Support Recovery in Mixture Models With Sparse Parameters

Arya Mazumdar, *Senior Member, IEEE*, and Soumyabrata Pal

***Abstract*— Mixture models are widely used to fit complex and multimodal datasets. In this paper we study mixtures with high dimensional sparse parameter vectors and consider the problem of support recovery of those vectors. While parameter learning in mixture models is well-studied, the sparsity constraint remains relatively unexplored. Sparsity of parameter vectors is a natural assumption in high dimensional settings, and support recovery is a major step towards parameter estimation. We provide efficient algorithms for support recovery that have a logarithmic sample complexity dependence on the dimensionality of the latent space, and also poly-logarithmic dependence on sparsity. Our algorithms, applicable to mixtures of many different canonical distributions including high dimensional Uniform, Poisson, Laplace, Gaussians, etc., are based on the *method of moments*. In most of these settings, our results are the first guarantees on the problem while in the rest, our results provide improvements on or are competitive with existing works.**

***Index Terms*— Mixture models, sparse approximation, method of moments.**

I. Introduction

MIXTURE models are standard tools for probabilistic modeling of heterogeneous data, and have been studied theoretically for more than a century. Mixtures are used in practice for modeling data across different fields, such as, astronomy, genetics, medicine, psychiatry, economics, and marketing among many others [2]. Mixtures with finite number of components are especially successful in modeling datasets having a group structure, or presence of a subpopulation within the overall population. Often, mixtures can handle situations where a single parametric family cannot provide a satisfactory model for local variations in the observed data [3].

The literature on algorithmically learning mixture distributions is quite vast and comes in different flavors. Computational and statistical aspects of learning mixtures perhaps starts with [4], and since then have been the subject of intense investigation in both computer science and statistics [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. A large portion of this literature is devoted to *density estimation* or PAC-learning, where the goal is simply to find a distribution that is close in some distance (e.g., TV distance) to the data-generating mechanism. The results on density estimation can be further subdivided into *proper* and *improper learning* approaches depending on whether the algorithm outputs a distribution from the given mixture family or not. These two guarantees turn out to be quite different.

A significant part of the literature on the other hand is devoted to *parameter estimation*, where the goal is to identify the mixing weights and the parameters of each component from samples. Apart from Gaussian mixtures, where all types of results exist, prior work for other mixture families largely focuses on density estimation, and very little is known for parameter estimation outside of Gaussian mixture models. In this paper, our focus is to facilitate parameter estimation in Gaussian mixtures and beyond. We consider the setting where the parameters of the mixture are themselves high dimensional, but sparse (i.e., have few nonzero entries). Sparsity is a natural regularizer in high dimensional parameter estimation problems and have been considered in the context of mixtures in [17], [18], and [19], where it is assumed only few dimensions of the component means are relevant for de-mixing. In this paper we consider a slightly different model where we assume the means themselves are sparse. The former problem can be reduced to our setting if one of the component means is known. We, in particular, focus on only recovering the support of the vectors.

An interesting application of learning mixtures with sparse parameters is in high-dimensional clustering problems where cluster centers actually belong to a low-dimensional space. This is similar in spirit with sparse-PCA [20]; our objective is to identify a few important input features, so one can easily interpret its meaning. Our techniques can also be seen as a novel method for feature selection that can significantly speed up a learning algorithm.

Another practical application comes up naturally in recommendation systems where multiple users rate or evaluate items. Since users can often be heterogenous with a wide variety of distinct tastes, it is important that recommendations are personalized. However, the set of users can be partitioned (see for example [21]) into significantly large clusters where users in the same cluster have relatively similar preferences. Note that the identity of each unknown cluster can be modeled by an unknown parameter vector. It makes sense for the unknown vectors to be sparse, because most users have an affinity towards a few particular features of items among many possible. Sparse mixtures were motivated with such an application in the query based setting in [22] and [23].

Received 7 November 2023; revised 20 June 2024; accepted 8 September 2024. Date of publication 18 September 2024; date of current version 22 January 2025. This work was supported in part by the National Science Foundation under Award 2217058, Award 2112665, and Award 2133484. An earlier version of this paper was presented at the 2022 International Conference on Artificial Intelligence and Statistics (AISTATS) [1]. *(Corresponding author: Soumyabrata Pal.)*

Arya Mazumdar is with the Halıcıoğlu Data Science Institute (HDSI), University of California at San Diego, La Jolla, CA 92093 USA (e-mail: arya@ucsd.edu).

Soumyabrata Pal is with Adobe Research, Bengaluru 560045, India (e-mail: soumyabratap@adobe.com).

Communicated by S. Jalali, Associate Editor for Machine Learning and Statistics.

Digital Object Identifier 10.1109/TIT.2024.3462937

Note that, support recovery is an effective way to reduce the dimension of the ambient space, and therefore can be considered as a key step towards parameter estimation. We provide two flavors of results for support recovery namely, 1) *Exact support recovery:* where we recover the supports of all unknown sparse parameters corresponding to all components of the mixture, 2) *Maximal support recovery:* where we recover the maximal supports from the poset of all supports of the parameter vectors (i.e., supports that are not subsets of any other).

A. Discussion on Our Results and Other Related Works

Note that the sample complexity guarantees that we present in this paper for different notions of support recovery in high dimensional mixtures of distributions scale poly-logarithmically with the ambient dimension n.

Our technique of learning the supports of the latent parameter vectors in mixture of simple distributions is based on the *method of moments* [16], [24]. This method works in general, as long as moments of the distribution of each coordinate can be described as a polynomial in the component parameters. It was shown in [7] (see Table II in [7]) that most common distributions, including Gaussian, Uniform, Poisson, and Laplace distributions, satisfy this assumption. Our results include sample complexity guarantees for both exact support recovery (see Theorem 1) and maximal support recovery (see Theorem 2), and are not only applicable to many canonical distributions but also makes progress towards quantifying the sufficient number of moments in the general problem defined in Sec. II-B.

An alternate approach to the support recovery problem is to first recover the *union* of supports of the unknown parameters and then apply known parameter estimation guarantees to identify the support of each of the unknown vectors after reducing the dimension of the problem. Note that this approach crucially requires parameter estimation results for the corresponding family of mixtures which may be unavailable. To the best of our knowledge, most constructive sample complexity guarantees for parameter estimation in mixture models without separability assumptions correspond to mixtures of Gaussians [6], [7], [9], [16], [25], [26], [27], [28]. Moreover, most known results correspond to mixtures of Gaussians with two components. The only known results for parameter estimation in mixtures of Gaussians with more than 2 components is [9] but as we describe later, using the alternate approach with the guarantees in [9] results in a polynomial dependence on the sparsity. On the contrary, our sample complexity guarantees scales poly-logarithmically with the sparsity or dimension (for constant ℓ), see Corollary 3, which is a significant improvement over the alternate approach (though not unexpected, as support recovery is supposed be an easier task).

For other than Gaussian distributions, [7], [29] studied parameter estimation under the same moment-based assumption that we use. However, [7] use non-constructive arguments from algebraic geometry because of which, their results did not include bounds on the sufficient number of moments for learning the parameters in a mixture model. Reference [29] resolve this question to a certain extent for these aforementioned families of mixture models as they quantify the sufficient number of moments for parameter estimation under the restrictive assumption that the latent parameters lie on an integer lattice. Therefore, our results for these distributions form the first guarantees for support recovery.

1) Main Technical Contribution: Our work is most related to [23], where the focus is also support recovery, but one essentially queries a mixed linear model to get labels for *designed* features. Our unsupervised setting is completely different from this query-based setting and we crucially develop on a general technique introduced in [23] (see Lemma 1) for exact support recovery. The central idea that we borrow is that, support recovery is possible if we can estimate some subset statistics.

But computing estimates of these subset statistics to invoke the guarantees given in Lemma 1 is a difficult problem. Our approach to compute the sufficient statistics involves a two-step approach with polynomial identities: 1) first, using the method of moments, we compute estimates of the power sum polynomial of certain degree involving the unknown variables from all subsets of the coordinates up to a certain size; 2) secondly, we use an elegant connection via Newton's identities to compute estimates on the elementary symmetric polynomial in the unknown variables which in turn allows us to compute the sufficient statistics.

Exact Support Recovery: Our moment-based approach results in an algorithm (Theorem 1) with sample complexity of $O(\text{poly} \log n)$ for exact support recovery - assuming that other parameters such as the number of components ℓ, range of parameter entries, the first $\log \ell$ moments of the univariate base distribution of the mixture are constants and do not scale with the ambient dimension n or sparsity k. The dependence of our sample complexity guarantee on both the sparsity and ambient dimension is poly-logarithmic. Our results hold for multivariate mixture analogues of many canonical distributions such as Gaussian, Uniform, Poisson, and Laplace distributions among others. In contrast, for mixtures of Gaussians, the trivial alternate approach of first estimating parameters followed by support recovery results in a similar sample complexity $O(\text{poly}(k) \log n)$ guarantee for exact support recovery that scales polynomially with the sparsity k. For other mixture models, to the best of our knowledge, parameter estimation is an unsolved problem making the alternate approach infeasible.

Maximal Support Recovery: Maximal support recovery is an alternate problem related to support recovery and is an easier objective than exact support recovery. Under certain conditions, maximal support recovery is equivalent to exact support recovery (Remark 1). Therefore, as expected, we can provide improved sample complexity guarantees for maximal support recovery for several canonical distributions- these are again poly logarithmic in sparsity and ambient dimension but have significantly better polynomial factors.

a) Organization: The rest of the paper is organized as follows: in Section II, we provide the necessary definition and notations, and also formally define the problem. In Section III, we provide the necessary preliminary lemmas for support recovery (exact and maximal). In Section IV, we provide our main results on exact support recovery and discuss our core approaches, for example, see Corollary 3. In Section V, we have provided additional results on maximal support recovery. In Appendix A-B, we provide detailed proofs of all our results. In Appendix B, we provide the missing proofs of lemmas in Section III and in Appendix C, we provide the proof of Lemma 1 proved in [23]. In Appendix D, we provide a few technical lemmas that are used in the main proofs.

