \alpha_i x^i$$

is a univariate polynomial of degree at most $\kappa \Delta$ with $\sum_i |\alpha_i| \leq 2^{2\kappa \Delta}$. This gives Equation (20). \square

5.4 Parameters in the main algorithm

Before presenting the main algorithm, let's list parameters that will be used for SHARPNOISE:

$$\ell = k^{2/3}, \quad \rho = 1 - \frac{1}{2\ell}, \quad \kappa = 10 \log \left(\frac{k'}{\varepsilon}\right) \quad \text{and} \quad \Delta = 10r \log \left(\frac{k'}{\varepsilon}\right)$$

and parameters that will be used for the local estimator:

$$L = \kappa \ell, \quad \tau = \frac{\varepsilon}{10} \quad \text{and} \quad r = \Theta \left(\sqrt{L \log L \log(1/\tau)} \right).$$

In the rest of the section, \mathcal{E} denotes the r -local estimator given in Lemma 35 with parameters τ and L . Another parameter needed is

$$N := k'^{O(r)} \cdot 2^{O(\kappa \Delta)} \cdot \frac{k'^2}{\varepsilon^2} \leq \exp \left(k^{1/3} \cdot \text{polylog} \left(\frac{k'}{\varepsilon} \right) \right). \quad (21)$$

We note that while we take ℓ to be $k^{2/3}$ throughout our quantum testing algorithm, we will think of it as a parameter throughout Section 5.5, as we will need to set $\ell = k^{2/3} \text{polylog}(k'/\varepsilon)$ in our classical tester.

5.5 Local estimation of junta correlations

For our quantum junta tester we only need the results of this section for functions $f : \{\pm 1\} \rightarrow \{\pm 1\}$, but in the next section we will need these results for functions $f : \{\pm 1\} \rightarrow [-1, 1]$; hence we handle this more general setting in this subsection.

Given a k -subset $U \subseteq [k']$ and $C \subseteq U$, a subroutine is needed in the main algorithm to estimate

$$\text{corr} \left(f^{\overline{C}}, \mathcal{J}_U \right), \quad \text{where } f^{\overline{C}} := \text{SHARPNOISE}_{\ell, \kappa, \Delta}^{\overline{C}} f.$$

Recalling [Equations \(4\) and \(5\)](#), we can express the correlation as

$$\text{corr} \left(f^{\overline{C}}, \mathcal{J}_U \right) = \mathbf{E}_{\mathbf{y} \sim \{\pm 1\}^U} \left[\left| \mathbf{E}_{\mathbf{z} \sim \{\pm 1\}^{\overline{U}}} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}}(\mathbf{z}) \right] \right| \right]. \quad (22)$$

We first show that the following is a good estimation of the correlation:

$$\text{Est} := \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\mathcal{E} \left(f_{U \rightarrow \mathbf{x}_U}^{\overline{C}} \big|_{B(\mathbf{x}_{\overline{U}}, r)} \right) \right]. \quad (23)$$

Lemma 37. Let $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$. For any k -subset $U \subseteq [k']$ and $C \subseteq U$, we have

$$\left| \text{Est} - \text{corr} \left(f^{\overline{C}}, \mathcal{J}_U \right) \right| \leq 2\tau.$$

Proof. To align with [Equation \(22\)](#), for clarity we can rewrite Est as

$$\mathbf{E}_{\mathbf{y} \sim \{\pm 1\}^U} \left[\mathbf{E}_{\mathbf{z} \sim \{\pm 1\}^{\overline{U}}} \left[\mathcal{E} \left(f_{U \rightarrow \mathbf{y}}^{\overline{C}} \big|_{B(\mathbf{z}, r)} \right) \right] \right].$$

By [Lemma 35](#), we have that for every $\mathbf{y} \in \{\pm 1\}^U$,

$$\mathbf{E}_{\mathbf{z}} \left[\mathcal{E} \left(f_{U \rightarrow \mathbf{y}}^{\overline{C}} \big|_{B(\mathbf{z}, r)} \right) \right] = \left| \mathbf{E}_{\mathbf{z}} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}}(\mathbf{z}) \right] \right| \pm \left(\tau \sqrt{\mathbf{Var} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}} \right]} + 5(k')^r \sqrt{\mathbf{W}^{\geq L} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}} \right]} \right),$$

so taking expectation over \mathbf{y} and recalling [Equation \(22\)](#), we have that

$$\text{Est} = \text{corr}(f^{\overline{C}}, \mathcal{J}_U) \pm \mathbf{E}_{\mathbf{y} \sim \{\pm 1\}^U} \left[\left(\tau \sqrt{\mathbf{Var} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}} \right]} + 5(k')^r \sqrt{\mathbf{W}^{\geq L} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}} \right]} \right) \right].$$

We proceed to bound the two contributions to the error term above. For the first, we have

$$\mathbf{E}_{\mathbf{y} \sim \{\pm 1\}^U} \left[\sqrt{\mathbf{Var} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}} \right]} \right] \leq \sqrt{\mathbf{E}_{\mathbf{y}} \left[\mathbf{Var} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}} \right] \right]} \leq \sqrt{\mathbf{Var} \left[f^{\overline{C}} \right]} \leq 1$$

where the first inequality is Cauchy-Schwarz, the second is the law of total variance, and the third is because f 's outputs lie in $[-1, 1]$. For the second, we have

$$\mathbf{E}_{\mathbf{y}} \left[\sqrt{\mathbf{W}^{\geq L} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}} \right]} \right] \leq \sqrt{\mathbf{E}_{\mathbf{y}} \left[\mathbf{W}^{\geq L} \left[f_{U \rightarrow \mathbf{y}}^{\overline{C}} \right] \right]} = \sqrt{\sum_{R: |R \setminus U| \geq L} \widehat{f^{\overline{C}}}(R)^2}$$

where the first inequality again is Cauchy-Schwarz and the second is by an application of Proposition 3.22 of [O'D14].

Since $C \subseteq U$, every R such that $|R \setminus U| \geq L$ also satisfies $|R \setminus C| \geq L$. As a result, we have

$$\sum_{R:|R \setminus U| \geq L} \widehat{f^{\overline{C}}}(R)^2 \leq \sum_{R:|R \setminus C| \geq L} \widehat{f^{\overline{C}}}(R)^2 \leq 2^{-2\Delta} \cdot \sum_R \widehat{f}(R)^2 \leq 2^{-2\Delta}$$

where the second inequality is by Lemma 36 and the third is because $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$. Combining everything, we have

$$\left| \text{Est} - \text{corr}(f^{\overline{C}}, \mathcal{J}_U) \right| \leq \tau + 5(k')^r \cdot 2^{-\Delta} \leq 2\tau,$$

by the choice of Δ . □

Given the above lemma, it suffices to estimate Est. For this we introduce a notion of a “sample bundle” for a given point x :

Definition 38. Given $x \in \{\pm 1\}^{k'}$, a *sample bundle* \mathcal{B} for x is a multiset of $\binom{k'}{\leq r} \cdot (\kappa\Delta + 1)$ many points $\mathcal{B}(y, i) \in \{\pm 1\}^{k'}$, where y ranges over $B(x, r)$ and i ranges over $[0 : \kappa\Delta]$.

We write $\mathcal{D}_{x,C}$ to denote the following distribution of sample bundles \mathcal{B} for x :

For each $y \in B(x, r)$ and $i \in [0 : \kappa\Delta]$, draw independently a sample $\mathcal{B}(y, i) \sim N_{\rho^i}^{\overline{C}}(y)$.

Now we give a subroutine which, given $C \subseteq U$ and sample bundles for N points, computes a high-confidence, high-accuracy estimate of $\text{corr}(f^{\overline{C}}, \mathcal{J}_U)$ when all the points and their sample bundles are drawn correctly. In addition, the subroutine queries f on points only in the sample bundles. Looking ahead, it is crucial that the way these samples are drawn is independent of U and thus they can be reused to estimate $\text{corr}(f^{\overline{C}}, \mathcal{J}_U)$ for all k -subsets U of $[k']$ that contain C .

Lemma 39. Let \mathcal{A} be a 1-bounded randomized algorithm for $h : \{\pm 1\}^{k'} \rightarrow [-1, 1]$. There exists an algorithm which, given any k -subset U , $C \subseteq U$, $x^{(1)}, \dots, x^{(N)} \in \{\pm 1\}^{k'}$ and a sample bundle $\mathcal{B}^{(i)}$ for $x^{(i)}$ for each $i \in [N]$, only calls \mathcal{A} on points in $\mathcal{B}^{(1)}, \dots, \mathcal{B}^{(N)}$ and returns a number \mathbf{Est}' . When $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)} \sim \{\pm 1\}^{k'}$ and $\mathcal{B}^{(i)} \sim \mathcal{D}_{\mathbf{x}^{(i)}, C}$ independently, we have

$$\left| \mathbf{Est}' - \text{corr}(h^{\overline{C}}, \mathcal{J}_U) \right| \leq \varepsilon/2$$

with probability at least $1 - 2^{-(k'/\varepsilon)^2}$. Moreover, the number of calls to \mathcal{A} that are made by the algorithm is at most $N \cdot \text{poly}((k')^r, \kappa\Delta, 1/\varepsilon) \leq \exp\left(k^{1/3} \cdot \text{polylog}\left(\frac{k'}{\varepsilon}\right)\right)$.

Proof. Given Lemma 37 (and the choice of $\tau = \varepsilon/10$), it suffices to give a $(\varepsilon/3)$ -estimation of Est.

By Equation (23) and Equation (19), we have that

$$\text{Est} = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\mathcal{E} \left(h_{U \rightarrow \mathbf{x}_U}^{\overline{C}} \mid_{B(\mathbf{x}_{\overline{U}}, r)} \right) \right] = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\left[g \left(h_{U \rightarrow \mathbf{x}_U}^{\overline{C}} \mid_{B(\mathbf{x}_{\overline{U}}, r)} \right) \right] \right].$$

Recalling Equations (6), (14), (15) and (20), we get that

$$\text{Est} = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\left[\sum_{y \in B(\mathbf{x}_{\overline{U}}, r)} \sum_{i=0}^{\kappa\Delta} \alpha_{y,i} \cdot \mathbf{E}[h(z_{y,i,\mathbf{x}})] \right] \right],$$

where each $\alpha_{y,i}$ is a fixed coefficient that satisfies $|\alpha_{y,i}| \leq (k')^{O(r)} \cdot 2^{2\kappa\Delta}$ and each $\mathbf{z}_{y,i,\mathbf{x}}$ is an independent draw from $N_{\rho^i}^{\overline{C}}(\mathbf{x}_U \circ y)$. Given that $y \in B(\mathbf{x}_U, r)$, we have $\mathbf{x}_U \circ y \in B(\mathbf{x}, r)$. Thus, we define the output of our algorithm (the estimator \mathbf{Est}' for Est) to be

$$\mathbf{Est}' = \frac{1}{N} \sum_{j=1}^N \mathbf{Est}'_j(x^{(j)}), \text{ where } \mathbf{Est}'_j(x^{(j)}) = \left| \sum_{y \in B(x^{(j)}, r)} \sum_{i=0}^{\kappa\Delta} \alpha_{y,i} \cdot \mathcal{A}_{\varepsilon', \delta'} \left(\mathcal{B}^{(j)}(x^{(j)} \circ y, i) \right) \right|, \quad (24)$$

where $\mathcal{A}_{\varepsilon', \delta'}$ is the ε' -accuracy, δ' -confidence algorithm for estimating h that is obtained from the randomized algorithm \mathcal{A} for h as described in [Claim 13](#). (Note that this algorithm indeed only calls \mathcal{A} on points in the sample bundles $\mathcal{B}^{(1)}, \dots, \mathcal{B}^{(N)}$ as claimed.) Here ε', δ' are parameters that we will set later (see [Equation \(27\)](#)).

It remains to argue that when $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)} \sim \{\pm 1\}^{k'}$ and $\mathcal{B}^{(i)} \sim \mathcal{D}_{\mathbf{x}^{(i)}, C}$ independently, we have

$$|\mathbf{Est}' - \text{Est}| \leq \varepsilon/3 \text{ except with failure probability at most } 2^{-(k'/\varepsilon)^2}. \quad (25)$$

Towards this end, we have the following claim:

Claim 40. For each $j \in [N]$, with probability 1 we have that

$$|\mathbf{Est}'_j(\mathbf{x}^{(j)})| \leq E_{\max} := \text{poly}((k')^r, 2^{\kappa\Delta})$$

and

$$\left| \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] - \text{Est} \right| \leq \frac{\varepsilon}{6}.$$

Proof. The bound on the magnitude of $\mathbf{Est}'_j(\mathbf{x}^{(j)})$ follows directly from the definition of \mathbf{Est}'_j given in [Equation \(24\)](#), the coefficient bounds $|\alpha_{y,i}| \leq (k')^{O(r)} \cdot 2^{2\kappa\Delta}$, and the absolute bound on the output of $\mathcal{A}_{\varepsilon', \delta'}$ given by [Claim 13](#).

We turn to bounding $\left| \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] - \text{Est} \right|$. Let us define

$$V(\mathbf{x}^{(j)}) := \left| \sum_{y \in B(\mathbf{x}_U^{(j)}, r)} \sum_{i=0}^{\kappa\Delta} \alpha_{y,i} \cdot \mathbf{E}_{\mathbf{z}_{y,i,\mathbf{x}^{(j)}} \sim N_{\rho^i}^{\overline{C}}(\mathbf{x}_U^{(j)} \circ y)} [h(\mathbf{z}_{y,i,\mathbf{x}^{(j)}})] \right|, \text{ so } \text{Est} = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} [V(\mathbf{x})].$$

We have

$$\left| \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] - \text{Est} \right| = \left| \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] - \mathbf{E}[V(\mathbf{x}^{(j)})] \right|$$

so our goal is to prove that

$$\left| \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] - \mathbf{E}[V(\mathbf{x}^{(j)})] \right| \leq \frac{\varepsilon}{6}.$$

We have

$$\begin{aligned} & \left| \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] - \mathbf{E}[V(\mathbf{x}^{(j)})] \right| \\ &= \left| \mathbf{E} \left[\left| \sum_{y \in B(\mathbf{x}_U^{(j)}, r)} \sum_{i=0}^{\kappa\Delta} \alpha_{y,i} \cdot \mathcal{A}_{\varepsilon', \delta'} \left(\mathcal{B}^{(j)}(\mathbf{x}_U^{(j)} \circ y, i) \right) \right| - \left| \sum_{y \in B(\mathbf{x}_U^{(j)}, r)} \sum_{i=0}^{\kappa\Delta} \alpha_{y,i} \cdot \mathbf{E}_{\mathbf{z}_{y,i,\mathbf{x}^{(j)}} \sim N_{\rho^i}^{\overline{C}}(\mathbf{x}_U^{(j)} \circ y)} [h(\mathbf{z}_{y,i,\mathbf{x}^{(j)}})] \right| \right] \right| \\ &\leq \mathbf{E} \left[\sum_{y \in B(\mathbf{x}_U^{(j)}, r)} \sum_{i=0}^{\kappa\Delta} |\alpha_{y,i}| \cdot \left| \mathcal{A}_{\varepsilon', \delta'} \left(\mathcal{B}^{(j)}(\mathbf{x}_U^{(j)} \circ y, i) \right) - \mathbf{E}_{\mathbf{z}_{y,i,\mathbf{x}^{(j)}} \sim N_{\rho^i}^{\overline{C}}(\mathbf{x}_U^{(j)} \circ y)} [h(\mathbf{z}_{y,i,\mathbf{x}^{(j)}})] \right| \right]. \end{aligned}$$

Recalling the distribution of our sample bundles, we have that for each $y \in B(\mathbf{x}_U^{(j)}, r), i \in [0 : \kappa\Delta]$ the distribution of $\mathcal{B}^{(j)}(\mathbf{x}_U^{(j)} \circ y, i)$ is identical to the distribution of $\mathbf{z}_{y,i,\mathbf{x}^{(j)}} \sim N_{\rho^i}^{\mathcal{C}}(\mathbf{x}_U^{(j)} \circ y)$ (in fact, this is true on an outcome-by-outcome basis for $\mathbf{x}^{(j)}$). Hence we may rewrite the above as

$$\begin{aligned} & \mathbf{E} \left[\sum_{y \in B(\mathbf{x}_U^{(j)}, r)} \sum_{i=0}^{\kappa\Delta} |\alpha_{y,i}| \cdot \left| \mathcal{A}_{\varepsilon', \delta'} \left(\mathcal{B}^{(j)}(\mathbf{x}_U^{(j)} \circ y, i) \right) - \mathbf{E}_{\mathcal{B}^{(j)}} [h(\mathcal{B}^{(j)}(\mathbf{x}_U^{(j)} \circ y, i))] \right| \right] \\ & \leq \mathbf{E}_{\mathbf{x}^{(j)}} \left[\sum_{y \in B(\mathbf{x}_U^{(j)}, r)} \sum_{i=0}^{\kappa\Delta} |\alpha_{y,i}| \cdot \mathbf{E}_{\mathcal{B}^{(j)}} \left[\left| \mathcal{A}_{\varepsilon', \delta'} \left(\mathcal{B}^{(j)}(\mathbf{x}_U^{(j)} \circ y, i) \right) - h(\mathcal{B}^{(j)}(\mathbf{x}_U^{(j)} \circ y, i)) \right| \right] \right]. \end{aligned}$$

Now, since on every input $\mathcal{A}_{\varepsilon', \delta'}$ is a $\pm\varepsilon'$ -accurate estimate of h except with failure probability δ' , and $\mathcal{A}_{\varepsilon', \delta'}$ and h both always output values in $[-1, 1]$, by [Claim 13](#), the above is at most

$$\binom{k'}{\leq r} \cdot 2^{2\kappa\Delta} \cdot (2\delta' + \varepsilon'). \quad (26)$$

Choosing

$$\varepsilon' = \varepsilon \cdot \text{poly} \left(\frac{1}{(k')^r}, \frac{1}{2\kappa\Delta} \right), \delta' = \varepsilon \cdot \text{poly} \left(\frac{1}{(k')^r}, \frac{1}{2\kappa\Delta} \right), \quad (27)$$

we get that [Equation \(26\)](#) is at most $\varepsilon/6$ as desired. \square

With [Claim 40](#) in hand we can prove [Equation \(25\)](#). By a Hoeffding bound, using the fact ([Claim 40](#)) that each $|\mathbf{Est}'_j(\mathbf{x}^{(j)})| \leq E_{\max}$, we have that

$$\begin{aligned} \Pr \left[\left| \frac{1}{N} \sum_{j=1}^N \mathbf{Est}'_j(\mathbf{x}^{(j)}) - \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] \right| \geq \frac{\varepsilon}{6} \right] &= \Pr \left[\left| \mathbf{Est}' - \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] \right| \geq \frac{\varepsilon}{6} \right] \\ &\leq 2 \exp \left(\frac{-2N(\frac{\varepsilon}{6})^2}{(E_{\max})^2} \right) \\ &\leq 2^{-(k'/\varepsilon)^2}, \end{aligned}$$

where the last inequality holds by our choice of N (recall [Equation \(21\)](#)). On the other hand, note that when $\left| \mathbf{Est}' - \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] \right| \leq \frac{\varepsilon}{6}$, recalling that [Claim 40](#) gives us $\left| \mathbf{E}[\mathbf{Est}'_j(\mathbf{x}^{(j)})] - \text{Est} \right| \leq \frac{\varepsilon}{6}$, by the triangle inequality we have that

$$\left| \mathbf{Est}' - \text{Est} \right| \leq \varepsilon/3,$$

which completes the proof of [Equation \(25\)](#).

