List-Decodable Sparse Mean Estimation

Shiwei Zeng
Department of Computer Science
Stevens Institute of Technology
szeng4@stevens.edu

Jie Shen
Department of Computer Science
Stevens Institute of Technology
jie.shen@stevens.edu

Abstract

Robust mean estimation is one of the most important problems in statistics: given a set of samples in $\mathbb{R}^d$ where an α fraction are drawn from some distribution D and the rest are adversarially corrupted, we aim to estimate the mean of D. A surge of recent research interest has been focusing on the list-decodable setting where $\alpha \in (0, \frac{1}{2}]$, and the goal is to output a finite number of estimates among which at least one approximates the target mean. In this paper, we consider that the underlying distribution D is Gaussian with k-sparse mean. Our main contribution is the first polynomial-time algorithm that enjoys sample complexity $O\big(\mathrm{poly}(k, \log d)\big)$, i.e. poly-logarithmic in the dimension. One of our core algorithmic ingredients is using low-degree *sparse polynomials* to filter outliers, which may find more applications.

1 Introduction

Mean estimation is arguably a fundamental inference task in statistics and machine learning. Given a set of samples $\{x_1, \dots, x_n\} \subset \mathbb{R}^d$ where an α fraction are drawn from some well-behaved (e.g. Gaussian) distribution D and the rest are adversarially corrupted, the goal is to estimate the mean of D. In the noiseless case where $\alpha = 1$, the problem can be easily solved in view of the concentration of measure phenomenon [LT91]. However, this is rarely the case as modern data sets are often contaminated by random noise or even by adversarial corruptions. Thus, a great deal of recent efforts are focused on efficiently and robustly estimating the target mean in the presence of outliers.

Generally speaking, there is a phase transition between $\alpha > 1/2$ and $0 < \alpha \leq 1/2$, and solving either problem in a computationally efficient manner is highly nontrivial. The problem that most of the samples are uncorrupted, i.e. $\alpha > 1/2$, has a very long history dating back to the 1960s [Tuk60, Hub64], yet only until recently have computationally efficient algorithms been established [DKK+16, LRV16]. The other yet more challenging regime concerns that an overwhelming fraction of the samples are corrupted, i.e. $\alpha \leq 1/2$, which even renders estimation impossible. This motivates a line of research on *list-decodable* mean estimation [CSV17], where in place of outputting one single estimate, the algorithm is allowed to generate a finite list of candidates and is considered to be successful if there exists at least one candidate in the list that is sufficiently close to the target mean.

In this work, we investigate the problem of list-decodable mean estimation, for which there have been a plethora of elegant results established in recent years. From a high level, most of them concern error guarantees and running time. For example, [CSV17] proposed the first tractable algorithm based on semidefinite programming, which runs in polynomial time and achieves optimal error rate for variance-bounded distributions. [DKS18b] developed a multi-filtering scheme and showed that the error rate can be improved by using high degree polynomials if the underlying distribution is Gaussian. The more recent works [CMY20, DKK+21a] further addressed the computational efficiency of this task and achieved almost linear running time in certain regimes.

Although all of these algorithms exhibit near-optimal guarantees on either error rate or computational complexity, it turns out that less is explored to improve another yet important metric: the sample

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

complexity. In particular, the sample complexity of all these algorithms is $O(\mathrm{poly}(d))$, hence they quickly break down for data-demanding applications such as healthcare where the number of available samples is typically orders of magnitude less than the dimension d [Wai19]. Therefore, a pressing question that needs to be addressed in such a high-dimensional regime is the following:

> Does there exist a provably robust algorithm for list-decodable mean estimation that runs in polynomial time and enjoys a sample complexity bound of $O(\mathrm{polylog}(d))$?

In this paper, we answer the question in the affirmative by showing that when the target mean is k-sparse, i.e. it has at most k non-zero elements, it is *attribute-efficiently* list-decodable.

Theorem 1 (Main result)**.** *Given parameter $\alpha \in (0, \frac{1}{2}]$, failure probability $\tau \in (0,1)$, a natural number $\ell \geq 1$, and a set T of $\Omega\big(\frac{\ell^4 \cdot k^{8\ell}}{\alpha^7} \cdot \log^{6\ell}(\frac{\ell d}{\alpha\tau})\big)$ samples in $\mathbb{R}^d$, of which at least a (2α)-fraction are independent draws from the Gaussian distribution $N(\mu, \mathbb{I}_d)$ where $\|\mu\|_0 \leq k$, there exists an algorithm that runs in time $\mathrm{poly}\left(|T|, d^\ell, \frac{1}{\alpha}\right)$, uses polynomials of degree at most 2ℓ, and returns a list of $O(1/\alpha)$ number of k-sparse vectors such that with probability $1-\tau$, the list contains at least one $\hat{\mu} \in \mathbb{R}^d$ with $\|\hat{\mu} - \mu\|_2 = \tilde{O}\big(\alpha^{-\frac{1}{2\ell}} \cdot \sqrt{\ell}(\ell + \log\frac{1}{\alpha})\big)$, where $\tilde{O}(\cdot)$ hides poly-logarithmic factors.*

Remark 2. The key message of the theorem is that when the true mean is k-sparse, it is possible to efficiently approximate it with $O(\mathrm{polylog}(d))$ samples. This is in stark contrast to existing list-decodable results [CSV17, DKS18b, CMY20, DKK20a, DKK⁺21a] where the sample complexity is $O(\mathrm{poly}(d))$. The only attribute-efficient robust mean estimators are [BDLS17, DKK⁺19, CDK⁺21], but their results hold only for the mild corruption regime where $\alpha > 1/2$.

Remark 3. Our algorithm and analysis hold for any degree $\ell \geq 1$. When $\ell = 1$, the sample complexity reads as $\tilde{O}(\alpha^{-7} k^8 \log^6 d)$ and the algorithm achieves error $\tilde{O}(\alpha^{-\frac{1}{2}})$. As opposed to an $O(1-\alpha)$ error rate obtained for $\alpha > 1/2$, the (non-vanishing) error rate $\tilde{O}(\alpha^{-\frac{1}{2}})$ is typically what one can expect for list-decodable mean estimation under bounded second order moment condition, in light of the lower bounds in [DKS18b]. When leveraging degree-2ℓ polynomials into algorithmic design, we obtain the improved $\tilde{O}\big(\alpha^{-\frac{1}{2\ell}} \sqrt{\ell}(\ell + \log\frac{1}{\alpha})\big)$ error guarantee. Specially, when taking $\ell = \Theta\big(\log\frac{1}{\alpha}\big)$, our algorithm achieves error rate of $\tilde{O}\big(\log^{\frac{3}{2}}(\frac{1}{\alpha})\big)$ in quasi-polynomial time. This is very close to the minimax error rate of $\Theta(\log^{\frac{1}{2}}(\frac{1}{\alpha}))$ established in [DKS18b].

Remark 4. If we further increase the sample size with an ℓ^ℓ multiplicative factor with $\ell = \Theta(\log\frac{1}{\alpha})$, our algorithm will achieve an $\tilde{O}\big(\log^{\frac{1}{2}}(\frac{1}{\alpha})\big)$ error guarantee, which matches the minimax lower bound. The proof follows the same pipeline and we leave it to interested readers.

1.1 Overview of Our Techniques

Our main algorithm is inspired by the multifiltering framework of [DKS18b], where the primary idea is to construct a sequence of polynomials to test the concentration of the samples to Gaussian so that the algorithm either certifies that the sample set behaves like Gaussian, or sanitizes it by removing a sufficient amount of outliers. Our key technical contribution lies into a new design of *sparse polynomials*, and new filtering rules tailored to the sparse polynomials.

Sparse polynomials and sparsity-induced filters. To ensure that our algorithm is attribute-efficient, we will only control the maximum eigenvalue of the sample covariance matrix on sparse directions. Since such computation is NP-hard in general, we first consider a sufficient condition which tests the maximum Frobenuis norm under a cardinality constraint, similar to the idea of [DKK⁺19]. If such Frobenuis norm is small, it implies a small restricted eigenvalue and hence the sample mean is returned. Otherwise, we construct sparse polynomials in the sense that they can be represented by a set of $O(\ell^2 k^{4\ell})$ basis polynomials and $O(\ell k^{2\ell})$ coordinates of the samples (see Definition 7), and measure the concentration of these sparse polynomials to the Gaussian. Now as the underlying polynomials are sparse, we also design new sparsity-induced filters to certify the sample set, as otherwise a large amount of clean samples will be removed. See Algorithm 3 and Algorithm 4.

Clustering by L_∞-norm. Technically, the success of our attribute-efficient multifiltering approach hinges on a condition that all the samples lie within a small L_∞-norm ball. It is not hard to see that all the Gaussian samples satisfy such condition, and we show that there is a simple scheme which can simultaneously prune and cluster the given samples into $O(1/\alpha)$ groups, such that the retained samples are close under the L_∞-norm and at least one group contains most of the Gaussian samples. We note that the use of the L_∞-norm as our metric ensures attribute efficiency of this step. An immediate implication of this clustering step is that the polynomials of Gaussian samples will be close enough, which facilitates the analysis of the performance of our filters. See Section 2.3.

1.2 Related Works

Breaking the barrier of the typical $O(\mathrm{poly}(d))$ sample complexity bound is one of the central problems across many fields of science and engineering. Motivated by real-world applications, a property termed sparsity is often assumed for this end, meaning that only k out of the d number of attributes contribute to the underlying inference problem. In this way, an improved bound of $O(\mathrm{poly}(k, \log d))$ can be obtained in many inference paradigms such as linear regression [CDS98, Tib96, CT05, Don06, SL17a, SL17b, SL18, WSL18], learning of threshold functions [Lit87, BHL95, STT12, PV13, ABHZ16, ZSA20, She20, SZ21], principal component analysis [Ma13, DKK+19], and mean estimation [BDLS17, DKK+19, CDK+21]. Unfortunately, the success of all these attribute-efficient algorithms hinges on the presumption that the majority of the data are uncorrupted.

Learning with mild corruption ($\alpha > 1/2$). Learning in the presence of noise has been extensively studied in a broad context. In supervised learning where a sample consists of an instance (i.e. feature vector) and a label, lots of research efforts were dedicated to robust algorithms under label noise [AL87, Slo88, MN06]. Recent years have witnessed significant progress towards optimal algorithms in the presence of label noise, see for example, [KKMS05, ABL17, DKTZ20, ZSA20, DKK+20b, ZS22] and the references therein. The regime that both instances and labels are corrupted turns out to be significantly more challenging. The problem of learning halfspaces under such setting was put forward in the 1980s [Val85, KL88], yet only until recently have efficient algorithms been established with near-optimal noise tolerance [ABL17, DKS18a, She21, SZ21]. In addition, [BJK15, KKM18, LSLC20] studied robust linear regression and [BDLS17] presented a set of interesting results under various statistical models. More in line with this work is the problem of robust mean estimation, see the breakthrough works of [DKK+16, LRV16] and many follow-up works [DKK+17, BDLS17, DKS17, SCV18, KSS18, DKK+19, HLZ20, CDK+21].

Learning with overwhelming corruption ($\alpha \le 1/2$). The agnostic label noise of [Hau92, KSS92] seems the earliest model that allows the adversary to arbitrarily corrupt any fraction of the data (say 70%), though it can only corrupt labels. Following [CSV17], a considerate number of of recent works have studied the scenario that both instances and labels are grossly corrupted, and the goal is to output a finite list of candidate parameters among which at least one is a good approximation to the target. This includes list-decodable learning of mixture models [DKS18b, DKK+21b], regression [KKK19, RY20a], and subspace recovery [RY20b, BK21]. Interestingly, there are some works studying the problem under crowdsourcing models, where the samples are collected from crowd workers and most of them behave adversarially [SVC16, ABHM17, MV18, ZS21].

It is worth noting that [DKK+22] concurrently and independently developed a polynomial-time algorithm to solve the same problem, with an interesting difference-of-pairs metric to filter outliers.

1.3 Roadmap

We collect useful notations, definitions, and some preliminary results in Section 2. Our main algorithms are described in Section 3 along with performance guarantees. We conclude the work in Section 4, and defer all proof details to the appendix.

2 Preliminaries

Vector, matrix, and tensor. For a d-dimensional vector $v = (v_1, \dots, v_d)$, denote by $\|v\|_2$ its L_2-norm, $\|v\|_1$ its L_1-norm, $\|v\|_0$ its L_0-"norm" that counts the number of non-zeros, and $\|v\|_\infty$ its infinity norm. The hard thresholding operator $\mathrm{trim}_k : \mathbb{R}^d \to \mathbb{R}^d$ keeps the k largest elements (in magnitude) of a vector and sets the remaining to zero. Let $[d] := \{1, 2, \dots, d\}$ for some natural number $d > 0$. For an index set $\Omega \subseteq [d]$, v_Ω is the vector of v restricted on Ω. We say a vector is k-sparse if it has at most k non-zero elements, and likewise for matrices and tensors. For a matrix M of size $d_1 \times d_2$, denote by $\|M\|_F$ its Frobenius norm and by $\|M\|_*$ its nuclear norm. For $U \subseteq [d_1] \times [d_2]$, denote by M_U the submatrix of M with entries restricted to U.

We also use tensors in our algorithms to ease expressions. Note that vectors and matrices can be seen as order-1 and order-2 tensors respectively. We say that an order-l tensor A is symmetric if $A_{i_1,\dots,i_l} = A_{\pi(i_1,\dots,i_l)}$ for all permutations π. Given two tensors A and B, denote by $A \otimes B$ the outer product (or tensor product) of A and B. We will slightly abuse $\|A\|_2$ to denote the L_2-norm of a tensor A by seeing it as a long vector.

Probability. We reserve the capital letter G for a random draw from $N(\mu, \mathbb{I}_d)$, i.e. $G \sim N(\mu, \mathbb{I}_d)$, where $\mu \in \mathbb{R}^d$ is the target mean that we aim to estimate which is assumed to be k-sparse. Suppose that T is a finite sample set. We use μ_T to denote the sample mean of T, i.e. $\mu_T = \frac{1}{|T|}\sum_{x \in T} x$, and use $p(T)$ to denote the random variable $p(x)$ where x is drawn uniformly from T.

Constants. The capital letter C and its subscript variants such as C_1, C_2 are used to denote positive absolute constants. However, their values may change from appearance to appearance.

2.1 Polynomials

Let $x = (x_1, \dots, x_d)$ be a d-dimensional vector in $\mathbb{R}^d$, and let $\boldsymbol{a} = (\boldsymbol{a}_1, \dots, \boldsymbol{a}_d) \in \mathbb{N}^d$ be a d-dimensional multi-index. A *monomial* of x is a product of powers of the coordinates of x with natural exponents, written as $x^{\boldsymbol{a}} := \prod_{j=1}^d x_j^{\boldsymbol{a}_j}$. A *polynomial* of x, $p(x)$, is a finite sum of its monomials multiplied by real coefficients; that is, $p(x) = \sum_{\boldsymbol{a} \in \mathcal{A}} c_{\boldsymbol{a}} x^{\boldsymbol{a}}$ where $\mathcal{A} \subset \mathbb{N}^d$ is a finite set of multi-indices and the $c_{\boldsymbol{a}}$'s are real coefficients. Note that the degree of $p(x)$ is given by $\max_{\boldsymbol{a} \in \mathcal{A}} \|\boldsymbol{a}\|_1$. We denote by $\mathbb{P}(\mathbb{R}^d, l)$ the class of polynomials on $\mathbb{R}^d$ with degree at most l. We will often use the probabilist's Hermite polynomials that form a complete orthogonal basis with respect to $N(0, \mathbb{I}_d)$.

Definition 5 (Hermite polynomials)**.** Let $x \in \mathbb{R}$ be a variate. For any natural number $l \in \mathbb{N}$, the degree-l Hermite polynomial is defined as $\mathrm{He}_l(x) = (-1)^l e^{\frac{x^2}{2}} \frac{d^l}{dx^l} e^{-\frac{x^2}{2}}$. For $\boldsymbol{a} \in \mathbb{N}^d$ and $x \in \mathbb{R}^d$, the d-variate Hermite polynomial is given by $\mathrm{He}_{\boldsymbol{a}}(x) := \prod_{i=1}^d \mathrm{He}_{\boldsymbol{a}_i}(x_i)$, which is of degree $\|\boldsymbol{a}\|_1$.