II. Definition and Problem Statement

A. Notations

We write $[n]$ to denote the set $\{1, 2, \dots, n\}$. We will use $\mathbf{1}_n, \mathbf{0}_n$ to denote an all one vector and all zero vector of dimension n respectively. We will use $\mathcal{Q}([n])$ to denote the power set of $[n]$ i.e. $\mathcal{Q}([n]) = \{\mathcal{C} \mid \mathcal{C} \subseteq [n]\}$. The default base for logarithms is 2, unless otherwise specified.

For any vector $\mathbf{v} \in \mathbb{R}^n$, we use $\mathbf{v}_i$ to denote the i^{th} coordinate of $\mathbf{v}$ and for any ordered set $\mathcal{S} \subseteq [n]$, we will use the notation $\mathbf{v}_{|\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ to denote the vector $\mathbf{v}$ restricted to the indices in $\mathcal{S}$. Furthermore, we will use $\mathsf{supp}(\mathbf{v}) \triangleq \{i \in [n] : \mathbf{v}_i \neq 0\}$ to denote the support of $\mathbf{v}$ and $||\mathbf{v}||_0 \triangleq |\mathsf{supp}(\mathbf{v})|$ to denote the size of the support. $||\mathbf{v}||_\infty$ denotes the largest magnitude across entries of vector $\mathbf{v}$. Let us refer to $\chi(\mathbf{v}) \in \{0, 1\}^n$ as a binary vector such that for all $i \in [n]$, we have $\chi(\mathbf{v})_i = 1$ if $\mathbf{v}_i \neq 0$ and $\chi(\mathbf{v})_i = 0$ otherwise. Let $\mathsf{sign} : \mathbb{R} \to \{-1, +1\}$ be a function that returns the sign of a real number i.e. for any input $x \in \mathbb{R}$,

$$\mathsf{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0. \end{cases}$$

Consider a multi-set of n-dimensional vectors $\mathcal{U} \equiv \{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(\ell)}\}$. We will write $\mathcal{S}_{\mathcal{U}}(i) \triangleq \{\mathbf{u} \in \mathcal{U} : \mathbf{u}_i \neq 0\}$ to denote the multi-set of vectors in $\mathcal{U}$ that has a non-zero entry at the i^{th} index. Furthermore, for an ordered set $\mathcal{C} \subseteq [n]$ and vector $\mathbf{a} \in \{0, 1\}^{|\mathcal{C}|}$, we will also write $\mathsf{occ}_{\mathcal{U}}(\mathcal{C}, \mathbf{a}) \triangleq \sum_{\mathbf{u}\in\mathcal{U}} \mathbf{1}[\chi(\mathbf{u})_{|\mathcal{C}} = \mathbf{a}]$ to denote the number of vectors in $\mathcal{U}$ whose supports equal $\mathbf{a}$ when restricted to the indices in $\mathcal{C}$. For a matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$, we will use $\mathbf{M}_i$ to denote the i^{th} column of $\mathbf{M}$.

Again, for a multi-set of n-dimensional vectors $\mathcal{V} \equiv \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(\ell)}\} \subseteq \mathbb{R}^n$, $\mathbf{A}_{\mathcal{V}} \in \{0, 1\}^{n\times \ell}$ denote the support matrix of $\mathcal{V}$ where each column vector $\mathbf{A}_i \in \{0, 1\}^n$ represents the support of the vector $\mathbf{v}^{(i)} \in \mathcal{V}$. For ease of notation, we will omit the subscript $\mathcal{V}$ when the set of vectors is clear from the context.

We write $\mathcal{N}(\mu, \sigma^2)$ to denote a Gaussian distribution with mean μ and variance σ^2. We will denote the cumulative distribution function of a random variable Z by $\phi : \mathbb{R} \to [0, 1]$ i.e. $\phi(a) = \int_{-\infty}^{a} p(z)dz$ where $p(\cdot)$ is the density function of Z. Also, we will denote $\mathsf{erf} : \mathbb{R} \to \mathbb{R}$ to be the error function defined by $\mathsf{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2)dt$. Since the error function erf is bijective, we define $\mathsf{erf}^{-1}(\cdot)$ to be the inverse of the $\mathsf{erf}(\cdot)$ function. Finally, for a fixed set $\mathcal{B}$ we will write $X \sim_{\mathsf{Unif}} \mathcal{B}$ to denote a random variable X that is uniformly sampled from the elements in $\mathcal{B}$.

B. Formal Problem Statements

Consider a class of distributions $\mathcal{P} \equiv \{\mathbf{P}(\theta)\}_{\theta\in\Theta}$ parameterized by some $\theta \in \Theta \subseteq \mathbb{R}$. We assume that all distributions in $\mathcal{P}$ satisfy the following property: $\mathbb{E}_{x\sim\mathbf{P}(\theta)} x^\ell$ can be written as a polynomial in θ of degree exactly ℓ. We emphasize that the class of distributions $\mathcal{P}$ is known and therefore the aforementioned polynomials (coefficients) of all degrees are $\mathbb{E}_{x\sim\mathbf{P}(\theta)} x^\ell$ are pre-computed and known. From Table II in [7], we know that many well-known distributions satisfy this property. For example,

1) $\mathbf{P}(\theta)$ can be a Gaussian distribution with mean θ and fixed known variance σ^2: for any positive integer ℓ, we have $\mathbb{E}x^\ell = \mu\mathbb{E}x^{\ell-1} + (\ell - 1)\sigma^2\mathbb{E}x^{\ell-2}$.
2) $\mathbf{P}(\theta)$ can be a uniform distribution with range $[\theta, b]$ for a fixed and known b.
3) $\mathbf{P}(\theta)$ can be a Poisson distribution with mean θ.
4) From Table II in [7], Laplace, Gamma, Exponential, Chi-squared distributions with appropriate parameterization also satisfy the condition.

Let $\mathcal{V}$ be a multi-set of ℓ unknown k-sparse vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(\ell)} \in \mathbb{R}^n$ such that $\left|\left|\mathbf{v}^{(i)}\right|\right|_0 \leq k$ for all $i \in [\ell]$. In our model, we observe samples from a n-dimensional distribution $\mathcal{P}_n$ that is a uniform mixture of ℓ distributions each of which is parameterized by one of these sparse unknown vectors. A sample $\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \sim \mathcal{P}_n$ is generated as follows:

$$t \sim_{\mathsf{Unif}} [\ell] \text{ and}$$
$$\mathbf{x}_i \mid t \sim \mathbf{P}(\mathbf{v}_i^{(t)}) \text{ independently } \forall i \in [n].$$

Consider $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)} \in \mathbb{R}^n$, m i.i.d. copies of $\mathbf{x}$, that we observe.

Our goal is to recover the support of unknown vectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(\ell)} \in \mathcal{V}$ with minimum number of samples m. More formally, we look at two distinct notions of support recovery:

Definition 1 (Exact Support Recovery): We will say that an algorithm achieves Exact Support Recovery if it can recover the supports of all the unknown vectors in $\mathcal{V}$ exactly.

Note that, $\{\mathsf{supp}(\mathbf{v}) : \mathbf{v} \in \mathcal{V}\}$ is a poset according to containment or set-inclusion ($\subseteq$). A maximal element of this poset is one that is not subset of any other element. When the supports are all different, each of them is maximal.

Let $\mathsf{Maximal}(\mathcal{V})$ be the unique set of all maximal elements of the poset $\{\mathsf{supp}(\mathbf{v}) : \mathbf{v} \in \mathcal{V}\}$.

Definition 2 (Maximal Support Recovery): We will say that an algorithm achieves Maximal Support Recovery if it can recover $\mathsf{Maximal}(\mathcal{V})$, i.e., all the maximal elements of the poset $\{\mathsf{supp}(\mathbf{v}) : \mathbf{v} \in \mathcal{V}\}$.

Note that in Definition 2, the objective is to recover supports of the largest set of vectors in $\mathcal{V}$, where no support is included completely in another support; this is easier than exact support recovery (Definition 1).

Remark 1: If every unknown vector $\mathbf{v} \in \mathcal{V}$ had a unique non-zero index $i \in [n]$ i.e. $\mathbf{v}_i \neq 0$ and $\mathbf{v}'_i = 0$ for all $\mathbf{v}' \in \mathcal{V} \setminus \{\mathbf{v}\}$, then maximal support recovery is equivalent to exact support recovery. This condition, also known as the separability condition, has been commonly used in the literature for example in unique non-negative matrix factorization [30],

[31], [32] and approximate parameter recovery in Mixtures of Linear Classifiers in the query-based setting [22].

III. Useful Results

In the first sub-section, we quote some useful results from [23] that will be useful in proving our results for exact support recovery. In the second sub-section, we will derive some properties of set systems and show sufficient statistics for maximal support recovery.

A. Results of [23]

To derive our support recovery results, we will crucially use the result of Lemma 1 below which has been proved in [23]. Recall the definition of $\text{occ}_{\mathcal{U}}(\mathcal{C},\mathbf{a})$ in Sec. II-A. Lemma 1 states that for a multi-set of binary vectors $\mathcal{U}$, if $\text{occ}_{\mathcal{U}}(\mathcal{C},\mathbf{a})$ is known for all sets $\mathcal{C}\subseteq[n]$ up to a cardinality of $\log\ell+1$, then it is possible to recover the unknown vectors in $\mathcal{U}$ up-to permutation. We restate the result according to our terminology.