It remains only to bound the number of calls to \mathcal{A} . Recalling [Equation \(24\)](#) and [Claim 13](#), this is easily seen to be at most $N \cdot \text{poly}((k')^r, \kappa\Delta, 1/\varepsilon)$, and the lemma is proved. \square

Remark 41. In the next section, when we use [Lemma 39](#) on a function h with real-valued outputs, in order to bound the query complexity we will need to account for the number of oracle calls that the randomized algorithm A for h makes to the underlying function $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$ (the function that is actually being tested and for which we have black-box oracle access) in the course of its

Input: A Boolean function $f : \{\pm 1\}^{k'} \rightarrow \{\pm 1\}$ and $\varepsilon \in [0, 1/2]$

Output: An estimate γ of $\text{corr}(f, \mathcal{J}_k)$

QUANTUM-TOLERANT-JUNTA-TESTER:

1. Set $\gamma = 0$ and draw $\mathcal{S}_1, \dots, \mathcal{S}_{(k'/\varepsilon)^4}$ independently from the spectral sample of f .
2. Repeat $(1/\varepsilon)^{O(k^{1/3})}$ times:
 - (a) Sample $\mathbf{A}_1, \dots, \mathbf{A}_{k^{1/3}} \subseteq [k']$ independently from the spectral sample of f .
 - (b) Let \mathcal{C} be the union of \mathbf{A}_i 's and define

$$f^{\overline{\mathcal{C}}} := \text{SHARPNOISE}_{\ell, \kappa, \Delta}^{\overline{\mathcal{C}}} f.$$

- (c) Sample $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)} \sim \{\pm 1\}^{k'}$ uniformly and independently at random.
- (d) For each $\mathbf{x}^{(i)}$, draw a sample bundle $\mathcal{B}^{(i)} \sim \mathcal{D}_{\mathbf{x}^{(i)}, \mathcal{C}}$.
- (e) For each set $U \subseteq [k']$ of size k such that $\mathcal{C} \subseteq U$ and

$$\frac{|i : \mathcal{S}_i \subseteq U \text{ and } |\mathcal{S}_i \setminus \mathcal{C}| \geq \ell|}{(k'/\varepsilon)^4} \leq \frac{\varepsilon^2}{10}, \quad (28)$$

run the algorithm of [Lemma 39](#) using $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}, \mathcal{B}^{(1)}, \dots, \mathcal{B}^{(N)}$ to get an $(\varepsilon/2)$ -estimation $\mathbf{Est}_{\mathcal{C}, U}$ of $\text{corr}(f^{\overline{\mathcal{C}}}, \mathcal{J}_U)$; update $\gamma = \max(\gamma, \mathbf{Est}_{\mathcal{C}, U})$.

3. Return γ .

Algorithm 2: A Quantum Tolerant Junta Testing Algorithm

execution on a given input. In the context of this section, though, we will only need [Lemma 39](#) for $h = f$, the original $\{\pm 1\}$ -valued function that is being tested. In this case there is no need for the randomized algorithm A and the number of calls to f that are required is simply the number of points in the sample bundles $\mathcal{B}^{(1)}, \dots, \mathcal{B}^{(N)}$, i.e. at most $N \cdot \binom{k'}{\leq r}$.

5.6 Putting It Together: A Quantum Tester and the Proof of [Theorem 2](#)

Our quantum tolerant junta tester is given in [Algorithm 2](#).

5.6.1 Efficiency

Lemma 42 (Query complexity). [Algorithm 2](#) makes $\exp(k^{1/3} \text{polylog}(k'/\varepsilon))$ (quantum) queries.

Proof. We first account for the queries to f that are “actually quantum.” As discussed in [Section 5.1](#) and [Remark 4](#), these correspond to the draws from the spectral sample \mathcal{P}_f of f in lines 1 and 2(a); since each draw from \mathcal{P}_f requires one quantum query, by inspection of [Algorithm 2](#), it makes at most $(1/\varepsilon)^{O(k^{1/3})}$ oracle calls that are “actually quantum”.

We proceed to bound the number of classical queries to f made by [Algorithm 2](#). The only step of [Algorithm 2](#) that make classical oracle calls to f is 2(e), where points in the bundles $\mathcal{B}^{(1)}, \dots, \mathcal{B}^{(N)}$ are queried. As remarked before [Lemma 39](#), for each outcome of \mathcal{C} we reuse the points for all the

sets $U \subseteq [k']$ of size k that contain \mathcal{C} that are considered in line 2(e). So the total number of classical queries to f that are made, recalling [Remark 41](#), is

$$\left(\frac{1}{\varepsilon}\right)^{O(k^{1/3})} \cdot N \cdot \binom{k'}{\leq r} \cdot O(\kappa\Delta) = \exp\left(k^{1/3} \cdot \text{polylog}(k'/\varepsilon)\right). \quad \square$$

5.6.2 Correctness

We will refer to [Equation \(28\)](#) in the algorithm as a *test* on the pair of \mathcal{C} and U . We say \mathcal{C} and U *pass the test* if [Equation \(28\)](#) holds.

The two main results of this section are [Lemma 43](#) and [Lemma 46](#). We start with [Lemma 43](#), which states that with probability at least 0.99, every pair of \mathcal{C} and U that passes the test during the execution of [Algorithm 2](#) satisfies that $\text{corr}(f, \mathcal{J}_U)$ and $\text{corr}(f^{\overline{\mathcal{C}}}, \mathcal{J}_U)$ are close.

Lemma 43. With probability at least 99/100, we have the following: For every pair \mathcal{C} and U in step 2(e) that passes the [Equation \(28\)](#) test, it is the case that

$$|\mathbf{Est}_{\mathcal{C},U} - \text{corr}(f, \mathcal{J}_U)| \leq \varepsilon/2. \quad (29)$$

Proof. We start with a sufficient condition for the test to fail with high probability:

Claim 44. Let U be a k -subset of $[k']$ and $C \subseteq U$ be such that

$$\sum_{R \subseteq U: |R \setminus C| \geq \ell} \widehat{f}(R)^2 \geq \frac{\varepsilon^2}{5}.$$

Then the test fails with probability at least $1 - e^{-\Omega(k'^4/\varepsilon^2)}$.

Proof. This just follows from a standard multiplicative Chernoff bound and the fact that there are $(k'/\varepsilon)^4$ many \mathcal{S}_i 's. \square

Using the above claim, by a union bound we may assume that every pair of \mathcal{C} and U that passes the test satisfies

$$\sum_{R \subseteq U: |R \setminus C| \geq \ell} \widehat{f}(R)^2 \leq \frac{\varepsilon^2}{5} \quad (30)$$

(at the cost of a $o(1)$ failure probability). Fix any such pair C and U . Recall from [Equation \(5\)](#) that

$$\text{corr}(f, \mathcal{J}_U) = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\left| \sum_{R \subseteq U} \widehat{f}(R) \chi_R(\mathbf{x}) \right| \right] \quad \text{and} \quad \text{corr}(f^{\overline{\mathcal{C}}}, \mathcal{J}_U) = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\left| \sum_{R \subseteq U} \widehat{f}^{\overline{\mathcal{C}}}(R) \chi_R(\mathbf{x}) \right| \right]. \quad (31)$$

So we have

$$\begin{aligned}
\left| \text{corr}(f, \mathcal{J}_U) - \text{corr}(f^{\overline{C}}, \mathcal{J}_U) \right| &\leq \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\left| \sum_{R \subseteq U} \left(\widehat{f}(R) - \widehat{f}^{\overline{C}}(R) \right) \chi_R(\mathbf{x}) \right| \right] \\
&\leq \sqrt{\mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\left(\sum_{R \subseteq U} \left(\widehat{f}(R) - \widehat{f}^{\overline{C}}(R) \right) \chi_R(\mathbf{x}) \right)^2 \right]} \\
&= \sqrt{\sum_{R \subseteq U} \left(\widehat{f}(R) - \widehat{f}^{\overline{C}}(R) \right)^2}. \tag{32}
\end{aligned}$$

We divide the sum inside the square root into two parts:

1. $R \subseteq U$ with $|R \setminus C| = |R \cap \overline{C}| \leq \ell$: For this part we have from [Lemma 36](#) that

$$\sum_{R \subseteq U: |R \setminus C| \leq \ell} \left(\widehat{f}(R) - \widehat{f}^{\overline{C}}(R) \right)^2 \leq \sum_{R \subseteq U: |R \setminus C| \leq \ell} \widehat{f}(R)^2 \cdot (\Delta 2^{-\kappa})^2 \leq (\Delta 2^{-\kappa})^2 \leq \frac{\varepsilon^2}{20}$$

by our choices of κ and Δ .

2. $R \subseteq U$ with $|R \setminus C| > \ell$: For this part we have from [Lemma 36](#) and [Equation \(30\)](#) that

$$\sum_{R \subseteq U: |R \setminus C| > \ell} \left(\widehat{f}(R) - \widehat{f}^{\overline{C}}(R) \right)^2 \leq \sum_{R \subseteq U: |R \setminus C| > \ell} \widehat{f}(R)^2 \leq \frac{\varepsilon^2}{5}.$$

As a result, we have $|\text{corr}(f, \mathcal{J}_U) - \text{corr}(f^{\overline{C}}, \mathcal{J}_U)| \leq \varepsilon/2$.

By another union bound, using [Lemma 39](#) we may assume that (except with failure probability $o(1)$) for every C and U that pass the test, we have

$$\left| \mathbf{Est}_{C,U} - \text{corr}(f^{\overline{C}}, \mathcal{J}_U) \right| \leq \varepsilon/2.$$

As a result, we have $|\mathbf{Est}_{C,U} - \text{corr}(f, \mathcal{J}_U)| \leq \varepsilon$ by combining the two inequalities. \square

[Lemma 43](#) implies that with very high probability the algorithm never overestimates $\text{corr}(f, \mathcal{J}_k)$:

Corollary 45. With probability at least 99/100, the algorithm returns $\gamma \leq \text{corr}(f, \mathcal{J}_k) + \varepsilon$.

Proof. For every pair that passes the test, we have $\mathbf{Est}_{C,U} \leq \text{corr}(f, \mathcal{J}_U) + \varepsilon \leq \text{corr}(f, \mathcal{J}_k) + \varepsilon$. \square

Next we show that with high probability, the algorithm returns γ satisfying $\gamma \geq \text{corr}(f, \mathcal{J}_k) - \varepsilon$. The correctness of the algorithm follows by combining [Corollary 45](#) and [Lemma 46](#).

Lemma 46. With probability at least 98/100, the algorithm returns $\gamma \geq \text{corr}(f, \mathcal{J}_k) - \varepsilon$.

Proof. Let U^* be a k -subset that achieves the optimal correlation: $\text{corr}(f, \mathcal{J}_{U^*}) = \text{corr}(f, \mathcal{J}_k)$.

We will assume in the rest of the proof of the lemma that

$$\sum_{R \subseteq U^*} \widehat{f}(R)^2 \geq \varepsilon^2. \tag{33}$$

If this is not the case, then it is easy to see that $\text{corr}(f, \mathcal{J}_k) = \text{corr}(f, \mathcal{J}_{U^*}) \leq \varepsilon$, since by Equation (5) and Cauchy-Schwarz we have

$$\text{corr}(f, \mathcal{J}_{U^*}) = \mathbf{E}_{\mathbf{x} \sim \{\pm 1\}^{k'}} \left[\left| \sum_{R \subseteq U^*} \hat{f}(R) \chi_R(\mathbf{x}) \right| \right] \leq \sqrt{\sum_{R \subseteq U^*} \hat{f}(R)^2} \leq \varepsilon.$$

In this case the conclusion trivially holds given that the algorithm always returns a nonnegative number.

It suffices to show that with probability at least 99/100, one of the \mathbf{C} 's sampled satisfies $\mathbf{C} \subseteq U^*$ and passes the test together with U^* . If this is the case, then by a union bound with Lemma 43, with probability at least 98/100, this pair of \mathbf{C} and U^* not only passes the test but also leads to

$$\mathbf{Est}_{\mathbf{C}, U} \geq \text{corr}(f, \mathcal{J}_{U^*}) - \varepsilon \geq \text{corr}(f, \mathcal{J}_k) - \varepsilon.$$

For this purpose we consider the distribution of the spectral sample \mathcal{P}_f conditioned on the set that is drawn being a subset of U^* ; we denote this conditioned distribution by \mathcal{P}_f^* . (Note that \mathcal{P}_f^* is well defined given Equation (33).)

We prove the following claim:

Claim 47. Let $\mathbf{A}_1, \dots, \mathbf{A}_{k^{1/3}}$ be independent samples from \mathcal{P}_f^* , and \mathbf{C} be their union. Then

$$\Pr_{\mathbf{R} \sim \mathcal{P}_f^*} \left[|\mathbf{R} \setminus \mathbf{C}| \geq k^{2/3} \right] \leq \frac{\varepsilon^2}{20}. \quad (34)$$

with probability at least $\varepsilon^{O(k^{1/3})}$.

We prove Claim 47 at the end. Combining it with Equation (33), we have that \mathbf{C} satisfies the condition given in Equation (34) when $\mathbf{A}_1, \dots, \mathbf{A}_{k^{1/3}} \sim \mathcal{P}_f$ with probability at least

$$(\varepsilon^2)^{k^{1/3}} \cdot \varepsilon^{O(k^{1/3})} = \varepsilon^{O(k^{1/3})}.$$

For such a \mathbf{C} , the pair \mathbf{C}, U^* would pass the Equation (28) test with high probability. Since the number of repetitions of the outer loop of line 2 (the number of \mathbf{C} we draw) is $(1/\varepsilon)^{O(k^{1/3})}$, the lemma follows. \square

Proof of Claim 47. The proof is similar to the proof of Lemma 29.

We draw the sets $\mathbf{A}_1, \dots, \mathbf{A}_{k^{1/3}} \sim \mathcal{P}_f^*$ one by one and always write \mathbf{C} to denote the union of $\mathbf{A}_1, \dots, \mathbf{A}_i$ sampled so far after i rounds. Initially (before drawing \mathbf{A}_1) we set $\mathbf{C} = \emptyset$.

Initially we examine the probability of $\mathbf{R} \sim \mathcal{P}_f^*$ with $|\mathbf{R} \setminus \mathbf{C}| = |\mathbf{R}| \geq k^{2/3}$. If this probability is already less than $\varepsilon^2/20$, then we are done (this is because as we draw more \mathbf{A}_i 's, their union \mathbf{C} can only grow and this probability can only decrease). Assuming this probability is at least $\varepsilon^2/20$, then we pay a probability of $\varepsilon^2/20$ to draw such a sample \mathbf{A}_1 with $|\mathbf{A}_1| \geq k^{2/3}$, and \mathbf{C} grows in size by at least $k^{2/3}$ after the first round.

For the second round, we again consider the probability of $\mathbf{R} \sim \mathcal{P}_f^*$ satisfying $|\mathbf{R} \setminus \mathbf{C}| \geq k^{2/3}$. Again if this probability is at most $\varepsilon^2/20$, then we are done. Otherwise we pay a probability of $\varepsilon^2/20$ to draw a sample \mathbf{A}_2 such that $|\mathbf{A}_2 \setminus \mathbf{C}| \geq k^{2/3}$ and again, \mathbf{C} grows in size by at least $k^{2/3}$.

We repeat the above argument. This can repeat no more than $k^{1/3}$ times since U^* is of size k and thus $\mathbf{C} \subseteq U^*$ can only grow by $k^{2/3}$ for at most $k^{1/3}$ rounds. So within $k^{1/3}$ rounds, there must be a round after which the probability of $\mathbf{R} \sim \mathcal{P}_f^*$ with $|\mathbf{R} \setminus \mathbf{C}| = |\mathbf{R}| \geq k^{2/3}$ is at most $\varepsilon^2/20$. The total probability we pay is $(\varepsilon^2/20)^{k^{1/3}}$. \square

6 A $2^{\tilde{O}(k^{1/3})}$ Query Classical Tolerant Junta Tester

In this section we prove our main tolerant junta testing result, [Theorem 3](#). Given [Section 5](#), the main challenge is that in the classical setting, we no longer have access to draws from the spectral sample. The main challenge, as alluded in [Section 2.3](#), is that we need to handle the case where there is more than ε^2 mass above level $k^{2/3}$. To do this, we will use the notion of normalized influences and an extension of the machinery developed in [\[ITW21\]](#). Recall that a draw from the level- ℓ normalized influences corresponds to first drawing a set \mathbf{S} from the spectral sample of f conditioned on $|\mathbf{S}| \geq \ell$, and then sampling a uniform random set $\mathbf{T} \subseteq \mathbf{S}$ of size ℓ .

As sketched in [Section 2.3](#), we will use these normalized influences to design and analyze our key technical tool. This tool is a filtering procedure which, roughly speaking, takes as input a set $C \subseteq [k']$ of coordinates and outputs a collection \mathcal{C} of $2^{\tilde{O}(k^{1/3})}$ subsets of C such that one of the sets $C' \in \mathcal{C}$ has half as many “irrelevant” variables as C , but still contains all of the “relevant” variables that are likely to appear above level $k^{2/3}$. Using this, we can start with $C = [k']$ and use our filtering procedure to get an initial collection of subsets \mathcal{C} . We then run the filtering procedure on each element of \mathcal{C} (for each element of \mathcal{C} , this results in a collection of subsets), and take the union of those collections to create a new collection \mathcal{C}' . We repeat this (running the filtering procedure on each element of \mathcal{C}' , etc.) $O(\log(k'))$ times; after doing this, we end up with a list of $2^{\tilde{O}(k^{1/3})}$ elements (subsets of $[k']$) where with high probability one of them contains only relevant variables. We can then apply SHARPNOISE to all variables outside of this subset, just as we did in the quantum tester, and use our local estimators to determine the junta correlation.