Harmonic and homogeneous polynomials. A polynomial $h(x) \in \mathbb{P}(\mathbb{R}^d, l)$ is called *harmonic* if it can be written as a linear combination of degree-l Hermite polynomials. A polynomial $\mathrm{Hom}(x) \in \mathbb{P}(\mathbb{R}^d, l)$ is called *homogeneous* if all of its monomials have degree exactly l.

Fact 6. If a polynomial is degree-l harmonic or homogeneous, then there is a one-to-one mapping between it and an order-l symmetric tensor.

To see this, we may define an operation "$\circ$" such that $\mathrm{He}_l(x_i) \circ \mathrm{He}_l(x_j) = \mathrm{He}_l(x_i) \cdot \mathrm{He}_l(x_j)$ if $i \neq j$ and equals $\mathrm{He}_{2l}(x_i)$ otherwise. Then any degree-l Hermite polynomial can be written as $\mathrm{He}_1(x_{i_1}) \circ \mathrm{He}_1(x_{i_2}) \cdots \circ \mathrm{He}_1(x_{i_l})$ where all the indices $i_t \in [d]$. We will consider that one such sequence $(i_1, \dots, i_l)$ exactly corresponds to one degree-l Hermite polynomial on $\mathbb{R}^d$, and there are d^l number of such sequences that form all degree-l Hermite polynomials. In this sense, any harmonic polynomial $h(x)$ can be written as $h(x) = \sum_{i_1,\dots,i_l} A_{i_1,\dots,i_l} \cdot \mathrm{He}_1(x_{i_1}) \circ \mathrm{He}_1(x_{i_2}) \cdots \circ \mathrm{He}_1(x_{i_l})$, where $A_{i_1,\dots,i_l}$'s are the coefficients which form an order-l tensor. If we choose A as symmetric, it is easy to see that A fully represents $h(x)$. Then, we can convert "$\circ$" back to the regular product by counting the number of times a particular index j appearing in $(i_1, \dots, i_l)$. If we denote this number as $c_j(i_1, \dots, i_l)$, we have

$$h(x) = \frac{1}{\sqrt{l!}} \sum_{i_1,\dots,i_l} A_{i_1,\dots,i_l} \prod_j \mathrm{He}_{c_j(i_1,\dots,i_l)}(x_j) =: h_A(x), \text{ with } \sum_{j=1}^d c_j(i_1, \dots, i_l) = l, \tag{2.1}$$

where the factor $1/\sqrt{l!}$ is only used to normalize the magnitude of A to ease our analysis.

Likewise, any homogeneous polynomial takes the form

$$\mathrm{Hom}_A(x) = \sum_{i_1,\dots,i_l} A_{i_1,\dots,i_l} \prod_j x_j^{c_j(i_1,\dots,i_l)}.$$

Sparse polynomials. In order to define sparse polynomials, we will first specify a set of basis polynomials $\{b_1, \dots, b_{d^l}\} \subset \mathbb{P}(\mathbb{R}^d, l)$. In this paper, we will either choose such set as all degree-l monomials or all degree-l Hermite polynomials.

Definition 7 (((κ, ψ)-sparse polynomials)**.** We say that a polynomial $p \in \mathbb{P}(\mathbb{R}^d, l)$ is (κ, ψ)-sparse if it can be represented by at most κ number of basis polynomials and ψ coordinates of the input vector. We denote by $\mathbb{P}(\mathbb{R}^d, l, \kappa, \psi)$ the class of (κ, ψ)-sparse polynomials.

Note that when κ and l are fixed, $p(x)$ will depend on at most $\kappa \cdot l$ coordinates. Thus, the introduction of the parameter ψ makes sense only when $\psi \le \kappa \cdot l$. In our algorithm, we will always have $l \le 2\ell$, $\kappa = 4\ell^2 k^{4\ell}$, and $\psi = 2\ell k^{2\ell}$ for some natural number $\ell \ge 1$.

2.2 Representative Set and Good Set

To ease our analysis, we will need a deterministic condition on the set of uncorrupted samples.

Definition 8 (Representative set)**.** Given $\alpha \in (0, \frac{1}{2}]$ and $\tau \in (0, 1)$, we say that a sample set $S_G \subset \mathbb{R}^d$ is representative with respect to $\mathbb{P} := \mathbb{P}(\mathbb{R}^d, 2\ell, 4\ell^2 k^{4\ell}, 2\ell k^{2\ell})$ if the following holds:

$$\sup_{p \in \mathbb{P}} |\Pr[p(G) \ge 0] - \Pr[p(S_G) \ge 0]| \le \epsilon_0, \text{ where } \epsilon_0 := \frac{\alpha^3}{100 k^{2\ell} \cdot \log^{2\ell}(\frac{\ell d}{\alpha \tau})}.$$

We show that a sufficiently large set drawn independently from $N(\mu, \mathbb{I}_d)$ is representative. The proof follows from the classic VC theory, and is deferred to Appendix A.1.

Proposition 9 (Sample complexity)**.** *Given $\alpha \in (0, \frac{1}{2}]$ and $\tau \in (0, 1)$, let S_G be a set consisting of $|S_G| = C \cdot \frac{(l \cdot \kappa + \psi) \log d}{\epsilon^2} \log \frac{(l \cdot \kappa + \psi) \log d}{\epsilon \tau}$ independent samples from $N(\mu, \mathbb{I}_d)$ where $C > 0$ is a sufficiently large absolute constant. Then, with probability $1 - \tau$,*

$$\sup_{p \in \mathbb{P}(\mathbb{R}^d, l, \kappa, \psi)} |\Pr[p(G) \ge 0] - \Pr[p(S_G) \ge 0]| \le \epsilon.$$

In particular, when $l = 2\ell$, $\kappa = 4\ell^2 k^{4\ell}$, $\psi = 2\ell k^{2\ell}$, and $\epsilon = \frac{\alpha^3}{100 k^{2\ell} \cdot \log^{2\ell}(\frac{\ell d}{\alpha \tau})}$ for some natural number $\ell \ge 1$, it suffices to pick $|S_G| = C' \cdot \frac{\ell^4 \cdot k^{8\ell}}{\alpha^6} \cdot \log^{6\ell}(\frac{\ell d}{\alpha \tau})$ for some sufficiently large constant C' so that S_G is a representative set.

Our algorithm will progressively remove samples from T, and a key property that ensures the success of the algorithm is that most corrupted samples are eliminated while almost all uncorrupted samples are retained. Alternatively, we hope that T contains a representative set that contributes to a nontrivial fraction. For technical reasons, we also require that all samples in T lie in a small L_∞-ball.

Definition 10 (α-good set)**.** A multiset $T \subset \mathbb{R}^d$ is α-good if the following holds:

1. There exists a set S_G which is representative and satisfies $|S_G \cap T| \ge \max\{(1 - \alpha/6) |S_G|, \alpha |T|\}$.
2. $\max_{x,y \in T} \|x - y\|_\infty \le C \cdot \sqrt{\log(d |S_G| / \tau)}$ for some constant $C > 0$.

It is not hard to verify that the initial sample set T satisfies the first condition, and will also fulfill the second one with a simple data pre-processing, as stated in the next section.

2.3 Clustering for the Initial List

Since the corrupted samples may behave adversarially, we will perform a preliminary step of clustering which splits T into an initial list of subsets, among which at least one is α-good in the sense of Definition 10. We first show that all Gaussian samples have bounded L_∞-norm with high probability, which simply follows from the Gaussian tail bound.

Lemma 11. *Given $\tau \in (0, 1)$, with probability $1 - \tau$, we have $\max_{x \in S_G} \|x - \mu\|_\infty \le \sqrt{2 \log \frac{d|S_G|}{\tau}}$, where S_G is a set of samples drawn independently from $N(\mu, \mathbb{I}_d)$.*

The above observation implies that for any $x, y \in S_G$, their distance under the L_∞-norm metric is at most $2\sqrt{2 \log(d |S_G| / \tau)} \le O\big(\sqrt{\ell \cdot \log \frac{\ell d}{\alpha \tau}}\big)$ as far as the size of S_G has the same order with the one in Proposition 9. To guarantee the existence of such S_G, it suffices to draw a corrupted sample set T that is $1/\alpha$ times larger than $|S_G|$. The lemma below further shows that this is sufficient to guarantee the existence of an α-good subset of T.

Algorithm 1 CLUSTER(T, α, τ, ℓ)

Require: A multiset of samples $T \subset \mathbb{R}^d$, parameter $\alpha \in (0, 1/2]$, failure probability $\tau \in (0, 1)$, degree of polynomials $\ell \geq 1$.

1: A set of centers $\mathcal{C} \leftarrow \emptyset$, radius $\gamma \leftarrow C_0 \cdot \sqrt{\ell \cdot \log \frac{\ell d}{\alpha \tau}}$ for some constant $C_0 > 0$.

2: For each $x \in T$, proceed as follows: **if** there are at least $\alpha \cdot |T|$ samples y in T that satisfy $\|x - y\|_\infty \leq 2\gamma$, and no sample $x' \in \mathcal{C}$ satisfies $\|x - x'\|_\infty \leq 6\gamma$ **then** $\mathcal{C} \leftarrow \mathcal{C} \cup \{x\}$.

3: For each $x_i \in \mathcal{C}$, let $T_i = \{y \in T : \|x_i - y\|_\infty \leq 6\gamma\}$.

4: **return** $\{T_1, \dots, T_{|\mathcal{C}|}\}$.

Lemma 12 (CLUSTER)**.** *Given $\alpha \in (0, \frac{1}{2}]$ and $\tau \in (0, 1)$, let T be the sample set given to the learner. If $|T| = C \cdot \frac{\ell^4 \cdot k^{8\ell}}{\alpha^7} \cdot \log^{6\ell}(\frac{\ell d}{\alpha \tau})$ and a (2α)-fraction are independent samples from $N(\mu, \mathbb{I}_d)$, Algorithm 1 returns a list of at most $1/\alpha$ many subsets of T, such that with probability at least $1 - \tau$, at least one of them is an α-good set.*

As will be clear in our analysis, the motivation of bounding the L_∞-distance is to make sure that the function value of any $p(x) = h_A(x - \mu_T) \in \mathbb{P}(\mathbb{R}^d, l, \kappa, \psi)$ is bounded for samples in the α-good subset T_i. This is because when there exist a significant fraction of good samples in T_i, we want to efficiently distinguish the corrupted and uncorrupted ones. A value-bounded polynomial function will facilitate our analysis on the function variance.

Lemma 13. *Suppose that T is α-good and a polynomial $p \in \mathbb{P}(\mathbb{R}^d, l, 4\ell^2 k^{4\ell}, 2\ell k^{2\ell})$ satisfies the following: there exists a symmetric order-l tensor A such that $\|A\|_2 \leq 1$ and $p(x) = h_A(x - \mu_T)$. Then, it holds that $\max_{x,y \in T} |p(x) - p(y)| \leq 2k^\ell \cdot \gamma^l$, where $\gamma = C_0 \cdot \sqrt{\ell \cdot \log(\frac{\ell d}{\alpha \tau})}$.*

3 Main Algorithms and Performance Guarantees

We start with a review of the multifiltering framework that has been broadly used in prior works [DKS18b, DKK20a, DKK+21b], followed by a highlight of our new techniques.

The multifiltering framework, i.e. Algorithm 2, includes three major steps. The first step is to invoke CLUSTER (Algorithm 1) to generate an initial list $\mathcal{L}$ which guarantees the existence of an α-good subset of T (see Lemma 12). We then imagine that there is a tree with root being the original contaminated sample set T and each child node of the root represents a member in $\mathcal{L}$. The algorithm iterates through these child nodes and performs one of the following: (1) creating a leaf node which is an estimate of the target mean; (2) creating one or two child nodes where are subsets of the parent node; (3) certifying that the set cannot be α-good and delete branch. In the end, if all leaves of the tree cannot be further split or deleted, the mean of the subsets on leaf nodes will be collected as a list M. It is worth noting that the goal of algorithmic design is to guarantee that there always exists a branch that includes only α-good subsets. In other words, at any level of the algorithm, at least one of the subsets of T is α-good, which ensures the existence of a good estimation in the returned list M. The final step is a black-box algorithm that reduces the size of M from $O(\mathrm{poly}(1/\alpha))$ to $O(1/\alpha)$, which is due to [DKS18b]. Our technical contributions lie into an attribute-efficient implementation of the first and second steps. In this section, we elaborate on the second step, i.e. the ATTRIBUTE-EFFICIENT-MULTIFILTER algorithm.

3.1 Overview of Attribute-Efficient Multifiltering

The ATTRIBUTE-EFFICIENT-MULTIFILTER algorithm is presented in Algorithm 3. The starting point of the algorithm is a well-known fact that if the adversary were to significantly deteriorate our estimate on μ, the spectral norm of a certain sample covariance matrix $\tilde{\Sigma}$ would become large [DKK+16, LRV16]. In order to achieve attribute-efficient sample complexity $O(\mathrm{poly}(k, \log d))$, it is however vital to control the spectral norm only on k^ℓ-sparse directions for some pre-specified polynomial degree $\ell \geq 1$, which can further be certified by a small Frobenius norm restricted on the largest $k^{2\ell}$ entries. If the restricted Frobenius norm is sufficiently small, it implies that the sample covariance matrix behaves as a Gaussian one, and the algorithm returns the empirical mean truncated to be k-sparse (see Step 4). Otherwise, the algorithm will invoke either BASICMF (i.e. Algorithm 4)

Algorithm 2 Main Algorithm: Attribute-Efficient List-Decodable Mean Estimation

Require: A multiset of samples $T \subset \mathbb{R}^d$, parameter $\alpha \in (0, 1/2]$, failure probability $\tau \in (0, 1)$, degree of polynomials $\ell \geq 1$.

1: $\{T_1, \ldots, T_m\} \leftarrow \textsc{Cluster}(T, \alpha, \tau, \ell)$, $\mathcal{L} \leftarrow \{(T_1, \alpha/2), \ldots, (T_m, \alpha/2)\}$, $M \leftarrow \emptyset$.
2: **while** $\mathcal{L} \neq \emptyset$ **do**
3: $\quad (T', \alpha') \leftarrow$ an element in $\mathcal{L}$, $\mathcal{L} \leftarrow \mathcal{L} \backslash \{(T', \alpha')\}$.
4: $\quad \mathrm{ANS} \leftarrow \textsc{Attribute-efficient-Multifilter}(T', \alpha', \tau / |T|, \ell)$.
(i) **if** ANS is a vector **then** add it into M.
(ii) **if** ANS is a list of (T_i, α_i) **then** append those with $\alpha_i \leq 1$ to $\mathcal{L}$.
(iii) **if** $\mathrm{ANS} = \mathrm{NO}$ **then** go to the next iteration.
5: **end while**
6: **return** $\textsc{ListReduction}(T, \alpha, \ell, M)$.

Algorithm 3 $\textsc{Attribute-efficient-Multifilter}(T, \alpha, \tau, \ell)$

Require: A multiset of samples $T \subset \mathbb{R}^d$, parameter $\alpha \in (0, 1/2]$, failure probability $\tau \in (0, 1)$, degree of polynomials $\ell \geq 1$.