Algorithm 1 Support Recovery

```
Require: $|\text{occ}_{\mathcal{V}}(C,\boldsymbol{a})|$ for every $C\subset[n]$, $|C|=t$, $t\in\{p,p+1\}$, $p=\lfloor\log\ell\rfloor$, and every $\mathbf{a}\in\{0,1\}^p\cup\{0,1\}^{p+1}$.
Set count $=1, i=1$.
while count $\le\ell$ do
    if there exists a vector $\mathbf{a}\in\{0,1\}^p\cup\{0,1\}^{p+1}$ and a positive integer $w$ such that $|\text{occ}_{\mathcal{V}}(C,\boldsymbol{a})|=w$, and $|\text{occ}_{\mathcal{V}}(C\cup\{j\},(\mathbf{a},1))|\in\{0,w\}$ for all $j\in[n]\setminus C$ then
        Construct binary vector $\mathbf{u}^i\equiv\{0\}^n$ and set $\mathbf{u}^i_{|C}=\mathbf{a}$.
        For every $j\in[n]\setminus C$, set $\mathbf{u}^i_{|j}=1$, if $|\text{occ}_{\mathcal{V}}(C\cup\{j\},(\boldsymbol{a},1))|=w$.
        Set Multiplicity$^i=w$.
        For all $\mathbf{t}\in\{0,1\}^p\cup\{0,1\}^{p+1}, S\subseteq[n]$ such that $|S|\in\{p,p+1\}$, update
            $|\text{occ}_{\mathcal{V}}(S,\mathbf{t})|\leftarrow|\text{occ}_{\mathcal{V}}(S,\mathbf{t})|-|\text{occ}_{\mathcal{V}}(C,\mathbf{a})|\times\mathbf{1}[\mathbf{u}^i_{|S}=\mathbf{t}]$
        count $=$ count $+w$.
        $i=i+1$.
    end if
end while
Return Multiplicity$^j$ copies of $\mathbf{u}^j$ for all $j<i$.
```

Lemma 1 (Corollary 1 in [23]): Let $\mathcal{V}$ be a multi-set of ℓ unknown vectors in $\mathbb{R}^n$. Then, if $\text{occ}_{\mathcal{V}}(\mathcal{C},\mathbf{a})$ is provided as input for all sets $\mathcal{C}\subset[n],|\mathcal{C}|\le\log\ell+1$ and for all $\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$, then there exists an algorithm (see Algorithm 1) that can recover the support of the unknown vectors in $\mathcal{V}$.

At a high level, the proof of Lemma 1 has two steps. First it can be shown (see Appendix C) that if the multi-set of ℓ unknown vectors $\mathcal{V}$ is p-identifiable (every unknown vector restricted to a certain set of p indices is unique), then computing $|\text{occ}(C,\mathbf{a})|$ for every subset of p and $p+1$ indices is sufficient to recover the supports. Note that Algorithm 1 describes the above methodology namely recovering the supports of all unknown vectors from knowledge of $|\text{occ}(\cdot,\cdot)|$. For a more detailed description of Algorithm 1 that first appeared in [23], we refer the reader to Section C-A. Secondly, it can be shown that any $n\times\ell$, (with $n>\ell$) binary matrix with all distinct columns is p-identifiable for some $p\le\log\ell$.

Algorithm 2 Exact Support Recovery Using Access to Estimates of $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (or Alternatively $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$) That Are Correct With High Probability

```
Require: Access to an oracle $\mathcal{O}$ that takes as input any set $\mathcal{C}\subseteq[n]$ and returns estimate of $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (or alternatively $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$) that are correct with high probability.
For each $i\in[n]$, compute an estimate of $|\mathcal{S}(i)|$ by providing $i$ as input to oracle $\mathcal{O}$.
Compute $\mathcal{T}=\{i\in[n]\mid\text{estimate}(|\mathcal{S}(i)|)>0\}$.
Compute estimates of $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (or alternatively $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$) for all subsets $\mathcal{C}\subseteq\mathcal{T},|\mathcal{C}|\le\log\ell+1$ (by providing set $\mathcal{C}$ as input to oracle $\mathcal{O}$).
Compute $\text{occ}(\mathcal{C},\mathbf{a})$ for all subsets $\mathcal{C}\subseteq\mathcal{T},|\mathcal{C}|\le\log\ell+1,\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$ using principle of inclusion and exclusion with the computed estimates of $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (or alternatively $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$) as input.
Use Algorithm 1 to recover the support of all unknown vectors in $\mathcal{V}$.
```

For the sake of completeness, we provided the detailed proof of Lemma 1 in Appendix C.

Remark 2: Lemma 1 provides an unconditional guarantee for recovering the supports of a multi-set of unknown vectors in $\mathcal{V}$. In other words, in the worst case, we only need to know $\text{occ}_{\mathcal{V}}(\mathcal{C},\mathbf{a})$ for all sets of size $|\mathcal{C}|\le\log\ell+1$. However, in [23][Theorems 1,2 and 4] have improved sample complexity guarantees for recovering the support of $\mathcal{V}$ under different additional structural assumptions. As noted in [23], these additional assumptions are mild - in most cases, if $\text{occ}_{\mathcal{V}}(\mathcal{C},\mathbf{a})$ is known for all sets $\mathcal{C}\subseteq[n]$ and all $\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$ up to a cardinality of 3, then it is possible to recover the supports of all the unknown vectors in $\mathcal{V}$. These algorithms are based on exact low rank integer tensor decomposition which is possible efficiently for tensors of order 3. Note that in this work, our goal has been to impose minimal assumptions on $\mathcal{V}$ and therefore we have used Lemma 1 to provide our (slightly worse) sample complexity guarantees on support recovery.

Next, we describe another result, Lemma 2, proved in [23] that is also going to be useful for us. The main takeaway from Lemma 2 is that computing $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ (which represents the number of unknown vectors in $\mathcal{V}$ having non-zero values in at least one entry corresponding to $\mathcal{C}$) for all sets smaller than a fixed size (say t) is sufficient to compute $\text{occ}(\mathcal{C},\mathbf{a})$ for all subsets $\mathcal{C}\subseteq[n],|\mathcal{C}|\le t$ and all vectors $\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$. In addition, we provide a result in Lemma 2 where we show that it is also possible to compute $\text{occ}(\mathcal{C},\mathbf{a})$ if the quantities $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ (which represents the number of unknown vectors in $\mathcal{V}$ having non-zero values in all entries corresponding to $\mathcal{C}$) are provided for all subsets $\mathcal{C}\subseteq[n]$ satisfying $|\mathcal{C}|\le t$.

Lemma 2: [23]: Let $\mathcal{V}$ be a multi-set of ℓ unknown vectors in $\mathbb{R}^n$. If $|\bigcup_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)|$ is provided as input for all sets $\mathcal{C}\subset[n],|\mathcal{C}|\le t$ or alternatively $|\bigcap_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)|$ is provided as input for all sets $\mathcal{C}\subset[n],|\mathcal{C}|\le t$, then we can compute $\text{occ}_{\mathcal{V}}(\mathcal{C},\mathbf{a})$ for all sets $\mathcal{C}\subseteq[n],|\mathcal{C}|\le t,\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$.

Next, we prove a corollary of Lemma 2 where we assume the existence of a randomized oracle. The oracle takes as input a set $\mathcal{C}$ and returns $|\cup_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)|$ (or alternatively $|\bigcap_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)|$) with probability $1-\gamma$ using $\mathsf{T}\log\gamma^{-1}$ samples for some known value of T. Assuming the existence of such an oracle, we characterize the sample complexity of exact support recovery. In other words, the following corollary implies that designing the oracle is sufficient for exact support recovery. In Lemma 6, we characterize the value of T in the context of exact support recovery when observing samples from an unknown mixtures of distributions.

Corollary 1: Let $\mathcal{V}$ be a multi-set of ℓ unknown k-sparse vectors in $\mathbb{R}^n$. Suppose, for each $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \log \ell + 1$, we can compute $|\cup_{i \in \mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)|$ (or alternatively $|\bigcap_{i \in \mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)|$) with probability $1 - \gamma$ using $\mathsf{T} \log \gamma^{-1}$ samples where T is independent of γ. Then, there exists an algorithm (see Algorithm 2) that can achieve Exact Support Recovery with probability at least $1-\gamma$ using $O(\mathsf{T} \log(\gamma^{-1}(n+(\ell k)^{\log \ell + 1})))$ samples.

Note that the main workhorse of Algorithm 2 is the principle of inclusion and exclusion (see Step 4). We refer the reader to Appendix B for a detailed proof of guarantees of Algorithm 2 that is included in this work for sake of completeness.

Algorithm 3 Maximal Support Recovery Using the Quantities $\mathbf{1}[|\cap_{i \in \mathcal{C}} \mathcal{S}(i)| > 0]$

```
Require: For every $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \ell$, the quantities $\mathbf{1}[|\cap_{i \in \mathcal{C}} \mathcal{S}(i)| > 0]$ are provided as input
 1: Set $\mathcal{T} = \phi$ to be the set of binary support vectors.
 2: Set $\mathcal{Z} = \{i \in [n] \mid \mathbf{1}[|\mathcal{S}(i)| > 0]\}$ to be the union of supports of unknown vectors.
 3: If $\mathbf{1}[\left|\cap_{i \in \mathcal{Z} \cup \{j\}} \mathcal{S}(i)\right| > 0] = 1$, then return $\mathcal{T}$ to be the vector $\mathbf{v} \in \{0,1\}^n$ where $\mathsf{supp}(\mathbf{v}) = \mathcal{Z}$
 4: while There exists a set $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \ell - 1$ such that 1) $\mathbf{v}'_{|\mathcal{C}} \neq \mathbf{1}_{|\mathcal{C}|}$ for all $\mathbf{v}' \in \mathcal{T}$ 2) $\mathbf{1}[|\cap_{i \in \mathcal{C}} \mathcal{S}(i)| > 0] = 1$ and 3)$\mathbf{1}[\left|\cap_{i \in \mathcal{C} \cup \{j\}} \mathcal{S}(i)\right| > 0] = 1$ for some $j \in \mathcal{Z} \setminus \mathcal{C}$. do
 5:     Set $\mathcal{U}' = \mathcal{C}$.
 6:     for $j \in [n] \setminus \mathcal{C}$ do
 7:         if $\mathbf{1}[\left|\cap_{i \in \mathcal{C} \cup \{j\}} \mathcal{S}(i)\right| > 0] = 1$ then
 8:             Set $\mathcal{U}' \leftarrow \mathcal{U}' \cup \{j\}$
 9:         end if
10:     end for
11:     Set $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{v}\}$ where $\mathbf{v} \in \{0,1\}^n$ and $\mathsf{supp}(\mathbf{v}) = \mathcal{U}'$.
12: end while
13: Return $\mathcal{T}$.
```

B. Properties of Set Systems

In this section, we describe the first contributions of this work. In the following preliminary results, we study the set $\mathsf{Maximal}(\mathcal{V})$ and its useful characteristic properties. Further, in the next few lemmas, we also show sufficient conditions for maximal support recovery. We start with the following definition:

Definition 3 (t-Good): A binary matrix $\mathbf{A} \in \{0,1\}^{n \times \ell}$ with all distinct columns is called t-good if for every column $\mathbf{A}_i$, there exists a set $S^{(i)} \subset [n]$ of t-indices such that $(\mathbf{A}_i)_{|S^{(i)}} = \mathbf{1}_t$, and $(\mathbf{A}_j)_{|S^{(i)}} \neq \mathbf{1}_t$ for all $j \neq i$. A set $U \subset \mathcal{Q}([n])$ is t-good if its $n \times |U|$ incidence matrix is t-good.

