Learning and Testing Junta Distributions with Subcube Conditioning

Xi Chen XICHEN@CS.COLUMBIA.EDU
Columbia University.

Rajesh Jayaram RKJAYARA@CS.CMU.EDU
Carnegie Mellon University.

Amit Levi AMIT.LEVI@UWATERLOO.CA
University of Waterloo.

Erik Waingarten EAW@CS.COLUMBIA.EDU
Stanford University.

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

We study the problems of learning and testing junta distributions on $\{-1,1\}^n$ with respect to the uniform distribution, where a distribution p is a k-junta if its probability mass function $p(x)$ depends on a subset of at most k variables. The main contribution is an algorithm for finding relevant coordinates in a k-junta distribution with subcube conditioning Bhattacharyya and Chakraborty (2018); Canonne et al. (2019). We give two applications:

- An algorithm for learning k-junta distributions with $\tilde{O}(k/\epsilon^2)\log n + O(2^k/\epsilon^2)$ subcube conditioning queries, and
- An algorithm for testing k-junta distributions with $\tilde{O}((k+\sqrt{n})/\epsilon^2)$ subcube conditioning queries.

All our algorithms are optimal up to poly-logarithmic factors.

Our results show that subcube conditioning, as a natural model for accessing high-dimensional distributions, enables significant savings in learning and testing junta distributions compared to the standard sampling model. This addresses an open question posed by Aliakbarpour et al. (2016).

Keywords: List of keywords

1. Introduction

We consider the problems of *learning and testing k-junta distributions*, as first studied by Aliakbarpour, Blais, and Rubinfeld (Aliakbarpour et al. (2016)). Given $n \in \mathbb{N}$ and $k \le n$, a distribution p supported on $\{-1,1\}^n$ is a k-junta distribution (with respect to the uniform distribution) if the probability mass function $p(x) = \mathbf{Pr}_{\boldsymbol{z}\sim p}[\boldsymbol{z} = x]$ is a k-junta.[1] The goal of the learning problem is to design algorithms which, given access to an unknown k-junta distribution p over $\{-1,1\}^n$, output a hypothesis distribution $\widehat{p}$ that satisfies $d_{\mathrm{TV}}(p,\widehat{p}) \le \epsilon$. In the testing problem, the goal is

[1] We say a function $f(x)$ over $\{-1,1\}^n$ is a k-junta (function) if it depends on a subset of no more than k variables. More generally, Aliakbarpour et al. (2016) defines k-junta distributions with respect to a fixed distribution q. For $n \in \mathbb{N}$, $k \le n$, and a fixed distribution q supported on $\{-1,1\}^n$, a distribution p over $\{-1,1\}^n$ is a k-junta distribution with respect to q if there exist k coordinates $i_1,\ldots,i_k \in [n]$ such that for every $x \in \{-1,1\}^k$, the distributions p and q conditioned on coordinates $i_1,\ldots,i_k$ being set according to x are equal. When q is the uniform distribution, the above definition is equivalent to the requirement that $p(x)$ is a k-junta function.

to design algorithms which, given access to an arbitrary distribution p, can distinguish between p being a k-junta distribution, and being ϵ-far from a k-junta distribution.[2]

The study of computational aspects of juntas has spawned a large body of work (for instance, see Mossel et al. (2003); Fischer et al. (2004); Chockler and Gutfreund (2004); Lipton et al. (2005); Arpe and Reischuk (2007); Arpe and Mossel (2008); Arvind et al. (2009); Valiant (2015); Blais (2008, 2009, 2010); Servedio et al. (2015); Bshouty and Costa (2016); Blais et al. (2019a); Chen et al. (2017); Saglam (2018); Liu et al. (2018); Levi and Waingarten (2019); De et al. (2019); Pallavoor et al. (2020) and references therein). These problems are motivated by the *feature selection* problem in machine learning (see e.g. Guyon and Elisseeff (2003); Liu and Motoda (2012); Chandrashekar and Sahin (2014)), and are classically referred to in theoretical computer science as "learning in the presence of irrelevant information" Blum (1994); Blum and Langley (1997). The landmark (open) problem is the "junta problem" Blum (2003); Mossel et al. (2003); Valiant (2015): given an unknown k-junta $f\colon \{-1,1\}^n \to \{-1,1\}$, an algorithm receives independent samples $(\boldsymbol{x}, f(\boldsymbol{x}))$ where $\boldsymbol{x} \sim \{-1,1\}^n$ is uniform, and the task is to learn f (with respect to the uniform distribution). Aliakbarpour et al. (2016) study the analogous problem for distributions: for an unknown k-junta distribution p over $\{-1,1\}^n$, an algorithm receives independent samples $\boldsymbol{x} \sim p$, and the task is to learn p to within small distance in total variation. They obtain an algorithm with sample complexity $\tilde{O}(2^{2k}) \log n/\epsilon^4$ and running time $\tilde{O}(2^{2k}) \min\{n^k, 2^n\}/\epsilon^4$, and observed that any algorithm for learning k-junta distributions may be used to solve the "junta problem." Hence, running time significantly better than n^k (in particular, polynomial upper bounds for $k = O(\log n)$) would constitute a major breakthrough in computational learning theory.

Turning to testing k-junta distributions, Aliakbarpour et al. (2016) give a tight bound of $\tilde{\Theta}(2^{n/2}/\epsilon^2)$ for the number of samples $\boldsymbol{x} \sim p$ needed. We note that this "curse of dimensionality" is not unique to the problem of testing junta distributions, and already appears for the most basic testing task: testing whether a distribution on $\{-1,1\}^n$ is uniform Paninski (2008); Valiant and Valiant (2017), which can be viewed as testing k-junta distributions with $k = 0$. Works addressing this state-of-affairs have proceeded by either analyzing restricted classes of high dimensional distributions Rubinfeld and Servedio (2009); Canonne et al. (2017); Daskalakis and Pan (2017); Daskalakis et al. (2019); Gheissari et al. (2018); Bezáková et al. (2020); Diakonikolas et al. (2019), or by augmenting the oracle Batu et al. (2005); Canonne and Rubinfeld (2014); Canonne et al. (2015); Chakraborty et al. (2016); Acharya et al. (2018); Bhattacharyya and Chakraborty (2018); Onak and Sun (2018).

Membership queries. It has been observed Blum and Langley (1997); Mossel et al. (2003); Blum (2003) that the classic "junta problem" becomes significantly easier when allowing *membership queries*.[3] In particular, a simple algorithm making $O(k \log n/\epsilon)$ queries will find at most k relevant variables such that the function is ϵ-close to a junta function over those variables.[4] For the problem of testing junta functions (with membership queries), the state-of-the-art algorithm Blais (2009) only has query complexity $\tilde{O}(k/\epsilon)$ with no dependency on n. This leads to the following question that motivates our work:

> *What is an appropriate "membership query" model for learning and testing junta distributions, and would such query access admit significant complexity savings?*

[2] Here, two distributions p and q are ϵ-far if $d_{\mathrm{TV}}(p,q) \geq \epsilon$, and p is ϵ-far from being a k-junta distribution if every k-junta distribution is ϵ-far from p.

[3] In learning theory, a membership query refers to an oracle which returns $f(x)$ upon a query $x \in \{-1,1\}^n$.

[4] The algorithm iteratively builds a set $J \subset [n]$ of relevant variables by sampling pairs of points $\boldsymbol{x}, \boldsymbol{y} \sim \{-1,1\}^n$ with $\boldsymbol{x}_J = \boldsymbol{y}_J$; when $f(\boldsymbol{x}) \neq f(\boldsymbol{y})$, the algorithm performs a binary search to find a new relevant variable to add to J.

Subcube conditioning queries. This paper considers the *subcube conditioning model*, first studied by Bhattacharyya and Chakraborty (2018). A subcube conditioning query on a distribution p over $\{-1,1\}^n$ is specified by a string (or a restriction as we call in the paper) $\rho \in \{-1,1,*\}^n$. The oracle returns a sample $\boldsymbol{x} \sim p$ conditioned on every $i \in [n]$ with $\rho_i \neq *$ having $\boldsymbol{x}_i = \rho_i$. Equivalently, ρ encodes a subcube of $\{-1,1\}^n$ by fixing non-$*$ coordinates in ρ; the oracle returns a sample $\boldsymbol{x} \sim p$ conditioned on $\boldsymbol{x}$ lying in the subcube.[5] When the subcube encoded by ρ is not supported in p, the oracle under the model of Bhattacharyya and Chakraborty (2018) returns a point drawn uniformly from the subcube. We remark that this modeling choice is not important for this paper: our algorithms only make queries ρ that are consistent with a sample x previously drawn from p (i.e., $\rho_i = x_i$ for every non-$*$ coordinate i).[6]

The subcube conditioning model seems particularly appropriate for computational tasks over distributions supported on (high-dimensional) product domains, and was suggested in Canonne et al. (2015) as an open direction for learning and testing distributions over $\{-1,1\}^n$. From the purely theoretical perspective, we find two aspects of subcube conditioning especially compelling. The first is that restrictions of distributions over product domains are themselves distributions over product domains, which enable algorithms and their analyses to proceed recursively. The second is that algorithms may proceed via the method of (random) restrictions, exploiting properties of distributions apparent only by considering subcubes. See more discussions on random restrictions in Section 1.2.

From a practical perspective, subcube conditional queries arise in a number of applications. An important example is sampling from large joins in a relational database. For database joins, subcube conditioning has a natural interpretation: a sample from a join conditioned on a subcube (defined by fixing certain attributes in the join) can be represented as a sample from another join, where conditioning is first applied to each relation individually.[7] Thus, subcube conditional sampling from a join can be implemented in the same time as uniform sampling from a join with a minor overhead. Moreover, efficiently sampling from joins is an important task in database theory Chaudhuri et al. (1999); Acharya et al. (1999); Zhao et al. (2018); Chen and Yi (2020), and can often be implemented substantially faster than the time required to compute the entire query (which may be exponential in the number of relations given as input to the join).

Other query models. We briefly discuss other proposed access oracles for distributions. The evaluation oracle Batu et al. (2005); Canonne and Rubinfeld (2014) allows algorithms to query the probability mass function of an input, in addition to receiving random samples. We note the same "binary search" strategy prescribed for finding relevant variables in a k-junta function works well in this setting, making it too strong for learning juntas. Onak and Sun (2018) considers probability-revealing samples, where the algorithm receives pairs $(\boldsymbol{x}, p(\boldsymbol{x}))$ with $\boldsymbol{x} \sim p$. This model is too

[5] We note that while this paper considers distributions supported on $\{-1,1\}^n$, Bhattacharyya and Chakraborty (2018) study subcube conditioning in a general product domain Σ^n. There, a subcube conditioning query is specified by a sequence of n subsets $A_1 \times \cdots \times A_n$ where each $A_i \subset \Sigma$, and a sample $\boldsymbol{x} \sim p$ conditioned on $\boldsymbol{x}_i \in A_i$ for all $i \in [n]$. Extending results from $\{-1,1\}^n$ to Σ^n is a direction for future work.

[6] This gives our algorithms a flavor of those under the *active learning/testing* model Dasgupta (2005); Settles (2009); Balcan et al. (2012), adapted to the setting of distribution testing: an algorithm can only zoom in onto a subcube using conditioning queries after it is discovered by samples drawn from the distribution. Our lower bounds, on the other hand, apply to the original subcube conditioning model, which only makes them stronger.

[7] For example, a sample from a large multi-way join $J = R_1 \bowtie \cdots \bowtie R_m$ of relations $R_1, \ldots, R_m$ conditioned on fixing a subset of attributes according to a restriction ρ corresponds to a sample from the join query $J' = R'_1 \bowtie \ldots R'_m$, where each R'_i is the restriction of the relation R_i where attributes are fixed according to ρ.

weak for the learning problem, since the reduction of Aliakbarpour et al. (2016) from the k-junta problem to the k-junta distribution problem applies to this oracle as well.[8] Lastly, and most relevant to this paper, is the (general) conditional sampling model, introduced in Chakraborty et al. (2013, 2016); Canonne et al. (2014, 2015), where an algorithm is allowed to specify a (arbitrary) subset A of the domain and receive a sample conditioned on it lying in A. This model is more powerful than subcube conditioning, yet, looking ahead, our lower bounds for learning k-junta distributions will apply to this model as well, showing that conditioning on arbitrary sets $A \subseteq \{-1,1\}^n$ is no more powerful than that on subcubes for the learning problem.

1.1. Our results

Learning k-junta distributions. Our main algorithmic contribution is a procedure that can, given subcube conditioning query access to a k-junta distribution p over $\{-1,1\}^n$, identify a set $J \subset [n]$ of at most k relevant variables such that p is close to a k-junta over J. The number of queries needed to identify each relevant variable, on average, is roughly $\log n/\epsilon^2$. (We emphasize though that the main idea behind the algorithm is not based on binary search; see Section 1.2 for an overview of the algorithm.)

Theorem 1 (Identifying relevant variables) *There is a randomized algorithm, which takes subcube conditioning query access to an unknown distribution p over $\{-1,1\}^n$, an integer $k \in \mathbb{N}$, and a parameter $\epsilon \in (0,1/4]$. The algorithm makes $\tilde{O}(k/\epsilon^2) \cdot \log n$ queries, runs in time $\tilde{O}(k/\epsilon^2) \cdot n \log n$ and outputs a set $\mathbf{J} \subset [n]$ with the following guarantee. If p is a k-junta distribution then $|\mathbf{J}| \le k$ and p is ϵ-close to a junta distribution over variables in $\mathbf{J}$ with probability at least $2/3$.*

It is known as folklore that, once such a set J is identified, the unknown k-junta distribution p can be learnt easily using another batch of $O(2^k/\epsilon^2)$ samples from p and the same amount of running time. Together we obtain the following corollary, showing that subcube conditioning queries enable significant speedup compared to state-of-the-art learning algorithms under the sampling model.

Corollary 2 (Learning junta distributions) *Under the subcube conditioning query model, there is a learning algorithm for k-junta distributions with query complexity $\tilde{O}(k/\epsilon^2) \cdot \log n + O(2^k/\epsilon^2)$ and running time $\tilde{O}(k/\epsilon^2) \cdot n \log n + O(2^k/\epsilon^2)$.*

We show that query complexities of both algorithms are almost tight. Indeed they are almost tight even under the more powerful *general conditioning query model*, which was introduced simultaneously by Chakraborty et al. (2013, 2016) and Canonne et al. (2014, 2015). A general conditioning query to p is specified by an arbitrary subset A of $\{-1,1\}^n$ (which is not necessarily a subcube) and the oracle returns a sample $\boldsymbol{x} \sim p$ conditioned on $\boldsymbol{x} \in A$.

Theorem 3 *Let $0 < \epsilon \le 1/8$, $n \in \mathbb{N}$ and $0 < k \le n-1$. Suppose an algorithm receives as input conditional query access to an unknown k-junta distribution p supported on $\{-1,1\}^n$ and outputs a set $\mathbf{J} \subset [n]$ with $|\mathbf{J}| \le k$ such that with probability at least $4/5$, p is ϵ-close to a junta distribution over $\mathbf{J}$. Then, the algorithm must make $\Omega(\log \binom{n}{k}/\epsilon^2)$ queries.*

[8] In particular, consider an unknown k-junta function $f\colon \{-1,1\}^n \to \{-1,1\}$, and notice that with $\mathrm{poly}(2^k)$ random samples, we may know exactly how many inputs $x \in \{-1,1\}^n$ have $f(x) = 1$. Then, the reduction of Aliakbarpour et al. (2016) constructs the distribution which is uniform over the inputs where $f(x) = 1$, so knowing the probability mass function at these points gives no additional information.

Theorem 4 *Let $0 < \epsilon \leq 1/120$, $n \in \mathbb{N}$ and $0 < k \leq n-1$. Suppose an algorithm receives as input conditional query access to an unknown k-junta distribution p over $\{-1,1\}^n$ and outputs a distribution $\widehat{\boldsymbol{p}}$ such that with probability at least $4/5$, p is ϵ-close to $\widehat{\boldsymbol{p}}$. Then, the algorithm must make $\Omega(\log\binom{n}{k}/\epsilon^2) + \Omega(2^k/\epsilon^2)$ queries.*

Testing k-junta distributions For the problem of testing junta distributions, we obtain matching upper and lower bounds for the query complexity under the subcube conditioning query model.

Theorem 5 (Testing junta distributions) *There is an algorithm, which takes subcube conditioning access to an unknown distribution p over $\{-1,1\}^n$, an integer $k \in \mathbb{N}$, and $\epsilon \in (0, 1/4]$. It makes*

$$\tilde{O}\left(\frac{k+\sqrt{n}}{\epsilon^2}\right)$$

queries, runs in time $\tilde{O}(n(k+\sqrt{n})^2/\epsilon^4)$ and achieves the following guarantee: It accepts with probability at least $2/3$ if p is a k-junta distribution, and rejects with probability at least $2/3$ if p is ϵ-far from a k-junta.

Theorem 6 (Lower bound for junta testing) *There exist two absolute constants $\epsilon_0 > 0$ and $C_0 \in \mathbb{N}$ such that for any setting of $0 < \epsilon \leq \epsilon_0$, $n \geq C_0$ and $0 \leq k \leq n/2$, any algorithm which receives as input subcube conditioning query access to an unknown distribution p supported on $\{-1,1\}^n$ and distinguishes with probability at least $2/3$ between the case when p is a k-junta distribution and the case when p is ϵ-far from any k-junta distribution must make at least $\tilde{\Omega}(k+\sqrt{n})/\epsilon^2$ many queries. Furthermore, the lower bound holds even when p is promised to be a product distribution.*

An open problem posed by Aliakbarpour et al. (2016) is whether their exponential lower bound for testing junta distributions under the sampling oracle can be bypassed using general conditioning queries. We answer the question positively with subcube conditioning queries.

1.2. Technical overview

We give an overview of our results for learning and testing junta distributions. All our algorithms heavily use *random restrictions* drawn using samples from the unknown distribution. We start with some notation for restrictions and how we apply them on a distribution.

Let p be a distribution over $\{-1,1\}^n$ and let $\rho \in \{-1,1,*\}^n$ be a restriction. We write $p_{|\rho}$ to denote the distribution obtained by applying the restriction ρ on p: it is supported on $\{-1,1\}^{\text{stars}(\rho)}$ where $\text{stars}(\rho)$ is the set of $i \in [n]$ with $\rho_i = *$, and $\boldsymbol{y} \sim p_{|\rho}$ is drawn by first drawing $\boldsymbol{x} \sim p$ conditioned on $\boldsymbol{x}_i = \rho_i$ for all $i \notin \text{stars}(\rho)$ and then setting $\boldsymbol{y} = \boldsymbol{x}_{\text{stars}(\rho)}$. There will be mainly two ways we draw a random restriction $\boldsymbol{\rho}$. In the first scenario, we fix a set $S \subset [n]$ and draw a random restriction $\boldsymbol{\rho}$ by first drawing $\boldsymbol{x} \sim p$ and then setting $\boldsymbol{\rho}_i = \boldsymbol{x}_i$ for each $i \notin S$ and $\boldsymbol{\rho}_i = *$ otherwise. We denote this distribution of restrictions by $\mathcal{D}_S(p)$. The more sophisticated way of drawing a random restriction $\boldsymbol{\rho}$, given a parameter $\sigma \in (0,1)$, is to first draw $\boldsymbol{x} \sim p$ and a random set $\mathbf{S} \subseteq [n]$ by including each element independently with probability σ. We then set $\boldsymbol{\rho}_i = \boldsymbol{x}_i$ for each $i \notin \mathbf{S}$ and $\boldsymbol{\rho}_i = *$ otherwise. We denote this distribution of restrictions by $\mathcal{D}_\sigma(p)$

Algorithm for identifying relevant variables. Given access to a distribution p, the algorithm proceeds by maintaining a set J (initially empty) of relevant[9] variables found, and iteratively adding

[9] Unlike the Boolean function setting, we only know that variables in J are relevant with high probability.

to J until no more relevant variables are found. Hence, the key challenge is discovering new relevant variables when p remains ϵ-far from any k-junta distribution over J. The latter condition implies

$$\mathbf{E}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\left[d_{\mathrm{TV}}\big(p_{|\boldsymbol{\rho}},\mathcal{U}\big)\right]\geq\epsilon,$$

where $\mathcal{U}$ denotes the uniform distribution (of the right dimension). Assume, for convenience, that the algorithm samples a restriction ρ with $d_{\mathrm{TV}}(p_{|\rho},\mathcal{U})\geq\epsilon$. The major difficulty is that arbitrary correlations among (yet unknown) k relevant variables may hide the non-uniform nature of $p_{|\rho}$.[10] For this, we leverage a set of recently-developed tools from Canonne et al. (2019) for analyzing mean vectors of random restrictions of distributions. Specifically, for an arbitrary distribution p over $\{-1,1\}^n$, we denote $\mu(p)\in[-1,1]^n$ as the *mean vector*,

$$\mu(p)\stackrel{\mathrm{def}}{=}\mathop{\mathbf{E}}_{\boldsymbol{x}\sim p}[\boldsymbol{x}]\in[-1,1]^n.$$

We prove the following structural lemma for distributions which are far-from k-juntas. At a high level, this lemma allows us to find relevant variables by only considering the marginal distributions on specific coordinates after applying random restrictions.

Lemma 7 (Main structural lemma) *There is a universal constant $c>0$ such that the following holds. Let p be any probability distribution supported over $\{-1,1\}^n$ for some $n\in\mathbb{N}$. Let $J\subset[n]$ be a subset of variables such that p is ϵ-far from being a junta distribution over variables in J for some $\epsilon\in(0,1/4]$.[11] Then for $\sigma=1/2$ we have*

$$\sum_{j=1}^{\lceil\log_2 2n\rceil}\mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}\left[\big\|\mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}})\big\|_2\right]\right]\geq\frac{\epsilon}{\log^c(n/\epsilon)}. \tag{1}$$

We will apply the main structural lemma to the distribution p projected onto its k relevant variables (so n in Lemma 7 becomes k), which suggests the following algorithm: for each $j=1,\ldots,\lceil\log_2 2k\rceil$, draw $\boldsymbol{\rho}$ and $\boldsymbol{\nu}$ as described above in the hopes that $\|\mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}})\|_2\geq\epsilon/\log^c(k/\epsilon)$. Once this occurs, since $\mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}})$ contains at most k non-zero coordinates, at least one coordinate $i\in\mathrm{stars}(\boldsymbol{\nu})$ will have mean at least $\epsilon/(\sqrt{k}\log^c(k/\epsilon))$ in magnitude. In other words, the i-th variable is relevant, and the marginal distribution on the i-th coordinate of $(p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}$ is biased by at least $\tilde{\Omega}(\epsilon/\sqrt{k})$. Taking $\tilde{O}(k/\epsilon^2)\cdot\log n$ random samples from $(p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}$ is enough to identify all relevant coordinates whose marginal is at least $\tilde{\Omega}(\epsilon/\sqrt{k})$ to include into J; furthermore, (by the extra $(\log n)$-factor), we never include a non-biased coordinate in J. Notice, however, that all guarantees are only in expectation, and we need to employ a budget doubling strategy to achieve the nearly-optimal bound.

[10] For example, consider the k-junta distribution p over $\{-1,1\}^n$ which is parameterized by a subset $S\subset[n]$ of size k (denoting the relevant variables). A sample $\boldsymbol{x}\sim p$ is uniform over all points $y\in\{-1,1\}^n$ where $\prod_{i\in S}y_i=1$. Notice that $d_{\mathrm{TV}}(p,\mathcal{U})\geq 1/2$, however, the distribution given by projecting p onto any subset of coordinates which does not completely include all S variables is exactly uniform. The silver lining (for this specific distribution) will be that if a restriction ρ fixes all but one variable in S, i.e., $S\cap\mathrm{stars}(\rho)=\{i\}$, then every sample $\boldsymbol{x}\sim p_{|\rho}$ will have $\boldsymbol{x}_i$ always set to the same value.

[11] We require $\epsilon\leq 1/4$ just so that $\log(n/\epsilon)\geq 2$ even when $n=1$; this helps avoid an extra multiplicative constant needed on the right hand side of (1).

Algorithm for testing junta distributions. The testing algorithm first runs the algorithm for identifying relevant variables, and then tests whether the distribution depends only on the relevant variables found. In particular, let J be the set of variables it returns, and notice that the algorithm may immediately reject if $|J| > k$, since every variable in J found by the algorithm is relevant (with high probability). The remaining task is distinguishing between the following two cases:

1. If p is ϵ-far from k-junta distributions, then by definition p is ϵ-far from any junta distribution over J. By the main structural lemma, there is some $j = 1, \dots, \lceil \log_2 2n \rceil$ such that $\|\mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}})\|_2$ is large (in expectation) when $\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p)$ and $\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})$.

2. If p is a k-junta distribution, then for every $j = 1, \dots, \lceil \log_2 2n \rceil$, $(p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}$ will (trivially) still be a k-junta distribution and $\|\mu((p_{|\boldsymbol{\rho}})_{\boldsymbol{\nu}})\|_2$ will tend to be small (in expectation). The intuition for the latter condition is that otherwise, the algorithm for finding relevant variables as sketched above would have identified more variables.

To this end, we design a "robust mean tester" for juntas distributions.

Theorem 8 (Robust mean testing for juntas) *There is an algorithm which, given sample access to a distribution p on $\{-1, 1\}^n$, $k \in \mathbb{N}$ and a parameter $\epsilon \in (0, 1)$, has the following behavior:*

1. *If p is a k-junta distribution with $\|\mu(p)\|_2 \le \epsilon\sqrt{n}/100$, the algorithm returns* "`Is a k-junta`" *with probability at least $2/3$;*

2. *If p is a distribution that satisfies $\|\mu(p)\|_2 \ge \epsilon\sqrt{n}$, the algorithm returns* "`Not a k-junta`" *with probability at least $2/3$.*

Moreover, the algorithms draws

$$q = O\left(\max\left\{\frac{k + \sqrt{n}}{\epsilon^2 n}, \frac{k + \sqrt{n}}{\epsilon\sqrt{n}}\right\}\right) \tag{2}$$

samples from p and runs in time $O(q^2 n)$.