Before embarking on the proof, we first note that we will assume throughout that $\varepsilon \geq 2^{-k^{0.001}}$. Note that this is essentially without loss of generality, since for smaller values of ε the $\text{polylog}(1/\varepsilon)$ factor in the exponent of [Theorem 3](#) is bigger than k (assuming a sufficiently large exponent). It then follows that the tolerant tester of [\[DMN19\]](#), which makes $2^k \cdot \text{poly}(k, 1/\varepsilon)$ queries, has the claimed query complexity. We will also assume throughout that k is at least some sufficiently large absolute constant (since otherwise we can use the [\[DMN19\]](#) tester).

6.1 Approximating Normalized Influences

The filtering procedure that we describe and analyze in [Section 6.2](#) crucially uses the normalized influences of a suitable function h , which is based on the function f that is being tolerantly tested. In [\[ITW21\]](#), Iyer et al. gave an algorithm that gives a modestly accurate estimate of the normalized influences; roughly speaking, their algorithm computes an approximation that is accurate to within a constant multiplicative error. This error in the [\[ITW21\]](#) estimate would cause some complications in our analysis. To bypass these complications, in this subsection we give a procedure that computes an *unbiased* estimator for the value of $\mathbf{NInf}_U[f]$, for a given input set $U \subseteq [k']$, to within any desired accuracy ε . We remark that while the [\[ITW21\]](#) approach to estimating $\mathbf{NInf}_U[f]$ is based on random restrictions, in contrast, our procedure takes a quite different approach and is based on random sampling. We carefully design a particular random variable, show that it has expected value exactly $\mathbf{NInf}_U[f]$ (see [Lemma 50](#)), and give an algorithm ([Algorithm 3](#)) which samples from that random variable and thus achieves an arbitrarily accurate approximation of $\mathbf{NInf}_U[f]$.

Input: A function $f : \{\pm 1\}^{k'} \rightarrow [-B, B]$, a positive real number B , a set $U \subseteq [k']$, an accuracy parameter ε , and a failure probability parameter δ

Output: An estimate for $\mathbf{NInf}_U[f]$

ESTIMATE-NINF:

1. $S \leftarrow \emptyset$
2. $g_S \leftarrow f$
3. For $u \in U$:
 - (a) $g_{S \cup \{u\}}(x) = g_S(x) - \frac{1}{2} \left(g_S(x) + g(x_S^{\oplus u}) \right)$
 - (b) $S \leftarrow S \cup \{u\}$
4. $g(x) = g_U(x) \cdot \chi_U(x)$
5. $M \leftarrow 1000(U!)^4 B^4 \cdot \frac{\log(1/\delta)}{\varepsilon^2}$
6. For $t \in [1, M]$:
 - (a) Sample $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{|U|} \in [0, 1]$ uniformly at random
 - (b) Sort the \mathbf{r}_i 's in decreasing order to get $\mathbf{y}_1 > \mathbf{y}_2 > \dots > \mathbf{y}_{|U|}$
 - (c) Set γ_t to be an estimate of $\mathbf{E}_{\mathbf{x}} \left[\left(T_{\sqrt{\mathbf{y}_{|U|}}} g(\mathbf{x}) \right)^2 \right]$ evaluated via [Lemma 14](#) with accuracy $\frac{\varepsilon}{4(|U|!)^2}$ and failure probability $\frac{\varepsilon}{100B^2(|U|!)^2}$.
7. Output $(|U|!)^2 \cdot \frac{1}{M} \sum_{t=1}^M \gamma_t$

Algorithm 3: An algorithm to approximate $\mathbf{NInf}_U[f]$.

The main lemma of this section is the following:

Lemma 48. Let B be a positive real number, $f : \{\pm 1\}^{k'} \rightarrow [-B, B]$, $\varepsilon \in [0, 1]$ and $\delta \in [0, 1/4]$. Then with probability at least $1 - \delta$, we have that

$$\left| \text{ESTIMATE-NINF}(f, B, U, \varepsilon, \delta) - \mathbf{NInf}_U[f] \right| \leq \varepsilon.$$

Moreover, $\text{ESTIMATE-NINF}(f, B, U, \varepsilon, \delta)$ makes at most $\text{poly}(|U|!, B, 1/\varepsilon, \log(1/\delta))$ calls to the randomized algorithm \mathcal{A} , where \mathcal{A} is a B -bounded randomized algorithm computing f .

We start by noting a few simple properties of g .

Lemma 49. For any $U \subseteq [k']$, the function g , as defined on line (4), satisfies

$$g(x) = \sum_{S: S \supseteq U} \hat{f}(S) \chi_{S \setminus U}(x).$$

Moreover, we have that

$$g(x) = \chi_U(x) \cdot \sum_{S: S \subseteq U} \alpha_S f(x^{\oplus S})$$

for some coefficients $(\alpha_S)_{S \subseteq U}$ which satisfy $\sum_{S \subseteq U} |\alpha_S| = 1$.

Proof. We will prove both claims by induction on $|U|$. To start, note that when $U = \emptyset$ we get that $g = f$, giving us the statement. We now assume that $|U| \geq 1$ and the statement is true for all sets of size $|U| - 1$. Fix some $u \in U$, and note that by the inductive hypothesis, $g_{U \setminus \{u\}} \cdot \chi_{U \setminus \{u\}}$ is bounded in absolute value by B and satisfies

$$g_{U \setminus \{u\}}(x) \cdot \chi_{U \setminus \{u\}}(x) = \sum_{S: S \supseteq U \setminus \{u\}} \widehat{f}(S) \chi_{S \setminus (U \setminus \{u\})}(x)$$

Multiplying both sides by $\chi_{U \setminus \{u\}}$ then gives

$$g_{U \setminus \{u\}}(x) = \sum_{S: S \supseteq U \setminus \{u\}} \widehat{f}(S) \chi_S(x).$$

We next note that

$$\begin{aligned} g_{U \setminus \{u\}}(x) + g_{U \setminus \{u\}}(x^{\oplus u}) &= \sum_{S: S \supseteq U \setminus \{u\}} \widehat{f}(S) \chi_S(x) + \sum_{S: S \supseteq U \setminus \{u\}} \widehat{f}(S) \chi_S(x^{\oplus u}) \\ &= 2 \sum_{S: S \supseteq U \setminus \{u\}, u \notin S} \widehat{f}(S) \chi_S(x). \end{aligned}$$

This implies that

$$g_U(x) = g_{U \setminus \{u\}}(x) - \frac{1}{2} \left(g_{U \setminus \{u\}}(x) + g_{U \setminus \{u\}}(x^{\oplus u}) \right) = \sum_{S: S \supseteq U} \widehat{f}(S) \chi_S(x)$$

which implies

$$g(x) = g_U(x) \cdot \chi_U(x) = \sum_{S: S \supseteq U} \widehat{f}(S) \chi_{S \setminus U}(x)$$

as desired.

For the ‘‘Moreover’’ claim, we note that for any x we have

$$g(x) \chi_U(x) = g_U(x) = \frac{1}{2} g_{U \setminus \{u\}}(x) - \frac{1}{2} g_{U \setminus \{u\}}(x^{\oplus u}).$$

By the inductive hypothesis there exist coefficients β_S for $g_{U \setminus \{u\}}$ such that $\sum_{S \subseteq U \setminus \{u\}} |\beta_S| = 1$. So we get that

$$g(x) \chi_U(x) = \frac{1}{2} \sum_{S: S \subseteq U \setminus \{u\}} \beta_S f(x^{\oplus S}) - \frac{1}{2} \sum_{S: S \subseteq U \setminus \{u\}} \beta_S f(x^{\oplus S \cup \{u\}})$$

Since the above coefficients have absolute values that sum to 1, we have completed the proof of the inductive step and the lemma. \square

Now we come to the key observation underlying our estimator:

Lemma 50.

$$\mathbf{E}_{\mathbf{y}_{|U|}} \left[\mathbf{E}_{\mathbf{x}} \left[\left(T \sqrt{y_{|U|}} g(\mathbf{x}) \right)^2 \right] \right] = \frac{1}{(|U|!)^2} \mathbf{NInf}_U[f].$$

Proof. We start by noting that the vector of sorted values $(\mathbf{y}_1, \dots, \mathbf{y}_{|U|})$ is distributed uniformly over the region $A := \{(a_1, \dots, a_{|U|}) \in [0, 1]^{|U|} : 1 \geq a_1 \geq a_2 \geq \dots \geq a_{|U|} \geq 0\}$ (see e.g. Lemma 3 of [BS80]). Thus, we have

$$\begin{aligned}
& \mathbf{E}_{\mathbf{y}_{|U|}} \left[\mathbf{E}_{\mathbf{x}} \left[\left(T_{\sqrt{\mathbf{y}_{|U|}}} g(\mathbf{x}) \right)^2 \right] \right] \\
&= \frac{1}{|U|!} \int_0^1 \int_0^{a_1} \int_0^{a_2} \dots \int_0^{a_{|U|-1}} \mathbf{E}_{\mathbf{x}} \left[\left(T_{\sqrt{a_{|U|}}} g(\mathbf{x}) \right)^2 \right] da_{|U|} \dots da_1 \\
&= \frac{1}{|U|!} \int_0^1 \dots \int_0^{a_{|U|-1}} \mathbf{E}_{\mathbf{x}} \left[\left(\sum_{S: S \supseteq U} \sqrt{a_{|U|}^{|S|-|U|}} \widehat{f}(S) \chi_{S \setminus U}(\mathbf{x}) \right)^2 \right] da_{|U|} \dots da_1 \\
&= \frac{1}{|U|!} \int_0^1 \dots \int_0^{a_{|U|-1}} \sum_{S: S \supseteq U} \left(a_{|U|}^{|S|-|U|} \widehat{f}(S)^2 \right) da_{|U|} \dots da_1 \\
&= \sum_{S: S \supseteq U} \frac{1}{|U|!} \widehat{f}^2(S) \int_0^1 \dots \int_0^{a_{|U|-1}} a_{|U|}^{|S|-|U|} da_{|U|} \dots da_1 \\
&= \sum_{S: S \supseteq U} \frac{1}{|U|!} \widehat{f}^2(S) \cdot \frac{1}{(|S| - |U| + 1)(|S| - |U| + 2) \dots |S|} \\
&= \sum_{S: S \supseteq U} \frac{1}{(|U|!)^2} \binom{|S|}{|U|}^{-1} \widehat{f}^2(S) \\
&= \frac{1}{(|U|!)^2} \mathbf{NInf}_U[f]
\end{aligned}$$

as desired. \square

It now simply remains to argue that we indeed approximate this expectation with high accuracy, while making not too many calls to \mathcal{A} . To do this, we first show that there are B -bounded randomized algorithms for g and $T_{\sqrt{\mathbf{y}_{|U|}}} g$:

Lemma 51. There are B -bounded randomized algorithms for g and for $T_{\sqrt{\mathbf{y}_{|U|}}} g$. Moreover both algorithms call \mathcal{A} at most $2^{|U|}$ times, where \mathcal{A} is the B -bounded randomized algorithm for f .

Proof. Note that by Lemma 49, we have that there are coefficients α_S with $\sum_S |\alpha_S| = 1$ such that

$$g(x) = \chi_U(x) \cdot \sum_{S: S \subseteq U} \alpha_S f(x^{\oplus S}).$$

Given an $x \in \{\pm 1\}^{k'}$, our randomized algorithm \mathcal{A}' for g simply outputs

$$\chi_U(x) \cdot \sum_{S: S \subseteq U} \alpha_S \mathcal{A}(x^{\oplus S}).$$

It is easy to see that this computes g in expectation, and that it makes at most $2^{|U|}$ queries to \mathcal{A} , as desired. Moreover, since $\sum_{S \subseteq U} |\alpha_S| = 1$, we see that \mathcal{A}' is also B -bounded.

Finally, since there is an algorithm \mathcal{A}' for g , by Lemma 15 we have that there is a B -bounded randomized algorithm for $T_{\sqrt{\mathbf{y}_{|U|}}} g$ that makes the same number of queries to \mathcal{A} . \square

Lemma 52. In any iteration $t \in [1, M]$ of the loop on line 6 of the algorithm, we have that $0 \leq \gamma_t \leq B^2$ and

$$\left| \mathbf{E}[\gamma_t] - \mathbf{E} \left[\left(T_{\sqrt{y_{|U|}}} g(\mathbf{x}) \right)^2 \right] \right| \leq \frac{\varepsilon}{2(|U|!)^2}.$$

Proof. Note that by [Lemma 14](#) and the fact that $T_{\sqrt{y_{|U|}}} g$ has a B -bounded randomized algorithm from [Lemma 51](#), we get that $0 \leq \gamma_t \leq B^2$.

For the claim on the expectation, fix any outcome of $y_{|U|}$ and let \mathcal{W} denote the algorithm from [Lemma 14](#) used on line 6(c) to compute γ_t , the estimate of $\mathbf{E}_{\mathbf{x}} \left[\left(T_{\sqrt{y_{|U|}}} g(\mathbf{x}) \right)^2 \right]$. Recalling the failure probability from line 6(c) and the fact that the estimate, like the true value of $\mathbf{E}_{\mathbf{x}} \left[\left(T_{\sqrt{y_{|U|}}} g(\mathbf{x}) \right)^2 \right]$, lies in $[0, B^2]$, by [Lemma 14](#) we get that our estimate γ_t satisfies

$$\mathbf{E}_{\mathcal{W}} \left[\left| \gamma_t - \mathbf{E}_{\mathbf{x}} \left[\left(T_{\sqrt{y_{|U|}}} g(\mathbf{x}) \right)^2 \right] \right| \right] \leq \frac{\varepsilon}{4(|U|!)^2} + \left(\frac{\varepsilon}{100B^2(|U|!)^2} \right) B^2 \leq \frac{\varepsilon}{2(|U|!)^2}.$$

We then note that using Jensen's inequality, we have

$$\left| \mathbf{E}_{\mathbf{y}_{U, \mathcal{W}}} [\gamma_t] - \mathbf{E}_{\mathbf{x}, \mathbf{y}_{|U|}} \left[\left(T_{\sqrt{y_{|U|}}} g(\mathbf{x}) \right)^2 \right] \right| \leq \mathbf{E}_{\mathbf{y}_U} \left[\left| \mathbf{E}_{\mathcal{W}} \left[\left| \gamma_t - \mathbf{E}_{\mathbf{x}} \left[\left(T_{\sqrt{y_{|U|}}} g(\mathbf{x}) \right)^2 \right] \right| \right] \right| \right] \leq \frac{\varepsilon}{2(|U|!)^2}$$

as desired. \square

We can now prove [Lemma 48](#):

Proof of Lemma 48. We begin by proving that the algorithm is correct. Indeed, note that by a Hoeffding bound, using the fact ([Lemma 52](#)) that each γ_t lies in $[0, B^2]$, we have that

$$\Pr \left[\left| \frac{1}{M} \sum_{t=1}^M \gamma_t - \mathbf{E}[\gamma_t] \right| \geq \frac{\varepsilon}{2(|U|!)^2} \right] \leq 2 \exp \left(\frac{-(M\varepsilon)^2}{2MB^4(|U|!)^4} \right) \leq \delta.$$

On the other hand, note that when $\left| \frac{1}{M} \sum_{t=1}^M \gamma_t - \mathbf{E}[\gamma_t] \right| \leq \frac{\varepsilon}{2(|U|!)^2}$, then

$$\left| (|U|!)^2 \cdot \frac{1}{M} \sum_{t=1}^M \gamma_t - (|U|!)^2 \cdot \mathbf{E}[\gamma_t] \right| \leq \frac{\varepsilon}{2}.$$

Applying [Lemma 50](#) and [Lemma 52](#) then yields

$$\left| (|U|!)^2 \cdot \frac{1}{M} \sum_{t=1}^M \gamma_t - \mathbf{NInf}_U[f] \right| \leq \varepsilon,$$

which completes the proof of correctness.

For the bound on the number of queries, note that in each iteration of the line 6 loop, the call to [Lemma 14](#) on line 6(c) requires $\text{poly}(1/\varepsilon, |U|, B)$ calls to the randomized algorithm, \mathcal{A}' , for $T_{\sqrt{y_{|U|}}} g$. By [Lemma 51](#) each call to \mathcal{A}' makes $2^{|U|}$ queries to \mathcal{A} . So all in all we make at most

$$M \cdot 2^{|U|} \cdot \text{poly}(1/\varepsilon, |U|, B) = \text{poly}(1/\varepsilon, |U|, B, \log(1/\delta))$$

calls to \mathcal{A} throughout the algorithm. \square

6.2 Filtering Coordinates

As mentioned above, the key tool for handling functions with mass on “high” levels is a filtering procedure which we call `REFINE-COORDINATES`. Intuitively, this procedure takes a set of coordinates $C \subseteq [k']$ as input and outputs a collection of subsets $C' \subseteq C$. (Actually, for each subset C' that it outputs, it also outputs an associated set I ; we will explain the role of these I sets later.) We will seek to show that for some C' returned by the algorithm, we have that $|C' \cap \overline{R}| \leq |C \cap \overline{R}|/2$, where $R \subseteq [k']$ is a set of size k . (It may be helpful to think of R as corresponding to the set U^* of k coordinates achieving the maximum junta correlation, but in our main lemma, [Lemma 53](#), R can be any k -set.)

Input: A function $f : \{\pm 1\}^{k'} \rightarrow \mathbb{R}$, a set of coordinates $C \subseteq [k']$, and a parameter $\varepsilon \in (0, 1)$

Output: A set \mathcal{F} of pairs (C', \mathbf{I}) with $C', \mathbf{I} \subseteq C$

`REFINE-COORDINATES`:

1. Set $\mathcal{F} \leftarrow \emptyset$ and let

$$\beta = \max\{|C|, k\}, \quad m = \frac{10\beta}{k^{2/3}} \log^4\left(\frac{k'}{\varepsilon}\right) \quad \text{and} \quad \gamma = k^{1/3} \log^3\left(\frac{k'}{\varepsilon}\right).$$

2. Repeat $\exp(O(k^{1/3} \log^5(k'/\varepsilon)))$ many times:

(a) Draw $\mathbf{i}_1, \dots, \mathbf{i}_m \sim C$ uniformly and independently, and set $\mathbf{I} = \{\mathbf{i}_1, \dots, \mathbf{i}_m\}$.