1: $\tilde{\Sigma} \leftarrow \mathbb{E}[P_{d,\ell}(T - \mu_T) \cdot P_{d,\ell}(T - \mu_T)^\top]$, and $P_{d,\ell}(x)$ is the column vector of all degree-ℓ Hermite polynomials of x.
2: $\{(i_t, j_t)\}_{t \geq 1}^{\frac{1}{2}(k^{2\ell} + k^\ell)} \leftarrow$ index set of the k^ℓ diagonal entries and $\frac{1}{2}(k^{2\ell} - k^\ell)$ entries above the main diagonal of $\tilde{\Sigma}$ with largest magnitude. $U \leftarrow \{(i_t, j_t)\}_{t \geq 1} \cup \{(j_t, i_t)\}_{t \geq 1}$, $U' \leftarrow I \times I$, with $I = \{i_t\}_{t \geq 1} \cup \{j_t\}_{t \geq 1}$.
3: $\lambda^*_{\text{sparse}} \leftarrow \left[C_1 \cdot (\ell + C_1 \log \frac{1}{\alpha}) \cdot \log^2(2 + \log \frac{1}{\alpha})\right]^{2\ell}$ for large enough constant $C_1 > 0$.
4: **if** $\left\|(\tilde{\Sigma})_U\right\|_F \leq \lambda^*_{\text{sparse}}$ **then return** $\hat{\mu} \leftarrow \mathrm{trim}_k(\mu_T)$.
5: $(\lambda^*, v^*) \leftarrow$ the largest eigenvalue and eigenvector of $(\tilde{\Sigma})_{U'}$.
6: **if** $\lambda^* \geq \lambda^*_{\text{sparse}}$ **then**
7: $\quad$ **if** $\ell = 1$ **then** $\mathrm{ANS} \leftarrow \textsc{BasicMF}(T, \alpha, \tau, p_1)$ **else** $\mathrm{ANS} \leftarrow \textsc{HarmonicMF}(T, \alpha, \tau, p_1)$ where $p_1(x) := v^* \cdot P_{d,\ell}(x - \mu_T)$.
8: **else**
9: $\quad p_2(x) \leftarrow \frac{1}{\|A'\|_F} \cdot \left(P_{d,\ell}(x - \mu_T)^\top \cdot A' \cdot P_{d,\ell}(x - \mu_T)\right)$ with $A' := (\tilde{\Sigma})_{U'}$.
10: $\quad \mathrm{ANS} \leftarrow \textsc{HarmonicMF}(T, \alpha, \tau, p_2)$.
11: **end if**
12: **return** ANS.

or $\textsc{HarmonicMF}$ (i.e. Algortihm 5) to examine the concentration of a polynomial of the empirical data to that of Gaussian. Both algorithms will either assert that the current sample set does not contain a sufficiently large amount of Gaussian samples, or will prune many corrupted samples to increase the fraction of Gaussian ones. A more detailed description of the two algorithms can be found in Section 3.2.1 and Section 3.2.2 respectively. What is subtle in Algorithm 3 is that we will check the maximum eigenvalue λ^* of the empirical covariance matrix $\tilde{\Sigma}$ restricted on a carefully chosen subset U', which corresponds to the maximum eigenvalue on a certain $(2k^{2\ell})$-sparse direction. If λ^* is too large, this indicates an *easy* problem since it must be the case that the adversary corrupted the samples in an aggressively way. Therefore, it suffices to prune outliers using a degree-ℓ polynomial p_1 which is simply the projection of $P_{d,\ell}(x - \mu_T)$ onto the span of the maximum eigenvector; see Step 7 in Algorithm 3. On the other hand, if λ^* is on a moderate scale, it indicates that the adversary corrupted the samples in a very delicate way so that it passes the tests of both Frobenius norm and spectral norm. Now the main idea is to check the concentration of higher degree polynomials induced by the sample set; we show that it suffices to construct a degree-2ℓ harmonic polynomial; see Step 10.

While sparse mean estimation has been studied in [DKK+19] and the idea of using restricted Frobenius norm and filtering was also developed, we note that their analysis only holds in the mild corruption regime where $\alpha > 1/2$. To establish the main results, we will leverage the tools from [DKS18b], with a specific treatment on the fact that μ is k-sparse, to ensure an attribute-efficient sample complexity bound. As we will show later, a key idea to this end is to utilize a sequence of carefully chosen sparse polynomials in the sense of Definition 7 along with sparsity-induced filters.

The performance guarantee of ATTRIBUTE-EFFICIENT-MULTIFILTER is as follows.

Theorem 14 (Algorithm 3)**.** *Consider Algorithm 3 and denote by* ANS *its output. With probability* $1-\tau$*, the following holds.* ANS *cannot be* TBD*. If* ANS *is a* k*-sparse vector and if* T *is* α*-good, then* $\|\mu-\hat{\mu}\|_2 \leq \tilde{O}\big(\alpha^{-\frac{1}{2\ell}}\sqrt{\ell}(\ell+\log\frac{1}{\alpha})\big)$*. If* $\mathrm{ANS}=\mathrm{NO}$*, then* T *is not* α*-good. If* $\mathrm{ANS}=\{(T_i,\alpha_i)\}_{i=1}^m$ *for some* $m\leq 2$*, then* $T_i\subset T$ *for all* $i\in[m]$ *and* $\sum_{i=1}^m \frac{1}{\alpha_i^2}\leq\frac{1}{\alpha^2}$*; if additionally* T *is* α*-good, then at least one* T_i *is* α_i*-good. Finally, the algorithm runs in time* $O\big(\mathrm{poly}(|T|,d^\ell)\big)$*.*

3.2 Analysis of ATTRIBUTE-EFFICIENT-MULTIFILTER

We first show that if the restricted Frobenius norm of the sample covariance matrix is small, then the sample mean is a good estimate of the target mean.

Lemma 15. *Consider Algorithm 3. If the algorithm returns a vector* $\hat{\mu}$ *at Step 4 and if* T *is* α*-good, we have that* $\|\hat{\mu}-\mu\|_2\leq O\big(\alpha^{-\frac{1}{2\ell}}\sqrt{\ell}\cdot(\ell+\log\frac{1}{\alpha})\cdot\log^2(2+\log\frac{1}{\alpha})\big)$*.*

Next, we give performance guarantees on the remaining steps of Algorithm 3, where we consider the case that the algorithm does not return at Step 4. Namely, the algorithm will either reach at Step 7 or Step 10, and will return the ANS obtained thereof. These two steps will invoke BASICMF or HARMONICMF on different sparse polynomials. Observe that both algorithms may return 1) "NO", which certifies that the current input set T is not α-good; 2) a list of subsets $\{(T_i,\alpha_i)\}_{i=1}^m$ for some $m\leq 2$, on which Algorithm 3 will be called in a recursive manner; or 3) TBD, which indicates that the algorithm is uncertain on T being α-good. In the following, we prove that the way that we invoke BASICMF and HARMONICMF ensures that they will never return TBD when being called within Algorithm 3. We then give performance guarantees on these two filtering algorithms when they return "NO" or $\{(T_i,\alpha_i)\}_{i=1}^m$, thus establishing Theorem 14.

Let us consider that the algorithm reaches Step 7, i.e. the largest eigenvalue on one sparse direction is larger than the threshold $\lambda^*_{\mathrm{sparse}}$. It is easy to see that when $\ell=1$, ANS cannot be TBD since the only way that BASICMF returns TBD is when $\mathrm{Var}[p(T)]$ is not too large, but this would violate the condition that $\lambda^*>\lambda^*_{\mathrm{sparse}}$ in view of our setting on $\lambda^*_{\mathrm{sparse}}$. Similarly, we show that under the large λ^* regime, HARMONICMF will not return TBD either. Thus, we have the following lemma.

Lemma 16. *Consider Algorithm 3. If it reaches Step 7, then* $\mathrm{ANS}\neq\mathrm{TBD}$*.*

Now it remains to consider the case that the algorithm reaches Step 10, which is more subtle since the evidence from the magnitude of the largest restricted eigenvalue is not so strong to prune outliers. Note that this could happen even when T contains many outliers, since λ^* is not the maximum eigenvalue on all sparse directions but on a submatrix indexed by U'. Fortunately, if λ^* is not large, we show that the algorithm can still make progress by calling HARMONICMF on degree-2ℓ sparse polynomials. This is because higher-degree polynomials are more sensitive to outliers than low-degree polynomials, as far as we can certify the concentration of high-degree polynomials on clean samples. As a result, we will have the following guarantee.

Lemma 17. *Consider Algorithm 3. If it reaches Step 10, then* $\mathrm{ANS}\neq\mathrm{TBD}$*.*

3.2.1 Basic Multifilter for Sparse Polynomials

The BASICMF algorithm (Algorithm 4) is a key ingredient in the multifiltering framework. It takes as input a sparse polynomial p and uses it to certify whether T is α-good and sufficiently concentrated. The central idea is to measure how $p(T)$ distributed and compare it to that of the distribution of $p(G)$. We require the input p has certifiable variance on G, i.e. $\mathrm{Var}[p(G)]\leq 1$, as otherwise, it could filter away a large number of the good samples. We note that the bounded variance condition is always satisfied for degree-1 Hermite polynomials under proper normalization, while for high-degree polynomials, one cannot invoke BASICMF directly (see Section 3.2.2 for a remedy).

The way that BASICMF certifies the input sample set T not being α-good is quite simple: if not all samples lie in a small L_∞-ball, it returns "NO" at Step 2, in that this contradicts Lemma 13. Otherwise, the algorithm will attempt to search for a finer interval $[a,b]$ such that it includes most of the samples. If such interval exists, then either the adversary corrupted the samples such that the sample variance is as small as that of Gaussian while the sample mean may deviate far from the target, in which case BASICMF returns TBD at Step 5; or the sample variance is large, in which case

Algorithm 4 BASICMF(T, α, τ, p)

Require: A multiset of samples $T \subset \mathbb{R}^d$, parameter $\alpha \in (0, 1/2]$, failure probability $\tau \in (0, 1)$, a polynomial $p \in \mathbb{P}(\mathbb{R}^d, l, 4\ell^2 k^{4\ell}, 2\ell k^{2\ell})$ such that $l \le 2\ell$, $\mathrm{Var}[p(G)] \le 1$, and $p(x) = h_A(x - \mu_T)$.

1: $R \leftarrow (C_1 \cdot \log \frac{1}{\alpha})^{l/2}$, $\gamma \leftarrow C_0 \cdot \sqrt{\ell \cdot \log \frac{\ell d}{\alpha \tau}}$.

2: **if** $\max_{x,y \in T} |p(x) - p(y)| > 2k^{\ell} \cdot \gamma^{l}$ **then return** "NO".

3: **if** there is an interval $[a, b]$ of length $C_1 \cdot R \cdot \log(2 + \log \frac{1}{\alpha})$ that contains at least $(1 - \frac{\alpha}{2})$-fraction of samples in $\{p(x) : x \in T\}$ **then**

4:     **if** $\mathrm{Var}[p(T)] \le C_1 \cdot \left(\ell + C_1 \log \frac{1}{\alpha}\right)^{l} \cdot \log^2(2 + \log \frac{1}{\alpha})$ **then**

5:         **return** "TBD".

6:     **else**

7:         Find a threshold $t > 2R$ such that

$$\Pr_{x \sim T}\left[\min\{|p(x) - a|, |p(x) - b|\} \ge t\right] > \frac{32}{\alpha} \exp(-(t - 2R)^{2/l}) + \frac{2\alpha^2}{k^{2\ell} \log^{l}(\frac{\ell d}{\alpha \tau})}.$$

8:         $T' \leftarrow \{x \in T : \min\{|p(x) - a|, |p(x) - b|\} \le t\}$, $\alpha' \leftarrow \alpha \cdot \left(\frac{(1-\alpha/8)|T|}{|T'|} + \frac{\alpha}{8}\right)$.

9:         **return** $\{(T', \alpha')\}$.

10:     **end if**

11: **else**

12:     Find $t \in \mathbb{R}$, $R' > 0$ such that the sets $T_1 := \{x \in T : p(x) > t - R'\}$ and $T_2 := \{x \in T : p(x) < t + R'\}$ satisfy

$$|T_1|^2 + |T_2|^2 \le |T|^2 (1 - \alpha/100)^2 \text{ and } |T| - \max(|T_1|, |T_2|) \ge \alpha |T| / 4.$$

13:     $\alpha_i \leftarrow \alpha \cdot (1 - \alpha^2/100) \cdot |T| / |T_i|$, for $i = 1, 2$.

14:     **return** $\{(T_1, \alpha_1), (T_2, \alpha_2)\}$.

15: **end if**

it is possible to construct a sparsity-induced filter to prune outliers (see Steps 7 and 8). We note that in Step 7, the first term on the right-hand side is derived from Chernoff bound for degree-l Gaussian polynomials and the second term is due to concentration of empirical samples to Gaussian (see Definition 8), both of which are scaled by a factor $8/\alpha$ so that the number of the samples removed from T is $8/\alpha$ times more than that of the good samples in the representative set $S_G \subset T$, which means most of the removed samples are outliers. We show by contradiction the existence of the threshold t (see Lemma 26). In fact, had such threshold t not existed, the set T must be sufficiently concentrated such that the algorithm would have returned at Step 5. This essentially relies on our result of the initial clustering of Algorithm 1, which guarantees that each subset T is bounded in a small L_∞-ball and the function value of p on the α-good T does not change drastically (Lemma 13). We then show that equipped with such threshold t, T' is a subset of T and it is α'-good if T is α-good (Lemma 28).

When there is no such short interval $[a, b]$, the algorithm splits T into two overlapping subsets $\{T_1, T_2\}$ such that $T_1 \cap T_2$ is large enough to contain most of the samples in S_G. This guarantees that most of the samples in S_G (if T is α-good) are always contained in one subset and thus there always exists an α-good subset of T. We show that an appropriate threshold t can also be found at Step 12 (Lemma 30), and at least one T_i is α_i-good if T is α-good.

As a result, we have the following guarantees for Algorithm 4; see Appendix B for the full proof.

Theorem 18 (BASICMF)**.** *Consider Algorithm 4. Denote by* ANS *its return. Suppose that T being α-good implies* $\mathrm{Var}[p(G)] \le 1$. *Then with probability $1 - \tau$, the following holds.* ANS *is either "NO", "TBD", or a list of $\{(T_i, \alpha_i)\}_{i=1}^m$ with $m \le 2$. 1) If* ANS = NO, *then T is not α-good. 2) If* ANS = TBD, *then* $\mathrm{Var}[p(T)] \le O\left(\left(\ell + \log \frac{1}{\alpha}\right)^{l} \cdot \log^2(2 + \log \frac{1}{\alpha})\right)$*; and if additionally T is α-good, then* $|\mathbb{E}[p(G)] - \mathbb{E}[p(T)]| \le O\left(\left(\ell + \log \frac{1}{\alpha}\right)^{\frac{l}{2}} \cdot \log(2 + \log \frac{1}{\alpha})\right)$*. 3) If* ANS $= \{(T_i, \alpha_i)\}_{i=1}^m$*, then $T_i \subset T$ and $\sum_i \frac{1}{\alpha_i^2} \le \frac{1}{\alpha^2}$ for all $i \in [m]$; if additionally T is α-good, then at least one T_i is α_i-good.*

Algorithm 5 HARMONICMF(T, α, τ, p)

Require: A multiset of samples $T \subset \mathbb{R}^d$, parameter $\alpha \in (0, 1/2]$, failure probability $\tau \in (0, 1)$, a polynomial $p \in \mathbb{P}(\mathbb{R}^d, l, 2\ell k^{2\ell}, 2\ell k^{2\ell})$ such that $p(x) = h_A(x - \mu_T)$ and $\|A\|_2 = 1$.

1: **for** $l' = 0, 1, \ldots, l$ **do**

2: Let $B^{(l')}$ be an order-$2l'$ tensor with

$$B^{(l')}_{i_1,\ldots,i_{l'},j_1,\ldots,j_{l'}} = \sum_{k_{l'+1},\ldots,k_l} A_{i_1\ldots,i_{l'},k_{l'+1},\ldots,k_l} A_{j_1\ldots,j_{l'},k_{l'+1},\ldots,k_l}.$$

3: Consider $B^{(l')}$ as a $d^{l'} \otimes d^{l'}$ symmetric matrix by grouping each of the $i_1, \ldots, i_{l'}$ and $j_1, \ldots, j_{l'}$ coordinates together. Apply eigenvalue decomposition on $B^{(l')}$ to obtain $B^{(l')} = \sum_i \lambda_i V_i \otimes V_i$.