The above theorem improves on a (non-robust) mean tester from Canonne et al. (2019) (which solves the case when $k = 0$) in two ways. The first is that since $k \neq 0$, the case p is a k-junta may have non-zero mean vector, and our algorithm distinguishes a constant factor gap between the ℓ_2-norm of mean vectors.[12] The second is that the algorithm runs in time $O(q^2 n)$ as opposed to $n^{O(\log n)}$, and gives optimal query complexity (whereas the result in Canonne et al. (2019) lost a triply-logarithmic factor).

Lower bounds for identifying relevant variables and learning junta distributions. Both proofs of Theorem 3 and Theorem 4 follow from a reduction to the one-way communication complexity of the indexing problem: Alice receives a uniformly random string $\boldsymbol{y} \sim \{-1, 1\}^m$; Bob receives a uniformly random index $\boldsymbol{i} \sim [m]$; Alice needs to send a message to Bob so that Bob

[12] This gives the robust mean tester a somewhat tolerant testing flavor. Removing the assumption of p being a k-junta in the completeness case, and allowing arbitrary distributions with small ℓ_2-norms on the mean vector would result in an $\Omega(1/\epsilon^2)$ lower bound (which is always much higher than (2)). Proof: for $x \in \{-1, 1\}^n$, let p_1 and p_2 be distributions over $\{x, -x\}$ where p_1 is uniform and p_2 samples x with probability $(1 + \epsilon)/2$. These exhibit a gap in the mean vectors, but are indistinguishable with significantly fewer than $1/\epsilon^2$ samples.

outputs $\boldsymbol{y_i}$. This problem has a well known $\Omega(m)$ lower bound for any public-coin protocol that succeeds with probability at least $2/3$ Miltersen et al. (1995).

We focus on Theorem 3, as the proof of Theorem 4 follows a similar plan. We assume that there is an algorithm $\mathcal{A}$ for identifying relevant variables of any k-junta distribution p over $\{-1,1\}^n$ with q general conditioning queries, and similarly to Blais et al. (2019b), we will give a communication protocol which simulates $\mathcal{A}$ to contradict communication complexity lower bounds. Given an input string $y \in \{-1,1\}^m$ where $m = \Omega(\log \binom{n}{k})$, Alice builds a k-junta distribution p_y over $\{-1,1\}^n$ such that Bob can decode y by learning relevant variables of p_y. By Harsha et al. (2010); Braverman and Garg (2014) (specifically, Corollary 7.7 in Rao and Yehudayoff (2020)) and the nature of distribution p_y, we compress the naive one-way communication protocol (where Alice sends q samples using qn bits) into a public-coin protocol with $O(q\epsilon^2) + O(1)$ communication bits.

Lower bound for testing junta distributions. Our lower bound instances will always consist of product distributions, which simplifies the lower bound proof in two ways. The first way is that subcube conditioning queries may be simulated by random samples, so that it suffices to prove a sample complexity lower bound. The second is that, even uniformity testing (which is the case of $k = 0$), has a lower bound of $\Omega(\sqrt{n}/\epsilon^2)$ samples Canonne et al. (2017, 2019), so that it suffices to prove a lower bound of $\tilde{\Omega}(k)/\epsilon^2$. We prove an $\tilde{\Omega}(n)/\epsilon^2$ sample complexity lower bound for testing k-junta product distributions with $k = n/2$, and extend the result to all $k \leq n/2$ with a padding argument.

The two distributions of "hard" instances, $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$, are quite delicate, as they must simultaneously satisfy the following guarantees. (i) A distribution $\boldsymbol{p} \sim \mathcal{D}_{\text{yes}}$ is an $(n/2)$-junta product distribution with probability at least $1 - o_n(1)$, i.e., $\mu(\boldsymbol{p})$ has at most $n/2$ non-zero coordinates (in particular, these are the relevant coordinates). (ii) A distribution $\boldsymbol{p} \sim \mathcal{D}_{\text{no}}$ is ϵ-far from any $(n/2)$-junta product distribution with probability $1 - o_n(1)$, i.e., letting μ' be $\mu(\boldsymbol{p})$ after zeroing out the top half of coordinates, $\|\mu'\|_2 \geq \epsilon$. (iii) The joint distributions over significantly fewer than n/ϵ^2 samples from a draw $\boldsymbol{p} \sim \mathcal{D}_{\text{yes}}$ and $\boldsymbol{p} \sim \mathcal{D}_{\text{no}}$, respectively, are $o_n(1)$ in total variation distance. The constructions proceed by randomly and independently setting $\mu(\boldsymbol{p})_i$ according to one of two possible distributions (one for $\mathcal{D}_{\text{yes}}$ and one for $\mathcal{D}_{\text{no}}$) such that the first $O(\log n/\log\log n)$ moments of each $\mu(\boldsymbol{p})_i$ match when $\boldsymbol{p} \sim \mathcal{D}_{\text{yes}}$ and $\boldsymbol{p} \sim \mathcal{D}_{\text{no}}$, which we show suffices for condition (iii).[13]

2. Preliminaries

We use boldface symbols to represent random variables, and non-boldface symbols for fixed values (potentially realizations of these random variables) — see, e.g., $\boldsymbol{\rho}$ versus ρ. Given $n \in \mathbb{N}$, we let $\mathcal{U}_n$ denote the uniform distribution over $\{-1,1\}^n$. Usually, as the support of $\mathcal{U}_n$ will be clear from the context, we will drop the subscript and simply write $\mathcal{U}$. We write $f(n) \lesssim g(n)$ if, for some $c > 0$, $f(n) \leq c \cdot g(n)$ for all $n \geq 1$ (the $\gtrsim$ symbol is defined similarly). We use the notation $\tilde{O}(f(n))$ to denote $O(f(n) \cdot \text{polylog}(f(n)))$, and $\tilde{\Omega}(f(n))$ to denote $\Omega(f(n)/(1 + |\text{polylog}(f(n))|))$. The notation $[k]$ denotes the set of integers $\{1, \ldots, k\}$.

We introduce two useful operations on a distribution p supported on $\{-1,1\}^n$.

[13] The method of matching moments for distribution testing tasks is a well-known technique Raskhodnikova et al. (2009); Valiant (2011), where the core is analyzing the solution of a Vandermonde system to construct hard instances. While our plan proceeds in a similar fashion, the specific technical details are rather intricate. In particular, seemingly innocuous changes to the Vandermonde system result in constructions which would not work.

Definition 9 (Projection) *For any set $S \subseteq [n]$, we write $\overline{S} = [n] \setminus S$ and define the* projected distribution *$p_{\overline{S}}$ supported on $\{-1,1\}^{\overline{S}}$ by letting $\boldsymbol{y} \sim p_{\overline{S}}$ be drawn as $\boldsymbol{y} = \boldsymbol{x}_{\overline{S}}$ for $\boldsymbol{x} \sim p$.*

Definition 10 (Restriction) *We refer to a string $\rho \in \{-1,1,*\}^n$ as a* restriction *and use* $\mathrm{stars}(\rho)$ *to denote the set of indices $i \in [n]$ with $\rho_i = *$. We denote by $p_{|\rho}$ the* restricted *distribution supported on $\{-1,1\}^{\mathrm{stars}(\rho)}$ given by $\boldsymbol{x}_{\mathrm{stars}(\rho)}$ where $\boldsymbol{x}$ is drawn from p* conditioned *on every $i \notin \mathrm{stars}(\rho)$ being set to ρ_i.*

The majority of the results in this work consider restrictions $\boldsymbol{\rho}$ drawn randomly from one of the distributions that we define next.

Definition 11 *Let $n \in \mathbb{N}$ and p be a distribution supported on $\{-1,1\}^n$. Given a set $S \subseteq [n]$ we let $\mathcal{D}_S(p)$ be the distribution over restrictions $\rho \in \{-1,1,*\}^n$ given by letting $\boldsymbol{\rho} \sim \mathcal{D}_S(p)$ be sampled according to a sample $\boldsymbol{x} \sim p$, and setting for all $i \in [n]$: $\boldsymbol{\rho}_i = *$ if $i \in S$ and $\boldsymbol{\rho}_i = \boldsymbol{x}_i$ if $i \notin S$.*

For any $\sigma \in (0,1)$ and a ground set T, we let $\mathcal{S}_\sigma(T)$ be the distribution supported on subsets $S \subseteq T$ given by letting $\mathbf{S} \sim \mathcal{S}_\sigma(T)$ be the set which includes each $i \in T$ in $\mathbf{S}$ independently with probability σ. We oftentimes write $\mathcal{S}_\sigma = \mathcal{S}_\sigma([n])$ when n is clear from context. We let $\mathcal{D}_\sigma(p)$ be the distribution supported on restrictions $\{-1,1,*\}^n$ given by letting $\boldsymbol{\rho} \sim \mathcal{D}_\sigma(p)$ be sampled by first sampling $\mathbf{S} \sim \mathcal{S}_\sigma$ and then outputting $\boldsymbol{\rho} \sim \mathcal{D}_{\mathbf{S}}(p)$.

3. Finding Relevant Variables

In this section we give our algorithm for identifying relevant variables from junta distributions. We restate our main structural lemma but delay its proof to Section F.

Lemma 12 (Main structural lemma) *There is a universal constant $c > 0$ such that the following holds. Let p be any probability distribution supported over $\{-1,1\}^n$ for some $n \in \mathbb{N}$. Let $J \subset [n]$ be a subset of variables such that p is ϵ-far from being a junta distribution over variables in J for some $\epsilon \in (0,1/4]$.*[14] *Then for $\sigma = 1/2$ we have*

$$\sum_{j=1}^{\lceil \log_2 2n \rceil} \mathop{\mathbf{E}}_{\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p)} \left[\mathop{\mathbf{E}}_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})} \left[\left\| \mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}) \right\|_2 \right] \right] \geq \frac{\epsilon}{\log^c(n/\epsilon)}. \tag{1}$$

We emphasize that the parameter n in our structural lemma will be set to be the junta parameter k later so we need it to hold for small n such as $n = 1$, which requires some care in its proof later.

We restate the main theorem of this section:

Theorem 1 (Identifying relevant variables) *There is a randomized algorithm, which takes subcube conditioning query access to an unknown distribution p over $\{-1,1\}^n$, an integer $k \in \mathbb{N}$, and a parameter $\epsilon \in (0,1/4]$. The algorithm makes $\tilde{O}(k/\epsilon^2) \cdot \log n$ queries, runs in time $\tilde{O}(k/\epsilon^2) \cdot n \log n$ and outputs a set $\mathbf{J} \subset [n]$ with the following guarantee. If p is a k-junta distribution then $|\mathbf{J}| \leq k$ and p is ϵ-close to a junta distribution over variables in $\mathbf{J}$ with probability at least $2/3$.*

Theorem 1 will follow by combining the main algorithmic component, Lemma 13 stated next, with the main structural lemma (Lemma 7).

[14] We require $\epsilon \leq 1/4$ just so that $\log(n/\epsilon) \geq 2$ even when $n = 1$; this helps avoid an extra multiplicative constant needed on the right hand side of (1).

Lemma 13 *There exists a randomized algorithm,* `FindRelevantVariables`*, which takes subcube conditional query access to an unknown distribution p supported on $\{-1,1\}^n$, an integer $k \in \mathbb{N}$ and a parameter $\epsilon \in (0, 1/4]$. The algorithm makes $\tilde{O}(k/\epsilon^2) \cdot \log n$ queries and outputs a set $\mathbf{J} \subset [n]$ that satisfies the following guarantees:*

1. *With probability at least $8/9$, for every $i \in \mathbf{J}$, there is a restriction $\rho \in \{-1, 1, *\}^n$ with $i \in \mathrm{stars}(\rho)$ such that $\mu(p_{|\rho})_i \neq 0$ (and thus, i is a relevant variable of p);*

2. *Suppose p is a k-junta distribution and let $\sigma = 1/2$. With probability at least $8/9$, $\mathbf{J}$ satisfies*

$$\mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{\mathbf{J}}}(p)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}\left[\left\|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)\right\|_2\right]\right] \leq \epsilon, \qquad \textit{for every } j = 1, \ldots, \lceil \log_2 2k \rceil. \tag{3}$$

Proof of Theorem 1 assuming Lemma 13: We execute `FindRelevantVariables`$(p, k, \tilde{\epsilon})$ for some parameter $\tilde{\epsilon}$ to be specified shortly, and upon receiving $\mathbf{J} \subset [n]$ outputs $\mathbf{J}$. We show that when p is a k-junta distribution, $\mathbf{J}$ satisfies the condition of Theorem 1 with probability at least $2/3$. For this purpose it suffices to show that the condition of Theorem 1 follows from the two conditions of Lemma 13 when $\tilde{\epsilon}$ is set appropriately.

Let $J \subset [n]$ be a set of variables for which both conditions of Lemma 13 hold (with $\tilde{\epsilon}$ on the right hand side in (2) instead of ϵ). Since p is a k-junta, we let $I = \{i_1, \ldots, i_k\} \subset [n]$ and $g\colon \{-1,1\}^k \to [0,1]$ be such that $p(x) = g(x_{i_1}, \ldots, x_{i_k})$. By the first condition, we have $J \subseteq I$ and $|J| \leq k$, since a restriction $\rho \in \{-1, 1, *\}^n$ with $i \in \mathrm{stars}(\rho)$ and $\mu(p_{|\rho})_i \neq 0$ certifies that each $i \in J$ is a relevant variable in p. Next consider the distribution $h = p_I$ supported on $\{-1,1\}^I$ and suppose for the sake of contradiction that h is ϵ-far from being a junta over variables in J. Then by applying Lemma 7 on h and J with $\sigma = 1/2$ (and noting that parameter n in Lemma 7 is set to k), we have

$$\frac{\epsilon}{\log^c(k/\epsilon)} \leq \sum_{j=1}^{\lceil \log_2 2k \rceil} \mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(h)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(h_{|\boldsymbol{\rho}})}\left[\left\|\mu\big((h_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)\right\|_2\right]\right], \tag{4}$$

where $c > 0$ is the universal constant from Lemma 7.

On the other hand, we claim that the right hand side of the inequality above is the same as

$$\sum_{j=1}^{\lceil \log_2 2k \rceil} \mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}\left[\left\|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)\right\|_2\right]\right],$$

after replacing h with p. This is because p is a k-junta over I and thus, the mean vector of $(p_{|\rho})_{|\nu}$ for any restrictions ρ and ν always has zeros in entries outside of those in I. As a result, we have

$$\frac{\epsilon}{\log^c(k/\epsilon)} \leq \sum_{j=1}^{\lceil \log_2 2k \rceil} \mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}\left[\left\|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)\right\|_2\right]\right] \leq \lceil \log_2 2k \rceil \cdot \tilde{\epsilon},$$

where we used the second condition of Lemma 13. Hence, choosing $\tilde{\epsilon} = \epsilon/\mathrm{polylog}(k/\epsilon)$ gives us a contradiction. This shows that h is ϵ-close to being a junta over variables in J. Since p is a junta over I and $h = p_I$, p is ϵ-close to being a junta over variables in J as well.

```
Subroutine FindRelevantVariables$(p, k, \epsilon)$

Input: Subcube conditioning access to a distribution $p$ supported on $\{-1,1\}^n$, an integer
$k \in \mathbb{N}$ and a proximity parameter $\epsilon \in (0,1)$.
Output: A set $J \subset [n]$ of variables.

  1. Initialize $J = \emptyset$ (and $B = 0$, which is used only in the analysis), and let
         $$\epsilon_0 = \frac{\epsilon}{100 \cdot \log^3(k/\epsilon)}.$$
  2. Execute the following while $|J| \le k$:
      (a) Initialize $b = 1$.
      (b) Repeat the following procedure while $b \le 2k$:
            Increase $B$ by $b$; run VariablesBudget$(p, k, \epsilon_0, b, J)$, which outputs
              $J' \subset [n] \setminus J$.
              A. If $|J'| \ge b$, update $J$ by adding $b$ elements of $J'$ to $J$ and go to step 2.
              B. If $|J'| < b$, update $b \leftarrow 2b$ and repeat the loop of step 2b.
      (c) If $b > 2k$, output $J$.
  3. Output $J$.
```

Figure 1: The FindRelevantVariables subroutine.

To finish the proof we note that the bound on the query complexity follows from the fact that we executed FindRelevantVariables$(p, k, \tilde{\epsilon})$ with $\tilde{\epsilon}$ picked as above. ■

We present FindRelevantVariables in Figure 1. It uses a subroutine VariablesBudget which we describe in Figure 2 and analyze in the lemma below, whose proof deferred to Appendix A.

Lemma 14 *There exists a randomized algorithm,* VariablesBudget, *which takes subcube conditional query access to an unknown distribution p over $\{-1,1\}^n$, an integer $k \in \mathbb{N}$, a parameter $\epsilon \in (0, 1/4]$, an integer $b \in [k]$, and a set $J \subset [n]$. It makes*

$$O\left(\frac{b}{\epsilon^2} \cdot \log^2\left(\frac{k}{\epsilon}\right) \cdot \log\left(\frac{n}{\epsilon}\right)\right)$$

subcube conditional queries, and outputs a set $\mathbf{J}' \subset [n] \setminus J$ satisfying the following guarantees:

1. *With probability at least $1 - (\epsilon/n)^9$, for every coordinate $i \in \mathbf{J}'$, there exists a restriction $\rho \in \{-1, 1, *\}^n$ with $i \in \mathrm{stars}(\rho)$ such that $\mu(p_\rho)_i \neq 0$.*

Subroutine `VariablesBudget` (p, k, ϵ, b, J)

Input: Subcube conditioning access to a distribution p supported on $\{-1,1\}^n$, an integer $k \in \mathbb{N}$, a proximity parameter $\epsilon \in (0, 1/4]$, a parameter $b \in [k]$ and a set $J \subset [n]$.
Output: A set $J' \subset [n] \setminus J$ which either has size at least b, or is empty.

- Repeat the following for $j \in [\lceil \log_2 2k \rceil]$ and $a \in \{0, \ldots, \lfloor \log_2(\sqrt{b}/\epsilon) \rfloor\}$ with $\alpha = 2^{-a}$:

  Sample t_α many pairs $\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p)$ and $\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})$, where

  $$t_a = 100 \cdot 2^a \cdot \log(k/\epsilon) = 100 \cdot \log(k/\epsilon)/\alpha$$

  (a) For each sampled pair $(\boldsymbol{\rho}, \boldsymbol{\nu})$, take s_a samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{s_a} \sim (p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}$ with

  $$s_a = 100 \cdot \left(\frac{\alpha^2 b}{\epsilon^2}\right) \cdot \log\left(\frac{n}{\epsilon}\right) \tag{6}$$

  (noting $\alpha^2 b/\epsilon^2 \geq 1$) and let $\widehat{\mu} \in \mathbb{R}^{\mathrm{stars}(\boldsymbol{\nu})}$ be their empirical mean given by

  $$\widehat{\mu} = \frac{1}{s_a} \sum_{\ell=1}^{s} \boldsymbol{x}_\ell.$$

  (b) Let $\mathbf{J}'$ be the set of coordinates $i \in \mathrm{stars}(\boldsymbol{\nu})$ satisfying

  $$|\widehat{\mu}_i| \geq \frac{\epsilon}{2\alpha\sqrt{b}}$$

  and output $\mathbf{J}'$ if $|\mathbf{J}'| \geq b$.

- If we have not yet produced an output at the end of the main loop, output $\emptyset$.

Figure 2: The `VariablesBudget` subroutine.

2. *If there exist $j \in [\lceil \log_2 2k \rceil]$ and a real number $\alpha > 0$ such that*[15]

$$\Pr_{\substack{\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p) \\ \boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}} \left[\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big) \textit{ contains at least } b \textit{ coordinates of magnitude} \geq \frac{\epsilon}{\alpha\sqrt{b}} \right] \geq \alpha \tag{5}$$

then the set $\mathbf{J}'$ has size at least b with probability at least $1 - (\epsilon/k)^9$.

[15] Note that a trivial necessary condition for the inequality to hold is $\alpha \leq 1$ and $\alpha \geq \epsilon/\sqrt{b}$.

References

Jayadev Acharya, Arnab Bhattacharyya, Constantinos Daskalakis, and Saravanan Kandasamy. Learning and testing causal models with interventions. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS '2018)*, 2018.

Swarup Acharya, Phillip B Gibbons, Viswanath Poosala, and Sridhar Ramaswamy. Join synopses for approximate query answering. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 275–286, 1999.

Maryam Aliakbarpour, Eric Blais, and Ronitt Rubinfeld. Learning and testing junta distributions. In *Proceedings of the 29th Annual Conference on Learning Theory (COLT '2016)*, pages 19–46, 2016.

Jan Arpe and Elchanan Mossel. Agnostically learning juntas from random walks. *arXiv preprint arXiv:0806.4210*, 2008.

Jan Arpe and Rüdiger Reischuk. Learning juntas in the presence of noise. *Theoretical Computer Science*, 384(1):2–21, 2007.

Vikraman Arvind, Johannes Köbler, and Wolfgang Lindner. Parameterized learnability of juntas. *Theoretical Computer Science*, 410(47-49):4928–4936, 2009.

Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '2012)*, pages 21–30, 2012.

Tugkan Batu, Sanjoy Dasgupta, Ravi Kumar, and Ronitt Rubinfeld. The complexity of approximating the entropy. *SIAM Journal on Computing*, 35(1):132–150, 2005.

Ivona Bezáková, Antonio Blanca, Zongchen Chen, Daniel Štefankovič, and Eric Vigoda. Lower bounds for testing graphical models: Colorings and antiferromagnetic ising models. *Journal of Machine Learning Research*, 21(25):1–62, 2020.

Rijirash Bhattacharyya and Sourav Chakraborty. Property testing of joint distributions using conditional samples. *ACM Transactions on Computation Theory*, 10(4), 2018.

Eric Blais. Improved bounds for testing juntas. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 317–330. Springer, 2008.

Eric Blais. Testing juntas nearly optimally. In *Proceedings of the 41st ACM Symposium on the Theory of Computing (STOC '2009)*, pages 151–158, 2009.

Eric Blais. Testing juntas: A brief survey. In *Property Testing - Current Research and Surveys*, pages 32–40. 2010.

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 Transactions on Computation Theory*, 11(4):1–33, 2019a.

Eric Blais, Clément L. Canonne, and Tom Gur. Distribution testing lower bounds via reductions from communication complexity. *ACM Transactions on Computation Theory*, 12(2):1–37, 2019b.

Avrim Blum. Relevant examples and relevant features–thoughts from computational learning theory. Technical report, AAAI Fall Symposium on Relevance, 1994.

Avrim Blum. Open problem: learning a function of r relevant variables. In *Proceedings of the 16th Annual Conference on Learning Theory (COLT '2003)*, 2003.

Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

Mark Braverman and Ankit Garg. Public vs private coin in bounded-round information. In *Automata, Languages, and Programming*, pages 502–513, 2014.

Nader H Bshouty and Areej Costa. Exact learning of juntas from membership queries. In *Proceedings of the 27th International Conference on Algorithmic Learning Theory (ALT '2016)*, 2016.

Clement Canonne and Ronitt Rubinfeld. Testing probability distributions underlying aggregated data. In *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP '2014)*, pages 283–295, 2014.

Clement L. Canonne, Dana Ron, and Rocco A. Servedio. Testing equivalence between distributions using conditional samples. In *Proceedings of the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA '2014)*, 2014.

Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing probability distributions using conditional samples. *SIAM Journal on Computing*, 44(3):540–616, 2015.

Clement L. Canonne, Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Testing bayesian networks. In *Proceedings of the 30th Annual Conference on Learning Theory (COLT '2017)*, 2017.

Clement L. Canonne, Xi Chen, Gautam Kamath, Amit Levi, and Erik Waingarten. Random restrictions of high-dimensional distributions and uniformity testing with subcube conditioning, 2019.

Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. In *ITCS2013*, pages 561–580, 2013.

Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. On the power of conditional samples in distribution testing. *SIAM Journal on Computing*, 45(4):1261–1296, 2016.

Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

Surajit Chaudhuri, Rajeev Motwani, and Vivek Narasayya. On random sampling over joins. *ACM SIGMOD Record*, 28(2):263–274, 1999.

Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the query complexity of non-adaptive junta testing. In *Proceedings of the 32nd Conference on Computational Complexity (CCC '2017)*, 2017.

Yu Chen and Ke Yi. Random sampling and size estimation over cyclic joins. In *23rd International Conference on Database Theory, ICDT 2020, March 30-April 2, 2020, Copenhagen, Denmark*, pages 7:1–7:18, 2020.

Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Information Processing Letters*, pages 301–305, 2004.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *Proceedings of Advances in Neural Information Processing Systems*, pages 337–344, 2005.

Constantinos Daskalakis and Qinxuan Pan. Square hellinger subadditivity for bayesian networks and its applications to identity testing. In *Proceedings of the 30th Annual Conference on Learning Theory (COLT '2017)*, pages 697–703, 2017.

Constantinos Daskalakis, Nishanth Dikkala, and Gautam Kamath. Testing ising models. *IEEE Transactions on Information Theory*, 65(11):6829–6852, 2019.

Anindya De, Elchanan Mossel, and Joe Neeman. Junta correlation is testable. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2019')*, 2019.

Ilias Diakonikolas, Daniel M. Kane, and John Peebles. Testing identity of multidimensional histograms. In *Proceedings of the 32nd Annual Conference on Learning Theory (COLT '2019)*, 2019.

Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samordinsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753–787, 2004.

Reza Gheissari, Eyal Lubetzky, and Yuval Peres. Concentration inequalities for polynomials of contracting ising models. *Electronic Communications in Probability*, 23, 2018.

Isabelle Guyon and Andr'e Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. *IEEE Trans. Inf. Theor.*, 56(1):438–449, 2010.

Amit Levi and Erik Waingarten. Lower bounds for tolerant junta and unateness testing via rejection sampling of graphs. In *Proceedings of the 2019 ACM Conference on Innovations in Theoretical Computer Science (ITCS '2019)*, 2019.

Richard J Lipton, Evangelos Markakis, Aranyak Mehta, and Nisheeth K Vishnoi. On the fourier spectrum of symmetric boolean functions with applications to learning symmetric juntas. In *CCC2005*, pages 112–119, 2005.

Huan Liu and Hiroshi Motoda. *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media, 2012.

Zhengyang Liu, Xi Chen, Rocco A Servedio, Ying Sheng, and Jinyu Xie. Distribution-free junta testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 2018.

Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. In *Proceedings of the 27th ACM Symposium on the Theory of Computing (STOC '1995)*, pages 103–111, 1995.

Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. Learning juntas. In *Proceedings of the 35th ACM Symposium on the Theory of Computing (STOC '2003)*, pages 206–212, 2003.

Krzysztof Onak and Xiaorui Sun. Probability–revealing samples. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS '2018)*, 2018.

Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. Approximating the distance to monotonicity of boolean functions. In *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms (SODA '2020)*, 2020.

Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Transactions on Information Theory*, 54(10):4750–4755, 2008.

Anup Rao and Amir Yehudayoff. *Communication Complexity and Applications*. Cambridge University Press, 2020.

Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam D. Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009.

Ronitt Rubinfeld and Rocco A. Servedio. Testing monotone high-dimensional distributions. *Random Structures and Algorithms*, 34(1):24–44, 2009.

Mert Saglam. Near log-convexity of measured heat in (discrete) time and consequences. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2018)*, 2018.

Rocco A Servedio, Li-Yang Tan, and John Wright. Adaptivity helps for testing juntas. In *Proceedings of the 30th Conference on Computational Complexity (CCC '2015)*, pages 264–279, 2015.

Burr Settles. Active learning literature survey. *Computer Sciences Technical Report*, 1648, 2009.

Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *Journal of the ACM*, 62(2):13, 2015.

Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. *Journal of the ACM*, 64(6):37:1–37:41, 2017.

Paul Valiant. Testing symmetric properties of distributions. *SIAM Journal on Computing*, 40(6): 1927–1968, 2011.

Charles F Van Loan. The ubiquitous kronecker product. *Journal of computational and applied mathematics*, 123(1-2):85–100, 2000.

Zhuoyue Zhao, Robert Christensen, Feifei Li, Xiao Hu, and Ke Yi. Random sampling over joins revisited. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1525–1539, 2018.

Appendix A. Proof of Lemma 14

We start with the first condition. We observe that, for the output $\mathbf{J}'$ to violate the condition, there must be an execution of step (a) for some j, a, ρ and ν such that $\mu((p_{|\rho})_{|\nu})_i = 0$ for some $i \in \mathrm{stars}(\nu)$ but the same coordinate in the average of s_a samples drawn from $(p_{|\rho})_{|\nu}$ has magnitude at least $\epsilon/(2\alpha\sqrt{b})$ with $\alpha = 2^{-a}$. Note that this coordinate in the average is just the average of s_a uniformly random bits.

Via a union bound over coordinates and a Chernoff bound, the probability that one round of step (a) gives a $\mathbf{J}'$ in step (b) that violates the condition is at most

$$n \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{s_a}\sim\{-1,1\}}\left[\left|\frac{1}{s}\sum_{\ell=1}^{s_a}\boldsymbol{z}_\ell\right| \geq \frac{\epsilon}{2\alpha\sqrt{b}}\right] \leq 2n\cdot\exp\left(-\frac{s_a\epsilon^2}{8\alpha^2 b}\right) \leq \left(\frac{\epsilon}{n}\right)^{11}. \tag{7}$$

With a union bound over all rounds of (a), the probability of $\mathbf{J}'$ violating the condition is at most

$$\lceil\log_2 2k\rceil\cdot\left(\sum_{a=0}^{\lfloor\log_2(\sqrt{b}/\epsilon)\rfloor}100\cdot 2^a\cdot\log(k/\epsilon)\right)\cdot\left(\frac{n}{\epsilon}\right)^{11} \leq O\left(\frac{\sqrt{b}}{\epsilon}\right)\cdot\log^2\left(\frac{k}{\epsilon}\right)\cdot\left(\frac{\epsilon}{n}\right)^{11} \leq \left(\frac{\epsilon}{n}\right)^{9}.$$

We now turn to the second condition. By assumption there are parameters $j \in [\lceil\log_2 k\rceil]$ and $\alpha^* > 0$ such that (5) holds (which implies that $\epsilon/\sqrt{b} \leq \alpha^* \leq 1$). Let

$$0 \leq a = \lfloor\log(1/\alpha^*)\rfloor \leq \lfloor\log(\sqrt{b}/\epsilon)\rfloor \quad\text{and}\quad \alpha = 2^{-a}$$

so that $\alpha^* \leq \alpha \leq 2\alpha^*$. It suffices to show that during the main loop of `VariablesBudget` with j and a, at least one of the t_a pairs $\boldsymbol{\rho}$ and $\boldsymbol{\nu}$ sampled leads to $\mathbf{J}'$ with $|\mathbf{J}'| \geq b$ with high probability.

For this purpose we say a pair (ρ, ν) of restrictions is *good* if the mean vector of $(p_{|\rho})_{|\nu}$ has at least b coordinates of magnitude at least $\epsilon/(\alpha^*\sqrt{b})$. It follows from (5) that $\boldsymbol{\rho} \in \mathcal{D}_{\overline{J}}(p)$ and $\boldsymbol{\nu} \in \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})$ are good with probability at least α^*. By virtue of step (a) being repeated

$$t_a = 100\cdot\log(k/\epsilon)/\alpha \geq 50\cdot\log(k/\epsilon)/\alpha^*$$

times, we have that with probability at least $1-(\epsilon/k)^{10}$, at least one of the pairs of restrictions $\boldsymbol{\rho}$ and $\boldsymbol{\nu}$ sampled in the main loop of j and a is good.

On the other hand, fix any such good pair (ρ, ν) and any coordinate $i \in \mathrm{stars}(\nu)$ with

$$\left|\mu((p_{|\rho})_\nu)_i\right| \geq \epsilon/(\alpha^*\sqrt{b}) \geq \epsilon/(\alpha\sqrt{b})$$

since $\alpha \geq \alpha^*$. It follows from a Chernoff bound similar to (7) that every such coordinate i is added to $\mathbf{J}'$ with probability at least $1-(\epsilon/n)^{10}$. By a union bound over the two bad events, the main loop with j and a outputs a set of size at least b with probability at least $1-(\epsilon/n)^{10}-(\epsilon/k)^{10} \geq 1-(\epsilon/k)^9$.

Finally, the query complexity is bounded by:

$$\begin{aligned}\lceil\log_2 2k\rceil\cdot\sum_{a=0}^{\lfloor\log_2(\sqrt{b}/\epsilon)\rfloor} t_a s_a &\leq 100^2\cdot\lceil\log_2 2k\rceil\sum_{a=0}^{\lceil\log_2(\sqrt{b}/\epsilon)\rceil}2^a\cdot\log\left(\frac{k}{\epsilon}\right)\cdot\frac{b}{2^{2a}\epsilon^2}\cdot\log\left(\frac{n}{\epsilon}\right)\\ &= O\left(\frac{b}{\epsilon^2}\cdot\log^2\left(\frac{k}{\epsilon}\right)\cdot\log\left(\frac{n}{\epsilon}\right)\right).\end{aligned}$$

as required. This finishes the proof of the lemma.

Finally we use Lemma 14 to analyze `FindRelevantVariables` and prove Lemma 13:

Proof of Lemma 13: To analyze the query complexity, consider an execution of `FindRelevantVariables`(p, k, ϵ). Given that all queries are made in calls to `VariablesBudget`, the number of queries made by the subroutine at any time is captured by

$$B \cdot O\left(\frac{1}{\epsilon_0^2} \cdot \log^2\left(\frac{k}{\epsilon_0}\right) \cdot \log\left(\frac{n}{\epsilon_0}\right)\right) = \frac{B}{\epsilon^2} \cdot \text{polylog}\left(\frac{k}{\epsilon}\right) \cdot \log n.$$

using $\epsilon_0 = \epsilon/\text{polylog}(k/\epsilon)$. So it suffices to show that $B = O(k)$ when the algorithm terminates. To see this is the case we prove by induction that at the end of each loop of (b), we have

$$B \le 2|J| + b.$$

This clearly holds at the beginning (before the first loop of (b)) because $B = 0$, $b = 1$ and $|J| = 0$. For the induction step, note that each iteration of step (b) either (A) increases both B and $|J|$ by b and resets b to 1; or (B) increases B by b, b gets doubled and $|J|$ remains the same. As a result, it suffices to bound b and $|J|$ when the algorithm terminates. If the algorithm terminates because of line (c), then we can bound b by $4k$ and $|J|$ by k; if the algorithm terminates because of line 3, then we can bound b by $2k$ and $|J|$ by $k + b \le 3k$.

In both cases we have $B \le 2|J| + b \le 8k$. This finishes the analysis of the query complexity.

Towards proving the first guarantee, note that the total number of executions of `VariablesBudget` is at most the value of B when the algorithm terminates, and we know from the analysis above that it is bounded by $8k$. We take a union bound over all executions of `VariablesBudget`, and deduce that with probability at least $8/9$, every execution satisfies the first condition in Lemma 14, from which J also satisfies the first condition in Lemma 13 since J only contains coordinates returned by calls to `VariablesBudget`.

To prove the second guarantee, suppose p is a k-junta distribution. We can similarly take a union bound over all executions of `VariablesBudget` and deduce that with probability at least $8/9$, every execution satisfies both conditions in Lemma 14. Let J be the output of `FindRelevantVariables`. Then similar to the argument above, the first condition in Lemma 14 implies that J contains only relevant variables of p and thus, $|J| \le k$. If $|J| = k$, the inequality (3) is immediate since all relevant variables of p have been identified in J and hence for every $\rho \in \text{supp}(\mathcal{D}_{\overline{J}}(p))$, $p_{|\rho}$ is uniform.

Suppose then that $|J| < k$ and note from Figure 1 that the algorithm terminates because of line (c). This implies that for J, step (b) executed `VariablesBudget`(p, k, ϵ_0, b, J) for every $b \le 2k$ being a power of 2 and $|J'| < b$ for every execution. It then follows from the second guarantee of Lemma 14 that, for every $j \in [\lceil \log_2 2k \rceil]$, $b = 2^\beta$ with $\beta = 0, \ldots, \lfloor \log_2 2k \rfloor$ and every $\alpha > 0$, (5) does not hold:

$$\Pr_{\substack{\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p) \\ \boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}} \left[\left| \mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)_i \right| \ge \frac{\epsilon_0}{\alpha\sqrt{b}} \text{ for at least } b \text{ coordinates} \right] \le \alpha. \tag{8}$$

We use (8) to show for each $j \in [\lceil \log_2 2k \rceil]$ that

$$\mathop{\mathbf{E}}_{\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p)} \left[\mathop{\mathbf{E}}_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})} \left[\left\| \mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big) \right\|_2 \right] \right] \le \epsilon.$$

To this end, we use

$$\mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^J}(p_{|\boldsymbol{\rho}})}\left[\big\|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)\big\|_2\right]\right]\le\epsilon_0+\int_{\epsilon_0}^{\sqrt{k}}\mathop{\mathbf{Pr}}_{\boldsymbol{\rho},\boldsymbol{\nu}}\left[\big\|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)\big\|_2\ge\gamma\right]d\gamma \tag{9}$$

and the following claim; the proof is elementary so we delay its proof to the end.

Claim 15 *Let $x\in[-1,1]^k$ with $\|x\|_2\ge\gamma$ for some $\gamma>0$. Let $t=\lfloor\log_2 2k\rfloor$. Then there must be a $\beta=0,1,\ldots,t$ such that the number of $i\in[k]$ with*

$$|x_i|\ge\frac{\gamma}{2\sqrt{2^\beta t}}$$

is at least 2^β.

Letting $t=\lfloor\log_2 2k\rfloor$. Lemma 15 implies that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{\rho},\boldsymbol{\nu}}\left[\big\|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)\big\|_2\ge\gamma\right]\le\sum_{\beta=0}^{t}\mathop{\mathbf{Pr}}_{\boldsymbol{\rho},\boldsymbol{\nu}}\left[\Big|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)_i\Big|\ge\frac{\gamma}{2\sqrt{2^\beta t}}\text{ for at least }2^\beta\text{ coordinates}\right]. \tag{10}$$

Combining (8), (9) and (10), we have that the left hand side of (9) is at most

$$\begin{aligned}\epsilon_0+&\sum_{\beta=0}^{t}\int_{\epsilon_0}^{\sqrt{k}}\mathop{\mathbf{Pr}}_{\boldsymbol{\rho},\boldsymbol{\nu}}\left[\Big|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)_i\Big|\ge\frac{\gamma}{2\sqrt{2^\beta t}}\text{ for at least }2^\beta\text{ coordinates}\right]d\gamma\\ &\le\epsilon_0+2\epsilon_0\sqrt{t}\cdot\sum_{\beta=0}^{t}\int_{\epsilon_0}^{\sqrt{k}}\frac{1}{\gamma}\,d\gamma\le\epsilon_0\left(1+2\sqrt{t}(t+1)\cdot\ln\left(\frac{\sqrt{k}}{\epsilon_0}\right)\right)\le\epsilon,\end{aligned}$$

using our choice of $\epsilon_0=\epsilon/(100\cdot\log^3(k/\epsilon))$. This finishes the proof of the lemma. ■

Proof of Claim 15: Assume for contradiction that this is not the case for every $\beta=0,1,\ldots,t$. In particular, it means that no coordinate has $|x_i|\ge\gamma/(2\sqrt{t})$ using the case with $\beta=0$. Therefore,

$$\gamma^2\le\|x\|_2^2<2\cdot\sum_{\beta=1}^{t}2^\beta\cdot\frac{\gamma^2}{4\cdot2^\beta t}+k\cdot\frac{\gamma^2}{4\cdot2^t t}\le\frac{\gamma^2}{2}+\frac{\gamma^2}{4t}<\gamma^2,$$

a contradiction. ■

Appendix B. Lower Bounds for Learning

The goal of this section is to prove the following lower bounds for the number of subcube conditioning queries needed by an algorithm to solve the following two tasks (1) to learn a set of relevant variables of a k-junta distribution and (2) to learn a distribution.

Note that our lower bounds hold for the general conditioning model Chakraborty et al. (2016); Canonne et al. (2015) which allows the algorithm to condition on arbitrary subsets of the domain $\{-1,1\}^n$, rather that only subcubes.

Theorem 3 *Let $0 < \epsilon \leq 1/8$, $n \in \mathbb{N}$ and $0 < k \leq n-1$. Suppose an algorithm receives as input conditional query access to an unknown k-junta distribution p supported on $\{-1,1\}^n$ and outputs a set $\mathbf{J} \subset [n]$ with $|\mathbf{J}| \leq k$ such that with probability at least $4/5$, p is ϵ-close to a junta distribution over $\mathbf{J}$. Then, the algorithm must make $\Omega(\log \binom{n}{k}/\epsilon^2)$ queries.*

Theorem 4 *Let $0 < \epsilon \leq 1/120$, $n \in \mathbb{N}$ and $0 < k \leq n-1$. Suppose an algorithm receives as input conditional query access to an unknown k-junta distribution p over $\{-1,1\}^n$ and outputs a distribution $\widehat{\boldsymbol{p}}$ such that with probability at least $4/5$, p is ϵ-close to $\widehat{\boldsymbol{p}}$. Then, the algorithm must make $\Omega(\log \binom{n}{k}/\epsilon^2) + \Omega(2^k/\epsilon^2)$ queries.*

Both proofs of Theorem 3 and Theorem 4 follow from reductions from the communication complexity lower bound of the following indexing problem:

- Alice receives a uniformly random string $\boldsymbol{y} \sim \{-1,1\}^m$.
- Bob receives a uniformly random index $\mathbf{i} \sim [m]$.
- The task is for Alice to send a message to Bob so that Bob outputs $\boldsymbol{y}_{\mathbf{i}}$.

This problem has a well known $\Omega(m)$ lower bound on the one-way communication of any protocol in order for Bob to succeed with probability at least $2/3$ Miltersen et al. (1995).

The plan for proving Theorem 3 is the following. Our main goal is to cast the indexing problem as the problem of finding relevant variables. Let $\mathcal{A}$ be a *deterministic* algorithm for the task described in Theorem 3 with q general conditioning queries; it will become clear in the proof later that this is without loss of generality (so $\mathcal{A}$ can be viewed as a depth-q decision tree; see Definition 21). Setting $m = \Omega(\log \binom{n}{k})$, we show that Alice can use its input string $y \in \{-1,1\}^m$ to construct a k-junta distribution p_y over $\{-1,1\}^n$ with the following recovery property: any subset $J \subset [n]$ of no more than k variables such that p_y is ϵ-close to a junta distribution over J can be used to recover y. Alice uses private randomness to simulate the execution of $\mathcal{A}$ on p_y and sends a message to Bob that contains the sequence of q samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q$. The recovery property guarantees that whenever Bob succeeds in finding relevant variables using $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q$, which happens with probability at least $4/5$, he can use them to recover Alice's string y and then y_i.

However, the naive protocol described above has communication complexity qn and we only get $q \geq \Omega(m/n)$ which is insufficient for our goal. To compress this protocol, we note that distributions p_y constructed from y are in some sense very close to the uniform distribution over $\{-1,1\}^n$. More formally, we give the following definition of *ϵ-almost uniform distributions*.

Definition 16 *Let p be a probability distribution over $\{-1,1\}^n$ and $\epsilon \in (0,1/2)$. We say that p is ϵ-almost uniform if for every $x \in \{-1,1\}^n$, $|p(x) - 2^{-n}| \leq \epsilon 2^{-n}$.*

The intuition behind the compression is that a sample from an ϵ-almost uniform distribution (even being conditioned on a subset of $\{-1,1\}^n$) carries with it very little information (roughly $O(\epsilon^2)$). One can then use results from Harsha et al. (2010); Braverman and Garg (2014) (also see Corollary 7.7 in Rao and Yehudayoff (2020)) to show that the naive one-way private-coin protocol described above can be compressed into a public-coin protocol with $O(q\epsilon^2) + O(1)$ one-way communication bits. Formally we state the following lemma:

Lemma 17 *Let $\mathcal{A}$ be a deterministic algorithm on distributions over $\{-1,1\}^n$ that makes q general conditioning queries. Then there is a one-way public-coin protocol such that, upon receiving an ϵ-almost uniform distribution p over $\{-1,1\}^n$, Alice sends a message $\mathbf{M}$ of length $O(q\epsilon^2)+O(1)$ in the worst case. Bob can use $\mathbf{M}$ to compute a sequence of q strings $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_q \in \{-1,1\}^n$ such that the distribution of $(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_q)$ is $(1/20)$-close to the distribution of the sequence of q samples $\mathcal{A}$ receives when running on p.*

We give a self-contained proof of Lemma 17 in Section B.3 since the setting we work on is more explicit compared to those of Harsha et al. (2010); Braverman and Garg (2014). The flow of the proof for Theorem 4 is similar. The key differences lie in the construction of p_y from y for Alice, and the way Bob recovers y_i using the hypothesis $\widehat{p}$ returned by the learning algorithm for k-junta distributions. We prove Theorem 3 and Theorem 4 in Section B.1 and B.2, respectively.

B.1. Proof of Theorem 3

Suppose that $\mathcal{A}^*$ is a randomized algorithm which, given general conditioning query access to any unknown k-junta distribution p supported on $\{-1,1\}^n$, makes q queries and outputs with probability at least $4/5$ a subset $J \subset [n]$ of at most k variables such that p is ϵ-close to a junta distribution over J. So $\mathcal{A}^*$ can be viewed as a distribution of deterministic algorithms $\mathcal{A}$. Let

$$m = \left\lfloor \log \binom{n}{k} \right\rfloor = \Omega\left(\log \binom{n}{k}\right). \tag{11}$$

Alice will interpret her input string $x \in \{-1,1\}^m$ in the indexing problem as a set $S \subset [n]$ of size k and use S to define the following probability distribution p_S over $\{-1,1\}^n$:

$$p_S(x) = \begin{cases} (1+4\epsilon)2^{-n} & \prod_{i\in S} x_i = 1 \\ (1-4\epsilon)2^{-n} & \text{o.w.} \end{cases}.$$

It follows directly from the definition that p_S is $O(\epsilon)$-almost uniform. The following claim gives us the recovery property discussed earlier:

Claim 18 *Suppose that $S \subset [n]$ is a set of size k and $J \neq S \subset [n]$ is a set of size at most k. Then we have $d_{\mathrm{TV}}(p_S, g) \geq 2\epsilon$ for any junta distribution over variables in J.*

Proof: Notice that since S is of size k and $|J| \leq k$ of size at most k, there exists an index $i \in S$ such that $i \notin J$. Consider this fixed $i \in S \setminus J$. We will write the probability mass functions p_S and g as functions $\{-1,1\}^J \times \{-1,1\}^{[n]\setminus(J\cup\{i\})} \times \{-1,1\} \to \mathbb{R}_{\geq 0}$, where the first $|J|$ indices correspond to settings of bits in J, the second $n-|J|-1$ coordinates correspond to settings of bits in $[n]\setminus(J\cup\{i\})$, and the last bit determines i. We notice that since g is a junta over variables in J, for any $y \in \{-1,1\}^J$ and any two $u_1,u_2 \in \{-1,1\}^{[n]\setminus(J\cup\{i\})}$ and $v_1,v_2 \in \{-1,1\}$, $g(y,u_1,v_1) = g(y,u_2,v_2)$. Furthermore, by definition of p_S, $|p_S(y,u_1,v_1) - p_S(y,u_1,v_2)| = 8\epsilon 2^{-n}$ whenever

$v_1 \neq v_2$. Hence,

$$\begin{aligned}
d_{\mathrm{TV}}(p_S, g) &= \frac{1}{2} \sum_{x \in \{-1,1\}^n} |p_S(x) - g(x)| \\
&= \frac{1}{2} \sum_{y \in \{-1,1\}^J} \sum_{u \in \{-1,1\}^{[n] \setminus (J \cup \{i\})}} \left(|p_S(y,u,1) - g(y,u,1)| + |p_S(y,u,-1) - g(y,u,-1)|\right) \\
&\geq \frac{1}{2} \sum_{y \in \{-1,1\}^J} \sum_{u \in \{-1,1\}^{[n] \setminus (J \cup \{i\})}} |p_S(y,u,1) - p_S(y,u,-1)| = 2\epsilon.
\end{aligned}$$

This finishes the proof of the claim. ■

As a consequence of Claim 18, we obtain the following corollary.

Corollary 19 *Let $S \subset [n]$ be any set of size k, and let J be any set of size at most k such that p_S is ϵ-close to a junta distribution over J. Then we must have $J = S$.*

Proof: Let g be the closest junta over J to p_S, and suppose for the sake of contradiction, that $J \neq S$. Then, we apply Claim 18 which says that $d_{\mathrm{TV}}(p_S, g) \geq 2\epsilon$, giving the desired contradiction. ■

We are now ready to prove Theorem 3 by following the plan described earlier.

Proof of Theorem 3: The proof proceeds via a reduction from the two-party one-way communication problem of indexing. With m chosen in (59) Alice and Bob agree on a fixed injective map from $\{-1,1\}^m$ to subsets of $[n]$ of size k. Alice will interpret her input string $x \in \{-1,1\}^n$ as a subset $S \subset [n]$ of size k using this map. Given that $\mathcal{A}^*$ is a distribution of deterministic algorithms, there exists a q-query deterministic algorithm $\mathcal{A}$ such that

$$\mathbf{Pr}_{\boldsymbol{x} \sim \{-1,1\}^m} \left[\mathcal{A}(p_{\mathbf{S}}) \text{ returns } \mathbf{S}\right] \geq 4/5, \tag{12}$$

where $\boldsymbol{x}$ is drawn uniformly at random and $\mathbf{S} \subset [n]$ is its corresponding subset of size k. Alice and Bob agree on such a q-query deterministic algorithm $\mathcal{A}$.

Now we describe the protocol. Given $x \in \{-1,1\}^m$, Alice uses it to construct p_S over $\{-1,1\}^m$ which is $O(\epsilon)$-almost uniform. She uses Lemma 17 to send a message $\mathbf{M}$ of length $O(q\epsilon^2) + O(1)$ to Bob so that Bob can use $\mathbf{M}$ to obtain a sequence of q strings $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q \in \{-1,1\}^n$ such that the latter has distribution $(1/20)$-close to the distribution of the sequence of q samples $\mathcal{A}$ receives when running on p_S. It follows from (12) that when $\boldsymbol{x} \sim \{-1,1\}^m$, Bob successfully recovers $\mathbf{S}$ (and thus, $\boldsymbol{x}$ using the map they agreed on) by simulating $\mathcal{A}$ on $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q$ with probability at least $4/5 - 1/20 > 2/3$. By the $\Omega(m)$ lower bound on the indexing problem, we obtain the desired claim using (59).

■

B.2. Proof of Theorem 4

The lower bound $\Omega(\log \binom{n}{k}/\epsilon^2)$ follows trivially from Theorem 3. To see this, we can first learn p to within $\epsilon/2$ total variation distance. Let $\widehat{p}$ be the hypothesis distribution that the algorithm returns. Then we can find its closest k-junta distribution p' and let S be the set of relevant variables of p' with $|S| \leq k$. The algorithm can return S since $d_{\mathrm{TV}}(p, p') \leq d_{\mathrm{TV}}(p, \widehat{p}) + d_{\mathrm{TV}}(\widehat{p}, p') \leq \epsilon$.

We focus on the second part of the lower bound $\Omega(2^k/\epsilon^2)$ in the rest of the proof. Note that we may assume that k is asymptotically large; otherwise the second part is dominated by the first part. We follow the same flow. Suppose that $\mathcal{A}^*$ is a randomized algorithm which, given general conditioning query access to any unknown k-junta distribution p supported on $\{-1,1\}^n$, makes q queries and outputs with probability at least $4/5$ a hypothesis distribution $\widehat{p}$ such that $d_{\rm TV}(p,\widehat{p}) \le \epsilon$.

We say a Boolean function $f : \{-1,1\}^k \to \{-1,1\}$ is *good* if the number of 1-entries in f is between $2^k/3$ and $2^{k+1}/3$. Let G_k be the set of good Boolean functions. Then it follows from Chernoff bound that $|G_k| \ge 2^{2^k}(1-o_k(1))$. We set $m = 2^k$ and Alice interprets her input string $y \in \{-1,1\}^m$ in the indexing problem as a good Boolean function $f : \{-1,1\}^k \to \{-1,1\}$ by fixing a bijection between $[m]$ and $\{-1,1\}^k$ and interpreting y as the truth table of f.