(b) Set $C' \leftarrow \emptyset$ and let

$$f_{\text{ave}}^{\mathbf{I}}(x) := \mathbf{E}_{\mathbf{y} \sim \{\pm 1\}^{k'}} \left[f(\mathbf{y}) \mid \mathbf{y}_{\mathbf{I}} = x_{\mathbf{I}} \right].$$

(c) While true do

- i. Let ℓ, κ and Δ be the following parameters:

$$\ell = k^{2/3} \log^3\left(\frac{k'}{\varepsilon}\right), \quad \kappa = 10 \log\left(\frac{k'}{\varepsilon}\right) \quad \text{and} \quad \Delta = 10.$$

- ii. Let h be the following function:

$$h := f_{\text{ave}}^{\mathbf{I}} - \text{SHARPNOISE}_{\ell, \kappa, \Delta}^{C \setminus C'} f_{\text{ave}}^{\mathbf{I}}.$$

iii. Compute an estimate $\hat{\sigma}^2$ of $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})]$ and exit the loop if $\hat{\sigma}^2 \leq \varepsilon^2/k^2$.

iv. Sample a set $\mathbf{T} \subseteq [k']$ among all sets $U \subseteq [k], |U| = \gamma$ with probability proportional to $\mathbf{NInf}_U[h]$.

v. Exit the loop if $\mathbf{T} \cap C \subseteq C'$.

vi. Otherwise, set $C' \leftarrow C' \cup (\mathbf{T} \cap C)$ and repeat.

(d) Add (C', \mathbf{I}) to \mathcal{F} .

3. Return \mathcal{F} .

Algorithm 4: An algorithm to refine coordinates.

The main lemma that we will want to prove is the following:

Lemma 53. Let $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$ be such that

$$\mathbf{W}^{\geq 100k \log(k'/\varepsilon)}[f] \leq \left(\frac{\varepsilon}{k'}\right)^5$$

and let $R \subset [k']$ be a set of size k . Given access to f via a 1-bounded randomized algorithm \mathcal{A} that computes it, and a set $C \subseteq [k']$ of coordinates such that $C \cap \bar{R} \neq \emptyset$, with probability at least $1 - 3/(k')^2$, REFINE-COORDINATES outputs a collection \mathcal{F} of at most $2^{k^{1/3} \cdot \text{polylog}(k'/\varepsilon)}$ pairs $(\mathbf{C}', \mathbf{I})$ with $\mathbf{C}', \mathbf{I} \subseteq C$ such that at least one pair satisfies

$$\mathbf{I} \subseteq \bar{R}, \quad |\mathbf{C}' \cap \bar{R}| \leq \frac{|C \cap \bar{R}|}{2}, \quad \text{and} \quad \sum_{\substack{S \subseteq [k'] \\ |S \cap C \setminus \mathbf{C}'| \geq \kappa \ell}} \widehat{f_{\text{ave}}^{\mathbf{I}}}(S)^2 \leq \frac{4\varepsilon^2}{k^2}. \quad (35)$$

The procedure REFINE-COORDINATES is given in [Algorithm 4](#). The reader may notice that details on how to compute an estimate $\widehat{\sigma}^2$ of $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})]$ (line 2(c)(iii)) and how to sample a set \mathbf{T} of size γ proportional to \mathbf{NInf} of h (line 2(c)(iv)) are missing in [Algorithm 4](#), and these are actually the only steps in REFINE-COORDINATES that are computation heavy. This is because the proof of [Lemma 53](#) is quite complex and for clarity, we will start with a proof under an idealized setting in [Section 6.2.1](#). Afterwards, we will explain details behind 2(c)(iii) and 2(c)(iv) and lift the proof in the idealized setting to the real setting in [Section 6.2.2](#).

6.2.1 The Idealized Setting

In the idealized setting, we prove [Lemma 53](#) under the following assumption:

Assumption 54. We make the following two assumptions regarding 2(c)(iii) and 2(c)(iv):

1. Whenever line 2(c)(iii) is reached, if $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})] \leq \varepsilon^2/(2k^2)$ then the algorithm always exits the loop; if $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})] \geq 2\varepsilon^2/k^2$, then the algorithm always continues in the loop.
2. Whenever line 2(c)(iv) is reached, a set \mathbf{T} is drawn with probability proportional to $\mathbf{NInf}_U[h]$ among all sets $U \subseteq [k']$ of size γ . An equivalent description of the distribution of \mathbf{T} is first drawing a set \mathbf{S} of size at least γ from the spectral sample of h and then drawing a uniform subset \mathbf{T} of \mathbf{S} of size γ . (Of course this is only well defined if h has mass at or above level γ , but as will become clear in the proof, this will always be the case given the first assumption.)

(In fact, under [Assumption 54](#) we will prove [Lemma 53](#) with a slightly stronger success probability of $1 - 1/(k')^2$; later we will use this stronger success probability to prove the actual [Lemma 53](#) with its $1 - 3/(k')^2$ success probability.)

We now gather several lemmas on the behavior of \mathbf{T} that will be useful for us. We begin by noting that the purpose of guessing \mathbf{I} is to get a “random looking” set of “irrelevant” coordinates in C , where we think of coordinates in R as relevant and coordinates outside R as irrelevant. Towards this, we show that we indeed expect to sample a set \mathbf{I} that is comprised of irrelevant variables with non-negligible probability.

Lemma 55. Let $R \subseteq [k']$ be a set of size at most k , and $C \subseteq [k']$ be such that $C \cap \bar{R} \neq \emptyset$. Then

$$\Pr_{\mathbf{I}}[\mathbf{I} \cap R = \emptyset] \geq \exp\left(-O\left(k^{1/3} \log^5(k'/\varepsilon)\right)\right).$$

Proof. Note that

$$\Pr_{\mathbf{I}}[\mathbf{I} \cap R = \emptyset] = \Pr_{\mathbf{I}}[\mathbf{I} \subseteq C \cap \overline{R}] = \left(1 - \frac{|R \cap C|}{|C|}\right)^m.$$

If $|C| \geq 2k$, then we have $\beta = |C|$ and $m = (10|C|/k^{2/3}) \log^4(k'/\varepsilon)$ and thus,

$$\Pr_{\mathbf{I}}[\mathbf{I} \subseteq C \cap \overline{R}] \geq \left(1 - \frac{k}{|C|}\right)^m \geq \exp\left(-\frac{2km}{|C|}\right) = \exp\left(-O\left(k^{1/3} \log^4(k'/\varepsilon)\right)\right),$$

where the inequality follows from $1 - x \geq e^{-2x}$ for $0 \leq x \leq 1/2$.

On the other hand, if $|C| \leq 2k$, then we have $\beta \leq 2k$ and $m = O(k^{1/3} \log^4(k'/\varepsilon))$ and thus,

$$\Pr_{\mathbf{I}}[\mathbf{I} \subseteq C \cap \overline{R}] \geq \left(1 - \frac{|C| - 1}{|C|}\right)^m \geq \exp\left(-m \log(|C|)\right) \geq \exp\left(-O\left(k^{1/3} \log^5(k'/\varepsilon)\right)\right),$$

where the first inequality uses the fact that $C \cap \overline{R} \neq \emptyset$. \square

Next we show that if \mathbf{I} were drawn from $C \cap \overline{R}$ rather than from C , then we would expect the Fourier mass of $f_{\text{ave}}^{\mathbf{I}}$ on terms with many irrelevant variables (i.e. variables in $C \cap \overline{R}$) to be heavily dampened. Let

$$\delta = \frac{k^{2/3}}{\beta \log^3(k')}$$

(this will be the last parameter needed in this subsection).

Lemma 56. Let $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$, $R \subseteq [k']$ be a subset of size k , and $C \subseteq [k']$ be such that $C \cap \overline{R} \neq \emptyset$. Let \mathbf{I} be a set of m coordinates drawn independently from $C \cap \overline{R}$. Then with probability at least 0.8, we have

$$\sum_{\substack{S \subseteq [k'] \\ |S \cap C \cap \overline{R}| \geq \delta |C \cap \overline{R}|}} \widehat{f_{\text{ave}}^{\mathbf{I}}}(S)^2 \leq \left(\frac{\varepsilon}{k'}\right)^5. \quad (36)$$

Proof. Fix a set $S \subseteq [k']$ such that $|S \cap C \cap \overline{R}| \geq \delta |C \cap \overline{R}|$. We have that

$$\Pr_{\mathbf{I}}[S \cap \mathbf{I} = \emptyset] = \left(1 - \frac{|S \cap C \cap \overline{R}|}{|C \cap \overline{R}|}\right)^m \leq \frac{1}{10} \cdot \left(\frac{\varepsilon}{k'}\right)^5,$$

by our choices of δ and m . Recalling from Equation (4) that $f_{\text{ave}}^{\mathbf{I}}(x) = \sum_{S: S \subseteq \mathbf{I}} \widehat{f}(S) \chi_S(x)$, we get that

$$\mathbf{E}_{\mathbf{I}} \left[\sum_{\substack{S \subseteq [k'] \\ |S \cap C \cap \overline{R}| \geq \delta |C \cap \overline{R}|}} \widehat{f_{\text{ave}}^{\mathbf{I}}}(S)^2 \right] \leq \frac{1}{10} \cdot \left(\frac{\varepsilon}{k'}\right)^5 \sum_{\substack{S \subseteq [k'] \\ |S \cap C \cap \overline{R}| \geq \delta |C \cap \overline{R}|}} \widehat{f}^2(S) \leq \frac{1}{10} \cdot \left(\frac{\varepsilon}{k'}\right)^5,$$

where the second inequality is by using $\mathbf{E}_{\mathbf{x}}[f^2(\mathbf{x})] \leq 1$. By Markov's inequality, Equation (36) occurs with probability at least 0.9. \square

Given the number of times line 2 is repeated, we have from Lemma 55 and Lemma 56:

Corollary 57. With probability at least $1 - 1/(2(k')^2)$, at least $10 \log(k')$ rounds of \mathbf{I} that are sampled in line 2(a) satisfy Equation (36).

We fix such a set I (satisfying Equation (36)) in the rest of the proof, and show that for the iteration with this I , the pair (C', I) that is returned satisfies the desired properties in Equation (35) with probability at least $1/2$. Let \mathcal{D}_h be the spectral sample of h conditioning on sets being of size at least γ (see Lemma 59 below which implies that \mathcal{D}_h is well defined). Recall that an equivalent view of step 2(c)(iv) of the algorithm, for the purpose of analysis, is that a set S is drawn from \mathcal{D}_h and then a uniform γ -subset T is drawn from S . Our subsequent analysis will heavily use this view.

To understand T , we prove a few lemmas to show that most likely S is *good*, as defined below:

Definition 58 (Good Sets). We say a set $S \subseteq [k']$ is *good* if it satisfies the following conditions:

- (i) $|S \cap R \cap C \setminus C'| \geq \ell/2$;
- (ii) $|S \cap C \cap \overline{R}| \leq \delta |C \cap \overline{R}|$; and
- (iii) $|S| \leq 100k \log(k'/\varepsilon)$.

In words, its size must be at most roughly k ; its intersection with $C \setminus R$ must be a small fraction of $C \setminus R$; and the part of its intersection with $C \cap R$ that lies outside C' must not be too small.

Lemma 59. If $\mathbf{E}_x[h^2(\mathbf{x})] \geq \varepsilon^2/(2k^2)$ and $\mathbf{E}_x[f^2(\mathbf{x})] \leq 1$, then we have

$$\mathbf{W}^{\geq \gamma}[h] \geq \frac{\varepsilon^2}{4k^2}.$$

Proof. By the properties of SHARPNOISE (Lemma 36), we have that

$$\mathbf{W}^{< \gamma}[h] \leq \mathbf{W}^{\leq \ell}[h] \leq (\Delta 2^{-\kappa})^2 \cdot \mathbf{E}_x[f^2(\mathbf{x})] \leq \varepsilon^2/(4k^2)$$

by our choices of Δ and κ . The lemma follows from $\varepsilon^2/(2k^2) \leq \mathbf{E}_x[h^2(\mathbf{x})] = \mathbf{W}^{< \gamma}[h] + \mathbf{W}^{\geq \gamma}[h]$. \square

We are now ready to show that most likely $S \sim \mathcal{D}_h$ is good:

Lemma 60. Suppose that f and h satisfy

$$\begin{aligned} \mathbf{E}_x[f^2(\mathbf{x})] &\leq 1, \\ \mathbf{W}^{\geq 100k \log(k'/\varepsilon)}[f] &\leq (\varepsilon/k')^5 \text{ (as in the hypothesis of Lemma 53),} \end{aligned} \tag{37}$$

$$\mathbf{W}^{\geq \gamma}[h] \geq \varepsilon^2/(4k^2) \text{ (as given by Lemma 59),} \tag{38}$$

and Equation (36). Then we have

$$\Pr_{S \sim \mathcal{D}_h} [S \text{ good}] \geq 1 - \frac{1}{(k')^2}.$$

Proof. By Lemma 36, we have that for any $S \subseteq [k']$

$$\widehat{h}^2(S) = \left(\widehat{f_{\text{ave}}^{\mathbf{I}}}(S) - \overline{\text{SHARPNOISE}_{\ell, \kappa, \Delta}^{C \setminus C'} f_{\text{ave}}^{\mathbf{I}}(S)} \right)^2 \leq \widehat{f_{\text{ave}}^{\mathbf{I}}}^2(S),$$

so it follows from Equation (36) that

$$\sum_{\substack{S \subseteq [k'] \\ |S \cap C \cap \overline{R}| \geq \delta |C \cap \overline{R}|}} \widehat{h}^2(S) \leq \left(\frac{\varepsilon}{k'} \right)^5.$$

It then follows [Equation \(38\)](#) that

$$\Pr_{\mathcal{S} \sim \mathcal{D}_h} \left[|\mathcal{S} \cap C \cap \overline{R}| \geq \delta |C \cap \overline{R}| \right] \leq \frac{(\varepsilon/k')^5}{\mathbf{W}^{\geq \gamma}[h]} \leq \frac{1}{3(k')^2},$$

so part (ii) of \mathcal{S} being good holds with high probability. We now consider part (i); towards this end, we will argue that it is unlikely that $|\mathcal{S} \cap C \setminus \mathcal{C}'| \leq \ell$. Indeed, note that by [Lemma 36](#)

$$\sum_{S: |\mathcal{S} \cap C \setminus \mathcal{C}'| \leq \ell} \widehat{h}^2(S) \leq (\Delta 2^{-\kappa})^2 \sum_{S: |\mathcal{S} \cap C \setminus \mathcal{C}'| \leq \ell} \widehat{f_{\text{ave}}^{\mathbf{I}}}^2(S) \leq (\Delta 2^{-\kappa})^2 \cdot \mathbf{E}_{\mathbf{x}}[f^2(\mathbf{x})] \leq \left(\frac{\varepsilon}{k'}\right)^5,$$

by our choices of Δ and κ . Similar to the earlier argument, this implies that

$$\Pr_{\mathcal{S} \sim \mathcal{D}_h} [|\mathcal{S} \cap C \setminus \mathcal{C}'| \leq \ell] \leq \frac{1}{3(k')^2},$$

and when $|\mathcal{S} \cap C \setminus \mathcal{C}'| \geq \ell$ and $|\mathcal{S} \cap C \cap \overline{R}| \leq \delta |C \cap \overline{R}|$, we have

$$|\mathcal{S} \cap C \cap R \setminus \mathcal{C}'| \geq \ell - \delta |C \cap \overline{R}| \geq \ell - \frac{k^{2/3}}{|C| \log^3(k')} \cdot |C \cap \overline{R}| \geq \frac{\ell}{2},$$

giving part (ii). Finally, for part (iii), again by [Lemma 36](#) we have

$$\sum_{\substack{S \subseteq [k'] \\ |S| \geq 100k \log(k'/\varepsilon)}} \widehat{h}^2(S) \leq \mathbf{W}^{\geq 100k \log(k'/\varepsilon)}[f] \leq \left(\frac{\varepsilon}{k'}\right)^5$$

(by [Equation \(37\)](#)). As in the two preceding arguments, this means that

$$\Pr_{\mathcal{S} \sim \mathcal{D}_h} [|\mathcal{S}| \geq 100k \log(k'/\varepsilon)] \leq \frac{(\varepsilon/k')^5}{\mathbf{W}^{\geq \gamma}[h]} \leq \frac{1}{3(k')^2}.$$

The lemma then follows by a union bound over (i), (ii) and (iii). \square

The following lemma makes it easier to consider the random process of growing \mathcal{C}' :

Lemma 61. Under the assumptions of [Lemma 60](#), with probability at least 0.9, every $\mathcal{S} \sim \mathcal{D}_h$ is a good set and when the loop of step 2(c) exits, it does so in step (iii) rather than step (v).