Notice that if any set is t-good then it must be r-good for all positive integers $n \geq r \geq t$. In Lemma 3, we show that $\mathsf{Maximal}(\mathcal{V})$ is $(\ell-1)$-good and in Lemma 5, we provide sufficient conditions for maximal support recovery of the set of unknown vectors $\mathcal{V}$.

Lemma 3: For any set of ℓ unknown vectors $\mathcal{V}$, $\mathsf{Maximal}(\mathcal{V})$ must be $(\ell-1)$-good.

Proof: Note that, any set of $\mathsf{Maximal}(\mathcal{V}) \subset \mathcal{Q}([n])$ is not contained in any other. For any two $A, A' \in \mathsf{Maximal}(\mathcal{V})$, there exists some $i \in A$ such that $i \notin A'$. Therefore, for a fixed $A \in \mathsf{Maximal}(\mathcal{V})$, for each $A' \in \mathsf{Maximal}(\mathcal{V}) \setminus \{A\}$, we can have at most $\ell - 1$ elements, that are all in A, but not all in any other set. We can also exploit the fact that if any set is t-good, then the set must be r-good for all $n \geq r \geq t$. To conclude, $\mathsf{Maximal}(\mathcal{V})$ must be at most $(\ell-1)$-good and is therefore $(\ell-1)$-good. □

Lemma 4: Let $\mathcal{V}$ be a multi-set of ℓ unknown vectors in $\mathbb{R}^n$. If it is known whether $|\cap_{i \in \mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)| > 0$ or not for all sets $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq s+1$, then there exists an algorithm (see Algorithm 3) that achieves maximal support recovery of the multi-set of unknown vectors $\mathcal{V}$ provided $\mathsf{Maximal}(\mathcal{V})$ is known to be s-good for $s \leq \ell - 1$ and $|\mathsf{Maximal}(\mathcal{V})| \geq 2$.

Proof of Lemma 4: As stated in the Lemma, suppose it is known if $|\cap_{i \in \mathcal{C}} \mathcal{S}(i)| > 0$ or not for all sets $\mathcal{C} \subseteq [n]$ satisfying $|\mathcal{C}| \leq s+1$. Assume that $\mathsf{Maximal}(\mathcal{V})$ is s-good Consider a set $A \in \mathsf{Maximal}(\mathcal{V})$. Since $\mathsf{Maximal}(\mathcal{V})$ is s-good, there must exist an ordered set $\mathcal{C} \subseteq [n], |\mathcal{C}| = s$ such that $\mathcal{C} \subseteq A$ but $\mathcal{C} \not\subseteq A'$ for all $A' \in \mathsf{Maximal}(\mathcal{V}) \setminus \{A\}$. Therefore, we must have $|\cap_{i \in \mathcal{C}} \mathcal{S}(i)| > 0$. But, on the other hand, notice that if $|\mathsf{Maximal}(\mathcal{V})| \geq 2$, there must exist an index $j \in \cup_{A \in \mathsf{Maximal}(\mathcal{V})} A$ such that $\left|\cap_{i \in \mathcal{C} \cup \{j\}} \mathcal{S}(i)\right| = 0$ since A does not contain the support of all other vectors. Algorithm 3 precisely checks for this condition in Step 3 and therefore this completes the proof. □

Lemma 5: Let $\mathcal{V}$ be a multi-set of ℓ unknown vectors in $\mathbb{R}^n$. If it is known whether $|\cap_{i \in \mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)| > 0$ or not for all sets $\mathcal{C} \subseteq [n], |\mathcal{C}| = \ell$, then there exists an algorithm (see Algorithm 3) that achieves maximal support recovery of the multi-set of unknown vectors $\mathcal{V}$.

Note that Lemma 5 just needs to handle the additional case where $|\mathsf{Maximal}(\mathcal{V})| = 1$.

Next, as in exact support recovery, we prove a corollary of Lemma 4 where we assume the existence of a randomized oracle. The oracle takes as input a set $\mathcal{C}$ and returns $|\cap_{i \in \mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)| > 0$ with probability $1 - \gamma$ using $\mathsf{T} \log \gamma^{-1}$ samples for some known value of T. Assuming the existence of such an oracle, we characterize the sample complexity of maximal support recovery. In other words, the following corollary implies that designing such an oracle is sufficient for maximal support recovery. In Lemma 7, we characterize the value of T in the context of exact support recovery when observing samples from an unknown mixtures of distributions.

Corollary 2: Let $\mathcal{V}$ be a multi-set of ℓ unknown k-sparse vectors in $\mathbb{R}^n$. Suppose with probability $1-\gamma$, for each $\mathcal{C} \subseteq [n], |\mathcal{C}| \leq \ell$, we can compute if $|\cap_{i \in \mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)| > 0$ correctly with $\mathsf{T} \log \gamma^{-1}$ samples where T is independent of γ. Then, there exists an algorithm (see Algorithm 4) that can achieve maximal support recovery with probability at least $1-\gamma$ using $O(\mathsf{T} \log(\gamma^{-1}(n + (\ell k)^{\ell})))$ samples.

The above corollary just takes into account the failure probability in computing estimates of all the statistical quantities sufficient for maximal support recovery.

Remark 3: Corollary 2 describes the sample complexity for maximal support recovery using Lemma 5 which provides the

Algorithm 4 Maximal Support Recovery Using Access to Estimates of $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|>0]$ That Are Correct With High Probability

```
Require: For $\mathcal{C}\subseteq[n]$, access to estimates of $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|>0]$ that are correct with high probability.
1: For each $i\in[n]$, compute an estimate of $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|>0]$.
2: Compute $\mathcal{T}=\{i\in[n]\mid \text{estimate}(\mathbf{1}[|\mathcal{S}(i)|>0]))=\text{True}\}$.
3: Compute estimates of $\mathbf{1}[|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|>0]$ for all subsets $\mathcal{C}\subseteq\mathcal{T},|\mathcal{C}|\le\ell$.
4: Use Algorithm 3 to recover the support of all unknown vectors in $\mathcal{V}$.
```

worst-case guarantees as $\mathsf{Maximal}(\mathcal{V})$ is $(\ell-1)$-good for all sets $\mathcal{V}$. We can also provide improved guarantees for maximal support recovery provided $\mathsf{Maximal}(\mathcal{V})$ is known to be s-good by using Lemma 4. However, for the sake of simplicity of exposition, we have only provided results for maximal support recovery in mixture models using Corollary 2.

All the missing proofs of this section (other than that of Lemma 1 and Lemma 4) can be found in Appendix B.

IV. Exact Support Recovery

In this section, we will present our main results and high level techniques for exact support recovery. The detailed proofs of all results in this section can be found in Section A-B. We will start by introducing some additional notations specifically for this setting. Recall that $\mathbb{E}_{x\sim\mathbf{P}(\theta)}x^t$ can be written as a polynomial in θ of degree t. We will write

$$q_t(\theta)\triangleq\mathbb{E}_{x\sim\mathbf{P}(\theta)}x^t=\sum_{i\in[t+1]}\beta_{t,i}\theta^{i-1}$$

to denote this aforementioned polynomial where we use $\{\beta_{t,i}\}_{i\in[t+1]}$ to denote its coefficients. For all sets $\mathcal{A}\subseteq[n]$, we will write $\mathcal{Q}_i(\mathcal{A})$ to denote all subsets of $\mathcal{A}$ of size at most i i.e. $\mathcal{Q}_i(\mathcal{A})=\{\mathcal{C}\mid\mathcal{C}\subseteq\mathcal{A},|\mathcal{C}|\le i\}$. Let us define the function $\pi:\mathcal{Q}([n])\times[n]\to[n]$ to denote a function that takes as input a set $\mathcal{C}\subseteq[n]$, an index $r\in\mathcal{C}$ and returns as output the position of r among all elements in $\mathcal{C}$ sorted in ascending order. In other words, for a fixed set $\mathcal{C}$ and all $j\in[|\mathcal{C}|]$, $\pi(\mathcal{C},\cdot)$ maps the j^{th} smallest index in $\mathcal{C}$ to j; for example, if $\mathcal{C}=\{3,5,9\}$, then $\pi(\mathcal{C},3)=1,\pi(\mathcal{C},5)=2$ and $\pi(\mathcal{C},9)=3$.

We will write $\mathbb{Z}^+$ to denote the set of non-negative integers and $(\mathbb{Z}^+)^n$ to denote the set of all n-dimensional vectors having entries which are non-negative integers. For two vectors $\mathbf{u},\mathbf{t}\in(\mathbb{Z}^+)^n$, we will write $\mathbf{u}\le\mathbf{t}$ if $\mathbf{u}_i\le\mathbf{t}_i$ for all $i\in[n]$; similarly, we will write $\mathbf{u}<\mathbf{t}$ if $\mathbf{u}_i<\mathbf{t}_i$ for all $i\in[n]$. For any fixed subset $\mathcal{C}\subseteq[n]$ and vectors $\mathbf{u},\mathbf{t}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$, we will write $\zeta_{\mathbf{t},\mathbf{u}}$ to denote the quantity $\zeta_{\mathbf{t},\mathbf{u}}\triangleq\prod_{i\in\mathcal{C}}\beta_{\mathbf{t}_{\pi(\mathcal{C},i)},\mathbf{u}_{\pi(\mathcal{C},i)}+1}$. For any $\mathbf{u},\mathbf{z}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u}<\mathbf{z}$, we will define a path M to be a sequence of vectors $\mathbf{z}_1>\mathbf{z}_2>\cdots>\mathbf{z}_m$ such that $\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_m\in(\mathbb{Z}^+)^n$, $\mathbf{z}_1=\mathbf{z}$ and $\mathbf{z}_m=\mathbf{u}$. Let $\mathcal{M}(\mathbf{z},\mathbf{u})$ be the set of all paths starting from $\mathbf{z}$ and ending at $\mathbf{u}$. We will also write a path $\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})$ uniquely as a set of $m-1$ ordered tuples $\{(\mathbf{z}_1,\mathbf{z}_2),(\mathbf{z}_2,\mathbf{z}_3),\ldots,(\mathbf{z}_{m-1},\mathbf{z}_m)\}$ where each tuple consists of adjacent vectors in the path sequence. We will also write $\mathcal{T}(\mathsf{M})\equiv\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_m\}$ to denote the set of elements in the path.