Given a string $y \in \{-1,1\}^m$ and its corresponding $f : \{-1,1\}^k \to \{-1,1\}$, letting $I(y)$ be the number of 1-entries in f, Alice constructs the following k-junta distribution p_y over $\{0,1\}^n$:

$$p_y(x) = \begin{cases} 2^{-n}\left(1 + 40\epsilon \cdot \frac{2^k}{I(y)}\right) & \text{if } f(x_1,\ldots,x_k) = 1 \\ 2^{-n}\left(1 - 40\epsilon \cdot \frac{2^k}{2^k - I(y)}\right) & \text{if } f(x_1,\ldots,x_k) = -1 \end{cases}$$

Note that when f is good, p_y is an $O(\epsilon)$-almost uniform k-junta distribution; as it becomes clear later Alice constructs p_y only when f is good. The following claim gives us the recovery property:

Claim 20 *Given a good $y \in \{-1,1\}^m$ and p_y defined above, let $\widehat{p}$ be any distribution on $\{-1,1\}^n$ which has $d_{\rm TV}(p_y,\widehat{p}) \le \epsilon$. Then,*

$$\Pr_{\boldsymbol{x}\sim\{-1,1\}^n}\left[\operatorname{sign}\left(\widehat{p}(\boldsymbol{x}) - 2^{-n}\right) \ne \operatorname{sign}\left(p_y(\boldsymbol{x}) - 2^{-n}\right)\right] \le \frac{1}{20}.$$

Proof: Notice that for every $x \in \{-1,1\}^n$ where $\operatorname{sign}\left(\widehat{p}(x) - 2^{-n}\right) \ne \operatorname{sign}\left(p_y(x) - 2^{-n}\right)$, we have $|\widehat{p}(x) - p_y(x)| \ge 40\epsilon \cdot 2^{-n}$. Hence,

$$\epsilon \ge d_{\rm TV}(p_y,\widehat{p}) = \frac{1}{2}\sum_{x\in\{-1,1\}^n} |p_y(x) - \widehat{p}(x)| \ge 20\epsilon \cdot \Pr_{\boldsymbol{x}\sim\{-1,1\}^n}\left[\operatorname{sign}\left(\widehat{p}(\boldsymbol{x}) - 2^{-n}\right) \ne \operatorname{sign}\left(p_y(\boldsymbol{x}) - 2^{-n}\right)\right].$$

This finishes the proof of the claim. ■

Proof of Theorem 4: Again, the proof proceeds via a reduction from the two-party one-way communication problem of indexing over $\{-1,1\}^m$ where $m = 2^k$. Let $y \in \{-1,1\}^m$ be the input string of Alice. As alluded to earlier, in the case that y is not good, Alice just aborts the protocol and they fail the task with probability $o_k(1)$ because y is drawn uniformly at random from $\{-1,1\}^m$. In the case that y is good, Alice uses it to construct p_y, a k-junta distribution over $\{-1,1\}^n$ that is $O(\epsilon)$-almost uniform. Given that $\mathcal{A}^*$ is a randomized algorithm for learning k-junta distributions over $\{-1,1\}^n$, there exists a deterministic algorithm with q general conditioning queries such that

$$\mathbf{Pr}_{\mathbf{y}}\left[\mathcal{A}(p_{\mathbf{y}}) \text{ returns a hypothesis that is } \epsilon\text{-close to } p_{\mathbf{y}}\right] \ge 4/5,$$

where $\mathbf{y}$ is uniform over good strings. Alice and Bob agree on such an $\mathcal{A}$.

The protocol goes as before. When y is good, Alice uses Lemma 17 to send a message $\mathbf{M}$ of length $O(q\epsilon^2) + O(1)$ to Bob so that Bob can use $\mathbf{M}$ to obtain a sequence of q strings $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_q \in \{-1,1\}^n$ such that their distribution is $(1/20)$-close to the distribution of the sequence of q samples $\mathcal{A}$ receives when running on p_S. It follows from (12) that when $\mathbf{y} \sim \{-1,1\}^m$, Bob successfully learns a hypothesis distribution $\widehat{\mathbf{p}}$ that is ϵ-close to $p_{\mathbf{y}}$, by simulating $\mathcal{A}$ on $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_q$, with probability at least $4/5 - 1/20 - o_k(1)$. We now apply Claim 20 to conclude that if this occurs, Bob can output the correct $\mathbf{i}$-th bit of $\mathbf{y}$ with probability at least $9/10$ given that $\mathbf{i}$ is independent and uniform.. As a result, over the randomness of $\mathbf{y}$ and $\mathbf{i}$, Bob outputs the correct $\mathbf{y_i}$ with probability at least $4/5 - 1/20 - o_k(1) - 1/20 \geq 2/3$. By the $\Omega(m) = \Omega(2^k)$ lower bound on the indexing problem, we obtain the desired claim. ■

B.3. Compressing batches of conditional samples

We prove Lemma 17 in the rest of the section. Recall that $\mathcal{A}$ is a deterministic (adaptive) algorithm, where each query (a subset of $\{-1,1\}^n$) depends on all samples received from previous queries.

We use the following definition to capture such a q-query deterministic algorithm:

Definition 21 *For $n, q \in \mathbb{N}$, we say a q-*query tree *$\mathcal{T}$ is a rooted depth-q tree. Every non-leaf node $v \in \mathcal{T}$ contains a subset $A_v \subseteq \{-1,1\}^n$, as well as a child node v_x for every $x \in A_v$. Given a distribution p over $\{-1,1\}^n$, an* execution *of $\mathcal{T}$ on p is a random walk $(v_1, \ldots, v_q)$ down the tree, specifying a sequence of q samples $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q)$: starting at the root node and proceeding down the tree, for the current node v_i, sample $\boldsymbol{x}_i \sim p$ conditioned on $\boldsymbol{x}_i \in A_{v_i}$, and let $v_{i+1} = (v_i)_{\boldsymbol{x}_i}$. Let $\mathcal{E}_{p,\mathcal{T}}$ be the distribution supported on $(\{-1,1\}^n)^q$ which outputs the samples $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q)$ of an execution of $\mathcal{T}$ on p.*

We consider a protocol, `SampleWalk` which, without communication, generates an execution of a given q-query tree $\mathcal{T}$, and Alice decides whether or not to "accept" the samples at the end. In more detail, `SampleWalk` takes as input a distribution p over $\{-1,1\}^n$, a q-query tree $\mathcal{T}$, and an error tolerance $\delta \in (0,1)$, and using public randomness, will output a root-to-leaf walk of $\mathcal{T}$ specified by nodes $(v_1, \ldots, v_q)$ and $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q)$, or "reject". The protocol, `SampleWalk` follows the "rejection sampling" paradigm. (See Figure 3 for a precise description of the protocol.)

Protocol $\texttt{SampleWalk}(p, \mathcal{T}, \delta)$

Input: A distribution p supported on $\{-1,1\}^n$, a q-query tree $\mathcal{T}$, and a parameter $\delta \in (0,1)$. Furthermore, we assume access to a public string of infinite uniformly random bits.
Output: A root-to-leaf walk down the decision tree $\mathcal{T}$ specified by nodes $(v_1, \dots, v_q)$ and samples $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$, or "reject".[16]

1. Starting at the root of $\mathcal{T}$ and walking down the tree, Alice considers the current node in $v \in \mathcal{T}$, and the query $A_v \subset \{-1,1\}^n$. She uses public randomness to generate a sample $\boldsymbol{x}_v \sim A_v$ drawn *uniformly* from A_v, and considers the child node of $\mathcal{T}$ specified by $\boldsymbol{x}_v$. Notice that this builds a walk $(v_1, \dots, v_q)$ and $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$, and in particular, this step is completely independent from p, and draws a sample from $\mathcal{E}_{\mathcal{U},\mathcal{T}}$.

2. Alice samples a private bit which is 1 with probability
$$\min\left(1, \delta \cdot \frac{\mathcal{E}_{p,\mathcal{T}}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)}{\mathcal{E}_{\mathcal{U},\mathcal{T}}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)}\right)$$
and -1 otherwise. If Alice's sampled bit is 1, Alice "accepts" the sample $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$ and the nodes $(v_1, \dots, v_q)$, if it is -1, Alice "rejects".

Figure 3: The $\texttt{SampleWalk}$ Protocol.

Definition 22 *For a q-query tree $\mathcal{T}$, we let $\mathcal{D}^\circ_{p,\mathcal{T},\delta}$ be a distribution supported on $(\{-1,1\}^n)^q \cup \{\bot\}$ given by the samples $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$ forming the output of one execution of $\texttt{SampleWalk}(p, \mathcal{T}, \delta)$, or $\bot$ if it outputs "reject". We let $\mathcal{D}_{p,\mathcal{T},\delta}$ be the distribution $\mathcal{D}^\circ_{p,\mathcal{T},\delta}$ conditioned on it not outputting $\bot$.*

Lemma 23 *There exists a sufficiently small constant $\zeta \in (0,1)$ such that for any $\epsilon, \delta \in (0, 1/2)$ and*
$$q \le \left\lfloor \frac{\zeta \log(1/\delta)}{\epsilon^2} \right\rfloor,$$
the following holds. Let $\mathcal{T}$ be a q-query tree and p be ϵ-almost uniform. Then,
$$d_{\mathrm{TV}}(\mathcal{D}_{p,\mathcal{T},\delta}, \mathcal{E}_{p,\mathcal{T}}) \le \delta \qquad \textit{and} \qquad \mathbf{Pr}\left[\mathcal{D}^\circ_{p,\mathcal{T},\delta} \textit{ outputs } \bot\right] \le 1 - \delta/2.$$

Proof: In particular, notice that in order for an execution of $\texttt{SampleWalk}(p, \mathcal{T}, \delta)$ to output "reject", two events must occur:

- The first event is that the samples $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$ sampled in Step 1 satisfy
$$\mathcal{E}_{p,\mathcal{T}}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) < \mathcal{E}_{\mathcal{U},\mathcal{T}}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q) \cdot \frac{1}{\delta}. \tag{13}$$

[16] We note that outputting $(v_1, \dots, v_q)$ is unnecessary, as the samples $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$ uniquely determine a root-to-leaf walk down the tree $\mathcal{T}$. We maintain the notation just for notational simplicity.

- The second event is that a random bit sampled in Step 2 is set to -1, and the probability that his occurs is

$$1 - \delta \cdot \frac{\mathcal{E}_{p,\mathcal{T}}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)}{\mathcal{E}_{\mathcal{U},\mathcal{T}}(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)}.$$

We let $\mathcal{R} \subset (\{-1,1\}^n)^q$ be the set of strings which satisfy (13), i.e.,

$$\mathcal{R} = \left\{ (x_1, \dots, x_q) \in (\{-1,1\}^n)^q : \mathcal{E}_{p,\mathcal{T}}(x_1, \dots, x_q) < \frac{1}{\delta} \cdot \mathcal{E}_{\mathcal{U},\mathcal{T}}(x_1, \dots, x_q) \right\},$$

and notice that

$$\mathbf{Pr}\left[\mathcal{D}^{\circ}_{p,\mathcal{T},\delta} \text{ outputs } \bot\right] = \sum_{x \in \mathcal{R}} \mathcal{E}_{\mathcal{U},\mathcal{T}}(x) \left(1 - \delta \cdot \frac{\mathcal{E}_{p,\mathcal{T}}(x)}{\mathcal{E}_{\mathcal{U},\mathcal{T}}(x)}\right) = \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{E}_{\mathcal{U},\mathcal{T}}}[\boldsymbol{x} \in \mathcal{R}] - \delta \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{E}_{p,\mathcal{T}}}[\boldsymbol{x} \in \mathcal{R}], \tag{14}$$

so for simplicity in the notation, let

$$\alpha \stackrel{\text{def}}{=} \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{E}_{\mathcal{U},\mathcal{T}}}[\boldsymbol{x} \in \mathcal{R}] \qquad \text{and} \qquad \beta \stackrel{\text{def}}{=} \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{E}_{p,\mathcal{T}}}[\boldsymbol{x} \in \mathcal{R}].$$

Furthermore, whenever $x \in \mathcal{R}$,

$$\begin{aligned}\mathcal{D}_{p,\mathcal{T},\delta}(x) &= \sum_{k=1}^{\infty} \mathcal{E}_{\mathcal{U},\mathcal{T}}(x) \cdot \left(\delta \cdot \frac{\mathcal{E}_{p,\mathcal{T}}(x)}{\mathcal{E}_{\mathcal{U},\mathcal{T}}(x)}\right) \cdot \mathcal{D}^{\circ}_{p,\mathcal{T},\delta}(\bot)^{k-1} = \delta \cdot \mathcal{E}_{p,\mathcal{T}}(x) \left(\frac{1}{1 - \mathcal{D}^{\circ}_{p,\mathcal{T},\delta}(\bot)}\right) \\ &= \left(\frac{\delta}{1 - \alpha + \delta\beta}\right) \mathcal{E}_{p,\mathcal{T}}(x),\end{aligned}$$

and whenever $x \notin \mathcal{R}$, Step 2 always accepts the sample, so

$$\mathcal{D}_{p,\mathcal{T},\delta}(x) = \left(\frac{1}{1 - \alpha + \delta\beta}\right) \cdot \mathcal{E}_{\mathcal{U},\mathcal{T}}(x).$$

Thus, we may write

$$\begin{aligned} d_{\mathrm{TV}}\left(\mathcal{D}_{p,\mathcal{T},\delta}, \mathcal{E}_{p,\mathcal{T}}\right) &= \frac{1}{2} \sum_{x \in (\{-1,1\}^n)^q} \left|\mathcal{D}_{p,\mathcal{T},\delta}(x) - \mathcal{E}_{p,\mathcal{T}}(x)\right| \\ &\leq \frac{1}{2} \sum_{x \notin \mathcal{R}} \left(\mathcal{D}_{p,\mathcal{T},\delta}(x) + \mathcal{E}_{p,\mathcal{T}}(x)\right) + \frac{1}{2} \sum_{x \in \mathcal{R}} \mathcal{E}_{p,\mathcal{T}}(x) \left|\frac{\delta}{1 - \alpha + \delta\beta} - 1\right| \\ &= \frac{1}{2}\left(\frac{1 - \alpha}{1 - \alpha + \delta\beta} + (1 - \beta)\right) + \frac{1}{2}\beta \left|\frac{\delta(1 - \beta) - (1 - \alpha)}{1 - \alpha + \delta\beta}\right|, \end{aligned} \tag{15}$$

so it suffices to show

$$1 - \delta^2/2 \leq \alpha, \beta \leq 1$$

in order to conclude that (15) is at most δ, and that (14) is at most $1 - \delta/2$. In order to do so, we use the fact that p is ϵ-almost uniform to upper bound $1-\alpha$ and $1-\beta$. Notice that if $x \notin \mathcal{R}$, then, considering the unique path $(v_1, \dots, v_q)$ in $\mathcal{T}$ specified by x, we have

$$\begin{aligned}\frac{1}{\delta} \leq \frac{\mathcal{E}_{p,\mathcal{T}}(x)}{\mathcal{E}_{\mathcal{U},\mathcal{T}}(x)} &= \prod_{i=1}^{q} \left(\frac{p(x_i)}{\frac{1}{|A_{v_i}|}\sum_{y \in A_{v_i}} p(y)} \right) = \prod_{i=1}^{q} \left(1 + \frac{p(x_i) - \mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]}{\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]} \right) \\ &\leq \exp\left(\sum_{i=1}^{q} \frac{p(x_i) - \mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]}{\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]} \right). \qquad (16)\end{aligned}$$

We first upper bound $1-\alpha$ by considering the random sequence $\mathbf{Y}_1, \dots, \mathbf{Y}_q$ generated by starting at the root v_1 and walking down the tree $\mathcal{T}$, while sampling $\boldsymbol{x}_i \sim A_{v_i}$, setting $\mathbf{Y}_i = (p(\boldsymbol{x}_i) - \mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})])/\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]$, and letting $v_{i+1} = (v_i)_{\boldsymbol{x}_i}$. We upper-bound $1-\alpha$ by giving an upper bound for the probability that $\sum_{i=1}^{q} \mathbf{Y}_i \geq \ln(1/\delta)$, which in turn upper bounds $1-\alpha$ by (16). Notice that partial sums $\{\sum_{i=1}^{t} \mathbf{Y}_i\}_{t \in [q]}$ form a 0-centered martingale, and since p is ϵ-almost uniform,

$$|\mathbf{Y}_i| \leq \max_{\substack{v \in \mathcal{T} \\ x \in A_v}} \left| \frac{p(x) - \mathbf{E}_{\boldsymbol{z} \sim A_v}[p(\boldsymbol{z})]}{\mathbf{E}_{\boldsymbol{z} \sim A_v}[p(\boldsymbol{z})]} \right| \leq \max_{\substack{v \in \mathcal{T} \\ x \in A_v}} \left| \frac{\mathbf{E}_{\boldsymbol{z} \sim A_v}[p(x) - p(\boldsymbol{z})]}{\mathbf{E}_{\boldsymbol{z} \sim A_v}[p(\boldsymbol{z})]} \right| \leq \frac{2\epsilon}{1-\epsilon} \leq 4\epsilon.$$

We may apply Azuma's inequality to conclude

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \mathcal{E}_{\mathcal{U},\mathcal{T}}} \left[\sum_{i=1}^{q} \mathbf{Y}_i \geq \ln(1/\delta) \right] \leq \exp\left(-\frac{\ln^2(1/\delta)}{2 \cdot 16\epsilon^2 \cdot q} \right) \leq \delta^2/2$$

by setting of q with ζ being a sufficiently small constant, and hence lower bounds α by $1 - \delta^2/2$. In order to upper bound $1-\beta$, we consider the sequence of random variables $\mathbf{Y}'_1, \dots, \mathbf{Y}'_q$ generated by starting at the root v_1 and walking down the tree $\mathcal{T}$, but now we sample $\boldsymbol{x}_i \sim p$ conditioned on $\boldsymbol{x}_i \in A_{v_i}$, setting $\mathbf{Y}_i = (p(\boldsymbol{x}_i) - \mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})])/\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]$, and writing

$$\mathbf{Y}'_i = \mathbf{Y}_i - \frac{\mathbf{E}_{\boldsymbol{z}' \sim p}[p(\boldsymbol{z}') \mid \boldsymbol{z}' \in A_{v_i}] - \mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]}{\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]},$$

where the subsequent node $v_{i+1} = (v_i)_{\boldsymbol{x}_i}$. Notice that now the partial sums $\{\sum_{i=1}^{t} \mathbf{Y}'_i\}_{t \in [q]}$ have expectation 0, form a martingale, where $\mathbf{Y}'_i$ are obtained by shifting $\mathbf{Y}_i$ by its expectation, $\mathbf{E}_{\boldsymbol{x} \sim \mathcal{E}_{p,\mathcal{T}}}[\mathbf{Y}]$. Furthermore, we may upper bound this shift by importance sampling,

$$\begin{aligned}\frac{\mathbf{E}_{\boldsymbol{z}' \sim p}[p(\boldsymbol{z}') \mid \boldsymbol{z}' \in A_{v_i}] - \mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]}{\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]} &= \frac{\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})^2] - \mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]^2}{\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]^2} \\ &= \frac{\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}\left[\left(p(\boldsymbol{z}) - \mathbf{E}_{\boldsymbol{z}' \sim A_{v_i}}[p(\boldsymbol{z}')] \right)^2 \right]}{\mathbf{E}_{\boldsymbol{z} \sim A_{v_i}}[p(\boldsymbol{z})]^2} \leq \frac{4\epsilon^2}{1-\epsilon} \leq 8\epsilon^2. \qquad (17)\end{aligned}$$

so that similarly to the computation above, $|\mathbf{Y}_i'| \leq 4\epsilon + 8\epsilon^2 \leq 12\epsilon$. We may again, apply Azuma's inequality, where we notice that the expectation of

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{E}_{p,\mathcal{T}}}\left[\sum_{i=1}^{q}\mathbf{Y}_i \geq \ln(1/\delta)\right] &\leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{E}_{p,\mathcal{T}}}\left[\sum_{i=1}^{q}\mathbf{Y}_i' \geq \ln(1/\delta) - 8q\epsilon^2\right] \\ &\leq \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{E}_{p,\mathcal{T}}}\left[\sum_{i=1}^{q}\mathbf{Y}_i' \geq \ln(1/\delta)/2\right] \leq \exp\left(-\frac{\ln^2(1/\delta)}{2\cdot 4\cdot 144\epsilon^2 q}\right) \leq \delta^2/2,\end{aligned}$$

where we used a small enough constant $\zeta > 0$ so that $8q\epsilon^2 \leq \ln(1/\delta)/2$, as well as for the final inequality to hold. ■

We now use Lemma 23 to prove Lemma 17:

Proof of Lemma 17: We start with the easy case when $q < 1/\epsilon^2$. In this case, we apply Lemma 23 with $\delta = 1/(40)^{1/\zeta}$. Notice that $q \leq \lfloor \zeta \log(1/\delta)/\epsilon^2 \rfloor$, so we let $\mathcal{T}$ be $\mathcal{A}$, and Lemma 23 implies a single call to SampleWalk$(p, \mathcal{A}, \delta)$ succeeds in outputting a sample $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$ from $\mathcal{D}_{p,\mathcal{A},\delta}$ with probability at least $\delta/2$, and if it does succeed, the output distribution is at most δ-far from the distribution producing a sequence of q samples an execution of $\mathcal{A}$ on p. Alice and Bob use public randomness to execute SampleWalk$(p, \mathcal{A}, \delta)$ for $t = O(1/\delta)$ iterations, and Alice communicates the index of the first execution where SampleWalk$(p, \mathcal{A}, \delta)$ did not output "reject", or the final index if all executions outputted "reject". Notice that the distribution of the first time SampleWalk$(p_S, \mathcal{T}, \delta)$ accepts is exactly $\mathcal{D}_{p_S,\mathcal{T},\delta}$. Furthermore, this uses $O(\log(1/\delta)) = O(1)$ bits of communication, and that the total variation distance between the samples $(\boldsymbol{x}_1, \dots, \boldsymbol{x}_q)$ from this protocol and an execution of $\mathcal{A}$ on p is at most $\delta + (1-\delta/2)^t \leq 1/20$, where the first δ captures the case when some SampleWalk$(p, \mathcal{A}, \delta)$ does not reject, and $(1-\delta/2)^t$ is the probability that all SampleWalk$(p, \mathcal{A}, \delta)$ output "reject".

When $q \geq 1/\epsilon^2$, we apply Lemma 23 with

$$\delta = \frac{1}{\epsilon^2 q \cdot 100^{1/\zeta}}.$$

As per setting of (what we refer to as q') from Lemma 23, where $q' = \lfloor \zeta \log(1/\delta)/\epsilon^2 \rfloor \geq 2$ and hence $q' \geq \zeta \log(1/\delta)/(2\epsilon^2)$. Alice and Bob break up the q-query algorithm $\mathcal{A}$ into $\lceil q/q' \rceil$ many q'-query trees. The trees are adaptively chosen so as to simulate an execution of $\mathcal{A}$. For each q'-query tree $\mathcal{T}$, Alice and Bob use public randomness to execute SampleWalk$(p_S, \mathcal{T}, \delta)$ for $O(1/\delta)$ iterations such that with probability at least $1/2$, at least one accepts. Alice then communicates $O(\log(1/\delta))$ bits to Bob, indicating the first index where SampleWalk$(p_S, \mathcal{T}, \delta)$ accepts, or a special message indicating none accepted. If some execution accepts, then Bob re-constructs the samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{q'}$ utilizes those samples to simulate the walk down $\mathcal{T}$. If SampleWalk$(p_S, \mathcal{T}, \delta)$ never accepts, Alice and Bob try again on the same tree.

Notice that by Lemma 23, since the distribution over the leaves of $\mathcal{T}$ is δ-close in total variation distance from that of a true execution of $\mathcal{T}$ on p, after $\lceil q/q' \rceil$ successive executions of Lemma 23, the distribution over the leaves of $\mathcal{A}$ is at most $\delta\lceil q/q' \rceil$-close to that of a true execution of $\mathcal{A}$ on p, where we have

$$\delta\left\lceil \frac{q}{q'} \right\rceil \leq \frac{1}{\epsilon^2 q \cdot 100^{1/\zeta}}\left(\frac{q \cdot 2\epsilon^2}{\zeta \log(1/\delta)} + 1\right) \leq \frac{3}{100}$$

In order to upper bound the communication complexity, notice that each round of $\lceil q/q' \rceil$ sends $O(\log(1/\delta))$ bits and succeeds with probability at least $1/2$; which means that the expected communication complexity of a round is $O(\log(1/\delta))$. Hence, the expected communication complexity of the whole protocol is therefore

$$O\left(\left\lceil \frac{q}{q'} \right\rceil \log(1/\delta)\right) \le O\left(\frac{q\log(1/\delta)}{q'} + \log(1/\delta)\right) = O\left(q\epsilon^2 + \log(q\epsilon^2)\right) \le O(q\epsilon^2).$$

In order to bound the worst-case communication complexity, we use Markov's inequality. Specifically, by losing another constant factor, we may assume the protocol sends $O(q\epsilon^2)$ bits except with probability at most $1/100$; in this case, Alice sends an arbitrary bits. Then, the distribution over the samples that Bob may reconstruct is $(3/100 + 1/100)$-close to that of a true execution of $\mathcal{A}$ on p.
■

Appendix C. Testing Algorithm

We use `FindRelevantVariables` and `MeanTester` to give an algorithm for testing k-junta distributions. The algorithm, `TestingJuntas`, is described in Figure 4; we prove the following theorem:

Theorem 5 (Testing junta distributions) *There is an algorithm, which takes subcube conditioning access to an unknown distribution p over $\{-1,1\}^n$, an integer $k \in \mathbb{N}$, and $\epsilon \in (0, 1/4]$. It makes*

$$\tilde{O}\left(\frac{k+\sqrt{n}}{\epsilon^2}\right)$$

queries, runs in time $\tilde{O}(n(k+\sqrt{n})^2/\epsilon^4)$ and achieves the following guarantee: It accepts with probability at least $2/3$ if p is a k-junta distribution, and rejects with probability at least $2/3$ if p is ϵ-far from a k-junta.

Proof of Theorem 5: We start with the soundness case to show that `TestingJuntas` rejects with probability at least $2/3$ when p is far from k-juntas. Assume without loss of generality that the set J returned by `FindRelevantVariables` has size at most k; otherwise `TestingJuntas` rejects.