4: $\mathrm{ANS}_i \leftarrow$ MULTILINEARMF$(T, V_i, l', \alpha, \tau/(ld^l))$ for every V_i. If $\mathrm{ANS}_i = \mathrm{NO}$ or a list of $\{(T_j, \alpha_j)\}$ for some i, then **return** ANS_i. If $\mathrm{ANS}_i = \mathrm{TBD}$, **continue**.

5: **end for**

6: $\mathrm{ANS} \leftarrow$ BASICMF$(T, \alpha, \tau, \frac{1}{\beta} h_A(x - \mu_T))$ with $\beta := \left(C_1 \cdot (1 + \log \frac{1}{\alpha}) \cdot \log^2(2 + \log \frac{1}{\alpha})\right)^{\frac{l}{2}}$. If $\mathrm{ANS} = \mathrm{NO}$ or a list of (T_j, α_j), **return** ANS. If $\mathrm{ANS} = \mathrm{TBD}$, still **return** "NO".

3.2.2 Harmonic Multifilter with Hermite Polynomials

Recall that applying BASICMF (Algorithm 4) on a polynomial p requires $\mathrm{Var}[p(G)] \le 1$. It is nontrivial to verify this condition for a high-degree polynomial p, as the variance of high-degree Gaussian polynomials depends on the distribution mean, i.e. $\mu - \mu_T$ in this case, which is unfortunately unknown. As a remedy, notice that for any harmonic polynomial $h_A(x)$, $\mathbb{E}_{x \sim N(\mu', \mathbb{I})}[h_A(x)^2]$ equals the summation of homogeneous polynomials of μ', which can also be seen as the expectation of multilinear polynomials over independent variables $X_{(i)} \sim N(\mu', \mathbb{I}_d)$. Thus, we only need to verify the expectation of these corresponding multilinear polynomials, whose variance on G does not hinge on μ'. The harmonic multifilter is presented in Algorithm 5, where the subroutine MULTILINEARMF can be found in Appendix D. We first present the guarantee when Algorithm 5 returns all TBD at Step 4 and reaches Step 6, where we can certify a bounded variance for $h_A(x - \mu_T)$ on G.

Lemma 19 (Variance of p)**.** *Consider Algorithm 5. If it reaches Step 6 and T is α-good, then we have $\mathbb{E}[h_A(G - \mu_T)^2] \le \beta^2$.*

Based on Lemma 19, we have that $\mathrm{Var}[h_A(G - \mu_T)/\beta] \le 1$, for which we can invoke BASICMF on $h_A(x - \mu_T)/\beta$ and Theorem 18 can be applied immediately. We are ready to elaborate the proof ideas for Lemma 16 and 17. First, observe that BASICMF returns "TBD" at Step 6 if and only if $\mathrm{Var}[h_A(T - \mu_T)/\beta] \le C_1 \cdot \left(\ell + C_1 \log \frac{1}{\alpha}\right)^l \cdot \log^2(2 + \log \frac{1}{\alpha})$. Now return to Algorithm 3. When $h_A(x - \mu_T) = p_1(x)$, this could not happen because $\mathrm{Var}[p_1(T)] = \mathrm{Var}[v^* \cdot P_{d,\ell}(T - \mu_T)] \ge \lambda^* \ge \lambda^*_{\mathrm{sparse}} = \left[C_1 \cdot (\ell + C_1 \log \frac{1}{\alpha}) \cdot \log^2(2 + \log \frac{1}{\alpha})\right]^{2\ell} \ge \beta^2 \cdot C_1 \cdot \left(\ell + C_1 \log \frac{1}{\alpha}\right)^\ell \cdot \log^2(2 + \log \frac{1}{\alpha})$. A contradiction that implies Lemma 16. When $h_A(x - \mu_T) = p_2(x)$, the case is more delicate. Here, we instead show that T must not be α-good and HARMONICMF will return "NO" correctly. This is because if T is α-good, Proposition 18 implies that $\mathbb{E}[p_2(G)]$ is close to $\mathbb{E}[p_2(T)]$, and together with Lemma 19 we can show that $\mathbb{E}[p_2(T)]$ is small. However, by construction $\left\|(\tilde{\Sigma})_U\right\| = \mathbb{E}[p_2(T)] \ge \lambda^*_{\mathrm{sparse}}$, a contradiction that gives Lemma 17. The detailed proof can be found in Appendix B.3.

4 Conclusion and Future Work

In this paper, we developed an attribute-efficient mean estimation algorithm which achieves sample complexity poly-logarithmic in the dimension with low-degree sparse polynomials under the list-decodable setting. A natural question is whether the current techniques could be utilized to attribute-efficiently solve the other list-decodable problems, such as learning of halfspaces and linear regression.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers and meta-reviewer for valuable discussions. This work is supported by NSF-IIS-1948133 and the startup funding from Stevens Institute of Technology.

References

[ABHM17] Pranjal Awasthi, Avrim Blum, Nika Haghtalab, and Yishay Mansour. Efficient PAC learning from the crowd. In *Proceedings of the 30th Annual Conference on Learning Theory*, pages 127–150, 2017.

[ABHZ16] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Proceedings of the 29th Annual Conference on Learning Theory*, pages 152–192, 2016.

[ABL17] Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. *Journal of the ACM*, 63(6):50:1–50:27, 2017.

[AL87] Dana Angluin and Philip D. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1987.

[BDLS17] Sivaraman Balakrishnan, Simon S. Du, Jerry Li, and Aarti Singh. Computationally efficient robust sparse estimation in high dimensions. In *Proceedings of the 30th Annual Conference on Learning Theory*, pages 169–212, 2017.

[BHL95] Avrim Blum, Lisa Hellerstein, and Nick Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *Journal of Computer and System Sciences*, 50(1):32–40, 1995.

[BJK15] Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. In *NIPS*, pages 721–729, 2015.

[BK21] Ainesh Bakshi and Pravesh K. Kothari. List-decodable subspace recovery: Dimension independent error in polynomial time. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms*, pages 1279–1297, 2021.

[CDK+21] Yu Cheng, Ilias Diakonikolas, Daniel M. Kane, Rong Ge, Shivam Gupta, and Mahdi Soltanolkotabi. Outlier-robust sparse estimation via non-convex optimization. *CoRR*, abs/2109.11515, 2021.

[CDS98] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[CMY20] Yeshwanth Cherapanamjeri, Sidhanth Mohanty, and Morris Yau. List decodable mean estimation in nearly linear time. In *61st IEEE Annual Symposium on Foundations of Computer Science*, pages 141–148. IEEE, 2020.

[CSV17] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60, 2017.

[CT05] Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[DKK+16] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 655–664, 2016.

[DKK+17] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 999–1008. PMLR, 2017.

[DKK+19] Ilias Diakonikolas, Daniel Kane, Sushrut Karmalkar, Eric Price, and Alistair Stewart. Outlier-robust high-dimensional sparse estimation via iterative filtering. In *NeurIPS*, pages 10688–10699, 2019.

[DKK20a] Ilias Diakonikolas, Daniel Kane, and Daniel Kongsgaard. List-decodable mean estimation via iterative multi-filtering. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, 2020.

[DKK+20b] Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. A polynomial time algorithm for learning halfspaces with Tsybakov noise. *CoRR*, abs/2010.01705, 2020.

[DKK+21a] Ilias Diakonikolas, Daniel Kane, Daniel Kongsgaard, Jerry Li, and Kevin Tian. List-decodable mean estimation in nearly-pca time. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems*, pages 10195–10208, 2021.

[DKK+21b] Ilias Diakonikolas, Daniel M. Kane, Daniel Kongsgaard, Jerry Li, and Kevin Tian. Clustering mixture models in almost-linear time via list-decodable mean estimation. *CoRR*, abs/2106.08537, 2021.

[DKK+22] Ilias Diakonikolas, Daniel M. Kane, Sushrut Karmalkar, Ankit Pensia, and Thanasis Pittas. List-decodable sparse mean estimation via difference-of-pairs filtering. *CoRR*, abs/2206.05245, 2022.

[DKS17] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science*, pages 73–84, 2017.

[DKS18a] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing*, pages 1061–1073, 2018.

[DKS18b] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical gaussians. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1047–1060. ACM, 2018.

[DKTZ20] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with Massart noise under structured distributions. In *Proceedings of the 33rd Annual Conference on Learning Theory*, pages 1486–1513, 2020.

[Don06] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[Hau92] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[HLZ20] Samuel B. Hopkins, Jerry Li, and Fred Zhang. Robust and heavy-tailed mean estimation made simple, via regret minimization. *CoRR*, abs/2007.15839, 2020.

[Hub64] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

[KKK19] Sushrut Karmalkar, Adam R. Klivans, and Pravesh Kothari. List-decodable linear regression. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, pages 7423–7432, 2019.

[KKM18] Adam R. Klivans, Pravesh K. Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. In *Proceedings of the 31st Annual Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1420–1430. PMLR, 2018.

[KKMS05] Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 11–20, 2005.

[KL88] Michael J. Kearns and Ming Li. Learning in the presence of malicious errors. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 267–280, 1988.

[KSS92] Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward efficient agnostic learning. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory*, pages 341–352, 1992.

[KSS18] Pravesh K. Kothari, Jacob Steinhardt, and David Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1035–1046, 2018.

[Lit87] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. In *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–77, 1987.

[LRV16] Kevin A. Lai, Anup B. Rao, and Santosh S. Vempala. Agnostic estimation of mean and covariance. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 665–674, 2016.

[LSLC20] Liu Liu, Yanyao Shen, Tianyang Li, and Constantine Caramanis. High dimensional robust sparse regression. In *The 23rd International Conference on Artificial Intelligence and Statistics,*, volume 108 of *Proceedings of Machine Learning Research*, pages 411–421. PMLR, 2020.

[LT91] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag Berlin Heidelberg, 1991.

[Ma13] Zongming Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

[MN06] Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, pages 2326–2366, 2006.

[MV18] Michela Meister and Gregory Valiant. A data prism: Semi-verified learning in the small-alpha regime. In *Proceedings of the 31st Conference On Learning Theory*, pages 1530–1546, 2018.

[PV13] Yaniv Plan and Roman Vershynin. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *IEEE Transactions on Information Theory*, 59(1):482–494, 2013.

[RY20a] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, pages 161–180, 2020.

[RY20b] Prasad Raghavendra and Morris Yau. List decodable subspace recovery. In *Proceedings of the 33rd Annual Conference on Learning Theory*, pages 3206–3226, 2020.

[SCV18] Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference*, pages 45:1–45:21, 2018.

[She20] Jie Shen. One-bit compressed sensing via one-shot hard thresholding. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence*, pages 510–519, 2020.

[She21] Jie Shen. Sample-optimal PAC learning of halfspaces with malicious noise. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9515–9524, 2021.

[SL17a] Jie Shen and Ping Li. On the iteration complexity of support recovery via hard thresholding pursuit. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3115–3124, 2017.

[SL17b] Jie Shen and Ping Li. Partial hard thresholding: Towards a principled analysis of support recovery. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 3127–3137, 2017.

[SL18] Jie Shen and Ping Li. A tight bound of hard thresholding. *Journal of Machine Learning Research*, 18(208):1–42, 2018.

[Slo88] Robert H. Sloan. Types of noise in data for concept learning. In *Proceedings of the First Annual Workshop on Computational Learning Theory*, pages 91–96, 1988.

[STT12] Rocco A. Servedio, Li-Yang Tan, and Justin Thaler. Attribute-efficient learning and weight-degree tradeoffs for polynomial threshold functions. In *Proceedings of the 25th Annual Conference on Learning Theory*, pages 1–19, 2012.

[SVC16] Jacob Steinhardt, Gregory Valiant, and Moses Charikar. Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 4439–4447, 2016.

[SZ21] Jie Shen and Chicheng Zhang. Attribute-efficient learning of halfspaces with malicious noise: Near-optimal label complexity and noise tolerance. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, pages 1072–1113, 2021.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[Tuk60] John W. Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485, 1960.

[Val85] Leslie G. Valiant. Learning disjunction of conjunctions. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 560–566, 1985.

[Wai19] Martin J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

[WSL18] Jing Wang, Jie Shen, and Ping Li. Provable variable selection for streaming features. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5158–5166, 2018.

[ZS21] Shiwei Zeng and Jie Shen. Semi-verified learning from the crowd with pairwise comparisons. *CoRR*, abs/2106.07080, 2021.

[ZS22] Shiwei Zeng and Jie Shen. Efficient PAC learning from the crowd with pairwise comparisons. In *Proceedings of the 39th International Conference on Machine Learning*, pages 25973–25993, 2022.

[ZSA20] Chicheng Zhang, Jie Shen, and Pranjal Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, pages 7184–7197, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Omitted Proofs from Section 2

A.1 Proof of Proposition 9

Proof. Fix a subset $\Omega \subset [d]$ with size ψ, and then fix a set of κ monomials on Ω with degree at most l, denoted by $\mathcal{M}(\Omega, l)$. Let $\mathbb{P}(\mathbb{R}^d, \mathcal{M}(\Omega, l), \Omega)$ be the induced class of polynomials. Note that $\mathbb{P}(\mathbb{R}^d, l, \kappa, \psi) = \cup_\Omega \cup_{\mathcal{M}(\Omega,l)} \mathbb{P}(\mathbb{R}^d, \mathcal{M}(\Omega, l), \Omega)$.

It is easy to see that for any $p \in \mathbb{P}(\mathbb{R}^d, \mathcal{M}(\Omega, l), \Omega)$, it can be represented by a linear combinations of the κ monomials. Thus, the VC dimension of this class equals $\kappa + 1$. Then, we note that there are $\sum_{j=0}^{\psi} \binom{d}{j}$ choices of Ω, and for any given Ω, there are $\sum_{j=0}^{\kappa} \binom{2d^l}{j}$ choices of $\mathcal{M}(\Omega, l)$. Therefore, the total number of the subclass $\mathbb{P}(\mathbb{R}^d, \mathcal{M}(\Omega, l), \Omega)$ is at most

$$\sum_{j=0}^{\psi} \binom{d}{j} \cdot \sum_{j=0}^{\kappa} \binom{2d^l}{j} \leq \left(\frac{ed}{\psi}\right)^{\psi} \cdot \left(\frac{2ed^l}{\kappa}\right)^{\kappa}. \tag{A.1}$$

The concept class union argument states that for $\mathcal{H} = \cup_{i=1}^m \mathcal{H}_i$, the VC dimension of $\mathcal{H}$ is upper bounded by $O(\max\{V, \log m + V \log \frac{\log m}{V}\})$, where V is an upper bound on the VC dimension of all $\mathcal{H}_i$. In our case, we have $V = \kappa + 1$ and $m \leq \left(\frac{ed}{\psi}\right)^{\psi} \cdot \left(\frac{2ed^l}{\kappa}\right)^{\kappa}$. By calculation, we can show that the VC dimension of $\mathbb{P}(\mathbb{R}^d, l, \kappa, \psi)$ is upper bounded by

$$\psi \log \frac{ed}{\psi} + \kappa \log \frac{2ed^l}{\kappa} + \kappa + 1 \leq (l\kappa + \psi) \log d =: d'. \tag{A.2}$$

Recall that the VC theory states that for any $\epsilon, \tau \in (0, 1)$, as long as $|S_G| \geq C \left(\frac{d'}{\epsilon^2} \log \frac{d'}{\epsilon} + \frac{1}{\epsilon^2} \log \frac{1}{\tau}\right)$ for some absolute constant $C > 0$, the following holds with probability $1 - \tau$:

$$\sup_{p \in \mathbb{P}(\mathbb{R}^d, l, \kappa, \psi)} \left| \Pr[p(G) \geq 0] - \Pr_{x \sim S_G}[p(x) \geq 0] \right| \leq \epsilon. \tag{A.3}$$

With the expression of d' in (A.2), it is not hard to see that we can set $|S_G| = C \cdot \frac{(l \cdot \kappa + \psi) \log d}{\epsilon^2} \log \frac{(l \cdot \kappa + \psi) \log d}{\epsilon \tau}$ for some absolute constant $C > 0$ to ensure that the above holds.