We start with the following assumption which states that the magnitude of every non-zero co-ordinate of all unknown vectors is bounded from above and below:

Algorithm 5 Estimate(m,B) Estimating $\mathbb{E}X$ for $X\sim\mathcal{P}$

```
Require: I.i.d samples $x^{(1)},x^{(2)},\ldots,x^{(m)}\sim\mathcal{P}$
1: Set $t=m/B$
2: for $i=1,2,\ldots,B$ do
3:     Set Batch $i$ to be the samples $x^{(j)}$ for $j\in\{it+1,it+2,\ldots,(i+1)t\}$.
4:     Set $S_1^i=\sum_{j\in\text{ Batch } i}\frac{x^{(j)}}{t}$
5: end for
6: Return median$(\{S_1^i\}_{i=1}^B)$
```

Algorithm 6 Recover $|\bigcap_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)|$

```
Require: Samples $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)}\sim\mathcal{P}_n$. Set $\mathcal{C}\subseteq[n]$.
1: For every $\mathbf{z}\le 2\ell\mathbf{1}_{|\mathcal{C}|}$, compute estimate $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ using Algorithm 5 on the set of samples $\{(\mathbf{x}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}\}_{j=1}^m$.
2: For every $\mathbf{z}\le 2\ell\mathbf{1}_{|\mathcal{C}|}$, compute an estimate $\widehat{V}^{\mathbf{z}}$ of $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ recursively using equation $\ell\widehat{U}^{\mathbf{z}}-\sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\widehat{V}^{\mathbf{u}}=\zeta_{\mathbf{z},\mathbf{z}}\cdot\widehat{V}^{\mathbf{z}}$.
3: For every $t\in[\ell]$, compute an estimate $\widehat{\mathsf{A}}_{\mathcal{C},t}$ of $\sum_{\substack{\mathcal{C}'\subseteq[\ell]\\|\mathcal{C}'|=t}}\prod_{\substack{i\in\mathcal{C}\\j\in\mathcal{C}'}}(\mathbf{v}_i^{(j)})^2$ recursively using Newton's identity $t\widehat{\mathsf{A}}_{\mathcal{C},t}=\sum_{p=1}^t(-1)^{p+1}\widehat{\mathsf{A}}_{\mathcal{C},t-p}\widehat{V}^{2p\mathbf{1}_{|\mathcal{C}|}}$.
4: Return $\max_{t\in[\ell]}t.\mathbf{1}[\widehat{\mathsf{A}}_{\mathcal{C},t}>0]$.
```

Assumption 1: We will assume that the magnitude of all non-zero entries of all unknown vectors in the set $\mathcal{V}$ are bounded from above by R and from below by δ that is $\left|\left|\mathbf{v}^{(i)}\right|\right|_\infty\le R$ for all $i\in[\ell]$ and $\min_{\mathbf{v}\in\mathcal{V}}\min_{i:\mathbf{v}_i\neq 0}|\mathbf{v}_i|\ge\delta$.

Now, we show our main lemma in this setting where we characterize the sufficient number of samples to compute $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)\right|$ for each set $\mathcal{C}\subseteq[n]$ with high probability in terms of the coefficients of the polynomials $\{q_t(\theta)\}_t$. Note that, doing so can allow us to directly invoke Corollary 1 and obtain high probability sample complexity guarantees for exact support recovery.

Lemma 6: Suppose Assumption 1 is true. Fix any set $\mathcal{C}\subseteq[n]$. Let

$$\Phi\triangleq\frac{\delta^{2\ell|\mathcal{C}|}}{2\Big(3\max(\ell R^{2\ell|\mathcal{C}|},2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(\ell-1)}\ell!}$$
$$\times\left(\max_{\mathbf{z}\le 2\ell\mathbf{1}_{|\mathcal{C}|}}\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}}+\sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\right)^{-1},$$
$$g_{\ell,\mathcal{V}}\triangleq\frac{\max_{\mathbf{z}\le 2\ell\mathbf{1}_{|\mathcal{C}|}}\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2},$$

where $g_{\ell,\mathcal{V}}$ is a constant that is independent of k and n but depends on ℓ. There exists an algorithm (see Algorithm 6) that can compute $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)\right|$ exactly for the set $\mathcal{C}$ with probability at least $1-\gamma$ using $O\Big(\log(\gamma^{-1}(2\ell)^{|\mathcal{C}|})g_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_n$.

In order to prove Lemma 6, we first show that (see Lemma 8) for each fixed ordered set $\mathcal{C}\subseteq[n]$ and each vector

$\mathbf{t} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$, we must have

$$\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{t}_{\pi(\mathcal{C},i)}} = \frac{1}{\ell}\sum_{\mathbf{u}\le\mathbf{t}} \zeta_{\mathbf{t},\mathbf{u}} \cdot \Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big). \tag{1}$$

Note that each summand in equation 1 is a product of the powers of the co-ordinates of the same unknown vector. In Lemma 9, we show that for each set $\mathcal{C} \subseteq [n]$ and any vector $\mathbf{t} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$, we can compute $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{t}_{\pi(\mathcal{C},i)}}$ via a recursive procedure provided for all $\mathbf{u} \in (\mathbb{Z}^+)^{\le|\mathcal{C}|}$ satisfying $\mathbf{u} \le \mathbf{t}$, the quantity $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ is pre-computed. This implies that we can compute $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{2p}$ for all $p \in [\ell]$ from the quantities $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{u} \le 2p\mathbf{1}_{|\mathcal{C}|}$. It is easy to recognize $\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p$ as the power sum polynomial of degree p in the variables $\{\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\}_{\mathbf{v}\in\mathcal{V}}$. Now, let us define the quantity $\mathsf{A}_{\mathcal{C},t}$ for a fixed ordered set $\mathcal{C}$ and parameter $t \in [\ell]$ as follows:

$$\mathsf{A}_{\mathcal{C},t} \triangleq \sum_{\substack{\mathcal{C}'\subseteq[\ell]\\ |\mathcal{C}'|=t}} \prod_{\substack{i\in\mathcal{C}\\ j\in\mathcal{C}'}} (\mathbf{v}_i^{(j)})^2$$

Notice that $\mathsf{A}_{\mathcal{C},t} > 0$ if and only if there exists a subset $\mathcal{C}' \subseteq [\ell], |\mathcal{C}'| = t$ such that $\mathbf{v}_i^{(j)} \neq 0$ for all $i \in \mathcal{C}, j \in \mathcal{C}'$. Hence, the maximum value of t such that $\mathsf{A}_{\mathcal{C},t} > 0$ is the number of unknown vectors in $\mathcal{V}$ having non-zero value in all the indices in $\mathcal{C}$. In other words, we have that

$$\Big|\bigcap_{i\in\mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)\Big| = \max_{t\in[\ell]} t\cdot \mathbf{1}[\mathsf{A}_{\mathcal{C},t} > 0].$$

Notice that $\mathsf{A}_{\mathcal{C},t}$ is the elementary symmetric polynomial of degree t in the variables $\{\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\}_{\mathbf{v}\in\mathcal{V}}$. We can use Newton's identities to state that for all $t \in [\ell]$,

$$t\mathsf{A}_{\mathcal{C},t} = \sum_{p=1}^{t}(-1)^{p+1}\mathsf{A}_{\mathcal{C},t-p}\Big(\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p\Big)$$

using which, we can recursively compute $\mathsf{A}_{\mathcal{C},t}$ for all $t \in [\ell]$ ($\mathsf{A}_{\mathcal{C},0} = 1$) and hence $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ if we were given $\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p$ as input for all $p \in [\ell]$ (see Lemma 10). Lemma 6 follows from making these set of computations robust.

Thus, from Lemma 6, we are now equipped with the knowledge of T that we need to set in Corollary 1. We next show Theorem 1 which follows from applying Lemma 6 and Corollary 1.

Theorem 1: Let $\mathcal{V}$ be a set of ℓ unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1. Recall that for a given set $\mathcal{A}$, $\mathcal{Q}_i(\mathcal{A})$ corresponds to all subsets of $\mathcal{A}$ of size at most i. For any positive integer m, let $\mathcal{F}_m = \mathcal{Q}_1([n]) \cup \mathcal{Q}_m(\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v}))$ and

$$\Phi_m \triangleq \frac{\delta^{2\ell m}}{2\Big(3\ell\max(R^{2\ell m}, 2^\ell R^{\ell+m})\Big)^{(\ell-1)}\ell!} \times \Bigg(\max_{\mathbf{z}\le 2\ell\mathbf{1}_m}\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Bigg)^{-1},$$

$$f_{\ell,\mathcal{V}} \triangleq \max_{\substack{\mathbf{z}\le 2\ell\mathbf{1}_{\log\ell+1}\\ \mathcal{C}\in\mathcal{F}_{\log\ell+1}}} \frac{\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2_{\log\ell+1}}.$$

Here $f_{\ell,\mathcal{V}}$ is a quantity that is independent of sparsity k and ambient dimension n. Then, there exists an algorithm (see Algorithm 6 and 2) that achieves Exact Support Recovery with probability at least $1-\gamma$ using $O\Big(\log(\gamma^{-1}(2\ell)^{\log\ell+1}(n + (\ell k)^{\log\ell+1}))f_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_n$.