Given $|J| \le k$ and p is ϵ-far from k-junta distributions, the main structural lemma implies that

$$\sum_{j=1}^{\lceil \log_2 2n \rceil} \mathop{\mathbf{E}}_{\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p)} \left[\mathop{\mathbf{E}}_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})} \left[\left\| \mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}) \right\|_2 \right] \right] \ge \frac{\epsilon}{\log^c(n/\epsilon)}.$$

As a result, there exists a $j \in \lceil \log_2 2n \rceil$ (using the choice of ϵ' in (18)) such that

$$\mathop{\mathbf{E}}_{\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p)} \left[\mathop{\mathbf{E}}_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})} \left[\left\| \mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}) \right\|_2 \right] \right] \ge \epsilon'.$$

Fix such a j and we apply the following claim (which is elementry and we delay its proof):

```
Subroutine TestingJuntas$(p, k, \epsilon)$

Input: Subcube conditioning access to a distribution $p$ supported on $\{-1,1\}^n$, an integer
$k \in \mathbb{N}$ and a proximity parameter $\epsilon \in (0, 1/4]$.
Output: Either accept or reject.

  1. Let $c$ be the universal constant in the main structural lemma. We let
         $\epsilon' = \frac{\epsilon}{\lceil \log_2 2n \rceil \cdot \log^c(n/\epsilon)}, \quad r = \left\lceil \log(2\sqrt{n}/\epsilon') \right\rceil \quad \text{and} \quad \epsilon^* = \frac{\epsilon'}{1600r}.$   (18)

  2. Execute FindRelevantVariables$(p, k, \epsilon^*)$ and let $J$ be the set it returns.

  3. If $|J| > k$, reject.

  4. For each $j \in [\lceil \log_2 2n \rceil]$ and $\ell \in [r]$ with $r = \lceil \log(2\sqrt{n}/\epsilon') \rceil$:

       Repeat the following $L \cdot R$ times, where
           $L = \frac{4r\sqrt{n}}{2^\ell \epsilon'} \quad \text{and} \quad R = O\left(\log\left(\frac{n}{\epsilon'}\right)\right)$

       (A) Sample $\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p)$ and $\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})$, execute
           MeanTester$((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}, k, 2^{-\ell})$ for
           $R$ times and take the majority of answers.

       Reject if for at least $R/2$ rounds of (A), the majority of answers is "Not a
         Junta".

  5. Accept if this line is reached.
```

Figure 4: The TestingJuntas algorithm for testing junta distributions.

Claim 24 *Let $\mathbf{X}$ be a random variable that takes values between 0 and 1. If $\mathbf{E}[\mathbf{X}] \geq \delta$ for some $\delta \in (0,1)$, then there exists an $\ell \in [\lceil \log(2/\delta) \rceil]$ such that*

$$\mathbf{Pr}\left[\mathbf{X} \geq 2^{-\ell}\right] \geq \frac{2^\ell \delta}{4\lceil \log(2/\delta) \rceil}$$

Scaling down by $\sqrt{n}$ and applying Claim 24, there is an $\ell \in [r]$ with $r = \lceil \log(2\sqrt{n}/\epsilon') \rceil$ such that

$$\mathbf{Pr}_{\boldsymbol{\rho},\boldsymbol{\nu}}\left[\left\|\mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}})\right\|_2 \geq \sqrt{n}/2^\ell\right] \geq \frac{2^\ell \epsilon'}{4r\sqrt{n}}. \tag{19}$$

It follows from a Chernoff bound that, with probability at least $1 - o_n(1)$, the number of rounds of (A) in which $\boldsymbol{\rho}, \boldsymbol{\nu}$ satisfy (19) is at least $2R/3$ (since the expectation is at least R). It follows from the promise we get from MeanTester (i.e., each run returns "Not a Junta" with probability at least $2/3$ when the event in (19) holds) that with probability at least $1 - o_n(1)$, the majority of answers returned by `MeanTester` is "`Not a Junta`" in each of these $2R/3$ rounds of (A). So overall the algorithm rejects with probability at least $1 - o_n(1)$. This finishes the soundness case.

Next we work on the completeness case to show that `TestingJuntas` accepts with probability at least $2/3$ when p is a k-junta distribution. Suppose p is a k-junta distribution, and let $K \subset [n]$ be the set of at most k relevant variables (which is unknown to the algorithm). First it follows from Lemma 13 that with probability at least 7/9, the output J of `FindRelevantVariables` satisfies both conditions of Lemma 25. So let $|J| \le k$, and for every $j \in [\lceil \log_2(2k) \rceil]$,

$$\mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}\left[\|\mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}})\|_2\right]\right] \le \epsilon^*. \tag{20}$$

We will now use this fact, as well as the following simple claim (whose proof we defer), to derive the bound (20) for all $j \in [\lceil \log_2(2n) \rceil]$, and not just up to $\lceil \log_2(2k) \rceil$.

Claim 25 *Fix $m \in \mathbb{N}$ and let h be any distribution over $\{-1,1\}^m$. For any $0 \le \sigma_2 \le \sigma_1 \le 1/m$, we have*

$$\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma_2}(h)}\left[\|\mu(h_{|\boldsymbol{\nu}})\|_2\right] \le \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma_1}(h)}\left[\|\mu(h_{|\boldsymbol{\nu}})\|_2\right]$$

For every $\rho \in \mathrm{supp}(\mathcal{D}_{\overline{J}}(p))$, let $h^{(\rho)}$ be the distribution over $\{-1,1\}^{K\setminus J}$ given by $(p_{|\rho})_{K\setminus J}$. Since p is a junta over variables in K, for every $\rho \in \mathrm{supp}(\mathcal{D}_{\overline{J}}(p))$, the distribution of $p_{|\rho}$ over variables outside of K is always uniform, irrespective of the restriction ρ. Hence, for any $\sigma' \in (0,1)$, the non-zero coordinates of the mean vector $\mu((p_{|\rho})_{\boldsymbol{\nu}})$ for $\boldsymbol{\nu} \sim \mathcal{D}_{\sigma'}(p_{|\rho})$ are always supported on those coordinates in K. Hence, for every $\sigma' \in (0,1)$,

$$\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma'}(p_{|\rho})}\left[\|\mu((p_{|\rho})_{\boldsymbol{\nu}})\|_2\right] = \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma'}(h^{(\rho)})}\left[\|\mu(h^{(\rho)}_{|\boldsymbol{\nu}})\|_2\right].$$

We let $j^* = \lceil \log_2(2k) \rceil$ and note that $\sigma^{j^*} \le 1/k$. By Claim 25, we have that for $j' \in [\lceil \log_2(2n) \rceil]$ with $j' \ge j^*$,

$$\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j'}}(p_{|\rho})}\left[\|\mu((p_{|\rho})_{|\boldsymbol{\nu}})\|_2\right] = \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j'}}(h^{(\rho)})}\left[\|\mu(h^{(\rho)}_{|\boldsymbol{\nu}})\|_2\right] \le \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j^*}}}\left[\|\mu(h^{(\rho)}_{|\boldsymbol{\nu}})\|_2\right] = \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j^*}}}\left[\|\mu((p_{|\rho})_{\boldsymbol{\nu}})\|_2\right].$$

Averaging over $\boldsymbol{\rho} \sim \mathcal{D}_{\overline{J}}(p)$ implies that for all $j \in [\lceil \log_2(2n) \rceil]$,

$$\mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}\left[\|\mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}})\|_2\right]\right] \le \epsilon^*,$$

which in turn, implies that for all $j \in [\lceil \log_2(2n) \rceil]$, and all $\ell \in [r]$,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{\rho},\boldsymbol{\nu}}\left[\|\mu((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}})\|_2 \ge \sqrt{n}/(100 \cdot 2^\ell)\right] \le \frac{2^\ell \epsilon^* \cdot 100}{\sqrt{n}} \le \frac{2^\ell \epsilon'}{16 r \sqrt{n}} \tag{21}$$

using our choice of ϵ^* in (18). Fix j and ℓ. It follows from a Chernoff bound that with probability at least $1 - e^{-\Omega(R)}$, the number of rounds of (A) that satisfy the event in (21) is at most $R/2$ (because the expectation is at most $R/4$). The latter implies that the number of rounds of (A) that violate the event in (21) is at least $LR - R/2$. For each of these $LR - R/2$ rounds of (A), the majority of runs of `MeanTester` in (A) returns "`Is a Junta`" with probability at least $1 - e^{-\Omega(R)}$ by a Chernoff bound. By a union bound we have that all these $LR - R/2$ rounds have majority being "`Is a Junta`" with probability at least $1 - (LR - R/2) \cdot e^{-\Omega(R)}$. It follows that the main loop with j and ℓ rejects with probability at most

$$1 - e^{-\Omega(R)} - (LR - R/2) \cdot e^{-\Omega(R)} \leq 1 - LR \cdot e^{-\Omega(R)}.$$

Using a union bound over all main loops, the algorithm rejects with probability at most

$$\frac{2}{9} + \lceil \log 2n \rceil \cdot r \cdot LR \cdot e^{-\Omega(R)} < \frac{1}{3}.$$

Finally we bound the number of queries. Notice that both ϵ' and ϵ^* are $\epsilon/\mathrm{polylog}(n/\epsilon)$. Hence the number of queries made by the call to `FindRelevantVariables`* is $\tilde{O}(k/\epsilon^2) \cdot \mathrm{polylog}(n)$. On the other hand, the number of queries made by calls to `MeanTester` is (using $r = \lceil \log_2(2\sqrt{n}/\epsilon') \rceil$)

$$\begin{aligned} &\lceil \log_2 2n \rceil \cdot \sum_{\ell=1}^{r} \frac{4r\sqrt{n}}{2^\ell \epsilon'} \cdot O\left(\log^2\left(\frac{n}{\epsilon'}\right)\right) \cdot (k + \sqrt{n}) \cdot \max\left\{\frac{2^{2\ell}}{n}, \frac{2^\ell}{\sqrt{n}}\right\} \\ &\qquad = (k + \sqrt{n}) \cdot \mathrm{polylog}\left(\frac{n}{\epsilon}\right) \cdot \sum_{\ell=1}^{r} \frac{\sqrt{n}}{2^\ell \epsilon} \cdot \max\left\{\frac{2^{2\ell}}{n}, \frac{2^\ell}{\sqrt{n}}\right\} = \tilde{O}\left(\frac{k + \sqrt{n}}{\epsilon^2}\right). \end{aligned}$$

The upper bound on the running time can simply be verified from Figure 4 and Theorem 8. This finishes the proof of the theorem. ■

Proof of Claim 24: Let $r = \lceil \log(2/\delta) \rceil$, and assume for contradiction that the claim is not true for any $\ell \in [r]$. Then we have

$$\delta \leq \mathbf{E}[\mathbf{X}] < \sum_{\ell=1}^{r} \frac{2^\ell \delta}{4r} \cdot \frac{2}{2^\ell} + 1 \cdot \frac{1}{2^r} = \delta,$$

a contradiction. ■

Proof of Claim 25: We simply note that for any restriction $\nu \in \{-1, 1, *\}^m$ with $\mathrm{stars}(\nu) = S$,

$$\Pr_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma_1}(h)}[\boldsymbol{\nu} = \nu] = \Pr_{\mathbf{S} \sim \mathcal{S}_{\sigma_1}}[\mathbf{S} = S] \cdot \Pr_{\boldsymbol{x} \sim h_{\overline{S}}}\left[\boldsymbol{x} = \nu_{\overline{S}}\right] \geq \Pr_{\mathbf{S} \sim \mathcal{S}_{\sigma_2}}[\mathbf{S} = S] \cdot \Pr_{\boldsymbol{x} \sim h_{\overline{S}}}\left[\boldsymbol{x} = \nu_{\overline{S}}\right] = \Pr_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma_2}(h)}[\boldsymbol{\nu} = \nu],$$

where we used the fact that

$$\frac{d}{d\sigma}\left[\Pr_{\mathbf{S} \sim \mathcal{S}_\sigma}[\mathbf{S} = S]\right] = \sigma^{|S|-1}(1 - \sigma)^{m-|S|-1}(|S| - \sigma m) > 0$$

whenever $0 \leq \sigma \leq 1/m$. ■

Appendix D. Lower Bound for Testing

In this section, we prove the following theorem showing a lower bound for testing whether a product distribution is an k-junta distribution with $k = n/2$. We first state the theorem and proceed to show it implies Theorem 6.

Theorem 26 *There exist two absolute constants $\epsilon_1 > 0$ and $C_1 \in \mathbb{N}$ such that for all $0 < \epsilon \leq \epsilon_1$ and $n \geq C_1^2$, any algorithm which receives samples from an unknown product distribution p supported on $\{-1,1\}^n$ and distinguishes with probability at least $2/3$ between the case p is an $(n/2)$-junta distribution and the case p is ϵ-far from being an $(n/2)$-junta distribution must observe at least $\tilde{\Omega}(n)/\epsilon^2$ many samples from p.*

Proof of Theorem 6 assuming Theorem 26: We first inspect the proof of Theorem 4.8 from Canonne et al. (2017), which presents a lower bound on the sample complexity of testing whether an unknown product distribution is uniform or far from uniform. Specifically, they show that there are two constants $\epsilon_2 > 0$ and $C_2 \in \mathbb{N}$ such that for any $\epsilon \in (0, \epsilon_2]$ and $n \geq C_2$, there are two distributions $\mathcal{Y}$ and $\mathcal{N}$, supported on product distributions over $\{-1,1\}^n$ such that no algorithm can determine whether a draw $\boldsymbol{p}$ belongs to $\mathcal{Y}$ or $\mathcal{N}$ with probability greater than 2/3 without observing $\Omega(\sqrt{n}/\epsilon^2)$ samples from $\boldsymbol{p}$. Moreover, the distribution $\mathcal{Y}$ always outputs $\mathcal{U}_n$ and the distribution $\mathcal{N}$ always outputs a distribution $\boldsymbol{p}$ that is ϵ-far from being a $(n/2)$-junta distribution. We are done if $k \leq \sqrt{n}$ so we are left with the case when $k \geq \sqrt{n}$. In the rest of the proof we prove a lower bound of $\tilde{\Omega}(k)/\epsilon^2$ with a reduction to Theorem 26.

We now prove Theorem 6 by setting the two constants $\epsilon_0 = \min(\epsilon_1, \epsilon_2)$ and $C_0 = \max(C_1^2, C_2)$. Let $\epsilon \in (0, \epsilon_0]$, $n \geq C_0$ and $0 \leq k \leq n/2$. Since $\mathcal{U}_n$ is trivially a k-junta distribution and $k \leq n/2$, the properties of $\mathcal{Y}$ and $\mathcal{N}$ from Canonne et al. (2017) imply a lower bound of $\Omega(\sqrt{n}/\epsilon^2)$ for distinguishing between the case p is a k-junta distribution and the case p is ϵ-far from a k-junta distribution.

Note that $k \geq \sqrt{n} \geq C_1$. Consider an unknown product distribution g over $\{-1,1\}^{2k}$ and the task of distinguishing the case g is a k-junta distribution and the case g is ϵ-far from a k-junta distribution. By Theorem 26, any algorithm for this task must observe $\tilde{\Omega}(k)/\epsilon^2$ samples from g. On the other hand, let g' be the distribution supported on $\{-1,1\}^n$ defined using g as follows: To draw $\boldsymbol{x} \sim g'$ we first draw a sample $\mathbf{y} \sim g$ and set $\mathbf{y}$ to be the first $2k$ bits of $\boldsymbol{x}$; the last $n - 2k$ bits of $\boldsymbol{x}$ are drawn independently and uniformly at random. Notice that if g is a k-junta, then g' is a k-junta, and if g is ϵ-far from a k-junta, then g' is ϵ-far from a k-junta. Given that sample access to g' can be simulated using sample access to g, the task of distinguishing between the case g' is a k-junta and the case g' is ϵ-far from k-junta is at least as hard as the task for g. From this reduction we get a sample complexity lower bound of $\tilde{\Omega}(k)/\epsilon^2$. ■

The proof of Theorem 26 follows from the following lemma by simply noticing that any algorithm which receives s independent samples from an unknown product distribution p over $\{-1,1\}^n$ can be simulated by an algorithm which receives a sample from the product distribution $\mathrm{Bin}(s, p_1) \times \cdots \times \mathrm{Bin}(s, p_n)$.

Lemma 27 *There exists an absolute constant $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0]$ and $n \in \mathbb{N}$, there exist two distribution $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$ supported on product distributions over $\{-1,1\}^n$ satisfying*

$$\Pr_{\boldsymbol{p} \sim \mathcal{D}_{yes}}\left[\boldsymbol{p} \in \texttt{Junta}(n/2)\right] \geq 1 - o_n(1) \quad \textit{and} \quad \Pr_{\boldsymbol{p} \sim \mathcal{D}_{no}}\left[d_{\mathrm{TV}}(\boldsymbol{p}, \texttt{Junta}(n/2)) \geq \epsilon\right] \geq 1 - o_n(1). \tag{22}$$

Moreover, letting $s = \lceil n/(\epsilon^2 \log^{12} n) \rceil$, the two distributions $\mathcal{R}_{yes} = \mathcal{R}(s, \mathcal{D}_{yes})$ and $\mathcal{R}_{no} = \mathcal{R}(s, \mathcal{D}_{no})$ supported on $\mathbb{N}^n$ satisfy $d_{\mathrm{TV}}(\mathcal{R}_{yes}, \mathcal{R}_{no}) = o_n(1)$, where $\mathcal{R}(s, \mathcal{D})$ is specified by letting

$$\Pr_{\boldsymbol{r} \sim \mathcal{R}(s,\mathcal{D})}\left[\boldsymbol{r} = r\right] = \mathop{\mathbf{E}}_{\boldsymbol{p} \sim \mathcal{D}}\left[\prod_{i=1}^{n} \Pr_{\boldsymbol{\ell} \sim \mathrm{Bin}(s, \boldsymbol{p}_i)}\left[\boldsymbol{\ell} = r_i\right]\right], \quad \textit{for every } r \in \mathbb{N}^n. \tag{23}$$

The proof of Lemma 27 constitutes the next two subsections. We give the construction of $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ and prove (22) in Section D.1, and bound the distance between $\mathcal{R}_{\text{yes}}$ and $\mathcal{R}_{\text{no}}$ in Section D.2.

D.1. Construction of $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$

Let p be a product distribution over $\{-1,1\}^n$. We prove the following lemma that lowerbounds $d_{\mathrm{TV}}(p,\mathcal{U}_n)$ using $\|\mu(p)\|_2$:

Lemma 28 *There is two constants $c_1^*, c_2^* > 0$ such that any product distribution p over $\{-1,1\}^n$ satisfies*

$$d_{\mathrm{TV}}(p,\mathcal{U}_n) \geq \left(\frac{1}{8} - \frac{c_1^* \|\mu(p)\|_\infty}{\|\mu(p)\|_2}\right) \cdot \min\left(c_2^*, \frac{\|\mu(p)\|_2}{4}\right).$$

We delay the proof of Lemma 28 to Section D.3. We fix the constant $\epsilon_0 \in \mathbb{R}_{\geq 0}$ in Lemma 27 to be

$$\epsilon_0 = \frac{c_2^*}{9}. \tag{24}$$

For $n \in \mathbb{N}$, let $\ell = \lceil \log n / \log\log n \rceil$. Given any vector $\alpha \in \mathbb{R}^\ell$ we let $A(\alpha)$ be the Vandermonde matrix defined with respect to α, and $e_1 \in \mathbb{R}^\ell$ be the first basis vector:

$$A(\alpha) = \begin{bmatrix} \alpha_1^0 & \alpha_2^0 & \alpha_3^0 & \cdots & \alpha_\ell^0 \\ \alpha_1^1 & \alpha_2^1 & \alpha_3^1 & \cdots & \alpha_\ell^1 \\ \alpha_1^2 & \alpha_2^2 & \alpha_3^2 & \cdots & \alpha_\ell^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{\ell-1} & \alpha_2^{\ell-1} & \alpha_3^{\ell-1} & \cdots & \alpha_\ell^{\ell-1} \end{bmatrix} \qquad \text{and} \qquad e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Recall the following closed form for the determinant of a Vandermonde matrix $A(\alpha)$:

$$\det\big(A(\alpha)\big) = \prod_{\substack{i,j\in[\ell] \\ i<j}} (\alpha_j - \alpha_i),$$

so that $\det(A(\alpha)) \neq 0$ whenever coordinates of α are distinct. For the rest of the section, consider the vector $\alpha \in \mathbb{R}^\ell$ given by letting

$$\alpha_j = j^3 \qquad \forall j \in [\ell], \tag{25}$$

and let $z \in \mathbb{R}^\ell$ be the unique solution to the system of ℓ linear equations where $A(\alpha)z = e_1$. Let

$$\mathcal{W} = \{j \in [\ell] : z_j \geq 0\} \qquad \text{and} \qquad \mathcal{V} = [\ell] \setminus \mathcal{W}.$$

We will need the following technical claim about z; we delay its proof to Subsection D.4.

Claim 29 *There is an absolute constant $C^* > 0$ such that for any $\ell \in \mathbb{N}$, the solution $z \in \mathbb{R}^\ell$ to the Vandermonde system $A(\alpha)z = e_1$ with α as in (25) satisfies $\|z\|_1 \leq C^*$.*

We now describe $\mathcal{D}_{\text{no}}$ and $\mathcal{D}_{\text{yes}}$ using α, $\mathcal{W}$ and $\mathcal{V}$ given above. Let $\tau \in \mathbb{R}_{\geq 0}$ be set as

$$\tau = \min\left\{36\sqrt{C^*} \cdot \epsilon, \frac{\sqrt{n}}{2\ell^3}\right\}, \tag{26}$$

and notice that for large n, $\tau = 36\sqrt{C^*}\epsilon = \Theta(\epsilon)$. First we let $\boldsymbol{p} \sim \mathcal{D}_{\text{no}}$ be the product distribution supported on $\{-1,1\}^n$ given by letting for each $i \in [n]$, be independently set to

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\boldsymbol{p}}\left[\boldsymbol{x}_i = 1\right] = \frac{1}{2} + \frac{\boldsymbol{\gamma}_i \cdot \tau}{\sqrt{n}} \qquad \text{such that } \boldsymbol{\gamma}_i = \begin{cases} 0 & \text{w.p. } 1 - \dfrac{\sum_{j\in\mathcal{W}} z_j}{\|z\|_1} \\ j^3 & \text{w.p. } \dfrac{z_j}{\|z\|_1} \text{ for } j \in \mathcal{W}. \end{cases} \tag{27}$$

Notice that probabilities above are smaller than 1 since $\boldsymbol{\gamma}_i \leq \ell^3$, for $\ell = \lceil \log n / \log\log n \rceil$ and the setting of τ. On the other hand, we let $\boldsymbol{q} \sim \mathcal{D}_{\text{yes}}$ be the product distribution supported on $\{-1,1\}^n$ given by letting for each $i \in [n]$, be independently set to

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\boldsymbol{q}}\left[\boldsymbol{x}_i = 1\right] = \frac{1}{2} + \frac{\boldsymbol{\delta}_i \cdot \tau}{\sqrt{n}} \qquad \text{such that } \boldsymbol{\delta}_i = \begin{cases} 0 & \text{w.p. } 1 - \dfrac{\sum_{j\in\mathcal{V}} -(z_j)}{\|z\|_1} \\ j^3 & \text{w.p. } \dfrac{-z_j}{\|z\|_1} \text{ for } j \in \mathcal{V}. \end{cases} \tag{28}$$

Again, we note that the probabilities are at most 1 since $\boldsymbol{\delta}_i \leq \ell^3$ as well. We record a claim that follows directly from the definition of z, $\mathcal{W}$ and $\mathcal{V}$:

Claim 30 *For all $k = 1, \ldots, \ell - 1$, we have*

$$\mathop{\mathbf{E}}_{\boldsymbol{\delta}_i}\left[\boldsymbol{\delta}_i^k\right] = \mathop{\mathbf{E}}_{\boldsymbol{\gamma}_i}\left[\boldsymbol{\gamma}_i^k\right]. \tag{29}$$

Proof: The proof follows from the fact that

$$\mathop{\mathbf{E}}_{\boldsymbol{\gamma}}\left[\boldsymbol{\gamma}_i^k\right] - \mathop{\mathbf{E}}_{\boldsymbol{\delta}_i}\left[\boldsymbol{\delta}_i^k\right] = \frac{1}{\|z\|_1}\sum_{j=1}^{\ell} \alpha_j^k z_j = \frac{1}{\|z\|_1}(A(\alpha)z)_{k+1} = 0,$$

since $A(\alpha)z = e_1$. ■

We show in the next two claims that (22) holds when n is sufficiently large.

Claim 31 *We have $\boldsymbol{p} \in \texttt{Junta}(n/2)$ with probability at least $1 - o_n(1)$ over the draw of $\boldsymbol{p} \sim \mathcal{D}_{yes}$.*

Proof: Let $\boldsymbol{p} \sim \mathcal{D}_{\text{yes}}$, and let $\mathbf{A} \subseteq [n]$ be the set of coordinates $i \in [n]$ with $\boldsymbol{\delta}_i \neq 0$. We will show that, when n is sufficiently large, $|\mathbf{A}| \leq n/2$ with probability $1 - o_n(1)$, which implies that $\boldsymbol{p} \sim \mathcal{D}_{\text{yes}}$ is an $(n/2)$-junta for $\mathcal{U}_n$ with probability at least $1 - o_n(1)$.