When $l = 2\ell$, $\kappa = 4\ell^2 k^{4\ell}$, $\psi = 2\ell k^{2\ell}$, and $\epsilon = \frac{\alpha^3}{100k^{2\ell} \cdot \log^{2\ell}(\frac{\ell d}{\alpha \tau})}$ for some natural number $\ell \geq 1$, by algebraic calculation, it suffices to pick $|S_G| = C' \cdot \frac{\ell^4 \cdot k^{8\ell}}{\alpha^6} \cdot \log^{6\ell}(\frac{\ell d}{\alpha \tau})$ for some sufficiently large constant C'. This completes the proof. □

A.2 Proof of Lemma 11

Proof. By the standard tail bound of Gaussian distribution, for any x drawn from $N(\mu, \mathbb{I}_d)$, it holds that for any given index $i \in [d]$, $\Pr[|x_i - \mu_i| \geq t] \leq 2\exp(-t^2/2)$. By taking union bound over both index i and sample $x \in S_G$, we have $\Pr[\max_{x \in S_G} \max_{i \in [d]} |x_i - \mu_i| \geq t] \leq 2d\,|S_G| \exp(-t^2/2)$. Choosing $t = \sqrt{2\log(d\,|S_G|\,/\tau)}$ completes the proof. □

A.3 Proof of Lemma 12

Proof. Let S_G be the subset of T containing the samples drawn i.i.d. from $N(\mu, \mathbb{I}_d)$. Since $|S_G| = 2\alpha \cdot |T|$, we know that S_G is a representative set with probability at least $1 - \tau$ in light of Prop. 9.

Consider Algorithm 1. If for all $x, y \in T$, we have

$$\|x - y\|_\infty \leq 6\gamma, \tag{A.4}$$

then the algorithm returns only one cluster and the lemma follows immediately.

If that is not the case, we first note that, with probability at least $1 - \tau$ all of the samples in S_G satisfy Eq. (A.4) due to Lemma 11. Let us condition on this event occurs from now on.

Algorithm 1 constructs a set of disjoint L_∞-balls of radius 2γ, of which each is centered at one sample in T and contains at least an α-fraction of samples in T. Therefore, the number of such balls is at most $m = \lfloor 1/\alpha \rfloor$. Denote the set by $\{\mathbb{B}_1, \ldots, \mathbb{B}_m\}$. Let $\mathbb{B}_i'$ be the ball that has the same center as $\mathbb{B}_i$ but with ℓ_∞-radius of 6γ. In the following, we show that there exists $i \in [m]$, such that $T_i = T \cap \mathbb{B}_i'$ is α-good.

Consider a sample $x \in S_G$, for which we know that $\|x - \mu\|_\infty \le \gamma$. Then, for the L_∞-ball $\mathbb{B}_x := \{y \in \mathbb{R}^d : \|y - x\|_\infty \le 2\gamma\}$, all of the samples in S_G will be contained in $\mathbb{B}_x$. In addition, there must exist one $\mathbb{B}_i$ that intersects $\mathbb{B}_x$, as otherwise $\mathbb{B}_x$ will be in the set $\{\mathbb{B}_1, \ldots, \mathbb{B}_m\}$. That is, $\exists z \in T, z \in \mathbb{B}_x \cap \mathbb{B}_i$. By construction, $\mathbb{B}_x$ must be containted in $\mathbb{B}_i'$. Therefore, all samples of S_G must be included in T_i and T_i is α-good. $\square$

A.4 Proof of Lemma 13

Proof. Recall that after running Algorithm 1, every subset T_i is contained in an L_∞-ball of radius 6γ. By Jensen's inequality and the convexity of the L_∞-norm, we have for all $x \in T$, $\|x - \mu_T\|_\infty \le 6\gamma$.

Recall that we assumed $p(x) = h_A(x - \mu_T)$. Thus $\mathbb{E}_{x \sim N(\mu_T, \mathbb{I})}[p(x)] = 0$ due to the definition of harmonic polynomials. Thus, $\mathrm{Var}_{x \sim N(\mu_T, \mathbb{I})}[p(x)] = \|A\|_2^2$. Denote $z = x - \mu_T$. Then,

$$|p(x)| = \left| \sum_{j \in [k^{2\ell}]} c_{\boldsymbol{a}^{(j)}} \frac{\mathrm{He}_{\boldsymbol{a}^{(j)}}(z)}{\sqrt{\left\|\boldsymbol{a}^{(j)}\right\|_1 !}} \right| \le \sqrt{\left(\sum_{j \in [k^{2\ell}]} c_{\boldsymbol{a}^{(j)}}^2 \right) \left(\sum_{j \in [k^{2\ell}]} \frac{\mathrm{He}_{\boldsymbol{a}^{(j)}}(z)^2}{\left\|\boldsymbol{a}^{(j)}\right\|_1 !} \right)}. \tag{A.5}$$

where $\boldsymbol{a}^{(j)}$ is a d-dimensional multi-index for the j-th monomial, and $c_{\boldsymbol{a}^{(j)}}$ denotes its coefficient. Observe that in the first step, $p(x)$ is written as a linear combination of $k^{2\ell}$ Hermite polynomials, since we are considering $p \in \mathbb{P}(\mathbb{R}^d, l, k^{2\ell}, 2\ell k^{2\ell})$. Note also that $\sum_{j \in [k^{2\ell}]} c_{\boldsymbol{a}^{(j)}}^2 = \|A\|_2^2 \le 1$.

To bound the second factor on the right-hand side of (A.5), we use Mehler's formula, which shows that for any u with $|u| < 1$ and any natural number a,

$$\sum_{a=0}^{\infty} \frac{\mathrm{He}_a^2(z_i) u^a}{a!} = \frac{1}{\sqrt{1 - u^2}} e^{\frac{u}{1+u} z_i^2},$$

Since each $\mathrm{He}_{\boldsymbol{a}^{(j)}}(z)$ has degree at most l, it can be decomposed as a product of at most l univariate Hermite polynomials. Thus, we take such product and sum over $j \in [k^{2\ell}]$ to obtain

$$\sum_{j \in [k^{2\ell}]} \frac{\prod_{\boldsymbol{a}_i^{(j)} \neq 0} \left(\mathrm{He}_{\boldsymbol{a}_i^{(j)}}(z_i)^2 \cdot u^{\boldsymbol{a}_i^{(j)}} \right)}{\left\|\boldsymbol{a}^{(j)}\right\|_1 !} \le k^{2\ell} \cdot (1 - u^2)^{-\frac{l}{2}} \cdot e^{\frac{u}{1+u} \|\mathrm{trim}_l(z)\|_2^2}.$$

To simplify the above expression, observe that $\prod_{\boldsymbol{a}_i^{(j)} \neq 0} u^{\boldsymbol{a}_i^{(j)}} = u^{\left\|\boldsymbol{a}^{(j)}\right\|_1} \ge u^l$. In addition, $\|\mathrm{trim}_l(z)\|_2^2 \le l \cdot \|z\|_\infty^2 \le 36 l \gamma^2$. Lastly, by algebra, $(1 - u^2)^{-\frac{l}{2}} \le e^{\frac{u^2 l}{2}}$. Putting all pieces together gives

$$\sum_{j \in [k^{2\ell}]} \frac{\mathrm{He}_{\boldsymbol{a}^{(j)}}(z)^2}{\left\|\boldsymbol{a}^{(j)}\right\|_1 !} \le u^{-l} \cdot k^{2\ell} \cdot e^{\frac{u^2 l}{2}} \cdot e^{\frac{36 l \gamma^2 u}{1+u}} = k^{2\ell} \cdot u^{-l} \cdot e^{\frac{u^2 l}{2} + \frac{36 l \gamma^2 u}{1+u}}.$$

We set $u = \frac{1}{\gamma}$; this is possible as $\gamma > 1$. Then the exponent $\frac{u^2 l}{2} + \frac{36 l \gamma^2 u}{1+u} = \frac{l}{2\gamma^2} + \frac{36 l}{1 + 1/\gamma^2} \le 37 l$. Without loss of generality, we may assume that $\gamma > e^{37}$; in fact, we can always ensure this by setting $\gamma = (C_0 + e^{37}) \cdot \sqrt{\ell \cdot \log \frac{\ell d}{\alpha \tau}}$ where C_0 is the constant given in Algorithm 1. Thus, it follows that

$$\sum_{j \in [k^{2\ell}]} \frac{\mathrm{He}_{\boldsymbol{a}^{(j)}}(z)^2}{\left\|\boldsymbol{a}^{(j)}\right\|_1 !} \le k^{2\ell} \cdot \gamma^l \cdot e^{37 l} \le k^{2\ell} \cdot \gamma^{2l}.$$

Plugging it into (A.5) completes the proof. $\square$

B Analysis of ATTRIBUTE-EFFICIENT-MULTIFILTER

We collect a few useful facts about Hermite polynomials.

Recall that for an order-l tensor $A \in \mathbb{R}^{d^l}$, $\|A\|_2$ denotes its L_2 norm by seeing it as a long vector, and for a polynomial $p : \mathbb{R}^d \to \mathbb{R}$, $\|p\|_2 := \mathbb{E}_{x\sim N(0,\mathbb{I}_d)}[p^2(x)])^{1/2}$.

The following can be easily seen from the definition of harmonic polynomials.

Fact 20. For all order-l symmetric tensors A and its corresponding harmonic polynomial h_A, we have that $\|h_A\|_2 = \|A\|_2$. Moreover, if $l > 0$, then $\mathbb{E}_{x\sim N(0,\mathbb{I}_d)}[h_A(x)] = 0$.

Claim 21. Let $v \in \mathbb{R}^d$ be a unit vector. For $x \in \mathbb{R}^d$, the polynomial $p(x) = \mathrm{He}_l(v \cdot x)$ is harmonic with respect to x with degree l. That is, there exists a tensor $A = \mathrm{tensor}(p)$ which is symmetric and with order l.

B.1 Proof of Lemma 15

Proof. Recall that we denoted $\lambda^*_{\text{sparse}} = C_1 \cdot \left[(\ell + C_1 \log \frac{1}{\alpha}) \cdot \log^2(2 + \log \frac{1}{\alpha})\right]^{2\ell}$ in Algorithm 3.

Observe that if $\left\|\tilde{\Sigma}_U\right\|_F \le \lambda^*_{\text{sparse}}$, then for any index set $\Omega \subset [d^\ell]$ with $|\Omega| \le k^\ell$, we have

$$\lambda_{\max}(\tilde{\Sigma}_{\Omega\times\Omega}) \le \left\|\tilde{\Sigma}_{\Omega\times\Omega}\right\|_F \le \left\|\tilde{\Sigma}_U\right\|_F \le \lambda^*_{\text{sparse}},$$

where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue and the second step follows from our choice of U which maximizes the restricted Frobenius norm.

Thus, for any $u \in \mathbb{R}^{d^\ell}$ with $\|u\|_0 \le k^\ell$,

$$u^\top \tilde{\Sigma} u \le \lambda_{\max}(\tilde{\Sigma}_{\Omega\times\Omega}) \le \lambda^*_{\text{sparse}}. \tag{B.1}$$

Let v be a k-sparse unit vector in $\mathbb{R}^d$. That is, $v \in \mathbb{R}^d, \|v\|_0 \le k, \|v\|_2 = 1$. Consider some symmetric order-ℓ tensor B such that $\mathrm{He}_\ell(v \cdot (x - \mu_T)) = h_B(x - \mu_T)$ (Claim 21). Due to the sparsity of v, we know that B is an outer product of ℓ number of k-sparse vectors; hence $\|B\|_0 \le k^\ell$. As $h_B(x - \mu_T)$ is a degree-ℓ harmonic polynomial and the vector $P_{d,\ell}(x - \mu_T)$ includes all Hermite polynomials with degree exactly ℓ, we know that we can write $h_B(x - \mu_T) = u_B \cdot P_{d,\ell}(x - \mu_T)$ for some $u_B \in \mathbb{R}^{d^\ell}, \|u_B\|_0 \le k^\ell$. Thus, we have that

$$\mathbb{E}[h_B(T - \mu_T)^2] = \mathbb{E}[(u_B \cdot P_{d,\ell}(T - \mu_T))^2] = u_B^\top \tilde{\Sigma} u_B \le \lambda^*_{\text{sparse}} \|u_B\|_2^2 = \lambda^*_{\text{sparse}} \|B\|_2^2 .$$

By Fact 20, observe that $\|B\|_2^2 = \mathbb{E}_{x\sim N(\mu_T,\mathbb{I}_d)}[h_B(x - \mu_T)^2] = \ell!$, and thus we have $\mathbb{E}[\mathrm{He}_\ell(v \cdot (T - \mu_T))^2] = \mathbb{E}[h_B(T - \mu_T)^2] \le \lambda^*_{\text{sparse}}\ell!$.

As a result, we have for any k-sparse unit vector $v \in \mathbb{R}^d$ that

$$\begin{aligned}
\mathbb{E}[\mathrm{He}_\ell(v \cdot (S_G \cap T - \mu_T))^2] &= \frac{1}{|S_G \cap T|} \sum_{x\in S_G\cap T} \mathrm{He}_\ell(v \cdot (x - \mu_T))^2 \\
&\le \frac{1}{\alpha \cdot |T|} \sum_{x\in T} \mathrm{He}_\ell(v \cdot (x - \mu_T))^2 \\
&= \frac{1}{\alpha} \cdot \mathbb{E}[\mathrm{He}_\ell(v \cdot (T - \mu_T))^2] \le \frac{\lambda^*_{\text{sparse}} \cdot \ell!}{\alpha},
\end{aligned} \tag{B.2}$$

where the first inequality follows from the condition that T is α-good, which, by Definition 10, implies $|S_G \cap T| / |T| \ge \alpha$.

The remaining analysis borrows the proof strategy from [DKS18b]. In particular, we will need the following lemma.

Lemma 22 (Lemma 3.34 of [DKS18b])**.** *For any $v \in \mathbb{R}^d$, the polynomial $\mathrm{He}_l(v \cdot (G - \mu_T))$ has mean $(v \cdot (\mu - \mu_T))^l$ and variance at most $2\max(l, v \cdot (\mu - \mu_T))^{2(l-1)}$.*

Now to ease the notation, write $\theta := v \cdot (\mu - \mu_T)$. By Cantelli's inequality we have

$$\Pr\left[\mathrm{He}_\ell(v \cdot (G - \mu_T)) \geq \theta^\ell - \sqrt{2}\max(\ell, \theta)^{(\ell-1)}\right]$$
$$\geq 1 - \frac{\mathrm{Var}[\mathrm{He}_\ell(v \cdot (G - \mu_T))]}{\mathrm{Var}[\mathrm{He}_\ell(v \cdot (G - \mu_T))] + \mathrm{Var}[\mathrm{He}_\ell(v \cdot (G - \mu_T))]} \geq 1 - \frac{1}{2} = \frac{1}{2}.$$

Since S_G is representative, by Definition 8

$$\Pr\left[\mathrm{He}_\ell(v \cdot (S_G - \mu_T)) \geq \theta^\ell - \sqrt{2}\max(\ell, \theta)^{(\ell-1)}\right] \geq \frac{1}{2} - \frac{\alpha^3}{100} \geq \frac{49}{100}.$$

Since T is α-good, due to Definition 10, $|S_G \cap T| \,/\, |S_G| \geq 1 - \frac{\alpha}{6} \geq \frac{80}{100}$, we have that

$$\Pr\left[\mathrm{He}_\ell(v \cdot (S_G \cap T - \mu_T)) \geq \theta^\ell - \sqrt{2}\max(\ell, \theta)^{(\ell-1)}\right] \geq \frac{49}{100} - \frac{20}{100} \geq \frac{1}{4}.$$

On the other hand, due to Eq. (B.2), applying Markov's inequality gives that for any k-sparse unit vector v,

$$\Pr\left[\mathrm{He}_\ell(v \cdot (S_G \cap T - \mu_T)) \geq \sqrt{\frac{4\lambda^*_{\text{sparse}} \cdot \ell!}{\alpha}}\right] \leq \frac{\mathbb{E}[\mathrm{He}_\ell(v \cdot (S_G \cap T - \mu_T))^2]}{\left(\sqrt{\frac{4\lambda^*_{\text{sparse}} \cdot \ell!}{\alpha}}\right)^2}$$
$$\leq \frac{\lambda^*_{\text{sparse}} \cdot \ell!/\alpha}{4\lambda^*_{\text{sparse}} \cdot \ell!/\alpha} = \frac{1}{4}. \tag{B.3}$$