Remark 4: $f_{\ell,\mathcal{V}}$ is a quantity that is a function of δ, R (range of magnitude of non-zero entries in parameter vectors belonging to $\mathcal{V}$) the first 2ℓ moments of the random variable $x \sim \mathbf{P}(\theta)$ where $\theta \in \{\mathbf{v}_i\}_{i\in[n],\mathbf{v}\in\mathcal{V}}$. If the aforementioned quantities are constants, then $f_{\ell,\mathcal{V}}$ is also a constant. In other words, we emphasize that $f_{\ell,\mathcal{V}}$ is independent of k, n (sparsity and ambient dimension).

Remark 5: We can relax Assumption 1 in Theorem 1 without much further work. For our proofs to work out verbatim, it is sufficient to just have the following condition be true: given the latent variable t denoting the mixture component, coordinates of the random vector $\mathbf{x} \sim \mathcal{P}_n$ must be $(\log\ell+1)$-wise independent (any $\log\ell+1$ co-ordinates are independent). However, for the sake of simplicity, we have provided the setting where all co-ordinates of $\mathbf{x} \mid t$ are independent.

Example: Consider the setting when we obtain m i.i.d samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^n$ from a high dimensional Gaussian mixture $\mathcal{D} = \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}^{(1)}, \sigma^2\mathbf{I}) + \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}^{(2)}, \sigma^2\mathbf{I})$ with two components where $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)} \in \mathbb{R}^n$ satisfying $\|\boldsymbol{\mu}^{(1)}\|_0, \|\boldsymbol{\mu}^{(2)}\|_0 \le k$ are unknown and $\sigma > 0$ is known. Our goal is to recover the support of $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}$ while minimizing the number of samples m. For $\mathbf{x} \sim \mathcal{D}$, for all $i \in [n]$, we have that $\mathbb{E}\mathbf{x}_i^2 = \sigma^2 + ((\boldsymbol{\mu}_i^{(1)})^2 + (\boldsymbol{\mu}_i^{(2)})^2)/2$; for all $i, j \in [n], i \neq j$, we have

$$\mathbb{E}\mathbf{x}_i^2\mathbf{x}_j^2 = \sigma^2(\mathbb{E}\mathbf{x}_i^2 + \mathbb{E}\mathbf{x}_j^2) - \sigma^4 + \Big(\frac{(\boldsymbol{\mu}_i^{(1)})^2(\boldsymbol{\mu}_j^{(1)})^2 + (\boldsymbol{\mu}_i^{(2)})^2(\boldsymbol{\mu}_j^{(2)})^2}{2}\Big).$$

Hence, in the first step, for all $i \in [n]$, with probability $1-\gamma$ we compute an estimate u_i of $\mathbb{E}\mathbf{x}_i^2$ (using Lemma 14) satisfying $\left|u_i - \mathbb{E}\mathbf{x}_i^2\right| \le \delta^4/(64\sigma^2)$ using $\widetilde{O}(\delta^{-8}\sigma^4\max_i(\sigma^4, (\boldsymbol{\mu}_i^{(1)})^4, (\boldsymbol{\mu}_i^{(2)})^4))$ samples. With this, we can infer the union of support correctly to be $\mathcal{S} \equiv \{i \in [n] \mid u_i - \sigma^2 \ge \delta^2/4\}$. This is because for any index i in the union of support, we must have $\mathbb{E}\mathbf{x}_i^2 \ge \sigma^2 + \delta^2/2$ while for any index i not in the union, we have $\mathbb{E}\mathbf{x}_i^2 = \sigma^2$. Next, in the second step, for all $i, j \in \mathcal{S}; i \neq j$, we compute an estimate u'_{ij} of $\mathbb{E}\mathbf{x}_i^2\mathbf{x}_j^2$ satisfying $|u'_{ij} - \mathbb{E}\mathbf{x}_i^2\mathbf{x}_j^2| \le \delta^4/16$ using $O(\delta^{-8}\max_{i,j}(\sigma, \boldsymbol{\mu}_i^{(1)}, \boldsymbol{\mu}_j^{(1)}, \boldsymbol{\mu}_i^{(2)}, \boldsymbol{\mu}_j^{(2)})^8\log(n\gamma^{-1}))$ samples with probability at least $1-\gamma$ (see Lemma 14). In that case, if i, j belongs to the support of the same vector, then we will have $|u'_{ij} - \sigma^2(u_i + u_j) + \sigma^4| \ge 13\delta^4/32$ while otherwise, we must have $|u'_{ij} - \sigma^2(u_i + u_j) + \sigma^4| \le 3\delta^4/32$. Hence, $\mathcal{T} = \{(i,j) \in \mathcal{S}, i \neq j \mid |u'_{ij} - \sigma^2(u_i + u_j) + \sigma^4| \ge 13\delta^4/32\}$. If there does not exist $i, j \in \mathcal{S}, i \neq j$ such that $(i,j) \notin \mathcal{T}$, then we return $\mathsf{supp}(\boldsymbol{\mu}^{(1)}) = \mathsf{supp}(\boldsymbol{\mu}^{(2)}) = \mathcal{S}$ implying that both supports are same. On the other hand, if there exists $i, j \in \mathcal{S}, i \neq j$ such that $(i,j) \notin \mathcal{T}$ then i belongs to the support of one vector while j belongs to the support of the other vector (both supports are not same). Let the support of one vector will be $\{s \in \mathcal{S}, s \neq i \mid (i,s) \in \mathcal{T}\}$ and the support of the other vector is $\{s \in \mathcal{S}, s \neq j \mid (j,s) \in \mathcal{T}\}$. Therefore, the sufficient sample complexity for recovering the support is $m = O(\delta^{-8} \max_{i,j}(\sigma, \boldsymbol{\mu}_i^{(1)}, \boldsymbol{\mu}_j^{(1)}, \boldsymbol{\mu}_i^{(2)}, \boldsymbol{\mu}_j^{(2)})^8 \log(n\gamma^{-1}))$. Note that in this example, the algorithm is slightly different from the one presented in Algorithm 6; in, fact the algorithm follows that of maximal support recovery (see Section V) which is equivalent to exact support recovery for $\ell = 2$ (see Remark 1).

Now, we provide a corollary of Theorem 1 specifically for mean-estimation in a mixture of distributions with constant number of components i.e. $\ell = O(1)$. In particular, consider the setting where

$$t \sim_{\mathsf{Unif}} [\ell] \text{ and } \mathbf{x}_i \mid t \sim \mathbf{P}(\mathbf{v}_i^{(t)}) \text{ independently } \forall i \in [n]$$
$$\mathbb{E}_{\mathbf{x} \sim \mathcal{P}_n}[\mathbf{x}_i \mid t = j] = \mathbf{v}_i^{(j)}$$

i.e. the mean of the i^{th} co-ordinate of the random vector $\mathbf{x}$ distributed according to $\mathcal{P}_n$ is $\mathbf{v}_i^{(j)}$.

Corollary 3: Consider the mean estimation problem described above. Let $\mathcal{V}$ be a set of $\ell = O(1)$ unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1 and $f_{\ell,\mathcal{V}}$ be as defined in Theorem 1. Then, there exists an algorithm (see Algorithm 6 and 2) that with probability at least $1-\gamma$, achieves Exact Support Recovery using $O\Big(\text{poly}\log(n\gamma^{-1})\text{poly}(\delta R^{-1}) f_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_n$.

We can compare the sample complexity presented in Corollary 3 with the alternate approach for support recovery namely the two stage process of recovering the union of support followed by parameter estimation restricted to the union of support. As discussed in Section I, most known results (other than [9]) for parameter estimation in Gaussian mixtures without separability assumptions hold for two mixtures and are therefore not applicable for $\ell > 2$. For general value of ℓ, the only known sample complexity guarantees for parameter estimation in mixture of Gaussians is provided in [9].

Note that computing the union of support is not difficult. In particular, in Lemma 6, the guarantees include the sample complexity of testing whether a particular index belongs to the union of support; this can be used to compute the union of support itself after taking a union bound over all indices leading to a multiplicative $\log n$ factor.

However, for one dimensional Gaussian mixture models (1D GMM), the parameter estimation guarantees in [9] (See Corollary 5) are polynomial in the inverse of the failure probability. Since parameter estimation in 1D GMM is used as a framework for solving the high dimensional problem, it can be extracted that the sample complexity in n dimensions must be polynomial in n with degree at least 1 to achieve a per coordinate error (error in ℓ_∞ norm). If restricted to the union of support of the unknown vectors in $\mathcal{V}$, then using the guarantees in [9] directly will lead to a polynomial dependence on ℓk. In essence, the sample complexity of the alternate approach has a logarithmic dependence on the latent space dimension and a polynomial dependence on sparsity k (for constant ℓ). Note that in the analogous high dimensional mixtures of Gaussians, our sample complexity guarantee has a similar poly-logarithmic dependence on the ambient dimension n and the sparsity k (for a constant ℓ).

For other distributions, to the best of our knowledge, the only known parameter estimation results that exist in literature are [7], [29]. In both of these works, the authors use the same assumption that $\mathbb{E}_{x \sim \mathbf{P}(\theta)} x^\ell$ can be written as a polynomial in θ of degree exactly ℓ. While the guarantees in [7] are non-constructive, the results in [29] need the restrictive assumption that the means must be multiple of some $\epsilon > 0$ and moreover, they have an exponential dependence on the noise variance and ϵ^{-1}. Our results do not have these limitations and are therefore widely applicable.