Proof. There are at most $k' + 1$ many rounds as line 2(c)(vi) always adds an element to \mathcal{C}' and there are at most k' elements. So, by a union bound over [Lemma 60](#), every \mathcal{S} sampled is good. Assuming \mathcal{S} is good, $|\mathcal{S} \cap C \setminus \mathcal{C}'| \geq \ell/2$ and thus, the probability of having $\mathcal{T} \cap (C \setminus \mathcal{C}') = \emptyset$ is at most

$$\left(1 - \frac{\ell/2}{|\mathcal{S}|}\right)^\gamma \leq \left(1 - \frac{\ell/2}{100k \log(k'/\varepsilon)}\right)^\gamma \leq \frac{1}{(k')^2}$$

and the lemma follows from a union bound. \square

Assuming the lemma above holds, the algorithm must exit the loop because $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})] \leq 2\varepsilon^2/k^2$ for the function h during that round. With this and [Lemma 36](#), we have

$$\frac{2\varepsilon^2}{k^2} \geq \mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})] \geq (1 - 2^{-\Delta})^2 \sum_{S: |\mathcal{S} \cap C \setminus \mathcal{C}'| \geq \kappa \ell} \widehat{f_{\text{ave}}^{\mathbf{I}}}^2(S) \geq \frac{1}{2} \sum_{S: |\mathcal{S} \cap C \setminus \mathcal{C}'| \geq \kappa \ell} \widehat{f_{\text{ave}}^{\mathbf{I}}}^2(S). \quad (39)$$

Assuming the condition of [Lemma 61](#), we finally work on how \mathbf{C}' evolves round by round and show that with probability at least $1/2$, the final \mathbf{C}' satisfies

$$|\mathbf{C}' \cap \overline{\mathbf{R}}| \leq |C \cap \overline{\mathbf{R}}|/2.$$

To this end we consider the following random process that captures how \mathbf{C}' , starting with $\mathbf{C}' = \emptyset$, evolves. The process takes at least one round and at most k' rounds to end. At the beginning of the i -th round, we receive three random numbers (which may depend on what happens in previous rounds) t_i, α_i and β_i such that

$$1 \leq t_i \leq \alpha_i + \beta_i, \quad \alpha_i \geq \ell/2 \quad \text{and} \quad \beta_i \leq \delta |C \cap \overline{\mathbf{R}}| \ll \ell.$$

(They corresponds to $|\mathbf{T} \cap C \setminus \mathbf{C}'|$, $|\mathbf{S} \cap R \cap C \setminus \mathbf{C}'|$ and $|\mathbf{S} \cap \overline{\mathbf{R}} \cap C \setminus \mathbf{C}'|$, respectively. The last two inequalities follow from the assumption that \mathbf{S} is good; the first inequality follows from $\mathbf{T} \subseteq \mathbf{S}$ and the assumption that $\mathbf{T} \cap C \setminus \mathbf{C}' \neq \emptyset$.)

Given t_i, α_i and β_i , we then draw \mathbf{X}_i , which is a random variable that corresponds to the number of blue balls we get if we draw t_i balls from a pool of α_i red balls and β_i blue balls without replacement. So $\mathbf{E}[\mathbf{X}_i] = t_i \cdot \beta_i / (\alpha_i + \beta_i)$. (This corresponds to the fact that, conditioning on the value of $|\mathbf{T} \cap C \setminus \mathbf{C}'|$, $|\mathbf{T} \cap \overline{\mathbf{R}} \cap C \setminus \mathbf{C}'|$ is distributed exactly the same as \mathbf{X}_i .) If the process ended before round i , then we just set $t_i = \mathbf{X}_i = 0$.

Our goal, then, is to show that with probability at least $1/2$, we have

$$\sum_{i \in [k']} \mathbf{X}_i \leq \frac{|C \cap \overline{\mathbf{R}}|}{2|C|} \cdot \sum_{i \in [k']} t_i. \quad (40)$$

(Note that $\sum_i \mathbf{X}_i$ corresponds to $|\mathbf{C}' \cap \overline{\mathbf{R}}|$ at the end, and $\sum_i t_i$ corresponds to $|\mathbf{C}'|$ at the end. Using $\mathbf{C}' \subseteq C$ and $|\mathbf{C}'| \leq |C|$, it follows from [Equation \(40\)](#) that

$$|\mathbf{C}' \cap \overline{\mathbf{R}}| \leq |C \cap \overline{\mathbf{R}}|/2.$$

So it suffices to prove that [Equation \(40\)](#) occurs with probability at least $1/2$.)

To this end, we let $\tau_i = \mathbf{E}[\mathbf{X}_i]$ for each i so $\tau_i = 0$ if the process ended before the i -th round and $\tau_i = t_i \cdot \beta_i / (\alpha_i + \beta_i)$ otherwise. The following claim is straightforward:

Claim 62. With probability at least 0.9 , we have $\mathbf{X}_i \leq O(\log(k')) \cdot \max(1, \tau_i)$ for every $i \in [k']$.

Proof. Fix an $i \in [k']$. If $t_i \geq (\alpha_i + \beta_i)/2$, then the bound is trivial because $\tau_i \geq \beta_i/2$.

Otherwise, \mathbf{X}_i is dominated by the sum of t_i indicator random variables, each taking 1 with probability at most $2\beta_i / (\alpha_i + \beta_i) \ll 1$ given that $\beta_i \ll \alpha_i$. If $\tau_i \geq 1/4$, the claim follows from a Chernoff bound. Otherwise, the probability of \mathbf{X}_i taking some value $a \geq 10 \log(k')$ is at most

$$(t_i)^a \cdot \left(\frac{2\beta_i}{\alpha_i + \beta_i} \right)^a \leq (1/2)^a$$

and summing over a shows that the probability of $\mathbf{X}_i \geq O(\log k')$ is at most $1/(k')^2$. The claim then follows by a union bound over the k' rounds. \square

Assume the condition of the above claim holds. Let $\mathbf{Z}_i = 1$ if $\tau_i \leq 1/2$ and $\mathbf{X}_i \geq 1$, and $\mathbf{Z}_i = 0$ otherwise. Then we have

$$\mathbf{X}_i \leq O(\log(k')) \cdot (\mathbf{Z}_i + \tau_i)$$

On the other hand, \mathbf{Z}_i is a sequence of indicator random variables, each taking 1 with probability at most $\tau_i \leq 1/2$ by Markov (or 0 when $\tau_i > 1/2$). (Of course it is still the case that τ_i is a random number that depends on previous rounds.) We prove the following claim:

Claim 63. With probability at least 0.9, we have

$$\sum_{i \in [k']} \mathbf{Z}_i \leq O(\log^2 k') \cdot \sum_{i \in [k']} \tau_i. \quad (41)$$

Proof. Consider the random process where in each of the k' rounds, an adversary picks a number $\tau_i \leq 1/2$ and then \mathbf{Z}_i is set to be 1 with probability τ_i and 0 with probability $1 - \tau_i$, and we would like to prove that Equation (41) holds with probability at least 0.9. We can further assume without loss of generality that every nonzero τ_i is 2^{-j} for some $j \geq 1$; to see this we just round τ_i up to the closest 2^{-j} but only charge 2^{-j-1} on the RHS of Equation (41). Also note from the definition of $\tau_i = \mathbf{t}_i \beta_i / (\alpha_i + \beta_i)$ that every nonzero τ_i is at least $1/k'$.

Let \mathbf{i}^* be the random variable that denotes the smallest \mathbf{i}^* such that $\sum_{i \leq \mathbf{i}^*} \tau_i \geq 0.01$, and set $\mathbf{i}^* = k' + 1$ if no such \mathbf{i}^* exists. Let E be the event such that $\sum_{i < \mathbf{i}^*} \mathbf{Z}_i = 0$. Then we have

$$\Pr[E] = (1 - \tau_1) \cdots (1 - \tau_{\mathbf{i}^*-1}) \geq \exp\left(-2 \sum_{i < \mathbf{i}^*} \tau_i\right) \geq e^{-0.02} \geq 0.98,$$

where we used $1 - x \geq e^{-2x}$ for $0 \leq x \leq 1/2$. It then suffices to show that the probability of

$$\sum_{i \geq \mathbf{i}^*} \mathbf{Z}_i \leq O(\log^2 k') \cdot \max\left(1, \sum_{i \geq \mathbf{i}^*} \tau_i\right)$$

is at least 0.98. Note this is the same claim we would like to prove, except that we now get to add a max with 1 on the RHS (and raising the probability from 0.9 to 0.98). For convenience, we will still use the same notation and show that

$$\sum_{i \in [k']} \mathbf{Z}_i \leq O(\log^2 k') \cdot \max\left(1, \sum_{i \in [k']} \tau_i\right). \quad (42)$$

with probability at least 0.98 in the rest of the proof.

To this end, we consider the following equivalent setting:

1. For each $j \leq \log(k')$, start by drawing a sequence of k' random bits $\mathbf{Z}_{j,1}, \dots, \mathbf{Z}_{j,k'} \in \{0, 1\}$, each set to 1 with probability 2^{-j} independently. The adversary does not get to see them.
2. In each of the k' rounds, the adversary gets to pick a j , set τ to 2^{-j} and \mathbf{Z} to the next unused bit in the sequence $\mathbf{Z}_{j,1}, \dots, \mathbf{Z}_{j,k'}$.

The following event would imply Equation (42): For every j and every $\ell \leq k'$, we have

$$\sum_{i' \in [\ell]} \mathbf{Z}_{j,i'} \leq O(\log k') \cdot \max\left(1, \frac{\ell}{2^j}\right).$$

It now follows from a standard Chernoff bound and union bound that the above event occurs with probability at least 0.98. \square

As a result of Equation (41), we have that

$$\sum_{i \in [k']} \mathbf{X}_i \leq O(\log^3 k') \cdot \sum_{i \in [k']} \tau_i = O(\log^3 k') \sum_{i \in [k']} \frac{\beta_i}{\alpha_i + \beta_i} \cdot \mathbf{t}_i.$$

But for every i , we have by the choice of δ that

$$\frac{\beta_i}{\alpha_i + \beta_i} \leq \frac{\delta|C \cap \overline{R}|}{\ell/2}.$$

Using $(2\delta/\ell) \cdot O(\log^3(k')) \leq 1/(2|C|)$ by our choices of δ and ℓ , it follows that Equation (40) holds with probability at least 0.8.

We briefly summarize how the results established above in this subsection give a proof of Lemma 53 in the idealized setting.

Proof of Lemma 53 under Assumption 54. It is clear that the collection \mathcal{F} of (C', \mathbf{I}) pairs has size at most $2^{\tilde{O}(k^{1/3})}$ and that each $C', \mathbf{I} \subseteq C$. By Corollary 57, except with failure probability at most $1/(2(k')^2)$ there are at least $10 \log k'$ rounds in which the \mathbf{I} sampled in line 2(a) lies entirely in \overline{R} . For each such \mathbf{I} we can apply Lemma 61 and Equation (39) to get that with probability at least 0.9, both

- the second inequality of Equation (35) holds, and
- Equation (40) holds with probability at least 0.8 (in which case we get $|C' \cap \overline{R}| \leq |C \cap \overline{R}|/2$, the first inequality of Equation (35)).

So assuming that there were at least $10 \log k'$ rounds in which $\mathbf{I} \subseteq \overline{R}$, the probability that none of those rounds yields a C' satisfying Equation (35) is at most $(1 - 0.9 \cdot 0.8)^{10 \log k'} < 1/(2(k')^2)$. The lemma follows by a union bound. \square

6.2.2 Handling the Approximations

We now turn to address the fact that we only have access to f via a randomized algorithm \mathcal{A} . To start, we begin by observing that we have a randomized algorithm that computes h in Line 2(c)(ii).

Lemma 64. In each iteration of the loop on line 2(c) in REFINE-COORDINATES, there is a $(2^{2\kappa\Delta} + 1)$ -bounded randomized algorithm \mathcal{A}'' for computing h that makes $O(\kappa\Delta)$ queries to the randomized algorithm \mathcal{A} for f .

Proof. Recall that we have that by Lemma 36

$$\text{SHARPNOISE}_{\ell, \kappa, \Delta}^{C \setminus C'} f_{\text{ave}}^{\mathbf{I}} = \sum_{i=0}^{\kappa\Delta} \alpha_i \mathbb{T}_{\rho^i}^{C \setminus C} f_{\text{ave}}^{\mathbf{I}}$$

for $\rho := 1 - \frac{1}{2\ell}$ and α_i satisfying

$$\sum_{i=0}^{\kappa\Delta} |\alpha_i| \leq 2^{2\kappa\Delta}.$$

We now design our $(2^{2\kappa\Delta} + 1)$ -bounded randomized algorithm \mathcal{A}'' for h as follows: Given x , output

$$\mathcal{A}'(x) = \sum_{i=0}^{\kappa\Delta} \alpha_i \mathcal{N}_{\rho^i}^{C \setminus C'}(x)$$

where \mathcal{A}' is the 1-bounded randomized algorithm for $f_{\text{ave}}^{\mathbf{I}}$ from [Lemma 16](#) and $\mathcal{N}_{\rho^i}^{C \setminus C'}$ is the 1-bounded randomized algorithm for computing $\mathbb{T}_{\rho^i}^{C \setminus C'} f_{\text{ave}}^{\mathbf{I}}$ given by [Lemma 15](#). To see that the mean is correct, note that for any $x \in \{\pm 1\}^{k'}$ we have

$$\mathbf{E}[\mathcal{A}''(x)] = \mathbf{E} \left[\mathcal{A}'(x) - \sum_{i=1}^{\kappa\Delta} \alpha_i \mathcal{N}_{\rho^i}^{C \setminus C'}(x) \right] = f_{\text{ave}}^{\mathbf{I}}(x) - \sum_{i=1}^{\kappa\Delta} \alpha_i \mathbb{T}_{\rho^i}^{C \setminus C'} f_{\text{ave}}^{\mathbf{I}}(x) = h(x)$$

as desired. We further note that boundedness follows from our bound on the sum of the absolute values of α_i . For the query bound, note that each query to \mathcal{A}' and $\mathcal{N}_{\rho^i}^{C \setminus C'}$ requires one query to \mathcal{A} , so we make at most $1 + \kappa\Delta$ queries. \square

With [Lemma 64](#) in hand, we can now describe our full implementation of REFINE-COORDINATES. In particular, to implement 2(c)(iii) we use the algorithm from [Lemma 14](#) with the bounded randomized algorithm \mathcal{A}'' given by [Lemma 64](#) to compute an estimate that has accuracy $\pm \frac{\varepsilon^2}{10k^2}$ with failure probability at most $2^{-(k')^2}$. In line 2(c)(iv), for every set $U \subseteq [k']$ of size γ , we compute an estimate of $\mathbf{NInf}_U[h]$, which we denote λ_U , by calling

$$\text{ESTIMATE-NINF} \left(h, 2^{2\kappa\Delta} + 1, U, \left(\frac{\varepsilon}{k'} \right)^{10} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1} \cdot \left(\frac{k'}{\gamma} \right)^{-2}, 2^{-(k')^2} \cdot \left(\frac{k'}{\gamma} \right)^{-1} \right) \quad (43)$$

where the reader should think of the $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ term above as corresponding to the number of times we repeat the loop on line 2 in REFINE-COORDINATES. We then sample $\tilde{\mathbf{T}} = U$ with probability proportional to λ_U . (Throughout this section, we will write $\tilde{\mathbf{T}}$ to denote sets drawn from the λ_U 's and \mathbf{T} denote the original sets drawn according to the true normalized influences.)

We begin our analysis with a very useful lemma which shows that in any iteration of the loop on Line 2(c) of REFINE-COORDINATES, the distributions of $\tilde{\mathbf{T}}$ and \mathbf{T} are extremely close to each other:

Lemma 65. In any iteration of the loop on Line 2(d) of REFINE-COORDINATES, the total variation distance between $\tilde{\mathbf{T}}$ and \mathbf{T} is at most $\left(\frac{\varepsilon}{k'} \right)^5 \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1}$.

Proof. Note that it suffices to prove that for any $U \in \binom{[k']}{\gamma}$ we have that

$$\Pr[\tilde{\mathbf{T}} = U] \geq \Pr[\mathbf{T} = U] - \left(\frac{\varepsilon}{k'} \right)^5 \cdot \left(\frac{k'}{\gamma} \right)^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1}$$

and

$$\Pr[\tilde{\mathbf{T}} = U] \leq \Pr[\mathbf{T} = U] + \left(\frac{\varepsilon}{k'} \right)^5 \cdot \left(\frac{k'}{\gamma} \right)^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1}.$$

To show this, we condition on the compound event that (i) for all $U \in \binom{[k']}{\gamma}$ we have

$$|\lambda_U - \mathbf{NInf}_U[h]| \leq \left(\frac{\varepsilon}{k'} \right)^{10} \left(\frac{k'}{\gamma} \right)^{-2} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1},$$

and (ii) $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})] \geq \frac{\varepsilon^2}{2(k')^2}$. Note that these both happen with probability at least $1 - 2 \cdot 2^{-(k')^2}$ by [Lemma 48](#) and [Lemma 14](#) by our choice of parameters in the invocation of each call to ESTIMATE-NINF.

By (i), recalling that

$$\sum_{U \in \binom{[k']}{\gamma}} \mathbf{NInf}_U[h] = W^{\geq \gamma}[h]$$

(see the comment immediately after [Definition 21](#)), we get that

$$\left| \sum_U \lambda_U - \mathbf{W}^{\geq \gamma}[h] \right| \leq \left(\frac{\varepsilon}{k'} \right)^{10} \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1}$$

Since we are conditioning on (ii), we can apply [Lemma 59](#) to get that

$$\mathbf{W}^{\geq \gamma}[h] \geq \frac{\varepsilon^2}{4(k')^2}.$$

We thus have

$$\left| \sum_U \lambda_U - \mathbf{W}^{\geq \gamma}[h] \right| \leq \left(\frac{\varepsilon}{k'} \right)^{10} \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1} \leq \left(\frac{\varepsilon}{k'} \right)^7 \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1} \cdot \mathbf{W}^{\geq \gamma}[h].$$

It then follows that conditioned on (i) and (ii), we have

$$\begin{aligned} \Pr[\tilde{\mathbf{T}} = U] &\leq \frac{\mathbf{NInf}_U[h] + (\varepsilon/k')^{10} \binom{k'}{\gamma}^{-2} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1}}{\mathbf{W}^{\geq \gamma}[h] \left(1 - \left(\frac{\varepsilon}{k'} \right)^7 \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1} \right)} \\ &\leq \Pr[\mathbf{T} = U] + 9 \left(\frac{\varepsilon}{k'} \right)^7 \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1}, \end{aligned}$$

where the second inequality uses $\frac{\mathbf{NInf}_U[h]}{\mathbf{W}^{\geq \gamma}[h]} = \Pr[\mathbf{T} = U]$ and the lower bound on $\mathbf{W}^{\geq \gamma}[h]$ from above.

For the lower bound, reasoning as in the upper bound using (i) and (ii) we get that

$$\begin{aligned} \Pr[\tilde{\mathbf{T}} = U] &\geq \frac{\mathbf{NInf}_U[f] - (\varepsilon/k')^{10} \binom{k'}{\gamma}^{-2} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1}}{\mathbf{W}^{\geq \gamma}[h] \left(1 + \left(\frac{\varepsilon}{k'} \right)^7 \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1} \right)} \\ &\geq \Pr[\mathbf{T} = U] - 9 \left(\frac{\varepsilon}{k'} \right)^7 \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1}. \end{aligned}$$

Since our assumptions fail with probability at most $2^{-(k')^2}$ we then get that

$$\Pr[\tilde{\mathbf{T}} = U] \geq \Pr[\mathbf{T} = U] - 9 \left(\frac{\varepsilon}{k'} \right)^7 \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1} - 2 \cdot 2^{-(k')^2}$$

and

$$\Pr[\tilde{\mathbf{T}} = U] \leq \Pr[\mathbf{T} = U] + 9 \left(\frac{\varepsilon}{k'} \right)^7 \binom{k'}{\gamma}^{-1} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \right)^{-1} + 2 \cdot 2^{-(k')^2}$$

which completes the proof. \square

With this strong bound on variation distance in hand, we can now prove [Lemma 53](#) for our non-idealized REFINE-COORDINATES procedure.