Recall that $\theta = v \cdot (\mu - \mu_T)$. From Eq. (B.1) and (B.3), we have that for any k-sparse unit vector $v \in \mathbb{R}^d$,

$$(v \cdot (\mu - \mu_T))^\ell - \sqrt{2}\max(\ell, v \cdot (\mu - \mu_T))^{(\ell-1)} \leq \sqrt{\frac{4\lambda^*_{\text{sparse}} \cdot \ell!}{\alpha}}.$$

Note that $\theta^\ell \leq \sqrt{2}\max(\ell, \theta)^{(\ell-1)}$ only when $\theta \leq 2\ell$, and so we have that for any k-sparse unit vector $v \in \mathbb{R}^d$,

$$v \cdot (\mu - \mu_T) \leq 2\ell + \left(\frac{4\lambda^*_{\text{sparse}} \cdot \ell!}{\alpha}\right)^{\frac{1}{2\ell}}$$
$$= O\Big(2\ell + \Big(\frac{4(C_1 \cdot \big[(\ell + C_1 \log\frac{1}{\alpha}) \cdot \log^2(2 + \log\frac{1}{\alpha})\big]^{2\ell}) \cdot \ell!}{\alpha}\Big)^{\frac{1}{2\ell}}\Big)$$
$$= O\Big(\alpha^{-\frac{1}{2\ell}} \cdot \sqrt{\ell}\Big(\ell + \log\frac{1}{\alpha}\Big) \cdot \log^2\Big(2 + \log\frac{1}{\alpha}\Big)\Big).$$

By choosing $v = \mathrm{trim}_k(\mu - \mu_T)$ and combining the above with Lemma 39, we complete the proof. □

B.2 Analysis of BASICMF

Recall the notations in BASICMF (Algorithm 4): $R = (C_1 \cdot \log(\frac{1}{\alpha}))^{l/2}, \gamma = C_0 \cdot \sqrt{\ell \cdot \log(\frac{\ell d}{\alpha \tau})}$, and the length of the interval $[a, b]$, i.e. $b - a$, equals $C_1 \cdot R \cdot \log(2 + \log\frac{1}{\alpha})$. We will need a series of results to prove Theorem 18. First, we note that if BASICMF returns at Step 2, then T must not be α-good in view of Lemma 11. Thus we only need to consider the remaining steps. In particular, we divide the output of BASICMF into three cases:

- CASE 1: it returns TBD at Step 5.
- CASE 2: it returns one subset $\{(T', \alpha')\}$ at Step 9.
- CASE 3: it returns two subsets $\{(T_1, \alpha_1), (T_2, \alpha_2)\}$ at Step 14.

We analyze the performance for each case in the following.

B.2.1 Analysis of Case 1

Proposition 23. *Consider Algorithm 4. If it returns* TBD *and if T is an α-good set, then* $|\mathbb{E}[p(G)] - \mathbb{E}[p(T)]| \le O\big((\ell + \log\frac{1}{\alpha})^{\frac{l}{2}} \cdot \log(2 + \log\frac{1}{\alpha})\big)$.

Proof. We first argue that most of the good samples in T have $p(x)$ value close to $\mathbb{E}[p(G)]$.

Claim 24. If T is α-good, then the samples $x \in T \cap S_G$ that satisfy $|p(x) - \mathbb{E}[p(G)]| < R$ constitute at least an $\left(\alpha - \frac{\alpha^3}{100}\right)$-fraction of T and an $(1 - \frac{\alpha}{6} - \frac{\alpha^3}{100})$-fraction of S_G.

Next, we claim that if there exists an appropriate interval $[a, b]$ in Step 3, then the mean of $p(G)$ is in the interval $[a - R, b + R]$.

Claim 25. If T is α-good, and the interval $[a, b]$ contains at least $(1 - \frac{\alpha}{2})$-fraction of values of $p(x)$ for $x \in T$, then $\mathbb{E}[p(G)] \in [a - R, b + R]$.

Now by construction, if Algorithm 4 returns TBD, then

$$\mathrm{Var}[p(T)] \le C_1 \cdot \left(\ell + C_1 \log\frac{1}{\alpha}\right)^l \cdot \log^2\left(2 + \log\frac{1}{\alpha}\right). \tag{B.4}$$

On the other hand, the interval $[a, b]$ contains at least $(1 - \frac{\alpha}{2})$ fraction of values of $p(x)$ for $x \in T$. Therefore, the contribution of the samples in $[a, b]$ to the variance gives

$$\mathrm{Var}[p(T)] \ge \left(1 - \frac{\alpha}{2}\right) \cdot \max\left\{0, \left|\mathbb{E}[p(T)] - \frac{a+b}{2}\right| - \frac{b-a}{2}\right\}^2. \tag{B.5}$$

To see this, note that $\frac{a+b}{2}$ is the midpoint and $\frac{b-a}{2}$ is the length of interval $[a, b]$. When $\mathbb{E}[p(T)]$ is inside the interval, $\left|\mathbb{E}[p(T)] - \frac{a+b}{2}\right| - \frac{b-a}{2} < 0$ and the variance is lowered bounded by 0. Otherwise, when $\mathbb{E}[p(T)]$ is outside the interval, the distance from any sample in $[a, b]$ to $\mathbb{E}[p(T)]$ is at least $\left|\mathbb{E}[p(T)] - \frac{a+b}{2}\right| - \frac{b-a}{2} \ge 0$.

Moreover, since $b - a \le O((\log(1/\alpha))^{l/2} \cdot \log(2 + \log(1/\alpha)))$,

$$|\mathbb{E}[p(T)] - (a+b)/2| \le \frac{b-a}{2} + \sqrt{\mathrm{Var}[p(T)]} = O((\ell + C\log(1/\alpha))^{l/2} \log(2 + \log(1/\alpha))). \tag{B.6}$$

From the Claim 25, we also have

$$|\mathbb{E}[p(G) - (a+b)/2]| \le \frac{b-a}{2} + R = O((\ell + C\log(1/\alpha))^{l/2} \log(2 + \log(1/\alpha))). \tag{B.7}$$

By the triangle inequality, we have that $|\mathbb{E}[p(G)] - \mathbb{E}[p(T)]| = O((\ell + C\log(1/\alpha))^{l/2} \log(2 + \log(1/\alpha)))$. □

Proof of Claim 24. Since T is α-good, and $\mathrm{Var}[p(G)] \le 1$. By degree-l Chernoff bound (Lemma 40) and definition of representative set (Definition 8), for $R = (C_1 \cdot \log(1/\alpha))^{l/2}$

$$\begin{aligned}\Pr[|p(S_G) - \mathbb{E}[p(G)]| \ge R] &\le e^{-\Omega(R^{2/l})} + \frac{\alpha^3}{100k^{2\ell} \cdot \log^l(\frac{\ell d}{\alpha\tau})} \\ &\le e^{-C \cdot \log(1/\alpha)} + \frac{\alpha^3}{100k^{2\ell} \cdot \log^l(\frac{\ell d}{\alpha\tau})} \\ &= \alpha^C + \frac{\alpha^3}{100k^{2\ell} \cdot \log^l(\frac{\ell d}{\alpha\tau})} \le \frac{\alpha^3}{100},\end{aligned}$$

for large enough constant $C > 0$. □

Proof of Claim 25. From Claim 24, at least an $(\alpha - \alpha^3/100)$-fraction of T is R-close to $\mathbb{E}[p(G)]$. Also we know that at most an $\frac{\alpha}{2}$-fraction of T are not in $[a,b]$ by the definition of the interval $[a,b]$. Then, there must be at least

$$\left(\alpha - \frac{\alpha^3}{100}\right) - \frac{\alpha}{2} = \frac{\alpha}{2} - \frac{\alpha^3}{100} = \frac{\alpha}{2}\left(1 - \frac{\alpha^2}{50}\right) > 0$$

fraction of samples in T that are in $[a,b]$ and R close to $\mathbb{E}[p(G)]$. Therefore, $\mathbb{E}[p(G)]$ must be in $[a-R, b+R]$. □

B.2.2 Analysis of CASE 2

Lemma 26. *Consider Algorithm 4. If it reaches Step 7, there must exist a threshold $t > 2R$ satisfying the inequality thereof.*

Proof. We will prove this lemma by contradiction. Assume that Algorithm 4 reaches Step 7, but for all $t > 2R$, we have

$$\Pr[\min\{|p(T) - a|, |p(T) - b|\} \geq t] \leq \frac{32}{\alpha}\exp(-(t-2R)^{2/l}) + \frac{2\alpha^2}{k^{2\ell}\log^l(\frac{\ell d}{\alpha\tau})}.$$

By change of variables, we have that for any $t > 2R + \frac{b-a}{2}$,

$$\Pr\left[\left|p(T) - \frac{a+b}{2}\right| \geq t\right] \leq \frac{32}{\alpha}e^{-(t-2R-\frac{b-a}{2})^{2/l}} + \frac{2\alpha^2}{k^{2\ell}\log^l(\frac{\ell d}{\alpha\tau})}.$$

Note that this inequality only holds non-trivially when $t \geq t_0$ where $t_0 = 2R + \frac{b-a}{2} + (\log\frac{32}{\alpha})^{l/2}$; namely, if $t < t_0$, the right-hand side is at least 1.

By Lemma 13, we have $\max_{x,y\in T}|p(x) - p(y)| \leq 2k^\ell \cdot \gamma^l$, where $\gamma = C_0 \cdot \sqrt{\ell \cdot \log(\frac{\ell d}{\alpha\tau})}$. Also note that the size of the interval $[a,b]$ equals $C_1 \cdot R \cdot \log(2 + \log\frac{1}{\alpha})$ which is less than $k^\ell \cdot \gamma^l$. Therefore,

$$\max_{x\in T}\left|p(x) - \frac{a+b}{2}\right| \leq 3k^\ell \cdot \gamma^l. \tag{B.8}$$

Then, we have that

$$\begin{aligned}
\mathrm{Var}[p(T)] &\leq \mathbb{E}\left[\left(p(T) - \frac{a+b}{2}\right)^2\right] \\
&= \int_0^\infty \Pr\left[(p(T) - \frac{a+b}{2})^2 \geq t^2\right]dt^2 \\
&\stackrel{\zeta_1}{=} 2\int_0^{3k^\ell\cdot\gamma^l} \Pr\left[\left|p(T) - \frac{a+b}{2}\right| \geq t\right] tdt \\
&= 2\int_0^{t_0} \Pr\left[\left|p(T) - \frac{a+b}{2}\right| \geq t\right] tdt + 2\int_{t_0}^{3k^\ell\cdot\gamma^l} \Pr\left[\left|p(T) - \frac{a+b}{2}\right| \geq t\right] tdt \\
&\leq t_0^2 + 2\int_{t_0}^{3k^\ell\cdot\gamma^l} \Big(\frac{32}{\alpha}e^{-(t-2R-\frac{b-a}{2})^{2/l}} + \frac{2\alpha^2}{k^{2\ell}\cdot\log^l(\frac{\ell d}{\alpha\tau})}\Big)tdt \\
&= t_0^2 + \frac{2\alpha^2}{k^{2\ell}\cdot\log^l(\frac{\ell d}{\alpha\tau})}\cdot 9k^{2\ell}\cdot\gamma^{2l} + \frac{32}{\alpha}\int_{(\log\frac{32}{\alpha})^{l/2}}^\infty e^{-t^{2/l}}\cdot(2t + 4R + b - a)dt \\
&= t_0^2 + 18C_0^2\cdot\alpha^2\cdot\ell^l + \frac{32}{\alpha}\int_{\log\frac{32}{\alpha}}^\infty e^{-u}\cdot(2u^{l/2} + 4R + b - a)\cdot\frac{l}{2}\cdot u^{\frac{l}{2}-1}du \\
&\stackrel{\zeta_2}{\leq} O\left((\log(1/\alpha))^l\cdot\log^2(2 + \log(1/\alpha))\right) + O\left(2\alpha^2\cdot\ell^l\right) \\
&\qquad + O((l + \log 1/\alpha)^l\cdot\log(2 + \log 1/\alpha)) \\
&\leq O\left((\ell + \log(1/\alpha))^l\cdot\log^2(2 + \log(1/\alpha))\right),
\end{aligned}$$

where ζ_1 holds in view of (B.8), and where ζ_2 follows since

$$
\begin{aligned}
&\frac{32}{\alpha}\int_{\log(\frac{32}{\alpha})}^{\infty} e^{-u}\cdot(2u^{\frac{l}{2}}+4R+b-a)\cdot\frac{l}{2}\cdot u^{\frac{l}{2}-1}du\\
&=\frac{32}{\alpha}\int_{\log(\frac{32}{\alpha})}^{\infty} e^{-u}\cdot 2u^{l-1}\cdot\frac{l}{2}du+\frac{32}{\alpha}\int_{\log(\frac{32}{\alpha})}^{\infty} e^{-u}(4R+b-a)\cdot\frac{l}{2}\cdot u^{\frac{l}{2}-1}du\\
&=\frac{32}{\alpha}\cdot 2\cdot\frac{l}{2}\int_{\log(\frac{32}{\alpha})}^{\infty} e^{-u}\cdot u^{l-1}du+\frac{32}{\alpha}\cdot(4R+b-a)\cdot\frac{l}{2}\int_{\log(\frac{32}{\alpha})}^{\infty} e^{-u}\cdot u^{\frac{l}{2}-1}du\\
&\overset{\zeta_3}{\leq}\frac{32}{\alpha}\cdot 2\cdot\frac{l}{2}\cdot e^{-\log\frac{32}{\alpha}}\cdot\Big(\log\frac{32}{\alpha}+l\Big)^{l-1}+\frac{32}{\alpha}\cdot(4R+b-a)\cdot\frac{l}{2}\cdot e^{-\log\frac{32}{\alpha}}\cdot\Big(\log\frac{32}{\alpha}+\frac{l}{2}\Big)^{\frac{l}{2}-1}\\
&\leq\Big(\log\frac{32}{\alpha}+l\Big)^{l}+(4R+b-a)\cdot\Big(\log\frac{32}{\alpha}+\frac{l}{2}\Big)^{\frac{l}{2}}\\
&\leq\Big(\log\frac{32}{\alpha}+l\Big)^{l}+\Big(4\big(C_1\cdot\log\frac{1}{\alpha}\big)^{l/2}+C_1\cdot R\cdot\log\big(2+\log\frac{1}{\alpha}\big)\Big)\cdot\Big(\log\frac{32}{\alpha}+\frac{l}{2}\Big)^{\frac{l}{2}}\\
&=O\Big(\Big(l+\log\frac{1}{\alpha}\Big)^{l}\cdot\log\Big(2+\log\frac{1}{\alpha}\Big)\Big),
\end{aligned}
$$

where ζ_3 is due to the incomplete gamma function (see Claim 3.11 of [DKS18b]), i.e. $\int_x^\infty e^{-t}\cdot t^{s-1}dt\leq e^{-x}(x+s)^{s-1}$, for $s\geq 1, x\geq 0$.

In other words, had we not found an appropriate threshold $t>2R$ at Step 7, Algorithm 4 would have returned at Step 5, which is a contradiction. This completes the proof. □

Once we have verified the existence of such threshold t, it is easy to see that the resultant T' is a subset of T, and $\alpha'\geq\alpha$ by algebraic calculation. This has been already shown in [DKS18b].

Lemma 27 (Lemma 3.13 of [DKS18b])**.** *Consider Algorithm 4. If it reaches Step 9, then the output* $\{(T',\alpha')\}$ *is such that* $T'\subset T$ *and* $\alpha'>\alpha$.

Next, we show that BASICMF sanitizes the sample set, i.e. it removes more corrupted samples than the uncorrupted ones.