V. Maximal Support Recovery

In this section, we will present our main results for maximal support recovery. The detailed proofs of all results in this section can be found in Section A-B.

```
Algorithm 7 Estimate if $|\bigcap_{i\in\mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)| > 0$
Require: Samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)} \sim \mathcal{P}_n$. Set $\mathcal{C} \subseteq [n]$.
1: For every $\mathbf{z} \le 2\mathbf{1}_{|\mathcal{C}|}$, compute estimate $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$
   using Algorithm 5 on the set of samples $\{(\mathbf{x}_i^j)^{\mathbf{z}_{\pi(\mathcal{C},i)}}\}_{j=1}^m$.
2: For every $\mathbf{z} \le 2\mathbf{1}_{|\mathcal{C}|}$, compute an estimate $\widehat{V}^{\mathbf{z}}$ of
   $\sum_{j\in[\ell]} \prod_{i\in\mathcal{C}} (\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi_{\mathcal{C},i}}}$ recursively using the following equation:
   $$\ell\widehat{U}^{\mathbf{z}} - \sum_{\mathbf{u}<\mathbf{z}} \zeta_{\mathbf{z},\mathbf{u}} \cdot \widehat{V}^{\mathbf{u}} = \zeta_{\mathbf{z},\mathbf{z}} \cdot \widehat{V}^{\mathbf{z}}.$$
3: If $\widehat{V}^{2\mathbf{1}_{|\mathcal{C}|}} \ge \delta^{2|\mathcal{C}|}/2$, return True and otherwise return False.
```

Now, we provide results on maximal support recovery in the MD setting. Note that from Lemma 5, for Maximal support recovery, we only need to estimate correctly if $\left|\bigcap_{i\in\mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)\right| > 0$ for ordered sets $\mathcal{C} \subseteq [n]$. Notice that $\left|\bigcap_{i\in\mathcal{C}} \mathcal{S}(i)\right| > 0$ if and only if $\sum_{\mathbf{v}\in\mathcal{V}} \prod_{i\in\mathcal{C}} \mathbf{v}_i^2 > 0$. From our previous arguments, $\sum_{\mathbf{v}\in\mathcal{V}} \prod_{i\in\mathcal{C}} \mathbf{v}_i^2$ can be computed if the quantities $\mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{u} \le 2\mathbf{1}_{|\mathcal{C}|}$ are pre-computed. The following lemma stems from making the aforementioned computation robust to the randomness in the dataset and thus provide a sample complexity guarantee for estimating $\left|\bigcap_{i\in\mathcal{C}} \mathcal{S}_{\mathcal{V}}(i)\right| > 0$ for ordered sets $\mathcal{C} \subseteq [n]$ with high probability. This in turn can allow use to invoke Corollary 2 to obtain sample complexity guarantees for maximal support recovery:

Lemma 7: Suppose Assumption 1 is true. Fix any set $\mathcal{C} \subseteq [n]$. Let

$$\Phi \triangleq \max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}} \frac{\delta^{2|\mathcal{C}|}}{2} \Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}} \sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}} \zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})} \zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}$$

$$h_{\ell,\mathcal{V}} \triangleq \frac{\max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}} \mathbb{E}\prod_{i\in\mathcal{C}} \mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}$$

where $h_{\ell,\mathcal{V}}$ is a constant independent of k and n but depends on ℓ. There exists an algorithm (see Algorithm 7) that can compute if $\left|\bigcap_{i\in\mathcal{C}} \mathcal{S}(i)\right| > 0$ correctly for the set $\mathcal{C}$ with probability at least $1-\gamma$ using $O(h_{\ell,\mathcal{V}} \log\gamma^{-1})$ samples generated according to $\mathcal{P}_n$.

The subsequent theorem follows from Lemma 7 and Corollary 2. Note that, compared to exact support recovery (Theorem 2) the sample complexity for maximal support recovery has significantly improved dependency on δ and furthermore, it is also independent of R.

Thus, from Lemma 7, we are now equipped with the knowledge of T that we need to set in Corollary 2. Hence we show our main result regarding maximal support recovery by invoking Corollary 2 and setting the value of T that we obtain from Lemma 7.

Theorem 2: Let $\mathcal{V}$ be a set of unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1. Recall that for a given set $\mathcal{A}$, $\mathcal{Q}_i(\mathcal{A})$ corresponds to all subsets of $\mathcal{A}$ of size at most i. For any positive integer m, let $\mathcal{F}_m = \mathcal{Q}_1([n]) \cup \mathcal{Q}_m(\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v}))$ and

$$\Phi_m = \max_{\mathbf{z}\le 2\mathbf{1}_m} \frac{\delta^{2m}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}} \sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}$$

$$h'_{\ell,\mathcal{V}} \triangleq \max_{\substack{\mathbf{z}\le 2\mathbf{1}_\ell\\ \mathcal{C}\in\mathcal{F}_\ell}} \frac{\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi_\ell^2}$$

where $h'_{\ell,\mathcal{V}}$ is a constant independent of sparsity k and ambient dimension n but depends on ℓ. Accordingly, there exists an algorithm (see Algorithm 7 and 4) that achieves maximal support recovery with probability at least $1-\gamma$ using $O\Big(h'_{\ell,\mathcal{V}}\log(\gamma^{-1}(n+(\ell k)^\ell))\Big)$ samples generated from $\mathcal{P}_n$.

Remark 6 (Computational Complexity): All our algorithms are efficient, namely their computational complexities are polynomial in the dimension n and sparsity k.

VI. Conclusion

In this paper, we considered the problem of learning the support of some sparse high-dimensional vectors when we see samples from a mixture model parameterized by those sparse vectors. The class of distribution we can handle requires coordinate-wise independence. It will be good to relax this assumption in future. On the other hand, the distribution of each coordinate must be such that the expectation operator maps to a polynomial. While this class is pretty large, it would be good to come up with methods that are applicable to different general classes of distributions that do not satisfy this assumption. It is most likely that one needs to look beyond method of moments for such classes.

Appendix A
Proofs for Exact and Maximal Support Recovery

A. Notations and Definitions

Recall that $\mathbb{E}_{x\sim\mathbf{P}(\theta)}x^t$ can be written as a polynomial in θ of degree t. We write

$$q_t(\theta) \triangleq \mathbb{E}_{x\sim\mathbf{P}(\theta)}x^t = \sum_{i\in[t+1]}\beta_{t,i}\theta^{i-1}.$$

to denote this aforementioned polynomial where we use $\{\beta_{t,i}\}_{i\in[t+1]}$ to denote its coefficients. For all sets $\mathcal{A}\subseteq[n]$, we will write $\mathcal{Q}_i(\mathcal{A})$ to denote all subsets of $\mathcal{A}$ of size at most i i.e. $\mathcal{Q}_i(\mathcal{A}) = \{\mathcal{C} \mid \mathcal{C}\subseteq\mathcal{A}, |\mathcal{C}|\le i\}$. Let us define the function $\pi:\mathcal{Q}([n])\times[n]\to[n]$ to denote a function that takes as input a set $\mathcal{C}\subseteq[n]$, an index $r\in\mathcal{C}$ and returns as output the position of r among all elements in $\mathcal{C}$ sorted in ascending order. In other words, for a fixed set $\mathcal{C}$ and all $j\in[|\mathcal{C}|]$, $\pi(\mathcal{C},\cdot)$ maps the j^{th} smallest index in $\mathcal{C}$ to j; for example, if $\mathcal{C}=\{3,5,9\}$, then $\pi(\mathcal{C},3)=1, \pi(\mathcal{C},5)=2$ and $\pi(\mathcal{C},9)=3$.

We will write $\mathbb{Z}^+$ to denote the set of non-negative integers and $(\mathbb{Z}^+)^n$ to denote the set of all n-dimensional vectors having entries which are non-negative integers. For two vectors $\mathbf{u},\mathbf{t}\in(\mathbb{Z}^+)^n$, we will write $\mathbf{u}\le\mathbf{t}$ if $\mathbf{u}_i\le\mathbf{t}_i$ for all $i\in[n]$; similarly, we will write $\mathbf{u}<\mathbf{t}$ if $\mathbf{u}_i<\mathbf{t}_i$ for all $i\in[n]$. For any fixed subset $\mathcal{C}\subseteq[n]$ and vectors $\mathbf{u},\mathbf{t}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$, we will write $\zeta_{\mathbf{t},\mathbf{u}}$ to denote the quantity $\zeta_{\mathbf{t},\mathbf{u}}\triangleq\prod_{i\in\mathcal{C}}\beta_{\mathbf{t}_{\pi(\mathcal{C},i)},\mathbf{u}_{\pi(\mathcal{C},i)}+1}$. For any $\mathbf{u},\mathbf{z}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u}<\mathbf{z}$, we will define a path M to be a sequence of vectors $\mathbf{z}_1>\mathbf{z}_2>\cdots>\mathbf{z}_m$ such that $\mathbf{z}_1,\mathbf{z}_2,\dots,\mathbf{z}_m\in(\mathbb{Z}^+)^n$, $\mathbf{z}_1=\mathbf{z}$ and $\mathbf{z}_m=\mathbf{u}$. Let $\mathcal{M}(\mathbf{z},\mathbf{u})$ be the set of all paths starting from $\mathbf{z}$ and ending at $\mathbf{u}$. We will also write a path $\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})$ uniquely as a set of $m-1$ ordered tuples $\{(\mathbf{z}_1,\mathbf{z}_2),(\mathbf{z}_2,\mathbf{z}_3),\dots,(\mathbf{z}_{m-1},\mathbf{z}_m)\}$ where each tuple consists of adjacent vectors in the path sequence. We will also write $\mathcal{T}(\mathsf{M})\equiv\{\mathbf{z}_1,\mathbf{z}_2,\dots,\mathbf{z}_m\}$ to denote the set of elements in the path.

B. Detailed Proofs

Consider a set of unknown k-sparse vectors $\mathcal{V}\equiv\{\mathbf{v}^{(1)},\dots,\mathbf{v}^{(\ell)}\}$ that parameterize the generative process $\mathcal{P}_n$. Recall that a sample $\mathbf{x}\sim\mathcal{P}_n$ is generated as follows:

$$t\sim_{\mathsf{Unif}}[\ell] \text{ and } \mathbf{x}_i\mid t\sim\mathbf{P}(\mathbf{v}_i^{(t)}) \text{ independently } \forall i\in[n].$$

In other words, $\mathbf{x}$ is generated according to a uniform mixture of distributions each having a sparse unknown parameter vector. It is important to note that conditioned on $t\in[\ell]$, the entries of $\mathbf{x}$ are independently generated.