To see this is the case, we notice that each $\boldsymbol{\delta}_i$ is 0 with probability

$$1 - \frac{\sum_{j\in\mathcal{V}} -z_j}{\|z\|_1} = \frac{1}{2}\left(1 + \frac{\sum_{j\in\mathcal{W}} z_j + \sum_{j\in\mathcal{V}} z_j}{\|z\|_1}\right) = \frac{1}{2} + \frac{1}{2\|z\|_1} \geq \frac{1}{2} + \frac{1}{2C^*},$$

where we used the fact that z was the solution to $(A(\alpha)z)_1 = 1$ to deduce that $\sum_j z_j = 1$. Hence, for large n, we apply a Chernoff bound to deduce that $|\mathbf{A}| \leq n/2$ except with probability $o_n(1)$. ■

Claim 32 *We have $\boldsymbol{p}$ is ϵ-far from $\texttt{Junta}(n/2)$ with probability at least $1 - o_n(1)$ over the draw of $\boldsymbol{p} \sim \mathcal{D}_{no}$.*

Proof: By a similar computation, as the proof of Claim 31, if we let $\mathbf{A}$ be the subset of coordinates $i \in [n]$ with $\boldsymbol{\gamma}_i = 0$ in $\boldsymbol{p} \sim \mathcal{D}_{\text{no}}$, we have

$$|\mathbf{A}| \leq n\left(\frac{1}{2} - \frac{1}{4C^*}\right)$$

except with probability $o_n(1)$. Consider a fixed distribution p in the support of $\mathcal{D}_{\text{no}}$ where the above event occurs, i.e., the set $A \subset [n]$ of coordinates with zero γ_i (specifying the marginal distributions of p as in (27)) is smaller than $n/2 - n/(4C^*)$. Let q be any $(n/2)$-junta distribution and let S be the influential variables of q's p.d.f with $|S| \leq n/2$. We have that, for each $i \in \overline{A} \cap \overline{S}$,

$$|\mu(p)_i| \geq 2\tau\gamma_i/\sqrt{n} \geq 2\tau/\sqrt{n}.$$

Let T be $\overline{A} \cap \overline{S}$ with

$$t \overset{\text{def}}{=} |T| = |\overline{A} \cap \overline{S}| \geq n\left(\frac{1}{2} + \frac{1}{4C^*}\right) - \frac{n}{2} \geq \frac{n}{4C^*}.$$

Consider the distributions p_T and q_T given by taking a sample and projecting onto the coordinates in T. Since $T \subset \overline{S}$, and the p.d.f of q is constant for any setting of variables in S, the distribution q_T is the uniform distribution over t bits. We note

$$\begin{aligned} d_{\text{TV}}(p_T, \mathcal{U}_t) &= \frac{1}{2} \sum_{x \in \{-1,1\}^T} |p_T(x) - q_T(x)| = \frac{1}{2} \sum_{x \in \{-1,1\}^T} \left| \sum_{y \in \{-1,1\}^{\overline{T}}} p(x,y) - q(x,y) \right| \\ &\leq \frac{1}{2} \sum_{z \in \{-1,1\}^n} |p(z) - q(z)| = d_{\text{TV}}(p,q), \end{aligned} \tag{30}$$

where $p(x,y) = p(z)$ with $z_i = x_i$ for $i \in T$ and $z_i = y_i$ for $i \notin T$, and $q(x,y)$ is defined analogously. We now apply Lemma 28 to deduce a lower bound on $d_{\text{TV}}(p_T, \mathcal{U}_t)$, and by (30) lower bound $d_{\text{TV}}(p,q)$. Since p is a product distribution, $\mu(p)_i = \mu(p_T)_i$ for all $i \in T$, and we have

$$\|\mu(p_T)\|_\infty \leq \frac{2\tau\ell^3}{\sqrt{n}} \qquad \text{and} \qquad \|\mu(p_T)\|_2 \geq \sqrt{t} \cdot \frac{2\tau}{\sqrt{n}} = \frac{\tau}{\sqrt{C^*}}. \tag{31}$$

Applying Lemma 28, we have

$$d_{\text{TV}}(p_T, \mathcal{U}_t) \geq \left(\frac{1}{8} - o_n(1)\right) \cdot \min\left(c_2^*, \frac{\tau}{4\sqrt{C^*}}\right) \geq \min\left(\frac{c_2^*}{9}, \frac{\tau}{36\sqrt{C^*}}\right),$$

once n is a large enough constant. Finally, by the setting of ϵ_0 in (24), and τ in (26), $d_{\text{TV}}(p_T, \mathcal{U}_t) \geq \min(\epsilon_0, \epsilon) = \epsilon$ for large enough n. Since the distribution q was an arbitrary $(n/2)$-junta distribution, this concludes the proof. ■

D.2. Statistical Distance Between $\mathcal{R}_{\text{yes}}$ and $\mathcal{R}_{\text{no}}$

Let $s = \lceil n/(\epsilon^2 \log^{12} n) \rceil$. We show that distributions $\mathcal{R}_{\text{yes}} = \mathcal{R}(s, \mathcal{D}_{\text{yes}})$ and $\mathcal{R}_{\text{no}} = \mathcal{R}(s, \mathcal{D}_{\text{no}})$ as defined in (23) using $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ satisfy

$$d_{\text{TV}}(\mathcal{R}_{\text{yes}}, \mathcal{R}_{\text{no}}) \leq o_n(1). \tag{32}$$

Recall that $\mathcal{R}_{\text{yes}}$ is the distribution supported on $\{0, \dots, s\}^n$ given by first sampling $\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_n$ independently according to (28) and then sampling from the product distribution

$$\boldsymbol{r} \sim \prod_{i=1}^{n} \text{Bin}\left(s, \boldsymbol{q}_i\right), \qquad \text{where} \quad \boldsymbol{q}_i \overset{\text{def}}{=} \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \boldsymbol{q}}\left[\boldsymbol{x}_i = 1\right] = \frac{1}{2} + \frac{\boldsymbol{\delta}_i \cdot \tau}{\sqrt{n}}. \tag{33}$$

Notice that we always have

$$\frac{1}{2} \leq \boldsymbol{q}_i \leq \frac{1}{2} + \frac{\tau \ell^3}{\sqrt{n}} \leq \frac{1}{2} + O\left(\frac{\epsilon \log^3 n}{\sqrt{n}}\right)$$

once n is a large enough constant.

Similarly, $\mathcal{R}_{\text{no}}$ is the distribution supported on $\{0, \dots, s\}^n$ given by first sampling $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_n$ according to (27), and then sampling from the product distribution

$$\boldsymbol{r} \sim \prod_{i=1}^{n} \text{Bin}\left(s, \boldsymbol{p}_i\right), \qquad \text{where} \qquad \boldsymbol{p}_i \overset{\text{def}}{=} \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \boldsymbol{p}}\left[\boldsymbol{x}_i = 1\right] = \frac{1}{2} + \frac{\boldsymbol{\gamma}_i \cdot \tau}{\sqrt{n}},$$

and similarly, we have $1/2 \leq \boldsymbol{p}_i \leq 1/2 + O(\epsilon \log^3 n/\sqrt{n})$. In particular, if we denote the set $B \subset \{0, \dots, s\}^n$ given by

$$B = \left\{ r = (r_1, \dots, r_n) \in \{0, \dots, s\}^n : \exists j \in [n], \left| r_j - \frac{s}{2} \right| \geq \sqrt{s} \log^2 n \right\}.$$

It follows from our choice of s, that for every $i \in [n]$ and any fixed setting of $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$ and $\boldsymbol{q}_1, \dots, \boldsymbol{q}_n$,

$$\frac{s}{2} \leq \mathop{\mathbf{E}}_{\boldsymbol{r}_i \sim \text{Bin}(s, \boldsymbol{p}_i)}\left[\boldsymbol{r}_i\right], \mathop{\mathbf{E}}_{\boldsymbol{r}_i \sim \text{Bin}(s, \boldsymbol{q}_i)}\left[\boldsymbol{r}_i\right] \leq \frac{s}{2} + O\left(\frac{s\epsilon \log^3 n}{\sqrt{n}}\right) = \frac{s}{2} + O\left(\sqrt{s}\right),$$

so that via a Chernoff bound and a union bound,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{r} \sim \mathcal{R}_{\text{yes}}}[\boldsymbol{r} \in B], \mathop{\mathbf{Pr}}_{\boldsymbol{r} \sim \mathcal{R}_{\text{no}}}[\boldsymbol{r} \in B] = o_n(1).$$

Therefore, in order to show $d_{\text{TV}}(\mathcal{R}_{\text{yes}}, \mathcal{R}_{\text{no}}) = o_n(1)$, it suffices to show that for every $r \notin B$,

$$\frac{\mathbf{Pr}_{\boldsymbol{r} \sim \mathcal{R}_{\text{yes}}}\left[\boldsymbol{r} = r\right]}{\mathbf{Pr}_{\boldsymbol{r} \sim \mathcal{R}_{\text{no}}}\left[\boldsymbol{r} = r\right]} = \frac{\mathbf{E}_{\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_n}\left[\prod_{i=1}^{n}\left(\binom{s}{r_i}\left(\frac{1}{2} + \frac{\boldsymbol{\delta}_i \tau}{\sqrt{n}}\right)^{r_i}\left(\frac{1}{2} - \frac{\boldsymbol{\delta}_i \tau}{\sqrt{n}}\right)^{s - r_i}\right)\right]}{\mathbf{E}_{\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_n}\left[\prod_{i=1}^{n}\left(\binom{s}{r_i}\left(\frac{1}{2} + \frac{\boldsymbol{\gamma}_i \tau}{\sqrt{n}}\right)^{r_i}\left(\frac{1}{2} - \frac{\boldsymbol{\gamma}_i \tau}{\sqrt{n}}\right)^{s - r_i}\right)\right]} \leq 1 + o_n(1). \tag{34}$$

Toward this goal, consider a fixed $r \notin B$, and notice that since $\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_n$ are drawn independently, the numerator in (34) is

$$\prod_{i=1}^{n} \mathop{\mathbf{E}}_{\boldsymbol{\delta}_i}\left[\binom{s}{r_i}\left(\frac{1}{2}+\frac{\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^{r_i}\left(\frac{1}{2}-\frac{\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^{s-r_i}\right]$$
$$=\prod_{i=1}^{n}\binom{s}{r_i}\cdot\frac{1}{2^s}\cdot\mathop{\mathbf{E}}_{\boldsymbol{\delta}_i}\left[\left(1-\left(\frac{2\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^2\right)^{m_i}\left(1-\mathrm{sgn}(t_i)\cdot\frac{2\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^{|t_i|}\right], \qquad (35)$$

where $t_i = s - 2r_i$ and $m_i = \min\{r_i, s-r_i\}$; notice that $|t_i| \le 2\sqrt{s}\log^2 n$ since $r \notin B$. Similarly, the denominator in (34) may be expressed as (35) by replacing $\boldsymbol{\delta}_i$ with $\boldsymbol{\gamma}_i$. We analyze (34) by considering each term in the product; in particular, it suffices to show that for every $i \in [n]$,

$$\frac{\mathbf{E}_{\boldsymbol{\delta}_i}\left[\left(1-4\boldsymbol{\delta}_i^2\tau^2/n\right)^{m_i}\left(1-\mathrm{sgn}(t_i)\cdot 2\boldsymbol{\delta}_i\tau/\sqrt{n}\right)^{|t_i|}\right]}{\mathbf{E}_{\boldsymbol{\gamma}_i}\left[\left(1-4\boldsymbol{\gamma}_i^2\tau^2/n\right)^{m_i}\left(1-\mathrm{sgn}(t_i)\cdot 2\boldsymbol{\gamma}_i\tau/\sqrt{n}\right)^{|t_i|}\right]} \le 1 + o_n(1/n). \qquad (36)$$

Using the choice of s and the fact that both $\boldsymbol{\delta}_i$ and $\boldsymbol{\gamma}_i$ are no larger than $\log^3 n$, we always have

$$\left(1-\frac{4\boldsymbol{\delta}_i^2\tau^2}{n}\right)^{m_i}, \left(1-\mathrm{sgn}(t_i)\cdot\frac{2\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^{|t_i|}, \left(1-\frac{4\boldsymbol{\gamma}_i^2\tau^2}{n}\right)^{m_i}, \left(1-\mathrm{sgn}(t_i)\cdot\frac{2\boldsymbol{\gamma}_i\tau}{\sqrt{n}}\right)^{|t_i|} = 1 \pm o_n(1). \qquad (37)$$

In addition, we have,

$$\left(1-\frac{4\boldsymbol{\delta}_i^2\tau^2}{n}\right)^{m_i} = \sum_{k=0}^{m_i}\binom{m_i}{k}\left(\frac{-4\boldsymbol{\delta}_i^2\tau^2}{n}\right)^k$$
$$= \sum_{k=0}^{\ell/4-1}\binom{m_i}{k}\left(\frac{-4\boldsymbol{\delta}_i^2\tau^2}{n}\right)^k + \sum_{k=\ell/4}^{m_i}\binom{m_i}{k}\left(\frac{-4\boldsymbol{\delta}_i^2\tau^2}{n}\right)^k. \qquad (38)$$

For each term in the second sum, we upperbound $\boldsymbol{\delta}_i \le \ell^3$ and use the approximation of $\binom{m_i}{k} \le (em_i/k)^k$. We also use $k \ge \ell/4$, $m_i \ge s/3$ and the choice of $\ell = \lceil \log n/\log\log n \rceil$. As a result, the absolute value of the kth term in the second sum is at most

$$\left(\frac{em_i \cdot 4\ell^6 \cdot O(\epsilon^2)}{kn}\right)^k \le \left(O\left(\frac{s\ell^5\epsilon^2}{n}\right)\right)^k \le \left(\frac{1}{\log^6 n}\right)^k. \qquad (39)$$

As a result, the absolute value of the second sum is at most

$$\sum_{k=\ell/4}^{m_i}\left(\frac{1}{\log^6 n}\right)^k \le 2\cdot\left(\frac{1}{\log^6 n}\right)^{\frac{\log n}{4\log\log n}} = o_n(1/n).$$

In fact, we have shown, by negating all terms in (38) of degree (in $\boldsymbol{\delta}_i$) at least $\ell/4$,

$$\left(1-\frac{4\boldsymbol{\delta}_i^2\tau^2}{n}\right)^{m_i} = \sum_{k=0}^{\ell/4-1}\binom{m_i}{k}\left(\frac{-4\tau^2}{n}\right)^k\cdot\boldsymbol{\delta}_i^{2k} \pm o_n(1/n). \qquad (40)$$

Similarly,

$$\begin{aligned}\left(1-\frac{2\mathrm{sgn}(t_i)\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^{|t_i|} &= \sum_{k=0}^{|t_i|}\binom{|t_i|}{k}\left(\frac{-2\mathrm{sgn}(t_i)\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^k \\ &= \sum_{k=0}^{\ell/2-1}\binom{|t_i|}{k}\left(\frac{-2\mathrm{sgn}(t_i)\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^k + \sum_{k=\ell/2}^{|t_i|}\binom{|t_i|}{k}\left(\frac{-2\mathrm{sgn}(t_i)\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^k.\end{aligned}$$

Analogously to (39), the absolute value of the second sum can be bounded from above by

$$\sum_{k=\ell/2}^{|t_i|}\left(O\left(\frac{|t_i|}{k}\cdot\frac{\epsilon\ell^3}{\sqrt{n}}\right)\right)^k \le 2\left(O\left(\frac{1}{\log^2 n\log^2(\log n)}\right)\right)^{\frac{\log n}{2\log\log n}} = o_n(1/n)$$

and we have

$$\left(1-\frac{2\mathrm{sgn}(t_i)\boldsymbol{\delta}_i\tau}{\sqrt{n}}\right)^{|t_i|} = \sum_{k=0}^{\ell/2-1}\binom{|t_i|}{k}\left(\frac{-2\mathrm{sgn}(t_i)\tau}{\sqrt{n}}\right)^k\cdot\boldsymbol{\delta}_i^k \pm o_n(1/n). \tag{41}$$

Analogously, we may conclude that

$$\begin{aligned}\left(1-\frac{4\boldsymbol{\gamma}_i^2\tau^2}{n}\right)^{m_i} &= \sum_{k=0}^{\ell/4-1}\binom{m_i}{k}\left(\frac{-4\tau^2}{n}\right)^k\cdot\boldsymbol{\gamma}_i^{2k}\pm o_n(1/n) \qquad \text{and} \\ \left(1-\frac{2\mathrm{sgn}(t_i)\boldsymbol{\gamma}_i\tau}{\sqrt{n}}\right)^{|t_i|} &= \sum_{k=0}^{\ell/2-1}\binom{|t_i|}{k}\left(\frac{-2\mathrm{sgn}(t_i)\tau}{\sqrt{n}}\right)^k\cdot\boldsymbol{\gamma}_i^k \pm o_n(1/n). \end{aligned}\tag{42}$$

It follows from (37) and all four approximations in (40), (41) and (42) that all four sums on the right hand side are $1\pm o_n(1)$, and note that all these inequalities hold with probability 1 (over the draw of $\boldsymbol{\delta}_i$ and $\boldsymbol{\gamma}_1$). Putting (40), (41), (42) and (37) together, we have

$$\begin{aligned}&\mathop{\mathbf{E}}_{\boldsymbol{\delta}_i}\left[\left(1-4\boldsymbol{\delta}_i^2\tau^2/n\right)^{m_i}\left(1-\mathrm{sgn}(t_i)\cdot 2\boldsymbol{\delta}_i\tau/\sqrt{n}\right)^{|t_i|}\right] \qquad (43)\\ &\quad\le \mathop{\mathbf{E}}_{\boldsymbol{\delta}_i}\left[\left(\sum_{k=0}^{\ell/4-1}\binom{m_i}{k}\left(\frac{-4\tau^2}{n}\right)^k\cdot\boldsymbol{\delta}_i^{2k}+o_n(1/n)\right)\left(\sum_{k=0}^{\ell/2-1}\binom{|t_i|}{k}\left(\frac{-2\mathrm{sgn}(t_i)\tau}{\sqrt{n}}\right)^k\cdot\boldsymbol{\delta}_i^k+o_n(1/n)\right)\right]\\ &\quad\le \mathop{\mathbf{E}}_{\boldsymbol{\delta}_i}\left[\left(\sum_{k=0}^{\ell/4-1}\binom{m_i}{k}\left(\frac{-4\tau^2}{n}\right)^k\cdot\boldsymbol{\delta}_i^{2k}\right)\left(\sum_{k=0}^{\ell/2-1}\binom{|t_i|}{k}\left(\frac{-2\mathrm{sgn}(t_i)\tau}{\sqrt{n}}\right)^k\cdot\boldsymbol{\delta}_i^k\right)\right]+o_n(1/n),\end{aligned}$$

$$\begin{aligned}&\mathop{\mathbf{E}}_{\boldsymbol{\gamma}_i}\left[\left(1-4\boldsymbol{\gamma}_i^2\tau^2/n\right)^{m_i}\left(1-\mathrm{sgn}(t_i)\cdot 2\boldsymbol{\gamma}_i\tau/\sqrt{n}\right)^{|t_i|}\right] \qquad (44)\\ &\quad\ge \mathop{\mathbf{E}}_{\boldsymbol{\gamma}_i}\left[\left(\sum_{k=0}^{\ell/4-1}\binom{m_i}{k}\left(\frac{-4\tau^2}{n}\right)^k\cdot\boldsymbol{\gamma}_i^{2k}\right)\left(\sum_{k=0}^{\ell/2-1}\binom{|t_i|}{k}\left(\frac{-2\mathrm{sgn}(t_i)\tau}{\sqrt{n}}\right)^k\cdot\boldsymbol{\gamma}_i^k\right)\right]-o_n(1/n).\end{aligned}$$

Hence, notice that (43) and (44) are both $1 \pm o_n(1)$, and can each be expressed as the same linear function of the first $\ell - 1$ moments of $\boldsymbol{\delta}_i$ and $\boldsymbol{\gamma}_i$ up to additive errors $\pm o_n(1/n)$. Since the first $\ell - 1$ moments of $\boldsymbol{\delta}_i$ and $\boldsymbol{\gamma}_i$ are equal by Claim 30, we have shown (34), which completes the proof of (32).

D.3. Proof of Lemma 28

We will use the fact that $e^{-x} \leq 1 - x/2$ for all $x \in [0, 1]$, which implies that

$$e^{-x} \leq \max\left(e^{-1}, 1 - x/2\right) \tag{45}$$

for all $x \geq 0$. We set the constant c^* in Lemma 28 to be $1 - e^{-1}$.

Let $\mu = \mu(p)$ for convenience and we assume without loss of generality that $\mu_i \geq 0$ for all $i \in [n]$. A sample $\boldsymbol{x} \sim p$ has all coordinates set independently, where the ith coordinate of $\boldsymbol{x}_i$ is 1 with probability $(1 + \mu_i)/2$ and -1 with probability $(1 - \mu_i)/2$. Given any $x \in \{-1, 1\}^n$, we have

$$p(x) = \prod_{\substack{i \in [n] \\ x_i = 1}} \left(\frac{1 + \mu_i}{2}\right) \cdot \prod_{\substack{i \in [n] \\ x_i = -1}} \left(\frac{1 - \mu_i}{2}\right) = \frac{1}{2^n} \cdot \prod_{\substack{i \in [n] \\ x_i = 1}} (1 + \mu_i) \cdot \prod_{\substack{i \in [n] \\ x_i = -1}} (1 - \mu_i).$$

We say a string $x \in \{-1, 1\}^n$ is *good* if

$$\sum_{i \in [n]} \mu_i x_i \leq -\frac{\|\mu\|_2}{2}.$$

The proof proceeds in two steps. First we show that there exists a constant $c_1^* > 0$ such that when $\boldsymbol{x}$ is drawn uniformly at random from $\{-1, 1\}^n$, $\boldsymbol{x}$ is good with probability at least

$$\frac{1}{4} - \frac{c_1^* \|\mu\|_\infty}{\|\mu\|_2}.$$

Next we show there exists a constant $c_2^* > 0$ that every good string $x \in \{-1, 1\}^n$ satisfies

$$\left| p(x) - \frac{1}{2^n} \right| \geq \frac{1}{2^n} \cdot \min\left(c_2^*, \frac{\|\mu\|_2}{4}\right).$$

The lemma follows since

$$\begin{aligned} d_{\mathrm{TV}}(p, \mathcal{U}_n) = \frac{1}{2} \sum_{x \in \{-1,1\}^n} \left| p(x) - \frac{1}{2^n} \right| &\geq \frac{1}{2} \sum_{\substack{x \in \{-1,1\}^n \\ \text{good } x}} \left| p(x) - \frac{1}{2^n} \right| \\ &\geq \frac{1}{2} \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \{-1,1\}^n} \left[\boldsymbol{x} \text{ is good}\right] \cdot \min\left(c^*, \frac{\|\mu\|_2}{4}\right). \end{aligned}$$

For the first step, we let $\boldsymbol{x} \sim \{-1, 1\}^n$ be drawn uniformly at random and write $\mathbf{y}_i = \mu_i \boldsymbol{x}_i$. We recall the Berry–Esséen theorem:

Theorem 33 (Berry–Esséen) *There exists a universal constant $c_1^* > 0$ such that letting $\boldsymbol{s} = \mathbf{y}_1 + \cdots + \mathbf{y}_n$, where $\mathbf{y}_1, \ldots, \mathbf{y}_n$ be independent real-valued random variables with $\mathbf{E}[\mathbf{y}_i] = 0$ and $\mathbf{Var}[\mathbf{y}_i] = \sigma_i^2$, and suppose that $|\mathbf{y}_i| \le \tau$ with probability 1 for all $i \in [n]$. Let $\boldsymbol{g}$ be a Gaussian random variable with mean 0 and variance $\sum_{i\in[n]} \sigma_j^2$, matching those of $\boldsymbol{s}$. Then for all $\theta \in \mathbb{R}$ we have*

$$\Big|\mathbf{Pr}[\boldsymbol{s} \le \theta] - \mathbf{Pr}[\boldsymbol{g} \le \theta]\Big| \le \frac{c_1^* \tau}{\sqrt{\sum_{i\in[n]} \sigma_i^2}}.$$

Note that in our case, $\tau = \|\mu\|_\infty$ and $\sigma_i^2 = \mu_i^2$ and thus, the variance of $\boldsymbol{g}$ is $\|\mu\|_2^2$.

Recall the following fact about Gaussian anti-concentration:

Fact 34 (Gaussian anti-concentration) *Let $\boldsymbol{g}$ be a Gaussian random variable with variance σ^2. Then for all $\kappa > 0$ it holds that*

$$\sup_{\theta\in\mathbb{R}} \Big\{ \mathbf{Pr}\left[|\boldsymbol{g} - \theta| \le \kappa\sigma\right] \Big\} \le \kappa.$$

Setting $\kappa = 1/2$ and $\theta = 0$ (and using the symmetry of $\boldsymbol{g}$), we have that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{g}\sim\mathcal{N}(0,\|\mu\|_2^2)} \Big[\boldsymbol{g} \le -\|\mu\|_2/2\Big] \ge 1/4.$$

It follows from Berry-Esséen that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\{-1,1\}^n} \left[\sum_{i\in[n]} \mu_i \boldsymbol{x}_i \le -\frac{\|\mu\|_2}{2}\right] \ge \frac{1}{4} - \frac{c_1^*\|\mu\|_\infty}{\|\mu\|_2}.$$

This finishes the proof of the first step. For the second we use the fact that $e^x \ge 1 + x$ for all $x \in \mathbb{R}$ and thus, for each good $x \in \{-1,1\}^n$ we have

$$2^n \cdot p(x) \le \prod_{\substack{i\in[n]\\ x_i=1}} e^{\mu_i} \cdot \prod_{\substack{i\in[n]\\ x_i=1}} e^{-\mu_i} = e^{\sum_{i\in[n]} \mu_i x_i} \le e^{-\|\mu\|_2/2} \le \max\left(e^{-1}, 1 - \|\mu\|_2/4\right),$$

where we used (45) in the last inequality. As a result, we have

$$\left|1 - 2^n \cdot p(x)\right| = 1 - 2^n \cdot p(x) \ge \min\left(c_2^*, \|\mu\|_2/4\right)$$

since we set $c_2^* = 1 - e^{-1}$. This finishes the proof of the lemma.