Proof of Lemma 53. In each fixed iteration of the loop on Line 2(d) of REFINE-COORDINATES, we have that $d_{TV}(\tilde{\mathbf{T}}, \mathbf{T}) \leq \left(\frac{\varepsilon}{k'}\right)^5 \cdot \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)}\right)^{-1}$ by [Lemma 65](#). By the coupling interpretation of total variation distance, this means that there exists a coupling between \mathbf{T} and $\tilde{\mathbf{T}}$ such that $\Pr[\mathbf{T} \neq \tilde{\mathbf{T}}] = d_{TV}(\mathbf{T}, \tilde{\mathbf{T}})$.

We now imagine drawing sets from $\tilde{\mathbf{T}}$ according to this coupling in line 2(c)(iv). Correspondingly, let $\tilde{\mathcal{F}}$ and \mathcal{F} be the sets built using draws of $\tilde{\mathbf{T}}$ and \mathbf{T} , respectively. Note that the estimate of $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})]$ in line 2(c)(iii) has accuracy $\pm \frac{\varepsilon^2}{10k^2}$ at every iteration with probability at least $1 - 2^{-(k')^2}$. As the loop on line 2(c) can never repeat more than $k' + 1$ times (as we must add a new element to \mathbf{C}' each time we reach step 2(c)(vi)), by a union bound with high probability the algorithm always exits the loop in line 2(c)(iii) when $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})] \leq \frac{\varepsilon^2}{2(k')^2}$ and never exits when $\mathbf{E}_{\mathbf{x}}[h^2(\mathbf{x})] \geq \frac{2\varepsilon^2}{(k')^2}$. This means that item (1.) of [Assumption 54](#) is satisfied with probability at least $1 - 2^{-k'}$. On the other hand, by [Lemma 65](#), the coupling, and a union bound over all iterations of line 2(c)(iv), we have that $\tilde{\mathbf{T}} = \mathbf{T}$ in every iteration of line 2(c)(iv) with probability at least

$$1 - (k') \cdot \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)}\right) \cdot \left(\frac{\varepsilon}{k'}\right)^5 \cdot \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)}\right)^{-1} \geq 1 - \left(\frac{\varepsilon}{k'}\right)^4;$$

when this happens we have that that item (2.) of [Assumption 54](#) is satisfied as well. Our earlier analysis proved [Lemma 53](#) under [Assumption 54](#), so with overall probability at least

$$1 - 2^{-k'} - \left(\frac{\varepsilon}{k'}\right)^4 \geq 1 - \frac{2}{(k')^2},$$

we have that the output $\tilde{\mathcal{F}}$ of the non-idealized procedure is identical to the output \mathcal{F} of the idealized procedure. By the idealized version of [Lemma 53](#) we get the non-idealized version of [Lemma 53](#) where now the success probability is at least $1 - \frac{3}{(k')^2}$. \square

Finally, we turn to proving that REFINE-COORDINATES does not make too many calls to the randomized algorithm \mathcal{A} .

Lemma 66. Let $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$ be a function that we have access to via a 1-bounded randomized algorithm \mathcal{A} , $\varepsilon \in [0, 1]$, and $C \subseteq [k']$ be a set of coordinates. Then REFINE-COORDINATES(f, C) makes at most $\text{poly}(2^{k^{1/3} \text{polylog}(k'/\varepsilon)})$ calls to \mathcal{A} .

Proof. Calls to \mathcal{A} are made in lines 2(c)(iii) and 2(c)(iv). In particular, since as noted earlier we can only repeat the loop of line 2(c) at most $k' + 1$ times, using the fact that we only run the loop on line 2 of REFINE-COORDINATES $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ times, it follows that we make at most $(k' + 1) \cdot 2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ many calls to each of lines 2(c)(iii) and 2(c)(iv).

By [Lemma 14](#), we have that each call of line 2(c)(iii) makes $\text{poly}(k', 2^{\kappa\Delta}, \varepsilon)$ queries to the randomized algorithm \mathcal{A}'' for h (recall that as discussed earlier we invoke [Lemma 14](#) in this context with accuracy parameter $\frac{\varepsilon^2}{10k^2}$ and failure probability $2^{-(k')^2}$). In turn, each call to \mathcal{A}'' makes $O(\kappa\Delta)$ queries to \mathcal{A} by [Lemma 64](#). Thus, in total throughout all executions of line 2(c)(iii) of REFINE-COORDINATES we make at most

$$k' \cdot 2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \cdot \text{poly}(k', 2^{\kappa\Delta}, \varepsilon) \cdot O(\kappa\Delta) \leq \text{poly}(2^{k^{1/3} \text{polylog}(k'/\varepsilon)})$$

queries to \mathcal{A} .

On the other hand, each execution of line 2(c)(iv) makes $\binom{k'}{\gamma}$ calls to ESTIMATE-NINF as in Equation (43). By Lemma 48 each of those calls makes

$$\text{poly}(\gamma!, 2^{\kappa\Delta}, 2^{k^{1/3}\text{polylog}(k'/\varepsilon)})$$

calls to \mathcal{A}'' , and hence $O(\kappa\Delta)$ times as many calls to \mathcal{A} by Lemma 64. Thus, over the entire execution of REFINE-COORDINATES, we make at most

$$k' \cdot 2^{k^{1/3}\text{polylog}(k'/\varepsilon)} \cdot \text{poly}(\gamma!, 2^{\kappa\Delta}, 2^{k^{1/3}\text{polylog}(k'/\varepsilon)}) \cdot O(\kappa\Delta) \leq 2^{k^{1/3}\text{polylog}(k'/\varepsilon)}$$

many calls to \mathcal{A} via 2(c)(iv). □

6.3 Finding a Pure Set of Coordinates

In this section, we explain how we can iteratively apply REFINE-COORDINATES so as to output a set $C' \subseteq C$, where C' contains no irrelevant variables but still captures almost all of the Fourier mass in R above degree $\tilde{\Omega}(k^{2/3})$.

Input: A function $g : \{\pm 1\}^{k'} \rightarrow [-1, 1]$ along with a 1-bounded randomized algorithm for g , a set of pairs \mathcal{F}_{i-1} , an integer i , and a parameter $\varepsilon \in [0, 1]$ that will correspond to the desired junta correlation accuracy

Output: A set \mathcal{F}'' of pairs (C', I') with $\mathcal{F}_{i-1} \subseteq \mathcal{F}''$

FIND-HIGH-LEVEL-COORDINATES-RECURSIVE:

1. Set $\mathcal{F}_i \leftarrow \emptyset$
2. For each (C, I) in \mathcal{F}_{i-1} :
 - (a) For each $(\mathbf{C}', \mathbf{I}')$ in $\text{REFINE-COORDINATES}(g_{\text{ave}}^I, C, \varepsilon)$:
 - i. Add $(\mathbf{C}', I \cup \mathbf{I}')$ to \mathcal{F}_i
3. If $i < \log(k')$, return $\mathcal{F}_{i-1} \cup \text{FIND-HIGH-LEVEL-COORDINATES-RECURSIVE}(g, \mathcal{F}_i, i + 1, \varepsilon)$
4. Return $\mathcal{F}_{i-1} \cup \mathcal{F}_i$

Algorithm 5: A recursive algorithm to find variables in high level Fourier coefficients.

Input: A function $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$ along with a 1-bounded randomized algorithm for f , and a parameter $\varepsilon \in [0, 1]$ that will correspond to the desired junta correlation accuracy

Output: A set \mathcal{F} of pairs (C, I) with $C, I \subseteq [k']$

FIND-HIGH-LEVEL-COORDINATES:

1. $g \leftarrow \mathbb{T}_{1-\frac{1}{2k}} f$
2. Return $\text{FIND-HIGH-LEVEL-COORDINATES-RECURSIVE}(g, \mathcal{F}_{-1} = \{([k'], \emptyset)\}, 0, \varepsilon)$

Algorithm 6: A wrapper method around FIND-HIGH-LEVEL-COORDINATES-RECURSIVE.

The main guarantee we will need about the FIND-HIGH-LEVEL-COORDINATES procedure given in [Algorithm 6](#), is the following:

Lemma 67. Let $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$ be a function computed by a 1-bounded randomized algorithm \mathcal{A} and let $R \subseteq [k']$ be a set of size k . With probability at least $1 - \frac{1}{k'}$, FIND-HIGH-LEVEL-COORDINATES(f, ε) outputs a set \mathcal{F} such that for some $(\mathbf{C}, \mathbf{I}) \in \mathcal{F}$ we have

- (i) $\mathbf{I} \subseteq \overline{R}$;
- (ii) $\mathbf{C} \subseteq R$;
- (iii)
$$\sum_{\substack{S \subseteq [k'] \\ |S \setminus \mathbf{C}| \geq k^{2/3} \text{polylog}(k'/\varepsilon), \\ |S| \leq k}} \widehat{f_{\text{ave}}^{\mathbf{I}}}(S) \leq \frac{\varepsilon^2}{100}.$$

Proof. We start with the simple observation that since f is a $[-1, 1]$ valued function, the attenuation of its high-degree weight by the noise operator yields

$$\mathbf{W}^{\geq 100k \log(k'/\varepsilon)} \left[\mathbb{T}_{1-\frac{1}{2k}} f \right] \leq \left(1 - \frac{1}{2k} \right)^{100k \log(k'/\varepsilon)} \mathbf{E}[f^2] \leq \left(\frac{\varepsilon}{k'} \right)^{20}.$$

Hence by [Equation \(4\)](#), for any set $I \subseteq [k']$ we have

$$\mathbf{W}^{\geq 100k \log(k'/\varepsilon)} [g_{\text{ave}}^I] \leq \left(\frac{\varepsilon}{k'} \right)^{20},$$

which means that we can apply [Lemma 53](#) to any function g_{ave}^I . Indeed, we will repeatedly apply [Lemma 53](#) to establish the following claim:

Claim 68. For $i \leq \log(k')$, let \mathcal{F}'_i denote the set \mathcal{F}_i that has been produced by the procedure FIND-HIGH-LEVEL-COORDINATES-RECURSIVE($g, \mathbf{C}, i, \varepsilon$) after line (2) has been completed, and let $\mathcal{F}_{-1} = \{([k'], \emptyset)\}$. Then with probability at least $1 - (i+1) \cdot \frac{10}{(k')^2}$, there exists an $\ell \leq \log(k')$ and pairs $(\mathbf{C}_j, \mathbf{I}_j) \in \mathcal{F}'_j$, for all $-1 \leq j \leq \min\{i, \ell\}$, such that $[k'] := \mathbf{C}_{-2} := \mathbf{C}_{-1} \supseteq \mathbf{C}_0 \supseteq \mathbf{C}_1 \supseteq \mathbf{C}_2 \supseteq \dots \supseteq \mathbf{C}_{\min\{i, \ell\}}$ and $\mathbf{I}_{\min\{i, \ell\}} \supseteq \mathbf{I}_{-1} \supseteq \dots \supseteq \mathbf{I}_{-1} := \emptyset$, which satisfy the following for all $-1 \leq j \leq \ell$:

- (a) $\mathbf{I}_j \subseteq \overline{R}$
- (b) $|\mathbf{C}_j \cap \overline{R}| \leq \begin{cases} \frac{|\overline{R}|}{2^{j+1}} & j < \ell \\ 0 & j = \ell \end{cases}$
- (c)
$$\sum_{S: |S \cap \mathbf{C}_{j-1} \setminus \mathbf{C}_j| \geq k^{2/3} \text{polylog}(k'/\varepsilon)} \widehat{g_{\text{ave}}^{\mathbf{I}_j}}(S) \leq \frac{4\varepsilon^2}{k^2}.$$

Before proving the claim, we first show how it implies [Lemma 67](#). Indeed, we'll apply the claim with $i = \log(k')$ and show that $(\mathbf{C}_\ell, \mathbf{I}_\ell)$ will satisfy conditions (i), (ii), and (iii) of the lemma. Let $[k'] := \mathbf{C}_{-2} := \mathbf{C}_{-1} \supseteq \mathbf{C}_0 \supseteq \mathbf{C}_1 \supseteq \mathbf{C}_2 \supseteq \dots \supseteq \mathbf{C}_{\min\{i, \ell\}}$ and $\mathbf{I}_{\min\{i, \ell\}} \supseteq \mathbf{I}_{-1} \supseteq \dots \supseteq \mathbf{I}_{-1} := \emptyset$ be the sequence of sets produced by [Claim 68](#) for $i = \log(k')$. To start, note that properties (a) and (b) immediately gives us that $(\mathbf{C}_\ell, \mathbf{I}_\ell)$ satisfies (i) and (ii).

It remains to show that property (iii) holds. To start, note that for every set $S \subseteq [k']$ of size at most k , we have that for any set $I \subseteq [k']$

$$\widehat{g_{\text{ave}}^I}^2(S) \geq \left(1 - \frac{1}{2k}\right)^{2|S|} \widehat{f_{\text{ave}}^I}^2(S) \geq \frac{1}{4} \widehat{f_{\text{ave}}^I}^2(S).$$

Thus, we can conclude that that property (c) implies that for all $-1 \leq j \leq \ell$ we have that

$$\sum_{\substack{S: |S \cap \mathbf{C}_{j-1} \setminus \mathbf{C}_j| \geq k^{2/3} \text{polylog}(k'/\varepsilon) \\ |S| \leq k}} \widehat{f_{\text{ave}}^{\mathbf{I}_j}}^2(S) \leq 4 \sum_{\substack{S: |S \cap \mathbf{C}_{j-1} \setminus \mathbf{C}_j| \geq k^{2/3} \text{polylog}(k'/\varepsilon) \\ |S| \leq k}} \widehat{g_{\text{ave}}^{\mathbf{I}_j}}^2(S) \leq \frac{16\varepsilon^2}{k^2}, \quad (44)$$

We now note that if $|S \setminus \mathbf{C}'_\ell| \geq k^{2/3}(\ell+2)\text{polylog}(k'/\varepsilon)$, then there must exist a $j \leq \ell$ such that $|S \cap \mathbf{C}_{j-1} \setminus \mathbf{C}_j| \geq k^{2/3}\text{polylog}(k'/\varepsilon)$, since

$$\bigcup_{j=0}^{\ell} S \cap \mathbf{C}'_{j-1} \setminus \mathbf{C}'_j = S \setminus \mathbf{C}'_\ell.$$

Using this and $\mathbf{I}_\ell \supseteq \mathbf{I}_{\ell-1} \supseteq \dots$ for the first inequality below, we get that

$$\begin{aligned} \sum_{\substack{S \subseteq [k']: \\ |S \setminus \mathbf{C}'| \geq k^{2/3}(\ell+2)\text{polylog}(k'/\varepsilon), \\ |S| \leq k}} \widehat{f_{\text{ave}}^{\mathbf{I}_\ell}}^2(S) &\leq \sum_{j=0}^{\ell} \sum_{\substack{S \subseteq [k']: \\ |S \cap \mathbf{C}_{j-1} \setminus \mathbf{C}_j| \geq k^{2/3} \text{polylog}(k'/\varepsilon) \\ |S| \leq k}} \widehat{f_{\text{ave}}^{\mathbf{I}_j}}^2(S) \\ &\leq \sum_{j=0}^{\log(k')} \sum_{\substack{S \subseteq [k']: \\ |S \cap \mathbf{C}_{j-1} \setminus \mathbf{C}_j| \geq k^{2/3} \text{polylog}(k'/\varepsilon) \\ |S| \leq k}} \widehat{f_{\text{ave}}^{\mathbf{I}_j}}^2(S) \\ &\leq \log(k') \cdot \frac{16\varepsilon^2}{k^2} \quad (\text{by Equation (44)}) \\ &\leq \frac{\varepsilon^2}{100}. \end{aligned}$$

Since $\ell \leq \log(k')$ this gives us property (iii) as desired.

We now turn to prove [Claim 68](#).

Proof of Claim 68. We prove the statement by induction on i . For the base case, note that when $i = -1$, taking $\ell = 0$ and $(\mathbf{C}_{-1}, \mathbf{I}_{-1}) = ([k'], \emptyset)$ satisfies the required conditions. We now assume the statement holds for i and wish to show that it is true for \mathcal{F}_{i+1} . By the inductive hypothesis, let $\ell \leq \log(k')$ and $\mathbf{C}_{-1} \supseteq \mathbf{C}_0 \supseteq \mathbf{C}_1 \supseteq \mathbf{C}_2 \supseteq \dots \supseteq \mathbf{C}_{\min\{i, \ell\}}$ be as promised by the lemma with corresponding sets $\mathbf{I}_{\min\{i, \ell\}} \supseteq \dots \supseteq \mathbf{I}_{-1}$.

We claim that if, in the inductive hypothesis, we can take $\ell \leq i$, then we are done. To see this, observe that in the execution of the loop on line (2) in the $(i+1)$ st recursive call corresponding to $(\mathbf{C}_i, \mathbf{I}_i)$, we will add a pair (\mathbf{C}, \mathbf{I}) to \mathcal{F}_{i+1} which by the guarantee of `REFINE-COORDINATES` ([Lemma 53](#)) satisfies $\mathbf{C} \subseteq \mathbf{C}_i$ and $\mathbf{I} \subseteq \mathbf{C}_i$. Indeed, in this iteration we always have that the set \mathbf{C}' computed by `REFINE-COORDINATES` on line 2(a) is a subset of \mathbf{C}_i by the output guarantee of `REFINE-COORDINATES`. Moreover, since the pair we are adding is of the form $(\mathbf{C}', \mathbf{I}_i \cup \mathbf{I}')$, we have

that $\mathbf{I} = \mathbf{I}_i \cup \mathbf{I}' \supseteq \mathbf{I}_i$. So taking $\mathbf{C}_{i+1} = \mathbf{C}$ and $\mathbf{I}_{i+1} = \mathbf{I}$, we have the new sequence of inclusions $[k'] := \mathbf{C}_{-2} := \mathbf{C}_{-1} \supseteq \mathbf{C}_0 \supseteq \mathbf{C}_1 \supseteq \mathbf{C}_2 \supseteq \cdots \supseteq \mathbf{C}_{\min\{i+1, \ell\}}$ and $\mathbf{I}_{\min\{i+1, \ell\}} \supseteq \mathbf{I}_{i-1} \supseteq \cdots \supseteq \mathbf{I}_{-1} := \emptyset$ as required, and we inherit that (a), (b) and (c) hold for all $-1 \leq j \leq \ell$ from the inductive hypothesis.