Lemma 28. *Consider Algorithm 4. If it reaches Step 9, and if T is α-good and* $\mathrm{Var}[p(G)]\leq 1$, *then the output* $\{(T',\alpha')\}$ *is such that T' is α'-good.*

Proof. Due to Algorithm 1, the ℓ_∞-distance among all pairs of the samples are bounded. It remains to show $|S_G\cap T'|\,/\,|T'|\geq\alpha'$ and $|S_G\cap T'|\,/\,|S_G|\geq 1-\alpha'/6$.

We claim that for any $t>2R$, the following holds:

$$\Pr[\min\{|p(S_G)-a|\,,|p(S_G)-b|\}\geq t]\leq 2e^{-(t-R)^{2/l}}+\frac{\alpha^3}{50k^{2\ell}\cdot\log^l(\frac{\ell d}{\alpha\tau})}. \tag{B.9}$$

To see the rationale, we note that by Claim 25, we have $\mathbb{E}[p(G)]\in[a-R,b+R]$. Since $\mathbb{E}[p(G)]-R\leq b$, we have

$$
\begin{aligned}
\Pr[p(S_G)-b\geq t]&\leq\Pr[p(S_G)-(\mathbb{E}[p(G)]-R)\geq t]\\
&=\Pr[p(S_G)-\mathbb{E}[p(G)]\geq t-R]\\
&\leq\Pr[p(G)-E[p(G)]\geq t-R]+\frac{\alpha^3}{100k^{2\ell}\cdot\log^l(\frac{\ell d}{\alpha\tau})}\\
&\leq e^{-(t-R)^{2/l}}+\frac{\alpha^3}{100k^{2\ell}\cdot\log^l(\frac{\ell d}{\alpha\tau})},
\end{aligned}
$$

where in the third step, we used the fact that S_G is representative (see Definition 8), and in the last step we applied Lemma 40.

The inequality (B.9) follows since $\min\{|p(S_G)-a|\,,|p(S_G)-b|\}\geq t$ is a subevent of $|p(S_G)-b|>t$.

Since T is α-good, we know that a $1-\frac{\alpha}{6} \geq \frac{1}{2}$ fraction of the samples in S_G is in $S_G \cap T$. Therefore,

$$\Pr[\min\{|p(S_G \cap T) - a|, |p(S_G \cap T) - b|\} \geq t] \leq 4e^{-(t-R)^{2/l}} + \frac{\alpha^3}{25k^{2\ell} \cdot \log^l(\frac{\ell d}{\alpha\tau})}. \quad \text{(B.10)}$$

Due to the inequality of Step 7 in Algorithm 4, we know that the above probability is at least $8/\alpha$ times larger for the samples in T. Therefore,

$$\begin{aligned}\frac{|S_G \cap T'|}{|T'|} &= \frac{|S_G \cap T'|}{|S_G \cap T|}\frac{|S_G \cap T|}{|T|}\frac{|T|}{|T'|} \\ &\geq \left(1 - \frac{\alpha}{8}\cdot\left(1-\frac{|T'|}{|T|}\right)\right)\cdot\alpha\cdot\frac{|T|}{|T'|} \\ &\geq \left(\left(1-\frac{\alpha}{8}\right)\cdot\frac{|T|}{|T'|}+\frac{\alpha}{8}\right)\cdot\alpha \\ &= \alpha',\end{aligned}$$

meaning that the remaining fraction of good samples in T' is at least α'.

On the other hand, since $|S_G \cap T|/|S_G| \geq 1-\alpha/6$ and $\left(1-\frac{\alpha}{8}\cdot\left(1-\frac{|T'|}{|T|}\right)\right)\alpha = \alpha'|T'|/|T|$, we have

$$\begin{aligned}\frac{|S_G \cap T'|}{|S_G|} &= \frac{|S_G \cap T'|}{|S_G \cap T|}\frac{|S_G \cap T|}{|S_G|} \\ &\geq \left(1-\frac{\alpha}{8}\cdot\left(1-\frac{|T'|}{|T|}\right)\right)\left(1-\frac{\alpha}{6}\right) \\ &= \left(1-\frac{\alpha}{8}\cdot\left(1-\frac{|T'|}{|T|}\right)\right) - \frac{\alpha'|T'|}{6|T|},\end{aligned}$$

thus,

$$\begin{aligned}\frac{|S_G \cap T'|}{|S_G|} - \left(1-\frac{\alpha'}{6}\right) &\geq 1-\frac{\alpha}{8}\left(1-\frac{|T'|}{T}\right)-\frac{\alpha'}{6}\frac{|T'|}{|T|}-\left(1-\frac{\alpha'}{6}\right) \\ &= \left(\frac{\alpha'}{6}-\frac{\alpha}{8}\right)\left(1-\frac{|T'|}{T}\right) > 0\end{aligned}$$

This proves that T' is α'-good. □

We summarize the performance of BASICMF in CASE 2 in the following proposition, which is an immediate combination of Lemma 26, Lemma 27, and Lemma 28.

Proposition 29. *Consider Algorithm 4. If it reaches Step 7, there must exist $t > 2R$ that satisfies the inequality of this step, and the algorithm will output $\{(T', \alpha')\}$ with $T' \subset T$ and $\alpha' \geq \alpha$. If, in addition, T is α-good and $\mathrm{Var}[p(G)] \leq 1$, then T' is α'-good.*

B.2.3 Analysis of CASE 3

Lemma 30 (Lemma 3.12 of [DKS18b])**.** *Consider Algorithm 4. If it reaches Step 12, there must exist a threshold t that satisfy the conditions thereof.*

Lemma 31 (Lemma 3.14 of [DKS18b])**.** *Consider Algorithm 4. If it reaches Step 12, then the output $\{(T_1, \alpha_1), (T_2, \alpha_2)\}$ is such that $T_1 \subset T$, $T_2 \subset T$, and $\frac{1}{\alpha_1^2}+\frac{1}{\alpha_2^2} \leq \frac{1}{\alpha^2}$.*

Lemma 32. *Consider Algorithm 4. If it reaches Step 12, and if T is α-good, then the output $\{(T_1, \alpha_1), (T_2, \alpha_2)\}$ is such that T_i is α_i-good for some $i \in \{1, 2\}$.*

Proof. Recall that Claim 24 lower bounds the fraction of the good samples (i.e. $x \in S_G \cap T$) that satisfy $|p(x) - \mathbb{E}[p(G)]| < R$. Since T_1 and T_2 overlap in an interval of length at least $2R$, the good samples must be contained in either one of both two clusters. We will show that the T_i with these good samples (in interval of length $2R$) is α_i-good.

Since T is α-good, we have $|S_G \cap T| / |T| \geq \alpha$ and $|S_G \cap T| / |S_G| \geq (1 - \alpha/6)$. We want to show that (i) $|S_G \cap T_i| / |T_i| \geq \alpha_i$ and (ii) $|S_G \cap T_i| / |S_G| \geq (1 - \alpha_i/6)$.

To show (i), note that $|S_G \cap T_i| \geq \left(\alpha - \frac{\alpha^3}{100}\right) |T|$ due to Claim 24. Thus,

$$\frac{|S_G \cap T_i|}{|T_i|} = \frac{|S_G \cap T_i|}{|T|} \cdot \frac{|T|}{|T_i|} \geq \left(\alpha - \frac{\alpha^3}{100}\right) \cdot \frac{|T|}{|T_i|} = \alpha_i,$$

where the last transition is by definition.

To show (ii), we only have to show that $\alpha_i/6 \geq \alpha/6 + \alpha^3/100$, i.e. $\alpha_i \geq \alpha + 3\alpha^3/50$. Note that $|T| - |T_i| \geq \frac{\alpha}{4} |T|, \forall i$. Thus, $|T| / |T_i| \geq \frac{1}{1-\alpha/4}$ and we can show that

$$\alpha_i \geq \alpha \cdot \frac{1 - \alpha^2/100}{1 - \alpha/4} \geq \alpha \cdot \frac{100 - \alpha}{100 - 25\alpha} \geq \alpha \left(1 + \frac{24\alpha}{100 - 25\alpha}\right) \geq \alpha \left(1 + \frac{3\alpha^3}{50}\right).$$

This completes the proof. □

Combining Lemma 30, Lemma 31, and Lemma 32, we immediately have the following.

Proposition 33. *Consider Algorithm 4. If it reaches Step 12, then there must exist a threshold t that satisfies the conditions in this step. Moreover, the output $\{(T_1, \alpha_1), (T_2, \alpha_2)\}$ is such that $T_1 \subset T$, $T_2 \subset T$, and $\frac{1}{\alpha_1^2} + \frac{1}{\alpha_2^2} \leq \frac{1}{\alpha^2}$. If, in addition, T is α-good, then T_i is α_i-good for some $i \in \{1, 2\}$.*

B.2.4 Proof of Theorem 18

Proof. Observe that now Theorem 18 is an immediate result by combining Proposition 23, Proposition 29, and Proposition 33. □

B.3 Analysis of HarmonicMF

B.3.1 Certifying the varaince of p on G

Proof of Lemma 19. The proof follows directly from Lemma 3.31 of [DKS18b]. □

B.3.2 Analysis for p_1

Proof of Lemma 16. First, if at any subroutine of HarmonicMF, it returns "NO" or a list of pairs $\{(T_i, \alpha_i)\}$, ANS $\neq$ TBD. If that is not the case, it means HarmonicMF reaches Step 6 and BasicMF returns "TBD", then we have

$$\mathrm{Var}[p_1(T)/\beta] \leq C_1 \cdot \left(\ell + C_1 \log \frac{1}{\alpha}\right)^{\ell} \cdot \log^2 \left(2 + \log \frac{1}{\alpha}\right),$$

because p_1 is of degree ℓ. However, recall that the condition of Step 6 in Algorithm 3 is satisfied, thus

$$\begin{aligned}
\mathrm{Var}[p_1(T)] &= \mathrm{Var}[v^* \cdot P_{d,\ell}(T - \mu_T)] \geq \lambda^* \geq \lambda^*_{\text{sparse}} \\
&= \left[C_1 \cdot \left(\ell + C_1 \log \frac{1}{\alpha}\right) \cdot \log^2 \left(2 + \log \frac{1}{\alpha}\right)\right]^{2\ell} \\
&\geq \left(C_1 \cdot \left(1 + \log \frac{1}{\alpha}\right) \cdot \log^2 \left(2 + \log \frac{1}{\alpha}\right)\right)^{\ell} \cdot C_1 \cdot \left(\ell + C_1 \log \frac{1}{\alpha}\right)^{\ell} \cdot \log^2 \left(2 + \log \frac{1}{\alpha}\right) \\
&= \beta^2 \cdot C_1 \cdot \left(\ell + C_1 \log \frac{1}{\alpha}\right)^{\ell} \cdot \log^2 \left(2 + \log \frac{1}{\alpha}\right),
\end{aligned}$$

which induces a contradition. We conclude that BasicMF will not return "TBD" at Step 6, which completes the proof. □

B.3.3 Analysis for p_2

Proof of Lemma 17. We see that the lemma holds as long as HarmonicMF returns either "NO" or a list of (T_i, α_i) for p_2 correctly. First, we claim that $p_2(x)$ is harmonic such that MultilinearMF multifilters correctly at Step 4. To prove the claim, simply note that p_2 is of degree 2ℓ and consists of a set of $k^{2\ell}$ Hermite polynomials. In addition, $p_2(x)$ only applies on a set of $2\ell k^{2\ell}$ coordinates.

Based on the correctness of MULTILINEARMF, it remains to show that if every subroutine of HARMONICMF returns "TBD", then T must not be α-good. Consider that Algorithm 5 reaches Step 6, and BASICMF returns "TBD". Applying Lemma 19, we know that $\mathbb{E}[p_2(G)^2] \le \beta^2 = \big(C_1 \cdot (1+\log(\frac{1}{\alpha})) \cdot \log^2(2+\log(\frac{1}{\alpha}))\big)^{2\ell}$. Therefore, $\mathrm{Var}[\frac{1}{\beta} \cdot p_2(G)] \le \frac{1}{\beta^2} \cdot \mathbb{E}[p_2(G)^2] \le 1$ and thus satisfies the preconditions of BASICMF. Then, if BASICMF also returns "TBD", we can show that $\mathrm{Var}[\frac{1}{\beta} \cdot p_2(T)] = O\big((\ell+\log(\frac{1}{\alpha}))^{2\ell} \cdot \log^2(2+\log(\frac{1}{\alpha}))\big)$ according to Theorem 18. Thus,

$$\begin{aligned}\mathrm{Var}[p_2(T)] &\le \beta^2 \cdot O(\ell+\log(1/\alpha))^{2\ell}\log^2(2+\log(1/\alpha)) \\ &\le O((\ell+\log(1/\alpha))\log^2(2+\log(1/\alpha)))^{4\ell}.\end{aligned}$$

We then show by contradiction. Assume the above holds and T is α-good. Due to Theorem 18, we can show that

$$\begin{aligned}|\mathbb{E}[p_2(G)] - \mathbb{E}[p_2(T)]| &\le \beta \cdot O(\ell+\log(1/\alpha))^{\ell}\log(2+\log(1/\alpha)) \\ &\le O((\ell+\log(1/\alpha))\log^2(2+\log(1/\alpha)))^{2\ell}.\end{aligned}$$

Additionally, since

$$|\mathbb{E}[p_2(G)]| \le \sqrt{\mathbb{E}[p_2^2(G)]} \le \beta^2 = O((\ell+\log(1/\alpha))\log^2(2+\log(1/\alpha)))^{2\ell},$$

Therefore, by Cauchy-Schwarz inequality, we conclude that $|\mathbb{E}[p_2(T)]| \le O((\ell+\log(\frac{1}{\alpha}))\log^2(2+\log(\frac{1}{\alpha})))^{2\ell}$. However, by construction, we have

$$\begin{aligned}|\mathbb{E}[p_2(T)]| &= \mathbb{E}\left[\mathrm{Tr}\left(\frac{(\tilde{\Sigma})_U}{\left\|(\tilde{\Sigma})_U\right\|_F}\left(P_{d,\ell}(T-\mu_T)P_{d,\ell}(T-\mu_T)^\top\right)\right)\right] \\ &= \mathrm{Tr}\left((\tilde{\Sigma})_U\tilde{\Sigma}\right) = \left\|(\tilde{\Sigma})_U\right\|_F \overset{\zeta_4}{\ge} \lambda^*_{\text{sparse}} \\ &\ge C_1 \cdot ((\ell + C_1\log(1/\alpha))\log^2(2+\log(1/\alpha)))^{2\ell},\end{aligned}$$

where ζ_4 is due to the condition in Step 4 of Algorithm 3. This is a contradiction.

Hence, we conclude that T cannot be α-good and we remove it from the list. Moreover, if BASICMF returns NO or a list $\{(T_i, \alpha_i)\}$, the guarantees follow from Theorem 18. The proof is complete. □

Lemma 34 (Algorithm 5)**.** *Consider Algorithm 5 with input polynomial being p_1 or p_2 in view of Algorithm 3, and denote by* ANS *its output. With probability $1-\tau$, the following holds. If* ANS = NO, *then T is not α-good. If* ANS $= \{(T_i,\alpha_i)\}_{i=1}^m$ *for some $m \le 2$, then $T_i \subset T$ for all $i \in [m]$ and $\sum_{i=1}^m \frac{1}{\alpha_i^2} \le \frac{1}{\alpha^2}$; if additionally T is α-good, then at least one T_i is α_i-good.*

Proof. Inside any subroutine of BASICMF or MULTILINEARMF called by HARMONICMF, if ANS is assigned "NO" or a list of pairs $\{(T_i,\alpha_i)\}$, the guarantees are ensured by Lemma 19, Theorem 18 and Lemma 36. It remains to show the correctness of the algorithm returning "NO" at Step 6 when BASICMF returns TBD, which is implied by Lemma 17. □

B.4 Proof of Theorem 14

Proof. The theorem follows from Lemma 15, Lemma 16, Lemma 17, Theorem 18 and Lemma 34. □

C Proof of Theorem 1

Theorem 1 directly follows from the guarantees of our initial clustering step (Lemma 12), the main subroutine (Theorem 14), and the black-box list reduction algorithm (Proposition 37).