D.4. Proof of Claim 29

Applying Cramer's rule, we have

$$|z_i| = \left|\frac{\det(A_i)}{\det(A)}\right| = \prod_{j\in[\ell]\setminus\{i\}} \left|\frac{\alpha_j}{\alpha_i - \alpha_j}\right|, \tag{46}$$

where A_i is the $\ell \times \ell$ matrix given by replacing the i-th column with e_1, and notice that A_i is the Vandermonde matrix $A(\alpha^{(i)})$, with $\alpha^{(i)} \in \mathbb{R}^\ell$ being the vector which is exactly α_j on all $j \ne i$ and

0 when $j = i$. We now show that there exists a constant $i_0 \in \mathbb{N}$ (which does not depend on ℓ) such that for all $\ell \in \mathbb{N}$, the sequence $\{|z_i|\}_{i \geq i_0}$ is geometrically decreasing with constant bounded away from 1. This suffices to bound $\|z\|_1$, since

$$\|z\|_1 = \sum_{i=1}^{\ell} |z_i| \leq \sum_{i=1}^{i_0-1} |z_i| + \sum_{i=i_0}^{\ell} |z_i| \leq (i_0 - 1) \max_{i<i_0} |z_i| + O(|z_{i_0}|),$$

and for every $i \in [\ell]$, we can upperbound the logarithm of (46) by

$$\log_2\left(|z_i|\right) \leq (i-1)\log_2 i + \sum_{j>i} \log_2\left(1 + \frac{i^3}{j^3 - i^3}\right) \leq i^3 \left(1 + \sum_{j>i} \frac{1}{j^3 - i^3}\right) \lesssim i^3.$$

The first inequality follows from the fact that $|j^3/(i^3 - j^3)| \leq i$ for $j < i$; the second inequality follows from upperbounding $\log(1+x) \leq x$ for $x \geq 0$; the last inequality follows from the fact $j^3 - i^3 > (j-i)^3$ for all $j > i$, and the sums to a constant. From the upper bound on $\log_2 |z_i|$, we may conclude $\|z\|_1 \leq 2^{O(i_0^3)}$.

In order to pick $i_0 \in \mathbb{N}$, notice that for all $i \in \mathbb{N}$, we use (46) on z_{i+1} and z_i to obtain

$$\frac{|z_{i+1}|}{|z_i|} = \prod_{j=0}^{i-1} \frac{i^3 - j^3}{(i+1)^3 - j^3} \cdot \prod_{j=i+2}^{\ell} \frac{j^3 - i^3}{j^3 - (i+1)^3}.$$

We first handle the case when $\ell \geq 2i + 1$. In this case we break the product into

$$\frac{|z_{i+1}|}{|z_i|} = \prod_{k=1}^{i} \left(\frac{i^3 - (i-k)^3}{(i+1)^3 - (i-k)^3} \cdot \frac{(i+1+k)^3 - i^3}{(i+1+k)^3 - (i+1)^3} \right) \prod_{j=2i+2}^{\ell} \frac{j^3 - i^3}{j^3 - (i+1)^3}. \tag{47}$$

Using $a^3 - b^3 = (a-b)(a^2 + ab + b^2)$, the factor for each $k \in [i]$ in the first product becomes

$$\frac{3i^2 - 3ki + k^2}{3i^2 - 3(k-1)i + k^2 - k + 1} \cdot \frac{3i^2 + 3(k+1)i + (k+1)^2}{3i^2 + 3(k+2)i + k^2 + 3k + 3}. \tag{48}$$

Noting that the denominator of the first factor is

$$(i+1)^2 + (i+1)(i-k) + (i-k)^2 \leq (2i + 1 - k)^2$$

we can bound the first factor of (48) by

$$1 - \frac{3i - k + 1}{(i+1)^2 + (i+1)(i-k) + (i-k)^2} \leq 1 - \frac{3i - k + 1}{(2i + 1 - k)^2} \leq 1 - \frac{1}{2i + 1 - k}.$$

Similarly we have that the second factor of (48) is

$$1 - \frac{3i + k + 2}{(i+1+k)^2 + (i+1+k)(i+1) + (i+1)^2} \leq 1 - \frac{3i + k + 2}{(2i + 2 + k)^2} \leq 1 - \frac{1}{2i + k + 2}.$$

As a result, the first product in (48) is at most (using $1 + x \leq e^x$)

$$\exp\left(- \sum_{k=1}^{i} \left(\frac{1}{2i + 1 - k} + \frac{1}{2i + k + 2} \right) \right).$$

We note that by re-indexing terms,

$$\sum_{k=1}^{i}\left(\frac{1}{2i+1-k}+\frac{1}{2i+k+2}\right)=\sum_{h=i+1}^{3i+2}\frac{1}{h}-\left(\frac{1}{2i+1}+\frac{1}{2i+2}\right)\geq\int_{i+1}^{3i+2}\frac{1}{x}\cdot dx-\frac{1}{i}\xrightarrow{i\to\infty}\ln(3)$$

where the sum approaches $\ln 3$ as i grows so we fix our i_0 to be sufficiently large so that when $i \geq i_0$ the above sum is at least $1+\frac{1}{20}$. For the second product of (48) we rewrite it as

$$\begin{aligned}\prod_{j=2i+2}^{\ell}\frac{j^3-i^3}{j^3-(i+1)^3}&=\prod_{k=i+1}^{\ell-i-1}\left(1+\frac{3i^2+3i+1}{(i+1+k)^3-(i+1)^3}\right)\\&\leq\prod_{k=i+1}^{\ell-i-1}\left(1+\frac{3i^2+3i+1}{3(i+1)k^2+k^3}\right)\\&\leq\prod_{k=i+1}^{\ell-i-1}\left(1+\frac{i}{k^2}\right)\leq\exp\left(\sum_{k\geq i+1}\frac{i}{k^2}\right)\leq e,\end{aligned}$$

where the third inequality used $3i^2+3i+1 \leq i(3i+3+k)$ and the last inequality used the fact that $\sum_{k\geq i+1} 1/k^2 \leq 1/i$. As a result, in this case ($i \geq i_0$ and $\ell \geq 2i+1$) we have that $|z_{i+1}|/|z_i| \leq e^{-1/20}$. We are almost done. For the case when $\ell < 2i+1$, we simply note that

$$\frac{|z_{i+1}|}{|z_i|}\leq\prod_{k=1}^{i}\frac{i^3-(i-k)^3}{(i+1)^3-(i-k)^3}\cdot\frac{(i+1+k)^3-i^3}{(i+1+k)^3-(i+1)^3}$$

since we added more factors that are at least 1. Since $i \geq i_0$, the same argument used earlier implies that the ratio is at most $e^{-1-1/20}$.

Appendix E. Robust Mean Testing for k-Juntas

In this section, we consider a robust distribution testing algorithm which distinguishes between a given distribution p having a mean vector $\mu(p)$ with large ℓ_2 norm, and p being a k-junta distribution and having a mean vector with small ℓ_2 norm. Our tester is similar to the mean testing algorithm of Canonne et al. (2019), however we will require a tighter analysis of the completeness case, which in our setting is more general. The goal of this section is to demonstrate an algorithm that draws a small number of samples from p to distinguish these two cases with probability at least $2/3$. We restate the main theorem of this section:

Theorem 8 (Robust mean testing for juntas) *There is an algorithm which, given sample access to a distribution p on $\{-1,1\}^n$, $k \in \mathbb{N}$ and a parameter $\epsilon \in (0,1)$, has the following behavior:*

1. *If p is a k-junta distribution with $\|\mu(p)\|_2 \leq \epsilon\sqrt{n}/100$, the algorithm returns* "`Is a k-junta`" *with probability at least $2/3$;*
2. *If p is a distribution that satisfies $\|\mu(p)\|_2 \geq \epsilon\sqrt{n}$, the algorithm returns* "`Not a k-junta`" *with probability at least $2/3$.*

Moreover, the algorithms draws

$$q = O\left(\max\left\{\frac{k+\sqrt{n}}{\epsilon^2 n}, \frac{k+\sqrt{n}}{\epsilon\sqrt{n}}\right\}\right) \tag{2}$$

samples from p and runs in time $O(q^2 n)$.

To describe the testing algorithm we start with some notation.

Definition 35 *Given $x \in \{-1,1\}^n$, we write $x \otimes y$ to denote the tensor product of x and y:*

$$x \otimes x = (x_1x_1, x_1x_2, \dots x_1x_n, x_2x_1, \dots x_nx_n) \in \{-1,1\}^{n^2}.$$

We also write $x^{\otimes r}$ to denote the tensor product of r copies of x:

$$x^{\otimes r} = \underbrace{x \otimes x \otimes \cdots \otimes x}_{r}.$$

Given a distribution p over $\{-1,1\}^n$, we define the tensor-distribution $\odot(p)$ *of p, a distribution over $\{-1,1\}^{n^2}$, as the distribution of $\boldsymbol{x} \otimes \boldsymbol{x}$ with $\boldsymbol{x} \sim p$. We define $\odot^r(p)$ recursively as $\odot^r(p) = \odot(\odot^{r-1}(p))$ with $\odot^0(p) = p$, which is a distribution of dimension n^{2^r}. We call $\odot^r(p)$ the* r-th order tensor distribution *of p and note that, equivalently, $\odot^r(p)$ is the distribution of $\boldsymbol{x}^{\otimes 2^r}$ with $\boldsymbol{x} \sim p$.*

The following claim follows from the definition of tensor-distributions since $\mu(\odot^{r+1}(p))$ is the vectorization of the covariance matrix $\Sigma(\odot^r(p))$.

Claim 36 *Given p over $\{-1,1\}^n$ and $r \geq 0$, we have $\|\mu(\odot^{r+1}(p))\|_2 = \|\Sigma(\odot^r(p))\|_F$.*

Let p be a distribution over $\{-1,1\}^n$. The main test statistic used by our algorithm first draws $2q$ samples $\mathbf{X}_1, \dots, \mathbf{X}_q$ and $\mathbf{Y}_1, \dots, \mathbf{Y}_q$ independently from p, for some q to be specified, and construct

$$\overline{\mathbf{X}} = \frac{1}{q}\sum_{i=1}^{q} \mathbf{X}_i \qquad \text{and} \qquad \overline{\mathbf{Y}} = \frac{1}{q}\sum_{i=1}^{q} \mathbf{Y}_i.$$

We then set $\mathbf{Z} = \langle \overline{\mathbf{X}}, \overline{\mathbf{Y}} \rangle$. We use the following lemma (Lemma 4.1) from Canonne et al. (2019):

Proposition 37 *Let p be a distribution over $\{-1,1\}^n$. Then we have*

$$\begin{aligned}\mathbf{E}\left[\mathbf{Z}\right] &= \left\|\mu(p)\right\|_2^2 \\ \mathbf{Var}\left[\mathbf{Z}\right] &\leq \frac{1}{q^2}\cdot\left\|\Sigma(p)\right\|_F^2 + \frac{4}{q}\cdot\left\|\mu(p)\right\|_2^2\cdot\left\|\Sigma(p)\right\|_F.\end{aligned}$$

We will use the above test statistic for higher order tensor distributions of p. For $r \geq 0$, given $2q$ samples $\mathbf{X}_1, \dots, \mathbf{X}_q$ and $\mathbf{Y}_1, \dots, \mathbf{Y}_q$ from p, we use them to obtain $2q$ samples $\mathbf{X}_1^{(r)}, \dots, \mathbf{X}_q^{(r)}$ and $\mathbf{Y}_1^{(r)}, \dots, \mathbf{Y}_q^{(r)}$ from $\odot^r(p)$, by setting

$$\mathbf{X}_i^{(r)} = \mathbf{X}_i^{\otimes 2^r} \in \{-1,1\}^{n^{2^r}}.$$

We can then similarly form their averages $\overline{\mathbf{X}}^{(r)}, \overline{\mathbf{Y}}^{(r)}$ and set $\mathbf{Z}^{(r)} = \langle \overline{\mathbf{X}}^{(r)}, \overline{\mathbf{Y}}^{(r)} \rangle$.

We record the following corollary from the above proposition:

```
Algorithm 1: Robust Junta Mean Tester
input : Sample access to distribution $p$ over $\{-1,1\}^n$ and a distance parameter $\epsilon \in (0,1)$
Set $r_0 = \lceil \log\log n \rceil$.
Draw a sequence of $2q$ samples $\mathbf{S} = (\mathbf{X}_1, \dots, \mathbf{X}_q, \mathbf{Y}_1, \dots, \mathbf{Y}_q)$ from $p$ independently
for $r = 0, 1, 2, \dots r_0$ do
    Using samples from $\mathbf{S}$ to compute $\overline{\mathbf{X}}^{(r)}$, $\overline{\mathbf{Y}}^{(r)}$ and $\mathbf{Z}^{(r)}$
    if $\mathbf{Z}^{(r)} > \tau_r$ then
        output: Not a $k$-Junta
    end
end
if All $r_0$ tests pass then
    output: Is a $k$-Junta
end
```

Figure 5: Robust Junta Mean Tester

Corollary 38 *Let p be a distribution over $\{-1,1\}^n$ and $r \geq 0$. Then we have*

$$\mathbf{E}\left[\mathbf{Z}^{(r)}\right] = \left\|\mu(\odot^r(p))\right\|_2^2$$

$$\mathbf{Var}\left[\mathbf{Z}^{(r)}\right] \leq \frac{1}{q^2} \cdot \left\|\Sigma(\odot^r(p))\right\|_F^2 + \frac{4}{q} \cdot \left\|\mu(\odot^r(p))\right\|_2^2 \cdot \left\|\Sigma(\odot^r(p))\right\|_F.$$

Next, we set

$$q = C \cdot \max\left\{\frac{k+\sqrt{n}}{\epsilon^2 n}, \frac{1+k/\sqrt{n}}{\epsilon}\right\}$$

for some sufficiently large constant $C > 0$, and define a sequence $(\tau_r)_{r\geq 0}$ with $\tau_0 = \epsilon^2 n/2$ and

$$\tau_r = \frac{1}{5000} \cdot q^2 \tau_{r-1}^2 \tag{49}$$

for each $r \geq 1$. Setting $a = 1/5000$, we have the following closed form for τ_r:

$$\tau_r = \frac{1}{aq^2}\left(\frac{aq^2\epsilon^2 n}{2}\right)^{2^r}. \tag{50}$$

Our main algorithm is presented in Figure 5 and we prove Theorem 8 in the rest of the section. We divide the proof of correctness into a soundness and completeness case. The two cases are addressed in Sections E.2 and E.1 respectively, where we prove the following two lemmas:

Lemma 39 (Soundness) *Suppose p is a distribution over $\{-1,1\}^n$ satisfying $\|\mu(p)\|_2 \geq \epsilon\sqrt{n}$. Then there exists an $r \in \{0,1,\dots,r_0\}$ such that*

$$\mathbf{Pr}\left[\mathbf{Z}^{(r)} > \tau_r\right] \geq \frac{2}{3}.$$

Lemma 40 (Completeness) *Suppose p is a k-junta distribution over $\{-1,1\}^n$ with $\|\mu(p)\|_2 \le \epsilon\sqrt{n}/100$. Then for every $r \in \{0,1,\dots,r_0\}$, we have*

$$\mathbf{Pr}\Big[\mathbf{Z}^{(r)} > \tau_r\Big] \le \frac{1}{25}\cdot\left(\frac{1}{2}\right)^{2^r-1}.$$

Proof of Theorem 8: The soundness case follows directly from Lemma 39. For completeness, we can apply a union bound over all $r \in \{0,1,\dots,r_0\}$, giving

$$\mathbf{Pr}\Big[\mathbf{Z}^{(r)} > \tau_r \text{ for some } r \in \{0,1,\dots,r_0\}\Big] \le \sum_{r\ge 0}\frac{1}{25}\cdot\left(\frac{1}{2}\right)^{2^r-1} < 1/3. \tag{51}$$

The sample complexity of the algorithm follows directly from our choice of q in (2). Finally, we demonstrate that $\mathbf{Z}^{(r)}$ from the r-th order tensor distribution can be computed in polynomial time in n and q — much faster than the naive $O(n^{2^r})$ time required to compute samples $\mathbf{X}_i^{(r)}$ from $\odot^r(p)$ using samples $\mathbf{X}_i$ from p. To do this, we will use the following *mixed-product* property of tensor products.

Fact 41 (Van Loan (2000)) *If A, B, C, D are matrices with such that the products AC and BD are well-defined, then we have $(A\otimes B)(C\otimes D) = (AC\otimes BD)$.*

Let $X_1,\dots,X_q,Y_1,\dots,Y_q$ be strings in $\{-1,1\}^n$. Then our target $Z^{(r)}$ can be written as

$$\begin{aligned}
Z^{(r)} &= \frac{1}{q^2}\left\langle \sum_{i=1}^{q} X_i^{\otimes 2^r}, \sum_{i=1}^{q} Y_i^{\otimes 2^r}\right\rangle \\
&= \frac{1}{q^2}\sum_{1\le i,j\le q}\left(X_i^{\otimes 2^r}\right)^T Y_j^{\otimes 2^r} \\
&= \frac{1}{q^2}\sum_{1\le i,j\le q}\left(X_i^T\otimes X_i^T\otimes\cdots\otimes X_i^T\right)\left(Y_j\otimes Y_j\otimes\cdots\otimes Y_j\right) \qquad (52)\\
&= \frac{1}{q^2}\sum_{1\le i,j\le q}\left(X_i^TY_j\otimes X_i^TY_j\otimes\cdots\otimes X_i^TY_j\right) \\
&= \frac{1}{q^2}\sum_{1\le i,j\le q}\langle X_i, Y_j\rangle^{2^r}.
\end{aligned}$$

To compute $Z^{(r)}$ for each $r = 0,1,\dots,r_0$, we can first construct the $q\times q$ matrix M with $M_{i,j} = \langle X_i, Y_j\rangle$ in time $O(q^2n)$. Then each $Z^{(r)}$ is just the average of 2^r-th power of entries of M, namely $Z^{(r)} = (1/q^2)\cdot\sum_{i,j} M_{i,j}^{2^r}$. The time needed to compute $Z^{(r)}$ from M for $r = 0,1,\dots,r_0$ is $o(q^2n)$ (recall that $r = \lceil\log\log n\rceil$). This completes the analysis of running time of our algorithm. ■

E.1. Soundness: Proof of Lemma 39

We first prove the following lemma, which we will iteratively apply in the soundness case.

Lemma 42 *Let p be a distribution supported on $\{-1,1\}^n$ and $r \geq 0$. Suppose that*

$$\|\mu(\odot^r(p))\|_2^2 \geq 2\tau$$

for some $\tau > 0$ and $\mathbf{Pr}[\mathbf{Z}^{(r)} \leq \tau] \geq 1/3$. Then we have $\|\mu(\odot^{r+1}(p))\|_2^2 \geq (\tau q/24)^2$.

Proof: By Proposition 37, we have $\mathbf{E}\left[\mathbf{Z}^{(r)}\right] = \|\mu(\odot^r(p))\|_2^2 \geq 2\tau$. Thus

$$\begin{aligned}\frac{1}{3} &\leq \mathbf{Pr}\left[\mathbf{Z}^{(r)} \leq \tau\right] \leq \mathbf{Pr}\left[\left|\mathbf{Z}^{(r)} - \mathbf{E}\left[\mathbf{Z}^{(r)}\right]\right| \geq \frac{\mathbf{E}\left[Z^{(r)}\right]}{2}\right] \\ &\leq \frac{4}{\|\mu(\odot^r(p))\|_2^4}\left(\frac{1}{q^2}\cdot\|\mu(\odot^{r+1}(p))\|_2^2 + \frac{4}{q}\cdot\|\mu(\odot^r(p))\|_2^2\cdot\|\mu(\odot^{r+1}(p))\|_2\right),\end{aligned} \tag{53}$$

where in the last inequality we applied Chebyshev's inequality. It follows that at least one of the two terms on the last line of equation (53) must be greater than $1/6$. Thus $\|\mu(\odot^{r+1}(p))\|_2^2 \geq \tau^2q^2/3$ or $\|\mu(\odot^{r+1}(p))\|_2 \geq \tau q/24$, from which the lemmas follows. ■

Proof of Lemma 39: Assume for the sake of contradiction that $\mathbf{Pr}[\mathbf{Z}^{(r)} \leq \tau_r] \geq 1/3$ for every $r = 0, 1, \ldots, r_0$. We apply Lemma 42 to prove by induction on r that $\|\mu(\odot^r(p))\|_2^2 \geq 2\tau_r$ for every $r = 0, 1, 2, \ldots, r_0 + 1$. The base case of $r = 0$ follows from the choice of $\tau_0 = \epsilon^2 n/2$ and the assumption that $\|\mu(\odot^0(p))\|_2 = \|\mu(p)\|_2 \geq \epsilon\sqrt{n}$. For the induction step, we have by the inductive hypothesis that $\|\mu(\odot^r(p))\|_2^2 \geq 2\tau_r$ for some $r \leq r_0$. It follows from Lemma 42 and $\mathbf{Pr}[\mathbf{Z}^{(r)} \leq \tau_r] \geq 1/3$ that

$$\left\|\mu(\odot^{r+1}(p))\right\|_2^2 \geq \left(\frac{\tau_r q}{24}\right)^2 \geq \frac{1}{2500}\cdot q^2\tau_r^2 = 2\tau_{r+1}.$$

Now to get a contradiction, we note that

$$\left\|\mu(\odot^{r_0+1}(p))\right\|_2^2 \geq \frac{2}{aq^2}\left(\frac{aq^2\epsilon^2 n}{2}\right)^{2^{r_0+1}} = q^{2^{r_0+2}-2}\cdot\left(\epsilon\sqrt{n}\right)^{2^{r_0+2}}\cdot\left(\frac{a}{2}\right)^{2^{r_0+1}-1}.$$

Given that $q \geq C/\epsilon$ and $q \geq C/(\epsilon^2\sqrt{n})$ in (2), we have

$$q^{2^{r_0+2}-2} \geq \left(\frac{C}{\epsilon}\right)^{2^{r_0+2}-4}\cdot\left(\frac{C}{\epsilon^2\sqrt{n}}\right)^2 = \left(\frac{1}{\epsilon}\right)^{2^{r_0+2}}\cdot\frac{1}{n}\cdot C^{2^{r_0+2}-2}$$

and thus,

$$\|\mu(\odot^{r_0+1}(p))\|_2^2 \geq n^{2^{r_0+1}}\cdot\frac{1}{n}\cdot C^{2^{r_0+2}-2}\cdot\left(\frac{a}{2}\right)^{2^{r_0+1}-1},$$

which, after setting C to be a large enough constant and recalling that $r_0 = \lceil\log\log n\rceil$, contradicts the fact that we always have $\|\mu(\odot^{r_0+1}(p))\|_2^2 \leq n^{2^{r_0+1}}$. This completes the proof of the lemma. ■

E.2. Completeness: Proof of Lemma 40

We will now need the following bound on the mean vector in the completeness case.

Proposition 43 *Suppose p is a k-junta distribution over $\{-1,1\}^n$. Then for each $r \geq 1$ we have*

$$\left\|\mu(\odot^r(p))\right\|_2^2 \leq \left(2 \cdot \max\{n, k^2\} \cdot 2^r\right)^{2^{r-1}}.$$

Proof: For $r = 0$, the result holds because $\mu(p)$ is k-sparse when p is a k-junta distribution.

Next consider the case when $r > 0$. Let $R = 2^r$ and let $S \subseteq [n]$ be the set of influential variables with $|S| = k$. (Note that if the number of influential variables is smaller than k we can always add more variables to S to make it size k.) Without loss of generality we assume $S = [k]$ and by the definition of k-junta distributions, there is a distribution p' over $\{-1,1\}^k$ such that $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \sim p$ can be drawn by first drawing $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \sim p'$ and then drawing each $\boldsymbol{x}_i$, $i > k$, independently and uniformly at random from $\{-1,1\}$.

Now we consider the mean vector $\mu(\odot^r(p))$. Note that it has n^R entries and each entry is indexed by an R-tuple $I = (i_1, \ldots, i_R) \in [n]^R$: the entry indexed by I is given by

$$\mathbf{E}_{\boldsymbol{x}\sim p}\left[\boldsymbol{x}_{i_1} \cdots \boldsymbol{x}_{i_R}\right].$$

We define $Q \subseteq [n]^R$ as the set of all R-tuples $I = (i_1, \ldots, i_R) \in [n]^R$ such that every $j \notin S$ appears an even number of times in I. Given that every $\boldsymbol{x}_j$, $j \notin S$, is drawn independently from other variables and is uniform over $\{-1,1\}$, we have that entries of $\mu(\odot^r(p))$ are zero outside of those indexed by tuples in Q. On the other hand, every nonzero entry of $\mu(\odot^r(p))$ trivially has magnitude no larger than 1. As a result, $\|\mu(\odot^r(p))\|_2^2 \leq |Q|$ and we bound $|Q|$ in the rest of the proof.

To this end, let $Q_i \subseteq Q$ be the set of $I = (i_1, \ldots, i_R) \in Q$ such that $\{\ell \in [R] \mid i_\ell \notin S\}| = i$. Then

$$|Q_i| \leq \binom{R}{i} \cdot k^{R-i} \cdot L_i,$$

where L_i is the number of ordered i-tuples, each entry selected from $[n]$ (note that we relaxed it from $[n] \setminus S$ to $[n]$ to simplify the presentation since this can only make L_i bigger), in which every $j \in [n]$ appears an even number of times. Note that L_i is trivially 0 when i is odd. We can bound L_i by noting that to pick a tuple $(i_1, \ldots, i_R) \in Q_j$, we can first pick $i_1 \in [n]$, and then pick an index i_j for some $j > 1$ and set $i_j = i_1$. Next, we pick $i_2 \in [n]$ (or i_3 if i_2 was chosen to be i_j in the first round) and then pick an unused index $i_{j'}$ for some $j' > 2$ and set $i_{j'} = i_2$, and so on. Thus,

$$L_i \leq n(i-1) \cdot n(i-3) \cdots n = \left(\frac{n}{2}\right)^{i/2} \cdot \frac{i!}{(i/2)!} \leq \left(\frac{n}{2}\right)^{i/2} \cdot i^{i/2}$$

when i is even. Using that $|Q_0| = k^R$, we have

$$|Q| \leq \sum_{\ell=0}^{R/2} |Q_{2\ell}| \leq k^R + \sum_{\ell=1}^{R/2} \binom{R}{2\ell} \cdot (n\ell)^\ell \cdot k^{R-2\ell}.$$

Letting $\alpha = \max\{k^2, n\}$ so that $k \leq \sqrt{\alpha}$ and $n \leq \alpha$, we have

$$|Q| \leq \alpha^{R/2} + \sum_{\ell=1}^{R/2} \binom{R}{2\ell} \cdot \ell^\ell \cdot \alpha^{R/2} \leq \alpha^{R/2}\left(1 + (R/2)^{R/2} \cdot \sum_{\ell=1}^{R/2} \binom{R}{2\ell}\right) \leq \alpha^{R/2} \cdot (R/2)^{R/2} \cdot 2^R,$$

which completes the proof. ■

We now start the proof of Lemma 40.