So we assume that in the inductive hypothesis we cannot take $\ell \leq i$, i.e. we must have $\ell > i$. This implies that $|\mathbf{C}_i \cap \overline{\mathbf{R}}| \geq 1$ (because if $\mathbf{C}_i \cap \overline{\mathbf{R}} = \emptyset$, we could have taken $\ell = i$). Now again consider the execution of the loop on line (2) of the $(i+1)$ st recursive call corresponding to $(\mathbf{C}_i, \mathbf{I}_i)$. By [Lemma 53](#), it follows that $\text{REFINE-COORDINATES}(g_{\text{avg}}^{\mathbf{I}_i}, \mathbf{C}_i)$ outputs a pair $(\mathbf{C}', \mathbf{I}')$ such that $\mathbf{I}' \subseteq \overline{\mathbf{R}}$, $\mathbf{C}' \subseteq \mathbf{C}_i$,

$$|\mathbf{C}' \cap \overline{\mathbf{R}}| \leq \frac{|\mathbf{C}_i \cap \overline{\mathbf{R}}|}{2},$$

and

$$\sum_{S: |S \cap \mathbf{C}_i \setminus \mathbf{C}'| \geq k^{2/3} \text{polylog}(k'/\varepsilon)} \widehat{g_{\text{avg}}^{\mathbf{I}_i \cup \mathbf{I}'}}(S) \leq \frac{4\varepsilon^2}{(k')^2}.$$

It thus would suffice for us to take $\mathbf{C}_{i+1} = \mathbf{C}'$ and $\mathbf{I}_{i+1} = \mathbf{I}_i \cup \mathbf{I}'$ if we can choose $\ell = i + 2$. If we cannot choose $\ell = i + 2$, then we must have that $i + 1 = \log(k')$. In this case, it follows that

$$|\mathbf{C}_{i+1} \cap \overline{\mathbf{R}}| \leq \frac{|\mathbf{C}_i \cap \overline{\mathbf{R}}|}{2} \leq \frac{|\overline{\mathbf{R}}|}{2^{i+1}} \leq \frac{|\overline{\mathbf{R}}|}{2k'} < 1.$$

So $|\mathbf{C}_{i+1} \cap \overline{\mathbf{R}}| = 0$ and we can take $\ell = i + 1$. We have the desired inclusions of the \mathbf{C}_j 's and the \mathbf{I}_j 's; the arguments above establish that (a), (b) and (c) all hold for $j = \ell = i + 1$; and we inherit (a), (b), (c) holding for smaller values of j from the inductive case. So we have achieved the inductive claim for \mathcal{F}_{i+1} .

Finally, note that by a union bound, both the inductive step and our call to $\text{REFINE-COORDINATES}$ corresponding to the pair $(\mathbf{C}_i, \mathbf{I}_i)$ succeed with probability $1 - (i+1)\frac{10}{(k')^2}$, completing the inductive step and the proof of [Claim 68](#). \square

This concludes the proof of [Lemma 67](#). \square

We will also need some simple bounds on the size of the family of pairs that we output and the number of oracle calls made to the randomized algorithm computing f .

Lemma 69. Let $f : \{\pm 1\}^{k'} \rightarrow [-1, 1]$ and $\varepsilon \in [0, 1]$, then $\text{FIND-HIGH-LEVEL-COORDINATES}(f, \varepsilon)$ outputs a family \mathcal{F} of size at most $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$.

Proof. Let \mathcal{F}'_i denote the set \mathcal{F}' computed in the loop on line 2 of the i th recursive call of $\text{FIND-HIGH-LEVEL-COORDINATES-RECURSIVE}$. Since each call to $\text{REFINE-COORDINATES}$ in line 2(a) of $\text{FIND-HIGH-LEVEL-COORDINATES-RECURSIVE}$ adds $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ pairs to \mathcal{F} by [Lemma 53](#), it follows that

$$|\mathcal{F}'_{i+1}| \leq 2^{k^{1/3} \text{polylog}(k'/\varepsilon)} \cdot |\mathcal{F}'_i|.$$

Since we have

$$|\mathcal{F}_0| \leq 2^{k^{1/3} \text{polylog}(k'/\varepsilon)},$$

it follows that $\text{FIND-HIGH-LEVEL-COORDINATES}(f, \varepsilon)$ outputs at most

$$|\{[k'], \emptyset\}| + \sum_{i=0}^{\log(k')} |\mathcal{F}_i| \leq 1 + \sum_{i=0}^{\log(k')} \left(2^{k^{1/3} \text{polylog}(k'/\varepsilon)}\right)^{i+1} \leq 2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$$

pairs as desired. \square

Lemma 70. $\text{FIND-HIGH-LEVEL-COORDINATES}(f, \varepsilon)$ makes at most $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ queries to \mathcal{A} , the 1-bounded randomized algorithm for f .

Proof. Recall that for any $I \subseteq [k']$ by [Lemma 16](#) and [Lemma 15](#) it follows that there is a randomized 1-bounded algorithm for g_{ave}^I that only needs to query \mathcal{A} once.

Let \mathcal{F}'_i denote the set \mathcal{F}' computed in the loop on line 2 of the i th recursive call of $\text{FIND-HIGH-LEVEL-COORDINATES-RECURSIVE}$. Note that in the $(i + 1)$ st iteration, all queries to f will come from calling $\text{REFINE-COORDINATES}$ in line 2(a) of $\text{FIND-HIGH-LEVEL-COORDINATES-RECURSIVE}$. It follows from [Lemma 66](#) that the $(i + 1)$ st recursive call makes

$$|\mathcal{F}'_i| \cdot 2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$$

calls to \mathcal{A} , where $|\mathcal{F}'_{-1}| = 1$. Since \mathcal{F}'_i is a subset of the output of $\text{FIND-HIGH-LEVEL-COORDINATES}(f, \varepsilon)$ it then follows that $|\mathcal{F}'_i| \leq 2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$. Combining this with the fact that there are at most $\log(k') + 1$ rounds then yields that we make at most $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ calls to \mathcal{A} as desired. \square

6.4 Putting It All Together

6.4.1 Parameters for the Main Algorithm

As before, we will need several parameters for our main algorithm. Throughout the section, we will write these parameters in terms of k' , which will correspond to the number of coordinate oracles given by [Theorem 24](#).

We now set ℓ to correspond to the set size limit in the summation limits of (iii) from [Lemma 67](#). In particular, this gives us that $\ell = k^{2/3} \text{polylog}(k'/\varepsilon)$.

We will also need a variety of parameters for our local estimators. In particular, we will set these to all be identical to their counterparts in [Section 5.5](#). That is, we choose κ, Δ as

$$\kappa = 10 \log \left(\frac{k'}{\varepsilon} \right) \quad \text{and} \quad \Delta = 10r \log \left(\frac{k'}{\varepsilon} \right).$$

respectively where

$$L = \kappa \ell, \quad \tau = \frac{\varepsilon}{10}, \quad \text{and} \quad r = \Theta(\sqrt{L \log L \log(1/\tau)})$$

and set

$$N := k'^{O(r)} \cdot 2^{O(\kappa \Delta)} \cdot \frac{k'^2}{\varepsilon^2} \leq \exp \left(k^{1/3} \cdot \text{polylog} \left(\frac{k'}{\varepsilon} \right) \right).$$

Because these parameters are set identically (as functions of k' , ε , and ℓ) as in [Section 5.5](#), we have that [Lemma 39](#) still holds for any function $g : \{\pm 1\}^{k'} \rightarrow \{\pm 1\}$ with a 1-bounded randomized algorithm computing it with our new choice of ℓ .

6.4.2 The Testing Algorithm

With the above setup in place, we can finally state our classical testing algorithm.

Input: A boolean function $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$, an integer k , and a parameter $\varepsilon \in [0, 1]$

Output: A estimate for $\text{corr}(f, \mathcal{J}_k)$

TOLERANT-JUNTA-TESTER:

1. Set $\gamma = 0$
2. Compute coordinate oracles $\{\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_{k'}\}$ by invoking [Theorem 24](#) on f with accuracy ε and failure probability $\frac{1}{k}$. Let $k' = \text{poly}(k, \varepsilon^{-1})$ denote the number of oracles.
3. Let $g : \{\pm 1\}^{k'} \rightarrow [-1, 1]$ be the function given by

$$g(y) = \mathbf{E}_{\mathbf{x}} \left[f(\mathbf{x}) \middle| \mathcal{O}_1(\mathbf{x}) = y_1, \mathcal{O}_2(\mathbf{x}) = y_2, \dots, \mathcal{O}_{k'}(\mathbf{x}) = y_{k'} \right],$$
 which has a 1-bounded randomized algorithm, which we call \mathcal{A}_g , by [Theorem 25](#).
4. $\mathcal{F} \leftarrow \text{FIND-HIGH-LEVEL-COORDINATES}(g, \varepsilon^2)$
5. For $(\mathbf{C}, \mathbf{I}) \in \mathcal{F}$:
 - (a) $\gamma' = 0, A = \emptyset$
 - (b) Define $\left(g_{\text{ave}}^{\mathbf{I}}\right)^{\overline{\mathbf{C}}} \leftarrow \text{SHARPNOISE}_{\ell, \kappa, \Delta}^{\overline{\mathbf{C}}}(g_{\text{ave}}^{\mathbf{I}})$.
 - (c) Sample $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)} \sim \{\pm 1\}^{k'}$ uniformly and independently at random.
 - (d) For each $\mathbf{x}^{(i)}$ draw a sample bundle $\mathcal{B}^{(i)} \sim \mathcal{D}_{\mathbf{x}^{(i)}, \mathbf{C}}$.
 - (e) For each set $U \subseteq [k']$ of size k such that $U \cap \mathbf{I} = \emptyset$ and $U \supseteq \mathbf{C}$, run the algorithm of [Lemma 39](#) using $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}, \mathcal{B}^{(1)}, \dots, \mathcal{B}^{(N)}$ to get an $(\varepsilon/2)$ -estimation $\mathbf{Est}_{\mathbf{C}, U}$ of $\text{corr} \left(\left(g_{\text{ave}}^{\mathbf{I}}\right)^{\overline{\mathbf{C}}}, \mathcal{J}_U \right)$; if $\mathbf{Est}_{\mathbf{C}, U} > \gamma'$, update $\gamma' = \mathbf{Est}_{\mathbf{C}, U}$ and $A \leftarrow U$.
 - (f) Let \mathbf{Est}_A be an estimate of $\text{corr}(g, \mathcal{J}_A)$ to accuracy $\pm \varepsilon/20$ with failure probability $2^{-\Omega((k')^2)}$ and update $\gamma \leftarrow \max(\gamma, \mathbf{Est}_A)$.
6. Return γ

Algorithm 7: The overall classical tolerant junta tester

We give some high level intuition. Let U^* denote the set of k coordinates that maximize the junta correlation. At a high level the algorithm begins by building coordinate oracles so as to reduce the problem to a question about g , which is a real-valued function over $\{\pm 1\}^{k'}$ where $k' = \text{poly}(k, 1/\varepsilon)$, rather than f , which is a Boolean-valued function on n coordinates. Next, we call $\text{FIND-HIGH-LEVEL-COORDINATES}(g, \varepsilon^2)$ with the goal of finding a pair $(\mathbf{C}^*, \mathbf{I}^*)$ such that \mathbf{C}^* contains all coordinates in U^* that appear at level ℓ and above and $\mathbf{I}^* \subseteq \overline{U^*}$. For this particular pair $(\mathbf{C}^*, \mathbf{I}^*)$, we will show that for every set U that we produce an estimate for in line 5(e), we have that with high probability $\mathbf{Est}_{\mathbf{C}^*, U}$ will be within ε of $\text{corr}(g, \mathcal{J}_U)$. As a result we will correctly update γ on line 5(f) with an accurate estimate of the true junta correlation of g , which by the properties of the coordinate oracles will roughly equal the junta correlation of f . On the other hand, we will not have such guarantees about our estimates of the junta correlation arising from pairs $(\mathbf{C}, \mathbf{I}) \in \mathcal{F}$ not equal to $(\mathbf{C}^*, \mathbf{I}^*)$. It is precisely for this reason that we will require line 5(f)

to check the correlation between g and the best junta over the set of variables A identified in line 5(e), so as to ensure that we do not spuriously update γ with a false estimate that is higher than the true junta correlation.

6.4.3 Efficiency

In this section, we will prove the following lemma:

Lemma 71. For $f : \{\pm 1\}^n \rightarrow \{\pm 1\}$ and k a positive integer, $\text{TOLERANT-JUNTA-TESTER}(f, k, \varepsilon)$ make at most $2^{k^{1/3} \text{polylog}(k/\varepsilon)}$ queries to f in expectation.

Before proving [Lemma 71](#), we first show that we can efficiently implement line 5(f).

Lemma 72. The junta correlation $\text{corr}(g, \mathcal{J}_A)$ on line 5f can be $\pm\varepsilon/20$ -approximated, with failure probability $2^{-\Omega((k')^2)}$, using $\text{poly}(k/\varepsilon)$ calls to the randomized algorithm \mathcal{A}_g for g .

Proof. First, recall from [Equation \(5\)](#) that

$$\text{corr}(g, \mathcal{J}_A) = \mathbf{E}_{\mathbf{x}} \left[\left| g_{\text{ave}}^{\bar{A}}(\mathbf{x}) \right| \right], \quad (45)$$

and that by [Equation \(4\)](#) the function $g_{\text{ave}}^{\bar{A}}$ is 1-bounded. Hence by [Lemma 16](#), there is a 1-bounded randomized algorithm \mathcal{A}'_g for $g_{\text{ave}}^{\bar{A}}$ that makes a single call to \mathcal{A}_g .

Our approach to approximating $\text{corr}(g, \mathcal{J}_A)$ is as follows: we (1) first sample $\mathbf{y}^1, \dots, \mathbf{y}^{(k')^2/\varepsilon^2} \sim \{\pm 1\}^{k'}$; then (2) compute, for each i , a value $z_i(y_i)$ that is a $\pm\varepsilon/100$ -accurate estimate (with failure probability $2^{-(k')^2}$) of $|g_{\text{ave}}^{\bar{A}}(\mathbf{y}^i)|$ using [Claim 13](#); and (3) finally output

$$\mathbf{Z} := \frac{\varepsilon^2}{(k')^2} \sum_{i=1}^{(k')^2/\varepsilon^2} z_i$$

as our approximation. Let \mathcal{A}''_g denote the (randomized) algorithm used to estimate $z_i(\mathbf{y}^i)$ in step (2) above.

Let us argue correctness for this approximation. Since for every possible outcome y_i of \mathbf{y}_i the random variable $z_i(y_i)$ is bounded in $[-1, 1]$, by [Claim 13](#) we have that

$$\left| \mathbf{E}_{\mathcal{A}''_g} [z_i(y_i)] - \left| g_{\text{ave}}^{\bar{A}}(\mathbf{y}^i) \right| \right| \leq \frac{\varepsilon}{100} + 2 \cdot 2^{-(k')^2} \leq \frac{\varepsilon}{50},$$

from which it follows that

$$\left| \mathbf{E}_{\mathbf{y}^i, \mathcal{A}''_g} [z_i(\mathbf{y}_i)] - \overbrace{\mathbf{E}_{\mathbf{y}^i} \left[\left| g_{\text{ave}}^{\bar{A}}(\mathbf{y}^i) \right| \right]}^{=\text{corr}(g, \mathcal{J}_A) \text{ by (45)}} \right| \leq \mathbf{E}_{\mathbf{y}^i} \left[\left| \mathbf{E}_{\mathcal{A}''_g} [z_i(\mathbf{y}_i)] - \left| g_{\text{ave}}^{\bar{A}}(\mathbf{y}^i) \right| \right| \right] \leq \frac{\varepsilon}{50}. \quad (46)$$

We apply a Hoeffding bound to \mathbf{Z} (a sum of independent 1-bounded random $z_i(\mathbf{y}^i)$'s) to infer that with failure probability at most $2^{-\Omega((k')^2)}$, we have

$$\left| \mathbf{Z} - \mathbf{E}_{\mathbf{y}_i, \mathcal{A}''_g} [z_i(\mathbf{y}^i)] \right| \leq \frac{\varepsilon}{50}. \quad (47)$$

By the triangle inequality, Equation (46) and Equation (47) give that $|\mathbf{Z} - \text{corr}(g, \mathcal{J}_A)| \leq \varepsilon/20$ except with failure probability $2^{-\Omega((k')^2)}$, as desired.

It remains to bound the number of queries this procedure makes. This is straightforward: each invocation of Claim 13 requires $\text{poly}(k', 1/\varepsilon)$ calls to \mathcal{A}_g and we make $(k')^2/\varepsilon^2$ such invocations. Thus we make $\text{poly}(k', 1/\varepsilon) = \text{poly}(k/\varepsilon)$ calls in total, as claimed. \square

With Lemma 72 in hand, we now prove Lemma 71.

Proof of Lemma 71. We begin by noting that by Theorem 24, TOLERANT-JUNTA-TESTER makes at most $\text{poly}(k, \varepsilon^{-1})$ queries in line 2.

We now bound the number of queries made to g in the rest of the algorithm. By Lemma 70, we have that the call to FIND-HIGH-LEVEL-COORDINATES(g, ε^2) in line 4, makes at most $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ queries to g .

Now fix a particular iteration of the loop on line 5, i.e. a particular (\mathbf{C}, \mathbf{I}) pair. Note that using the local estimators to compute the junta correlation for all of the sets U in line 5e can be implemented using $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ queries to g_{ave}^I by Lemma 39. (Similar to before, this crucially uses the fact that we can reuse the sample bundles for all k -subsets U of $[k']$ such that $U \cap \mathbf{I} = \emptyset$ and $U \supseteq \mathbf{C}$.) In turn this corresponds to $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ queries to g by applying Lemma 16. Finally, estimating $\text{corr}(g, \mathcal{J}_A)$ on line 5f makes at most $\text{poly}(k/\varepsilon)$ queries by Lemma 72. Since we repeat the loop on line 5 at most $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ times by Lemma 69, we conclude that we make at most $2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$ queries to g throughout this loop.