Proof of Theorem 1. Consider Algorithm 2. By Lemma 12, T will be divided into at most $\frac{1}{2\alpha}$ number of subsets, at least one of which is $\frac{\alpha}{2}$-good. Algorithm 2 then maintains a list $\mathcal{L}$ of pairs $\{(T_i, \alpha_i)\}$ on which Algorithm 3 is called repetitively until the list becomes empty. Theorem 14 implies that when Algorithm 3 is called on some $T_i \in \mathcal{L}$ which is α_i-good, if a list of pairs $\{(T_j, \alpha_j)\}$ is returned, then at least one of $\{T_j\}$ is α_j-good ($\alpha_j > \alpha_i$). This ensures that there always exists an $\frac{\alpha}{2}$-good subset T_i in list $\mathcal{L}$, except that a leaf node has been created for this branch and the empirical mean of an $\frac{\alpha}{2}$-good data set is returnd. We then argue that Algorithm 2 eventually returns an estimated mean at the branch that includes only $\frac{\alpha}{2}$-good subsets. Since the subsets are $\frac{\alpha}{2}$-good, ANS never equals to NO. In addition, the branch will not create child nodes forever: note that the true multifiltering step is in BASICMF, and both Step 8 and 12 reduce the subset size $|T_i|$ by at least 1; since α_i is non-decreasing, the algorithm cannot remove only inliers; by Definition 10, $|S_G \cap T_i| \geq (1 - \alpha_i/6)\,|S_G| \geq \frac{1}{2}\,|S_G|$. Therefore, the algorithm must return an estimated mean when there is no outliers to filter.

We then bound the list size of the returned list of estimated means. Since during the process of multifiltering, $\sum_i \alpha_i^{-2}$ is non-increasing, we have that $\sum_{i=1}^{|\mathcal{L}|} \alpha_i^{-2} \leq \frac{1}{2\alpha} \cdot \alpha^{-2}$ at any point of Algorithm 2. In addition, $\alpha_i \leq 1, \forall i$, meaning that the list size will never be larger than $O(\alpha^{-3})$. So does the size of M. Then, by applying LISTREDUCTION on M with $|M| \leq O(\alpha^{-3})$, the list size can be reduced to $O(\alpha^{-1})$ in view of Proposition 37.

Finally, note that CLUSTER runs in time $O\big(\mathrm{poly}(|T|, d)\big)$, ATTRIBUTE-EFFICIENT-MULTIFILTER runs in time $O\big(\mathrm{poly}(|T|, d^{\ell})\big)$ in view of Theorem 14, and LISTREDUCTION runs in time $O\big(\mathrm{poly}(|T|, d)\big)$. Moreover, there are at most $O(|T|/\alpha^3)$ number of calls to ATTRIBUTE-EFFICIENT-MULTIFILTER, and only one call to CLUSTER and one call to LISTREDUCTION, we conclude that the time complexity of Algorithm 2 is $O\big(\mathrm{poly}\big(|T|, d^{\ell}, \frac{1}{\alpha}\big)\big)$. □

D Omitted Algorithms

In the following, we present the omitted algorithms. In particular, MULTILINEARMF (Algorithms 6) is an important component of HARMONICMF, for which we tailor the algorithms in [DKS18b] to our sparse setting. MULTILINEARMF will further invoke DEGREE2HOMOGENEOUS (Algorithm 7). Algorithm 8, due to [DKS18b], is the black-box list reduction approach that was invoked in Algorithm 2.

D.1 MULTILINEARMF

We introduce useful facts about multilinear polynomial here. For $d, l \in \mathbb{N}$, a polynomial $p(x_1, \dots, x_l) : \mathbb{R}^{dl} \to \mathbb{R}$, where $x_i \in \mathbb{R}^d$, is called multilinear if it is linear in each of its l arguments, i.e. if holds that $p(a \cdot x_1 + b \cdot x_1', x_2, \dots, x_l) = a \cdot p(x_1, x_2, \dots, x_l) + b \cdot p(x_1, \dots, x_l)$, for all $a, b \in \mathbb{R}$ and $x_i, x_i' \in \mathbb{R}^d$, and similarly for all the other arguments. Moreover, a polynomial p is called symmetric if $p(x_1, \dots, x_l) = p(x_{\pi(1)}, \dots, x_{\pi(l)})$ for any permutation $\pi : [l] \to [l]$. Any degree-l multilinear polynomial $p : \mathbb{R}^{dl} \to \mathbb{R}$ can be expressed as $A(x_1, \dots, x_l)$ for an order-l tensor A over $\mathbb{R}^d$. Moreover, A is symmetric if p is symmetric.

Algorithm 6 MULTILINEARMF

Require: A multiset of samples $T \subset \mathbb{R}^d$, parameter $\alpha \in (0, 1/2]$, failure probability $\tau \in (0, 1)$, a degree-l multilinear polynomial $V(x_1, \dots, x_l)$ over $\mathbb{R}^{dl}$ with $\|V\|_2 \leq 1$, where V is the outer product of l number of ψ-sparse vectors.

1: If $l = 1$, run BASICMF on $V(x - \mu_T)$, and **return** its output.
2: Compute the quadratic polynomial $q(x) = \|V(x - \mu_T)\|_2^2$, where $x \in \mathbb{R}^d$ and Vx is an order-$(l-1)$ tensor with $(Vx)_{i_2, \dots, i_l} = \sum_{i_1} x_{i_1} V_{i_1, \dots, i_l}$.
3: Run DEGREE2HOMOGENEOUS on $q(x)$. If it returns NO or a list $\{T_i, \alpha_i\}$, then **return** the same result.
4: Sample a set Φ of $m = 200 \cdot \alpha^{-1} \log(4/\tau)$ instances uniformly at random from T.
5: $\forall x \in \Phi$, let $V_x = \frac{1}{\sqrt{q(x)}} \cdot V(x - \mu_T)$. ANS $\leftarrow$ MULTILINEARMF on $(T, Vx, l-1, \alpha, \tau/2)$. If it returns NO or a list $\{(T_i, \alpha_i)\}$, then **return** the same result.
6: Otherwise, **return** TBD.

Algorithm 7 DEGREE2HOMOGENEOUS(T, α, τ, A)

Require: A multiset of samples $T \subset \mathbb{R}^d$, parameter $\alpha \in (0, 1/2]$, failure probability $\tau \in (0, 1)$, homogeneous polynomial $x^\top Ax$, where A is a $d \times d$ matrix with $\|A\|_* \leq 1$.

Compute the k^2 largest eigenvalues λ_i and eigenvectors v_i of A.
for $i = 1, \ldots, k^2$ **do**
  $\text{ANS}_i \leftarrow$ BASICMF(T, α, τ, p) with $p(x) = v_i \cdot x$.
  if ANS_i is a list of $\{T_i, \alpha_i\}$ or ANS $=$ NO **then return** ANS_i.
end for
if all $\text{ANS}_i =$ TBD **then return** TBD.

The MULTILINEARMF works in the following way: Given degree-l multilinear polynomial $V(x_1 - \mu_T, \ldots, x_l - \mu_T)$, where V is an order-l symmetric tensor and x_i's are l number of independent variables. The goal is to show that the polynomial has small absolute expectation over l number of i.i.d. draws from $G \sim N(\mu, \mathbb{I}_d)$ if the algorithm does not filter any samples and returns "TBD"; and otherwise, the algorithm multifilters correctly. Since all subroutine of MULTILINEARMF multifilter the data set by calling BASICMF on linear polynomials, we know that if it returns "NO" or a list of pairs $\{(T_i, \alpha_i)\}$, the correctness is guaranteed by Theorem 18. It remains to bound the expected value of the multilinear polynomial when all subroutines return "TBD" and T is α-good.

The idea is to sub-sample a large enough sample Φ from T. If T is α-good, then with sufficiently high probability, $\exists x \in \Phi$ that is from G. By recursively doing this, with sufficiently high probability, we construct a multilinear polynomial $V(G_1 - \mu_T, G_2 - \mu_T, \ldots, G_l - \mu_T)$, the expectation of which is what we concerned about. The upper bound is then shown by induction. When the polynomial is linear, $|\mathbb{E}[V_{x^{l-1}}(G - \mu_T)]| \leq O(\sqrt{1 + \log(1/\alpha)} \cdot \log(2 + \log(1/\alpha))$. Here, we use V_{x^i} to denote taking i times of inner product between tensor V and a vector x. Note that x can be different in each time of the inner product. Then, without loss of generality, assume that for order-$(l-1)$ tensor V_x, $|\mathbb{E}[V_x(G_1 - \mu_T, \ldots, G_{l-1} - \mu_T)]| \leq f(l-1, \alpha)$, we can show that $|\mathbb{E}[V(G_1 - \mu_T, \ldots, G_l - \mu_T)]| \leq f(l, \alpha)$, provided that S_G is sufficiently epresentative with respect to G on any linear polynomial $V_{x^{l-1}}(x - \mu_T)$ and any quadratic polynomial $q(x) = \|V_{x^i}(x - \mu_T)\|_2^2, \forall i \in [l]$. In this analysis, the only difference between our setting and that of [DKS18b] is the definition of representative set S_G (Definition 8). Fortunately, since all polynomials in our algorithm apply to at most $\psi = 2\ell k^{2\ell}$ coordinates, the linear polynomials must be in $\mathbb{P}(\mathbb{R}^d, 1, 2\ell k^{2\ell}, 2\ell k^{2\ell})$, and the quadratic polynomials must be in $\mathbb{P}(\mathbb{R}^d, 2, 4\ell^2 k^{4\ell}, 2\ell k^{2\ell})$. Therefore, our definition of representative set sufficies. The proof follows the same pipeline as that of Lemma 3.27 in [DKS18b]. As a result, it can be shown that $f(l, \alpha) = f(l-1, \alpha) \cdot O(\sqrt{1 + \log(1/\alpha)} \cdot \log(2 + \log(1/\alpha))$, which renders $|\mathbb{E}[V(G_1 - \mu_T, \ldots, G_l - \mu_T)]| \leq O\big((1 + \log(1/\alpha))^{l/2} \cdot \log^l(2 + \log(1/\alpha)\big)$.

Definition 35 (Multifilter condition)**.** We say that a list of pairs $\{(T_i, \alpha_i)\}$, where $T_i \subset T$ and $\alpha_i \in (0, 1)$, satisfies the multifilter condition for (T, α) if the following hold:

1. $\sum_i \frac{1}{\alpha_i^2} \leq \frac{1}{\alpha^2}$, and
2. If T is α-good, then at least one T_i is α_i-good.

Lemma 36 (MULTILINEARMF, Lemma 3.27 of [DKS18b])**.** *Given $\alpha \in (0, \frac{1}{2}]$ and $\tau \in (0, 1)$, let T be the input sample set, and a degree-l multilinear polynomial $V(x_1, \ldots, x_l)$ over $\mathbb{R}^{dl}$ with $\|V\|_2 = 1$. Algorithm 6 returns one of the following with guarantees: (1)* TBD*, and we have that, if T is α-good, then with probability $1 - \tau$, $|\mathbb{E}[V(G_1 - \mu_T, \ldots, G_l - \mu_T)]| = O\big((1 + \log(\frac{1}{\alpha}) \log^2(2 + \log(\frac{1}{\alpha})))\big)^{l/2}$, where G_i are independent copies of G. (2)* NO*, then T is not α-good. (3) A list of pairs $\{(T_i, \alpha_i)\}$, $T_i \subset T$, satisfying the multifilter condition for (T, α).*

D.2 LISTREDUCTION

Proposition 37 (LISTREDUCTION, Proposition B.1 of [DKS18b])**.** *Fix $\alpha, \beta, \delta, t > 0$ and let $\mu^* \in \mathbb{R}^d$ be finite, and let $S \subseteq T$ be so that (i) $|S| / |T| \geq \alpha$, and (ii) for all unit vectors $v \in \mathbb{R}^d$, we have $\Pr[v \cdot (S - \mu^*) > t] < \delta$. Then, given $M = \{\mu_1, \ldots, \mu_n\} \subset \mathbb{R}^d$ so that $\delta n = o(1)$ and there*

Algorithm 8 LISTREDUCTION(T, α, ℓ, M)

Require: A multiset of samples $T \subset \mathbb{R}^d$, parameter $\alpha \in (0, 1/2]$, degree $\ell \geq 1$, a list $M \subset \mathbb{R}^d$.

1: $\beta \leftarrow C_4 \cdot \alpha^{-\frac{1}{2\ell}} \sqrt{\ell}(\ell + \log \frac{1}{\alpha})$, $\delta \leftarrow \frac{1}{C_5 \log \frac{1}{\alpha}}$, $t \leftarrow \sqrt{\log(C_5 \log \frac{1}{\alpha})}$, $n \leftarrow |M|$.

2: For all $\mu_i, \mu_j \in M$, let v_{ij} denote the unit vector in the $\mu_i - \mu_j$ direction.

3: Let $T_i = \cap_{j \neq i} \{x \in T : |v_{ij} \cdot (x - \mu_i)| < \beta + t\}$.

4: $M' \leftarrow \emptyset$.

5: $\forall i \in [n]$, if $|T_i| \geq \alpha(1 - \delta n)\, |T|$, and $\nexists \mu_j \in M'$ such that $\|\mu_i - \mu_j\|_2 < 2(\beta + t)$, then $M' \leftarrow M' \cup \mu_i$.

6: **return** M'.

is some i so that $\|\mu_i - \mu^*\|_2 \leq \beta$ for some $\mu_i \in M$, Algorithm 8 outputs $M' \subseteq M$ so that $|M'| \leq \frac{1}{\alpha}(1 + O(\delta n))$ and $\|\mu' - \mu^*\|_2 \leq 3(\beta + t)$ for some $\mu' \in M'$.

Remark 38. Under the setting of Algorithm 8, we have that $n = O(\alpha^{-3})$. This combined with the parameter settings in LISTREDUCTION shows that the size of M' is $O(1/\alpha)$ and there is at least one $\mu_i \in M'$ that has comparable error guarantee to those in M.

E Useful Lemmas

Lemma 39 (Lemma 3.2 of [CDK+21])**.** *Fix two vectors x, y with $\|x\|_0 \leq k$ and $\|\mathrm{trim}_k(x - y)\|_2 \leq \delta$. We have that $\|x - \mathrm{trim}_k(y)\|_2 \leq \sqrt{5}\delta$.*

Lemma 40 (degree-l Chernoff bound, Fact 2.8 of [DKS18b])**.** *Let $G \sim N(\mu, \mathbb{I}_d)$, $\mu \in \mathbb{R}^d$. Let $p : \mathbb{R}^d \to \mathbb{R}$ be a degree-l polynomial. For any $t > 0$, we have that $\Pr\big[\, |p(G) - \mathbb{E}[p(G)]| \geq t \cdot \sqrt{\mathrm{Var}[p(G)]}\big] \leq \exp\big(-\Omega(t^{2/l})\big)$.*

Lemma 41 (Harmonic and multilinear polynomials, Lemma 3.24 of [DKS18b])**.** *Let $X, X_{(1)}, \ldots, X_{(\ell)}$ be i.i.d random variables distributed as $N(\mu, \mathbb{I})$ for some $\mu \in \mathbb{R}^d$. Then, for any symmetric matrix A, we have*

$$\sqrt{\ell!} \cdot \mathbb{E}[h_A(X)] = \mathrm{Hom}_A(\mu) = \mathbb{E}[A(X_{(1)}, \ldots, X_{(\ell)})],$$

and

$$\mathbb{E}[h_A(X)^2] = \sum_{\ell'=0}^{\ell} \left(\binom{\ell}{\ell - \ell'} / \ell'! \right) \cdot \mathrm{Hom}_{B^{(\ell')}}(\mu)$$

where $B^{(\ell')}$ is the order-$2\ell'$ tensor with

$$B^{(\ell')}_{i_1, \ldots, i_{\ell'}, j_1, \ldots, j_{\ell'}} = \sum_{k_{\ell'+1}, \ldots, k_\ell} A_{i_1 \ldots, i_{\ell'}, k_{\ell'+1}, \ldots, k_\ell} A_{j_1 \ldots, j_{\ell'}, k_{\ell'+1}, \ldots, k_\ell}.$$