Proof of Lemma 40: Again, we set $R = 2^r$. We show for each $r \in \{0, 1, \ldots, r_0\}$ that

$$\mathbf{E}\left[\mathbf{Z}^{(r)}\right] = \left\|\mu(\odot^r(p))\right\|_2^2 \le \frac{1}{100}\left(\frac{1}{2}\right)^{R-1} \cdot \tau_r \qquad \text{and} \qquad \mathbf{Var}\left[\mathbf{Z}^{(r)}\right] \le \frac{1}{100}\left(\frac{1}{2}\right)^{R-1} \tau_r^2. \tag{54}$$

Assuming this, by Chebyshev's inequality we have

$$\mathbf{Pr}\left[\mathbf{Z}^{(r)} > \tau_r\right] \le \mathbf{Pr}\left[\left|\mathbf{Z}^{(r)} - \mathbf{E}\left[\mathbf{Z}^{(r)}\right]\right| > \tau_r/2\right] \le 4 \cdot \frac{\mathbf{Var}[\mathbf{Z}^{(r)}]}{\tau_r^2} \le \frac{1}{25} \cdot \left(\frac{1}{2}\right)^{R-1} \tag{55}$$

and this finishes the proof of the lemma.

We start with the case when $r = 0$. The first part of (54) follows trivially from the assumption that $\|\mu(p)\|_2 \le \epsilon\sqrt{n}/100$, and the second part follows from Lemma 43. To see the latter, we have from Claim 36 and Lemma 43 that

$$\mathbf{Var}\left[\mathbf{Z}^{(0)}\right] \le \frac{1}{q^2} \cdot \left(4 \cdot \max(n, k^2)\right) + \frac{4}{q} \cdot \frac{\epsilon^2 n}{10000} \cdot \sqrt{(4 \cdot \max(n, k^2))} \le \frac{1}{100}\left(\frac{1}{2}\right)^{R-1} \cdot \tau_1^2,$$

where the last inequality used the choice of τ_1, $\epsilon \le 1$, and $q \ge C(k + \sqrt{n})/(\epsilon^2 n)$ for some sufficiently large constant C.

Moving to the general case when $r \ge 1$, we have $R = 2^r \ge 2$. Letting $\beta = \max(n, k^2)$ and using $q \ge C\sqrt{\beta}/(\epsilon^2 n)$ and $q \ge C\sqrt{\beta}/(\epsilon\sqrt{n})$, we have

$$q^{2R-2} = q^{2R-4} \cdot q^2 \ge \left(\frac{C\sqrt{\beta}}{\epsilon\sqrt{n}}\right)^{2R-4} \cdot \left(\frac{C\sqrt{\beta}}{\epsilon^2 n}\right)^2 = \left(C^2\beta\right)^{R-1} \cdot \left(\frac{1}{\epsilon^2 n}\right)^R.$$

Plugging this in the closed form (50) of τ_r, we have

$$\tau_r = \frac{1}{aq^2}\left(\frac{aq^2\epsilon^2 n}{2}\right)^R \ge \frac{1}{2} \cdot \left(\frac{aC^2\beta}{2}\right)^{R-1}.$$

Using Proposition 43, we have $\mathbf{E}\left[\mathbf{Z}^{(r)}\right] \le (2R\beta)^{R/2}$ and thus,

$$\frac{\mathbf{E}[\mathbf{Z}^{(r)}]}{\tau_r} \le \left(2R \cdot \left(\frac{2}{aC^2}\right)^{R-1} \cdot 2^{R/2}\right) \cdot \left(\frac{R}{\beta}\right)^{R/2-1}.$$

Note that $r \le r_0 = \lceil \log\log n \rceil$ and thus $R/\beta < 1$ when n is sufficiently large. As a result we have

$$\frac{\mathbf{E}[\mathbf{Z}^{(r)}]}{\tau_r} \le 2R \cdot \left(\frac{2}{aC^2}\right)^{R-1} \cdot 2^{R/2} \le 2R \cdot \left(\frac{4}{aC^2}\right)^{R-1} \le \frac{1}{100}\left(\frac{1}{2}\right)^{R-1},$$

when C is sufficiently large. This completes the proof of the first part of (54). For the second part, by Corollary 37 and using the first part of (54) and the recursive definition of τ_r in (49), we have

$$\begin{aligned}\mathbf{Var}\left[\mathbf{Z}^{(r)}\right] &\le \frac{1}{q^2} \cdot \left\|\mu(\odot^{r+1}(p))\right\|_2^2 + \frac{4}{q} \cdot \left\|\mu(\odot^r(p))\right\|_2^2 \cdot \left\|\mu(\odot^{r+1}(p))\right\|_2 \\ &\le \frac{1}{100 \cdot q^2 \cdot 2^{2R-1}} \cdot \tau_{r+1} + \frac{1}{250 \cdot q \cdot 2^{R-1}} \cdot \tau_r \cdot \sqrt{\tau_{r+1}} \\ &= \frac{1}{100 \cdot q^2 \cdot 2^{2R-1}} \cdot \left(\frac{q^2\tau_r^2}{5000}\right) + \frac{1}{250 \cdot q \cdot 2^{R-1}} \cdot \tau_r \cdot \sqrt{\frac{q^2\tau_r^2}{5000}} < \frac{1}{100}\left(\frac{1}{2}\right)^{R-1} \cdot \tau_r^2.\end{aligned}$$

This finishes the proof of the lemma. ■

Appendix F. Proof of the Main Structural Lemma: Lemma 7

In this section, we prove the main structural lemma. The goal is to relate the distance in total variation from a distribution which is far from being a k-junta to the expected Euclidean distance of its mean vector after applying random restrictions.

The proof of Lemma 7 uses the following results from Canonne et al. (2019), which we reproduce below.

Lemma 44 (Lemma 1.4 in Canonne et al. (2019)) *Let p be a distribution over $\{-1,1\}^n$. For any $\sigma \in (0,1)$,*

$$d_{\mathrm{TV}}(p,\mathcal{U}) \le \mathop{\mathbf{E}}_{\mathbf{S}\sim\mathcal{S}_\sigma}\left[d_{\mathrm{TV}}(p_{\overline{\mathbf{S}}},\mathcal{U})\right] + \mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_\sigma(p)}\left[d_{\mathrm{TV}}(p_{|\boldsymbol{\rho}},\mathcal{U})\right].$$

Lemma 45 (Implicit in Canonne et al. (2019)) *Let p be a distribution over $\{-1,1\}^n$. Then we have*

$$\frac{d_{\mathrm{TV}}(p,\mathcal{U})}{n\log n} \lesssim \mathop{\mathbf{E}}_{\substack{\mathbf{i}\sim[n]\\ \boldsymbol{\rho}\sim\mathcal{D}_{\{\mathbf{i}\}}(p)}}\left[\left\|\mu(p_{|\boldsymbol{\rho}})\right\|_2\right].$$

Proof: We follow Subsection 1.1.2 in Canonne et al. (2019). Let $f\colon \{-1,1\}^n \to [-1,\infty)$ be

$$f(x) = 2^n \cdot p(x) - 1.$$

Then by the first part of (4) in Canonne et al. (2019) (scaled by $1/n$), we have

$$\begin{aligned}
\frac{d_{\mathrm{TV}}(p,\mathcal{U})}{n\log n} &\lesssim \frac{1}{n}\cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}\left[\sqrt{\sum_{i=1}^n \left(\left(f(\boldsymbol{x})-f(\boldsymbol{x}^{(i)})\right)^+\right)^2}\,\right]\\
&= \frac{1}{n}\cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim p}\left[\sqrt{\sum_{i=1}^n \left(\frac{\left(f(\boldsymbol{x})-f(\boldsymbol{x}^{(i)})\right)^+}{f(\boldsymbol{x})+1}\right)^2}\,\right]\\
&\le \frac{1}{n}\cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim p}\left[\sum_{i=1}^n \left|\frac{\left(f(\boldsymbol{x})-f(\boldsymbol{x}^{(i)})\right)^+}{f(\boldsymbol{x})+1}\right|\right]\\
&\le \frac{2}{n}\cdot \sum_{i=1}^n \mathop{\mathbf{E}}_{\boldsymbol{x}\sim p}\left[\left|\frac{p(\boldsymbol{x})-p(\boldsymbol{x}^{(i)})}{p(\boldsymbol{x})+p(\boldsymbol{x}^{(i)})}\right|\right] = 2\mathop{\mathbf{E}}_{\substack{\mathbf{i}\sim[n]\\ \boldsymbol{\rho}\sim\mathcal{D}_{\{\mathbf{i}\}}(p)}}\left[\left|\mu(p_{|\boldsymbol{\rho}})_{\mathbf{i}}\right|\right],
\end{aligned}$$

where the first inequality uses a robust version of Pisier's inequality on f (see Theorem 1.7 and (3) in Canonne et al. (2019)); the next equation follows from importance sampling; the third inequality uses Jensen's inequality. Finally we note that since $p_{|\boldsymbol{\rho}}$ is supported on a single bit, the absolute value is the same as the Euclidean norm. ■

We point out that the two lemmas above hold even when n is a small constant. The next theorem from Canonne et al. (2019) holds only when n is sufficiently large.

Theorem 46 (Theorem 1.5 in Canonne et al. (2019)) *Let p be a distribution over $\{-1,1\}^n$. For any $\sigma \in (0,1)$,*

$$\mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_\sigma(p)}\Big[\big\|\mu(p_{|\boldsymbol{\rho}})\big\|_2\Big] \geq \frac{\sigma}{\mathrm{poly}(\log n)} \cdot \tilde{\Omega}\left(\mathop{\mathbf{E}}_{\mathbf{S}\sim\mathcal{S}_\sigma}\left[d_{\mathrm{TV}}(p_{\overline{\mathbf{S}}},\mathcal{U})\right] - 2e^{-\min(\sigma,1-\sigma)n/10}\right). \tag{56}$$

We are now ready to prove Lemma 7.

Proof of Lemma 7: Let q be the junta distribution on J such that its projection q_J is the same as p_J (equivalently, one can draw $\boldsymbol{x} \sim q$ by first drawing a string from $\{0,1\}^J$ from p_J and then drawing every other bit independently and uniformly at random). Given our assumption that p is ϵ-far from every junta distribution over J, we have

$$\epsilon \leq d_{\mathrm{TV}}(p,q) = \mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\Big[d_{\mathrm{TV}}\big(p_{|\boldsymbol{\rho}}, q_{|\boldsymbol{\rho}}\big)\Big] = \mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\Big[d_{\mathrm{TV}}\big(p_{|\boldsymbol{\rho}},\mathcal{U}\big)\Big]. \tag{57}$$

In the rest of the proof we consider a restriction $\rho \in \{-1,1,*\}^n$ with $\mathrm{stars}(\rho) = \overline{J}$ and lowerbound $d_{\mathrm{TV}}(p_{|\rho},\mathcal{U})$. For simplicity of notation, we let $g = p_{|\rho}$ be the distribution supported over $\{-1,1\}^{\overline{J}}$. The goal is to obtain a lower bound for $d_{\mathrm{TV}}(g,\mathcal{U})$ in terms of mean vectors of random restrictions of g, which is then plugged into (57) to finish the proof of Lemma 7.

Let $m = |\overline{J}|$. We start with the case when m satisfies $m \leq C \cdot \log(m/\epsilon)$ for some constant $C > 0$. We apply Lemma 45 on g (with the parameter n set to m). There is a constant $\widehat{c}$ such that

$$d_{\mathrm{TV}}(g,\mathcal{U}) \leq \widehat{c}\log^2(m/\epsilon) \cdot \mathop{\mathbf{E}}_{\substack{\mathbf{i}\sim[n]\\ \boldsymbol{\nu}\sim\mathcal{D}_{\{\mathbf{i}\}}(p)}}\Big[\big\|\mu(g_{|\boldsymbol{\nu}})\big\|_2\Big].$$

Letting $j = \lceil \log_2 2m \rceil$, the probability of $\boldsymbol{\rho} \sim \mathcal{D}_{\sigma^j}(g)$ having exactly one $*$ is at least

$$m \cdot \sigma^j \cdot (1-\sigma^j)^{m-1} \geq m \cdot \frac{1}{4m} \cdot \left(1 - \frac{1}{2m}\right)^{m-1} \geq \frac{1}{8},$$

and when this happens, the $*$ is distributed uniformly at random. As a result, we have

$$d_{\mathrm{TV}}(g,\mathcal{U}) \leq \widehat{c}\log^2(m/\epsilon) \cdot \mathop{\mathbf{E}}_{\substack{\mathbf{i}\sim[n]\\ \boldsymbol{\nu}\sim\mathcal{D}_{\{\mathbf{i}\}}(p)}}\Big[\big\|\mu(g_{|\boldsymbol{\nu}})\big\|_2\Big] \leq 8\widehat{c}\log^2(m/\epsilon) \cdot \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_\sigma(g)}\Big[\big\|\mu(g_{|\boldsymbol{\nu}})\big\|_2\Big] \tag{58}$$

The lemma then follows by combining (57) and (58). We now turn to the case when

$$|\overline{J}| = m \geq C \cdot \log(m/\epsilon) \tag{59}$$

for some sufficiently large constant $C > 0$. We first prove by induction that for any $t \in \mathbb{N}$,

$$d_{\mathrm{TV}}(g,\mathcal{U}) \leq \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^t}(g)}\Big[d_{\mathrm{TV}}\big(g_{|\boldsymbol{\nu}},\mathcal{U}\big)\Big] + \sum_{j=1}^{t}\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j-1}}(g)}\left[\mathop{\mathbf{E}}_{\mathbf{S}\sim\mathcal{S}_\sigma(\mathrm{stars}(\boldsymbol{\nu}))}\Big[d_{\mathrm{TV}}\big((g_{|\boldsymbol{\nu}})_{\overline{\mathbf{S}}},\mathcal{U}\big)\Big]\right]. \tag{60}$$

Lemma 44 provides the base case when $t = 1$, as a draw from the distribution $\mathcal{D}_1(g)$ always outputs the all-$*$ restriction $(*, *, \ldots, *)$. For the induction step with $t > 1$, notice that

$$d_{\mathrm{TV}}(g, \mathcal{U}) \leq \mathop{\mathbf{E}}_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^{t-1}}(g)} \Big[d_{\mathrm{TV}}\big(g_{|\boldsymbol{\nu}}, \mathcal{U}\big) \Big] + \sum_{j=1}^{t-1} \mathop{\mathbf{E}}_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^{j-1}}(g)} \left[\mathop{\mathbf{E}}_{\mathbf{S} \sim \mathcal{S}_\sigma(\mathrm{stars}(\boldsymbol{\nu}))} \Big[d_{\mathrm{TV}}\big((g_{\boldsymbol{\nu}})_{\overline{\mathbf{S}}}, \mathcal{U}\big) \Big] \right] \tag{61}$$

$$\leq \mathop{\mathbf{E}}_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^{t-1}(g)}} \left[\mathop{\mathbf{E}}_{\mathbf{S} \sim \mathcal{S}_\sigma(\mathrm{stars}(\boldsymbol{\nu}))} \Big[d_{\mathrm{TV}}\big((g_{|\boldsymbol{\nu}})_{\overline{\mathbf{S}}}, \mathcal{U}\big) \Big] + \mathop{\mathbf{E}}_{\boldsymbol{\nu}' \sim \mathcal{D}_\sigma(g_{|\boldsymbol{\nu}})} \Big[d_{\mathrm{TV}}\big((g_{|\boldsymbol{\nu}})_{|\boldsymbol{\nu}'}, \mathcal{U}\big) \Big] \right] \tag{62}$$

$$+ \sum_{j=1}^{t-1} \mathop{\mathbf{E}}_{\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^{j-1}}(g)} \left[\mathop{\mathbf{E}}_{\mathbf{S} \sim \mathcal{S}_\sigma(\mathrm{stars}(\boldsymbol{\nu}))} \Big[d_{\mathrm{TV}}\big((g_{\boldsymbol{\nu}})_{\overline{\mathbf{S}}}, \mathcal{U}\big) \Big] \right],$$

where we first applied the inductive hypothesis in (61) and then Lemma 44 to the distribution $g_{|\boldsymbol{\nu}}$ supported on $\{-1, 1\}^{\mathrm{stars}(\boldsymbol{\nu})}$ in (62). We get (60) by noticing that the distribution over distributions $(g_{|\boldsymbol{\nu}})_{|\boldsymbol{\nu}'}$ where $\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^{t-1}}(g)$ and $\boldsymbol{\nu}' \sim \mathcal{D}_\sigma(g_{|\boldsymbol{\nu}})$ is equivalent to $g_{|\boldsymbol{\nu}}$ with $\boldsymbol{\nu} \sim \mathcal{D}_{\sigma^t}(g)$.

Next for each restriction $\nu \in \{-1, 1, *\}^n$ we let

$$\alpha(\nu) = \mathop{\mathbf{E}}_{\mathbf{S} \sim \mathcal{S}_\sigma(\mathrm{stars}(\nu))} \Big[d_{\mathrm{TV}}\big((g_{|\nu})_{\overline{\mathbf{S}}}, \mathcal{U}\big) \Big],$$

and let $G_t \subset \{-1, 1, *\}^n$ for each $t \in \mathbb{N}$ be the set of restrictions $\nu \in \{-1, 1, *\}^n$ that satisfy

$$\alpha(\nu) \geq \max\left\{ \frac{\epsilon}{6t}, 4e^{-|\mathrm{stars}(\nu)|/20} \right\}.$$

For each restriction $\nu \notin G_t$ we trivially have

$$\alpha(\nu) \leq \frac{\epsilon}{6t} + 4e^{-|\mathrm{stars}(\nu)|/20}.$$

For each $\nu \in G_t$ we have

$$\alpha(\nu) - 2e^{-|\mathrm{stars}(\nu)|/20} \geq \alpha(v)/2 \geq \epsilon/(12t).$$

We can then apply Theorem 46 to get

$$\alpha(\nu) \leq \big(c_0 \cdot \big(\log n \cdot \log(12t/\epsilon)\big)^{c_1}\big) \cdot \mathop{\mathbf{E}}_{\boldsymbol{\nu}' \sim \mathcal{D}_\sigma(g_{|\nu})} \Big[\big\| \mu\big((g_{|\nu})_{|\boldsymbol{\nu}'}\big) \big\|_2 \Big]$$

for some universal constants c_0 and c_1. Therefore, we have for every $\nu \in \{-1, 1, *\}^n$ that

$$\alpha(\nu) \leq \big(c_0 \cdot \big(\log n \cdot \log(12t/\epsilon)\big)^{c_1}\big) \cdot \mathop{\mathbf{E}}_{\boldsymbol{\nu}' \sim \mathcal{D}_\sigma(g_{|\nu})} \Big[\big\| \mu\big((g_{|\nu})_{|\boldsymbol{\nu}'}\big) \big\|_2 \Big] + \frac{\epsilon}{6t} + 4e^{-|\mathrm{stars}(\nu)|/20}.$$

Combining this bound with (60), we get

$$d_{\mathrm{TV}}(g,\mathcal{U}) \le \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^t}(g)}\Big[d_{\mathrm{TV}}\big(g_{|\boldsymbol{\nu}},\mathcal{U}\big)\Big] \tag{63}$$

$$+\left(c_0\cdot\big(\log n\cdot\log(12t/\epsilon)\big)^{c_1}\right)\cdot\sum_{j=1}^{t}\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j-1}}(g)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}'\sim\mathcal{D}_{\sigma}(g_{|\boldsymbol{\nu}})}\Big[\big\|\mu\big((g_{|\boldsymbol{\nu}})_{|\boldsymbol{\nu}'}\big)\big\|_2\Big]\right] \tag{64}$$

$$+\frac{\epsilon}{6}+4\sum_{j=1}^{t}\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j-1}}(g)}\Big[e^{-|\mathrm{stars}(\boldsymbol{\nu})|/20}\Big]. \tag{65}$$

Setting (where C is the constant from (59))

$$t=\left\lfloor\log\left(\frac{m}{C\cdot\log(m/\epsilon)}\right)\right\rfloor+1 \tag{66}$$

in the rest of the proof. We upper bound the right-hand side of (65) by noting that $|\mathrm{stars}(\boldsymbol{\nu})|$, when $\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j-1}}(g)$ is a sum of n independent random variables, where each is set to 1 with probability σ^{j-1}. Thus, we have

$$\sum_{j=1}^{t}\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j-1}}(g)}\Big[e^{-|\mathrm{stars}(\boldsymbol{\nu})|/20}\Big]=\sum_{j=1}^{t}\left(\mathop{\mathbf{E}}_{\mathbf{X}\sim\mathrm{Ber}(\sigma^{j-1})}\Big[e^{-\mathbf{X}/20}\Big]\right)^m=\sum_{j=1}^{t}\left(1-\sigma^{j-1}\left(1-e^{-1/20}\right)\right)^m$$

$$\le\sum_{j=1}^{t}\left(1-\frac{\sigma^{j-1}}{100}\right)^m\le t\cdot\exp\left(-\frac{\sigma^{t-1}m}{100}\right)\le\frac{\epsilon}{24},$$

using our choice of t with $\sigma^{t-1}m\ge C\cdot\log(m/\epsilon)$ and a sufficiently large constant C. Therefore, the right-hand side of (65) can be bounded from above by $\epsilon/3$.

Next we upperbound (64). Using again the fact that $(g_{|\boldsymbol{\nu}})_{|\boldsymbol{\nu}'}$ with $\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^{j-1}}(g)$ and $\boldsymbol{\nu}'\sim\mathcal{D}_{\sigma}(g_{|\boldsymbol{\nu}})$ is distributed as $g_{|\boldsymbol{\nu}}$ with $\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(g)$, the right-hand side of (64) may be upper bounded by

$$\left(c_0\cdot\big(\log n\cdot\log(12t/\epsilon)\big)^{c_1}\right)\cdot\sum_{j=1}^{t}\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(g)}\Big[\big\|\mu(g_{|\boldsymbol{\nu}})\big\|_2\Big]. \tag{67}$$

Finally we bound the right-hand side of (63) by considering the set of restrictions $F\subset\{-1,1,*\}^n$ where $\nu\in\{-1,1,*\}^n$ is in F iff $|\mathrm{stars}(\nu)|\le 2C\cdot\log(m/\epsilon)$, and note that by the setting of t,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^t}(g)}\big[\boldsymbol{\nu}\notin F\big]\le\frac{\epsilon}{6}.$$

Using the trivial bound of $d_{\mathrm{TV}}(g_{|\nu},\mathcal{U})\le 1$, we have

$$\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^t}(g)}\Big[d_{\mathrm{TV}}\big(g_{|\boldsymbol{\nu}},\mathcal{U}\big)\Big]\le\frac{\epsilon}{6}+\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^t}(g)}\Big[d_{\mathrm{TV}}\big(g_{|\boldsymbol{\nu}},\mathcal{U}\big)\cdot\mathbf{1}\{\boldsymbol{\nu}\in F\}\Big]$$

We apply Lemma 45 to every $g_{|\boldsymbol{\nu}}$ with $\boldsymbol{\nu} \in F$. So there exists a universal constant c_2 such that

$$\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^t}(g)}\left[d_{\mathrm{TV}}\big(g_{|\boldsymbol{\nu}},\mathcal{U}\big)\cdot\mathbf{1}\left\{\boldsymbol{\nu}\in F\right\}\right] \le c_2\cdot\log^2(m/\epsilon)\cdot \mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^t}(g)}\left[\mathop{\mathbf{E}}_{\substack{\mathbf{i}\sim\mathrm{stars}(\boldsymbol{\nu})\\ \boldsymbol{\nu}'\sim\mathcal{D}_{\{\mathbf{i}\}}(g_{|\boldsymbol{\nu}})}}\left[\left\|\mu\big((g_{|\boldsymbol{\nu}})_{|\boldsymbol{\nu}'}\big)\right\|_2\right]\right].$$

Note that the distribution on $(g_{|\boldsymbol{\nu}})_{|\boldsymbol{\nu}'}$ is equivalent to the distribution $g_{|\boldsymbol{\nu}}$ which draws $\mathbf{i}\sim[n]$ and then sets $\boldsymbol{\nu}\sim\mathcal{D}_{\{\mathbf{i}\}}(g)$. Hence, we can upperbound (63) by

$$\frac{\epsilon}{6}+c_2\cdot\log^2(m/\epsilon)\cdot\mathop{\mathbf{E}}_{\substack{\mathbf{i}\sim[n]\\ \boldsymbol{\nu}\sim\mathcal{D}_{\{\mathbf{i}\}}(g)}}\left[\left\|\mu\big(g_{|\boldsymbol{\nu}}\big)\right\|_2\right] \le \frac{\epsilon}{6}+4c_2\cdot\log^2(m/\epsilon)\cdot\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^r}(g)}\left[\left\|\mu\big(g_{|\boldsymbol{\nu}}\big)\right\|_2\right]$$

where $r=\lceil\log_2 m\rceil$. The inequality used the fact that $\nu\sim\mathcal{D}_{\sigma^r}(g)$ has $\mathrm{stars}(\nu)=1$ with probability at least $1/4$ and when this happens, the star is distributed uniformly at random.

Finally, noting that $t<r$, we combine the upper bounds for (63), (64), and (65) to get

$$d_{\mathrm{TV}}(g,\mathcal{U}) \le \frac{\epsilon}{2}+c_3\cdot\log^{c_4}(n/\epsilon)\cdot\sum_{j=1}^{\lceil\log_2 n\rceil}\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(g)}\left[\left\|\mu(g_{|\boldsymbol{\nu}})\right\|_2\right]$$

for some universal constants c_3 and c_4. It follows from (57) that

$$\epsilon \le \frac{\epsilon}{2}+\mathrm{polylog}(n/\epsilon)\cdot\sum_{j=1}^{\lceil\log_2 n\rceil}\mathop{\mathbf{E}}_{\boldsymbol{\rho}\sim\mathcal{D}_{\overline{J}}(p)}\left[\mathop{\mathbf{E}}_{\boldsymbol{\nu}\sim\mathcal{D}_{\sigma^j}(p_{|\boldsymbol{\rho}})}\left[\left\|\mu\big((p_{|\boldsymbol{\rho}})_{|\boldsymbol{\nu}}\big)\right\|_2\right]\right],$$

which completes the proof. ■