Lemma 71 now follows because each query to g makes at most $\text{poly}(k')$ queries to f in expectation, by Theorem 25. In particular, applying linearity of expectation gives that we make at most

$$2^{k^{1/3} \text{polylog}(k'/\varepsilon)} = 2^{k^{1/3} \text{polylog}(k/\varepsilon)}$$

queries to f in expectation, as desired, where the equality used the fact that $k' = \text{poly}(k, \varepsilon^{-1})$. \square

6.4.4 Correctness

We now turn to prove that γ will indeed approximate $\text{corr}(f, \mathcal{J}_k)$ up to $O(\varepsilon)$ error. We will prove this in two parts. For the first part, we show that γ cannot significantly overestimate the junta correlation:

Lemma 73. With probability $1 - o(1)$,

$$\gamma \leq \text{corr}(f, \mathcal{J}_k) + \varepsilon.$$

Proof. We condition on the event that the coordinate oracles that we compute in line 2 outputs a set of oracles satisfying Theorem 24. We further condition on the event that all of our estimates on line 5f have error at most $\varepsilon/20$. By Lemma 72 and a union bound over all elements of \mathcal{F} , these events both happen with probability at least

$$1 - \frac{1}{k} - |\mathcal{F}| 2^{-\Omega((k')^2)} = 1 - o(1),$$

where we also used the fact that $|\mathcal{F}| \leq 2^{k^{1/3} \text{polylog}(k'/\varepsilon)}$, which follows from Lemma 69.

Now let $\mathcal{S} \subseteq [n]$ denote the set of coordinates corresponding to our coordinate oracles. We now note that since $g = f_{\text{ave}}^{\mathcal{S}}$, we get that in line 5f we always have

$$\text{corr}(g, \mathcal{J}_A) = \text{corr}(f_{\text{ave}}^{\mathcal{S}}, \mathcal{J}_A) = \text{corr}(f, \mathcal{J}_A) \leq \text{corr}(f, \mathcal{J}_k)$$

where the second equality used the fact that $A \subseteq \mathcal{S}$. It thus follows that we always have

$$\mathbf{Est}_A \leq \text{corr}(f, \mathcal{J}_k) + \frac{\varepsilon}{20}$$

whenever we run line 5f, and consequently $\gamma \leq \text{corr}(f, \mathcal{J}_k) + \frac{\varepsilon}{20}$ as desired. \square

It remains to bound γ in the other direction:

Lemma 74. With probability $1 - o(1)$, we have that

$$\gamma \geq \text{corr}(f, \mathcal{J}_k) - 4\varepsilon.$$

Proof. We will assume throughout the proof that we successfully created the coordinate oracles on line 2 of TOLERANT-JUNTA-TESTER (note that this happens with $1 - o(1)$ probability), and we let $\mathcal{S} \subseteq [n]$ denote the corresponding set of coordinates.

We now choose $R = U^* \subseteq [k']$ to be a set of size k that achieves $\text{corr}(g, \mathcal{J}_R) = \text{corr}(g, \mathcal{J}_k)$. Applying Lemma 67, we get that with $1 - o(1)$ probability there exists a pair $(\mathbf{C}^*, \mathbf{I}^*) \in \mathcal{F}$ such that $\mathbf{I}^* \subseteq \bar{R}$, $\mathbf{C}^* \subseteq R$, and

$$\sum_{\substack{S \subseteq [k'] \\ |S \setminus \mathbf{C}^*| \geq \ell, \\ |S| \leq k}} \widehat{g_{\text{ave}}^{\mathbf{I}^*}}(S)^2 \leq \frac{\varepsilon^2}{100}. \quad (48)$$

Our aim is to show that in the iteration of the loop on line 5 corresponding to $(\mathbf{C}^*, \mathbf{I}^*)$, we will update γ to be large. Towards this end, we assume that the conclusion of Lemma 39 holds for every set U that we compute in this iteration of line 5, that is

$$\left| \mathbf{Est}_{\mathbf{C}^*, U} - \text{corr} \left(\left(g_{\text{ave}}^{\mathbf{I}^*} \right)^{\bar{\mathbf{C}}^*}, \mathcal{J}_U \right) \right| \leq \frac{\varepsilon}{2}. \quad (49)$$

Note that this also holds with $1 - o(1)$ probability, as we output a bad estimate for each individual U with probability at most $2^{-(k'/\varepsilon)^2}$ by Lemma 39, and there are at most $2^{k'}$ sets U .

We will require the following claim:

Claim 75. For any k -subset $U \subseteq [k']$ satisfying $U \supseteq \mathbf{C}^*$ and $U \cap \mathbf{I}^* = \emptyset$, we have that

$$\left| \text{corr} \left(\left(g_{\text{ave}}^{\mathbf{I}^*} \right)^{\bar{\mathbf{C}}^*}, \mathcal{J}_U \right) - \text{corr}(g, \mathcal{J}_U) \right| \leq \frac{\varepsilon}{2}.$$

Before proving the claim, we first use it to finish the proof of Lemma 74. Together with Equation (49), we get that for all k -subsets U of $[k']$ satisfying $U \supseteq \mathbf{C}^*$ and $U \cap \mathbf{I}^* = \emptyset$, we have

$$|\mathbf{Est}_{\mathbf{C}^*, U} - \text{corr}(g, \mathcal{J}_U)| \leq \varepsilon; \quad (50)$$

specializing to $U = R$, we get that

$$|\mathbf{Est}_{\mathbf{C}^*, R} - \text{corr}(g, \mathcal{J}_R)| \leq \varepsilon.$$

So it follows that after the completion of line 5e, we have

$$\gamma' \geq \mathbf{Est}_{\mathbf{C}^*, R} \geq \text{corr}(g, \mathcal{J}_R) - \varepsilon = \text{corr}(g, \mathcal{J}_k) - \varepsilon.$$

If A is the set used on line 5f, it follows from Equation (50) and the above inequality that

$$\text{corr}(g, \mathcal{J}_A) + \varepsilon \geq \mathbf{Est}_{\mathcal{C}^*, A} = \gamma' \geq \text{corr}(g, \mathcal{J}_k) - \varepsilon.$$

So we conclude that

$$\text{corr}(g, \mathcal{J}_A) \geq \text{corr}(g, \mathcal{J}_k) - 2\varepsilon.$$

Thus assuming that the error in the estimate on line 5f is indeed at most $\varepsilon/20$ (which happens with overwhelming probability), we have that

$$\gamma \geq \mathbf{Est}_A \geq \text{corr}(g, \mathcal{J}_k) - 3\varepsilon.$$

Finally, since we successfully created the coordinate oracles, we have that

$$\text{corr}(g, \mathcal{J}_k) = \max_{S \subseteq \mathcal{S}: |S| \leq k} \text{corr}(f_{\text{ave}}^{\overline{S}}, \mathcal{J}_S) = \max_{S \subseteq \mathcal{S}: |S| \leq k} \text{corr}(f, \mathcal{J}_S) \geq \max_{S \subseteq [n]: |S| \leq k} \text{corr}(f, \mathcal{J}_S) - \varepsilon$$

where the final step used property (2) from Theorem 24. Thus, we get that

$$\gamma \geq \text{corr}(f, \mathcal{J}_k) - 4\varepsilon$$

with probability $1 - o(1)$, as desired.

So it only remains to prove the claim:

Proof of Claim 75. The proof is almost identical to the proof of Lemma 43. Fix any U as specified in Claim 75. We run the argument of Lemma 43's proof starting at Equation (31), now with $\text{corr}(g, \mathcal{J}_U)$ in place of $\text{corr}(f, \mathcal{J}_U)$ and with $\text{corr}\left(\left(g_{\text{ave}}^{\mathbf{I}^*}\right)^{\overline{\mathcal{C}^*}}, \mathcal{J}_U\right)$ in place of $\text{corr}\left(f^{\overline{\mathcal{C}}}, \mathcal{J}_U\right)$. That argument gives us that

$$\left| \text{corr}\left(\left(g_{\text{ave}}^{\mathbf{I}^*}\right)^{\overline{\mathcal{C}^*}}, \mathcal{J}_U\right) - \text{corr}(g, \mathcal{J}_U) \right| \leq \sqrt{\sum_{S \subseteq U} \left(\widehat{\left(g_{\text{ave}}^{\mathbf{I}^*}\right)^{\overline{\mathcal{C}^*}}}(S) - \widehat{g}(S) \right)^2}, \quad (51)$$

which is exactly analogous to Equation (32) in the earlier argument. As in the earlier argument, we now divide the sum inside the square root into two parts:

1. $S \subseteq U$ with $|S \setminus \mathcal{C}^*| = |S \cap \overline{\mathcal{C}^*}| \leq \ell$: For this part, exactly as in the earlier argument, we have from Lemma 36 that

$$\sum_{S \subseteq U: |S \setminus \mathcal{C}^*| \leq \ell} \left(\widehat{\left(g_{\text{ave}}^{\mathbf{I}^*}\right)^{\overline{\mathcal{C}^*}}}(S) - \widehat{g}(S) \right)^2 \leq \sum_{S \subseteq U: |S \setminus \mathcal{C}^*| \leq \ell} \widehat{g}(S)^2 \cdot (\Delta 2^{-\kappa})^2 \leq (\Delta 2^{-\kappa})^2 \leq \frac{\varepsilon^2}{20}$$

by our choices of κ and Δ .

2. $S \subseteq U$ with $|S \setminus \mathcal{C}^*| > \ell$: For this part we have from Lemma 36 and Equation (48) that

$$\sum_{S \subseteq U: |S \setminus \mathcal{C}^*| > \ell} \left(\widehat{\left(g_{\text{ave}}^{\mathbf{I}^*}\right)^{\overline{\mathcal{C}^*}}}(S) - \widehat{g}(S) \right)^2 \leq \sum_{S \subseteq U: |S \setminus \mathcal{C}^*| > \ell} \widehat{g}^2(S) = \sum_{S \subseteq U: |S \setminus \mathcal{C}^*| > \ell} \widehat{g_{\text{ave}}^{\mathbf{I}^*}}^2(S) \leq \frac{\varepsilon^2}{100}$$

where the equality step crucially used that $U \cap \mathbf{I}^* = \emptyset$ and the final inequality used that we only sum over sets with $|S| \leq |U| \leq k$ (in this inequality Equation (30) plays a role analogous to Equation (29) in the earlier argument).

Combining the bounds on these two parts, we get that

$$\left| \text{corr} \left(\left(g_{\text{ave}}^{\mathbf{I}^*} \right)^{\overline{C^*}}, \mathcal{J}_k \right) - \text{corr} (g, \mathcal{J}_k) \right| \leq \frac{\varepsilon}{2}$$

as desired, and [Claim 75](#) is proved. □

This concludes the proof of [Lemma 74](#). □

6.4.5 Proof of [Theorem 3](#)

We can finally reap the benefits of our labor and prove [Theorem 3](#).

Proof of [Theorem 3](#). We run `TOLERANT-JUNTA-TESTER`($f, k, \varepsilon/2$) to compute a value γ . If the number of queries it makes exceeds $k \times$ (the bound on the expected number of calls to f given in [Lemma 71](#)), then we abort the execution of the algorithm and set $\gamma = 0$. We then output $\frac{1}{2}(1 - \gamma)$ as our estimate of $\text{dist}(f, \mathcal{J}_k)$.

By [Lemma 71](#), we always make at most $2^{k^{1/3} \text{polylog}(k/\varepsilon)}$ queries. By Markov’s inequality, the probability that we abort is at most $1/k$. Assuming we do not abort, by [Lemma 73](#) and [Lemma 74](#), we have that

$$|\gamma - \text{corr}(f, \mathcal{J}_k)| \leq 2\varepsilon \tag{52}$$

with probability $1 - o(1)$. So assuming that we do not abort and that [Equation \(52\)](#) holds (which happens with overall probability $1 - o(1)$), we correctly estimate the junta distance $\text{dist}(f, \mathcal{J}_k)$ of f to within additive error $\pm\varepsilon$, where we used the fact that

$$\text{dist}(f, \mathcal{J}_k) = \frac{1}{2}(1 - \text{corr}(f, \mathcal{J}_k)).$$

This completes the proof of the theorem. □

Acknowledgements

X.C. is supported by NSF grants IIS-1838154, CCF-2106429, and CCF-2107187. S.P. is supported by NSF grants CCF-2106429, CCF-2107187, CCF-2218677, ONR grant ONR-13533312, and a NSF Graduate Student Fellowship. R.A.S. is supported in part by NSF awards CCF-2106429 and CCF-2211238.

References

- [BCE⁺19] Eric Blais, Clément L. Canonne, Talya Eden, Amit Levi, and Dana Ron. Tolerant junta testing and the connection to submodular optimization and function isomorphism. *ACM Trans. Comput. Theory*, 11(4):24:1–24:33, 2019.
- [BGMdW13] Harry Buhrman, David García-Soriano, Arie Matsliah, and Ronald de Wolf. The non-adaptive query complexity of testing k-parities. *Chic. J. Theor. Comput. Sci.*, 2013, 2013.
- [Bla08] Eric Blais. Improved bounds for testing juntas. In *Proc. RANDOM*, pages 317–330, 2008.

- [Bla09] Eric Blais. Testing juntas nearly optimally. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 2009.
- [BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47:549–595, 1993. Earlier version in STOC’90.
- [Blu03] A. Blum. Machine learning: a tour through some favorite results, directions, and open problems. FOCS 2003 tutorial slides, available at <http://www-2.cs.cmu.edu/~avrim/Talks/FOCS03/tutorial.ppt>, 2003.
- [BS80] Jon Louis Bentley and James B. Saxe. Generating sorted lists of random numbers. *ACM Transactions on Mathematical Software*, 6(3):359–364, September 1980.
- [Bsh19] Nader H. Bshouty. Almost optimal distribution-free junta testing. In Amir Shpilka, editor, *34th Computational Complexity Conference, CCC 2019, July 18–20, 2019, New Brunswick, NJ, USA*, volume 137 of *LIPICs*, pages 2:1–2:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- [CDL⁺24] Xi Chen, Anindya De, Yuhao Li, Shivam Nadimpalli, and Rocco A Servedio. Mildly exponential lower bounds on tolerant testers for monotonicity, unateness, and juntas. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4321–4337. SIAM, 2024.
- [CG04] H. Chockler and D. Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, 90(6):301–305, 2004.
- [CP23] Xi Chen and Shyamal Patel. New lower bounds for adaptive tolerant junta testing. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1778–1786. IEEE, 2023.
- [CST⁺18] Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the query complexity of non-adaptive junta testing. *J. ACM*, 65(6):40:1–40:18, 2018.
- [DDS14] Anindya De, Ilias Diakonikolas, and Rocco A Servedio. Learning from satisfying assignments. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 478–497. SIAM, 2014.
- [DKR25] Ilias Diakonikolas, Daniel M. Kane, and Lisheng Ren. Faster algorithms for agnostically learning disjunctions and their implications. *CoRR*, abs/2504.15244, 2025.
- [DMN19] Anindya De, Elchanan Mossel, and Joe Neeman. Junta correlation is testable. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9–12, 2019*, pages 1549–1563. IEEE Computer Society, 2019.
- [Fel12] Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer and System Sciences*, 78(5):1444–1459, 2012.
- [FKR⁺04] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. *J. Computer & System Sciences*, 68(4):753–787, 2004.

- [Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.
- [GKK20] Aravind Gollakota, Sushrut Karmalkar, and Adam R. Klivans. The polynomial method is universal for distribution-free correlational SQ learning. *CoRR*, abs/2010.11925, 2020.
- [Hau92] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.
- [HMP⁺93] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46:129–154, 1993.
- [ITW21] Vishnu Iyer, Avishay Tal, and Michael Whitmeyer. Junta distance approximation with sub-exponential queries. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPICs*, pages 24:1–24:38. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- [Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.
- [KKMS08] Adam Tauman Kalai, Adam R Klivans, Yishay Mansour, and Rocco A Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.
- [KLS96] Jeff Kahn, Nathan Linial, and Alex Samorodnitsky. Inclusion-exclusion: Exact and approximate. *Combinatorica*, 16:465–477, 1996.
- [KS04] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. *Journal of Computer & System Sciences*, 68(2):303–318, 2004.
- [KS07] Adam R. Klivans and Alexander A. Sherstov. A lower bound for agnostically learning disjunctions. In *Conference on Learning Theory (COLT’07)*, pages 409–423, 2007.
- [KSS94] M. J. Kearns, R. E. Schapire, and L. Sellie. Toward Efficient Agnostic Learning. *Machine Learning*, 17(2-3):115–141, 1994.
- [LN90] N. Linial and N. Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10(4):349–365, 1990.
- [LW19] Amit Levi and Erik Waingarten. Lower bounds for tolerant junta and unateness testing via rejection sampling of graphs. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, volume 124 of *LIPICs*, pages 52:1–52:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- [Mel57] Herman Melville. *The Confidence-Man: His Masquerade*. Dix, Edwards & Co., 1857. Available at <https://www.gutenberg.org/files/21816/21816-h/21816-h.htm>.
- [NC16] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition)*. Cambridge University Press, 2016.

- [NP24] Shivam Nadimpalli and Shyamal Patel. Optimal non-adaptive tolerant junta testing via local estimators. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1039–1050, 2024.
- [O’D14] R. O’Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.
- [Pat92] R. Paturi. On the degree of polynomials that approximate symmetric Boolean functions. In *Proceedings of the 24th Symposium on Theory of Computing*, pages 468–474, 1992.
- [PRR06] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006.
- [PRW22] Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. Approximating the distance to monotonicity of boolean functions. *Random Struct. Algorithms*, 60(2):233–260, 2022.
- [Sağ18] Mert Sağlam. Near log-convexity of measured heat in (discrete) time and consequences. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 967–978. IEEE Computer Society, 2018.
- [Sch90] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
- [Ser10] Rocco A Servedio. Testing by implicit learning: A brief survey. *Property testing: current research and surveys*, pages 197–210, 2010.
- [Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997.
- [STW15] R.A. Servedio, L.-Y. Tan, and J. Wright. Adaptivity helps for testing juntas. In *Proceedings of the 30th IEEE Conference on Computational Complexity*, pages 264–279, 2015.
- [Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.