Lemma 8: For each fixed set $\mathcal{C}\subseteq[n]$ and each vector $\mathbf{t}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$, we must have

$$\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{t}_{\pi(\mathcal{C},i)}} = \frac{1}{\ell}\sum_{\mathbf{u}\le\mathbf{t}}\zeta_{\mathbf{t},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big).$$

Proof: We will have

$$\begin{aligned}
&\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{t}_{\pi(\mathcal{C},i)}}\\
&=\frac{1}{\ell}\sum_{j\in[\ell]}\Big(\prod_{i\in\mathcal{C}}q_{\mathbf{t}_{\pi(\mathcal{C},i)}}(\mathbf{v}_i^{(j)})\Big)\\
&=\frac{1}{\ell}\sum_{j\in[\ell]}\Big(\prod_{i\in\mathcal{C}}\Big(\sum_{s\in[\mathbf{t}_{\pi(\mathcal{C},i)}+1]}\beta_{\mathbf{t}_{\pi(\mathcal{C},i)},s}(\mathbf{v}_i^{(j)})^{s-1}\Big)\Big).
\end{aligned}$$

From the above equations, note that each outer summand is a product of polynomials in $\mathbf{v}_i^{(j)}$ for a fixed j. Consider any vector $\mathbf{u}\le\mathbf{t}$ where $\mathbf{u}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ - for each such unique vector $\mathbf{u}$, we will obtain a corresponding monomial (fixing j) on expansion. Thus, expanding the polynomial and using the fact that $\zeta_{\mathbf{t},\mathbf{u}}=\prod_{i\in\mathcal{C}}\beta_{\mathbf{t}_{\pi(\mathcal{C},i)},\mathbf{u}_{\pi(\mathcal{C},i)}+1}$ is the coefficient of the monomial $\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ for all $j\in[\ell]$, we obtain proof of the lemma. □

Lemma 9: For each fixed set $\mathcal{C} \subseteq [n]$ and each vector $\mathbf{t} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$, we can compute $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{t}_{\pi(\mathcal{C},i)}}$ provided for all $\mathbf{u} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u} \le \mathbf{t}$, the quantities $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ are pre-computed.

Proof: We will prove this lemma by induction. For the base case, we have from Lemma 8 that $\ell\mathbb{E}\mathbf{x}_i = \beta_{1,2}\sum_{j\in[\ell]}\mathbf{v}_i^{(j)} + \beta_{1,1}$. Hence $\sum_{j\in[\ell]}\mathbf{v}_i^{(j)}$ can be computed from $\mathbb{E}\mathbf{x}_i$ by using the following equation:

$$\sum_{j\in[\ell]}\mathbf{v}_i^{(j)} = \frac{1}{\beta_{1,2}}\Big(\ell\mathbb{E}\mathbf{x}_i - \beta_{1,1}\Big).$$

Now suppose for all vectors $\mathbf{u} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u} \le \mathbf{t}$, the lemma statement is true. Consider another vector $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ such that there exists an index $j \in |\mathcal{C}|$ for which $\mathbf{z}_j = \mathbf{t}_j + 1$ and $\mathbf{z}_i = \mathbf{t}_i$ for all $i \neq j$. From the statement of Lemma 8, we know that

$$\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}} = \frac{1}{\ell}\sum_{\mathbf{u}\le\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big)$$

where $\zeta_{\mathbf{z},\mathbf{u}} = \prod_{i\in\mathcal{C}}\beta_{\mathbf{z}_{\pi(\mathcal{C},i)},\mathbf{u}_{\pi(\mathcal{C},i)}+1}$. From our induction hypothesis, we have already computed $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{u} < \mathbf{z}$ (the set $\{\mathbf{u} \in (\mathbb{Z}^+)^{|\mathcal{C}|} \mid \mathbf{u} < \mathbf{z}\}$ is equivalent to the set $\{\mathbf{u} \in (\mathbb{Z}^+)^{|\mathcal{C}|} \mid \mathbf{u} \le \mathbf{t}\}$). Since $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ is already pre-computed, we can compute $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ as follows:

$$\begin{aligned}&\ell\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}} - \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big)\\&= \zeta_{\mathbf{z},\mathbf{z}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}\Big).\end{aligned}$$

This completes the proof of the lemma. □

For a set of vectors $\mathcal{U}$, recall that we defined $\mathcal{S}_{\mathcal{U}}(i) \triangleq \{\mathbf{u} \in \mathcal{U} : \mathbf{u}_i \neq 0\}$ to denote the multi-set of vectors in $\mathcal{U}$ that has a non-zero entry at the i^{th} index. We will use the aforementioned notation for the set of unknown vectors $\mathcal{V}$ - we will ignore the subscript $\mathcal{V}$ for simplicity of notation.

Lemma 10: For each fixed set $\mathcal{C} \subseteq [n]$, we can compute $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right|$ provided for all $p \in [\ell]$, the quantity $\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p$ is pre-computed.

Proof: Let us fix a particular subset $\mathcal{C} \subseteq [n]$. Now, let us define the quantity

$$\mathsf{A}_{\mathcal{C},t} = \sum_{\substack{\mathcal{C}'\subseteq[\ell]\\|\mathcal{C}'|=t}}\prod_{\substack{i\in\mathcal{C}\\j\in\mathcal{C}'}}(\mathbf{v}_i^{(j)})^2$$

Notice that $\mathsf{A}_{\mathcal{C},t} > 0$ if and only if there exists a subset $\mathcal{C}' \subseteq [\ell], |\mathcal{C}'| = t$ such that $\mathbf{v}_i^{(j)} \neq 0$ for all $i \in \mathcal{C}, j \in \mathcal{C}'$. Hence, the maximum value of t such that $\mathsf{A}_{\mathcal{C},t} > 0$ is the number of unknown vectors in $\mathcal{V}$ having non-zero value in all the indices in $\mathcal{C}$. In other words, we have that

$$\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right| = \max_{t\in[\ell]} t\cdot\mathbf{1}[\mathsf{A}_{\mathcal{C},t} > 0].$$

Let $t^\star$ be the maximum value of t for which $\mathsf{A}_{\mathcal{C},t} > 0$. We will have $\mathsf{A}_{\mathcal{C},t^\star} \ge \delta^{2\ell|\mathcal{C}|}$ (from Assumption 1). It is easy to recognize $\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p$ as the power sum polynomial of degree p in the variables $\{\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\}_{\mathbf{v}\in\mathcal{V}}$. On the other hand, $\mathsf{A}_{\mathcal{C},t}$ is the elementary symmetric polynomial of degree t in the variables $\{\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\}_{\mathbf{v}\in\mathcal{V}}$. We can use Newton's identities to state that for all $t \in [\ell]$,

$$t\mathsf{A}_{\mathcal{C},t} = \sum_{p=1}^{t}(-1)^{p+1}\mathsf{A}_{\mathcal{C},t-p}\Big(\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p\Big)$$

using which, we can recursively compute $\mathsf{A}_{\mathcal{C},t}$ for all $t \in [\ell]$ if we were given $\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p$ as input for all $p \in [\ell]$. We can also express $\mathsf{A}_{\mathcal{C},t}$ as a complete exponential Bell polynomial B_t

$$\begin{aligned}\mathsf{A}_{\mathcal{C},t} = \frac{(-1)^n}{n!}\mathsf{B}_t\Big(&-\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2, -1!\Big(\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^2,\\&-2!\Big(\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^3,\dots,-(t-1)!\Big(\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^t\Big).\end{aligned}$$

□

We are now ready to prove Lemma 6.

lemma (Restatement of Lemma 6): Suppose Assumption 1 is true. Fix any set $\mathcal{C} \subseteq [n]$. Let

$$\begin{aligned}\Phi &\triangleq \frac{\delta^{2\ell|\mathcal{C}|}}{2\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(\ell-1)}\ell!}\\&\quad\times\left(\max_{\mathbf{z}\le2\ell\mathbf{1}_{|\mathcal{C}|}}\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\right)^{-1}\\g_{\ell,\mathcal{V}} &\triangleq \frac{\max_{\mathbf{z}\le2\ell\mathbf{1}_{|\mathcal{C}|}}\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}\end{aligned}$$

where $g_{\ell,\mathcal{V}}$ is a constant that is independent of k and n but depends on ℓ. There exists an algorithm (see Algorithm 6) that can compute $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}_{\mathcal{V}}(i)\right|$ exactly for the set $\mathcal{C}$ with probability at least $1-\gamma$ using $O\Big(\log(\gamma^{-1}(2\ell)^{|\mathcal{C}|})g_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_n$.

Proof: Fix a particular set $\mathcal{C} \subseteq [n]$. Suppose, for every vector $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z} \le 2\ell\mathbf{1}_{|\mathcal{C}|}$, we compute an estimate $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ such that $\left|\widehat{U}^{\mathbf{z}} - \mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}\right| \le \Phi_{\mathbf{z}}$ where $\Phi_{\mathbf{z}}$ is going to be determined later. Recall that in Lemma 10, we showed

$$\begin{aligned}&\ell\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}} - \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{u}_{\pi(\mathcal{C},i)}}\Big)\\&= \zeta_{\mathbf{z},\mathbf{z}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}\Big).\end{aligned}\quad(2)$$

Using the computed $\widehat{U}^{\mathbf{z}}$'s, we can recursively compute an estimate $\widehat{V}^{\mathbf{z}}$ of $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z} \le 2\ell\mathbf{1}_{|\mathcal{C}|}$. Let us denote the error in estimation by $\epsilon_{\mathbf{z}}$ i.e. we have $\left|\widehat{V}^{\mathbf{z}} - \sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}\right| \le \epsilon_{\mathbf{z}}$. Now, we prove the following claim.

Claim 1: We must have

$$\epsilon_{\mathbf{z}} \le \frac{\ell\Phi_{\mathbf{z}}}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\Phi_{\mathbf{u}}\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}$$

Proof: We will prove this lemma by induction. Let $\mathbf{e}_i$ be the standard basis vector having a non-zero entry at the i^{th} index and is zero everywhere else. For the base case, we have from Lemma 8 that $\ell\mathbb{E}\mathbf{x}_i = \beta_{1,2}\sum_{j\in[\ell]}\mathbf{v}_i^j + \beta_{1,1}$. Since β's are known we are going to compute $\widehat{\mathbf{V}}^{\mathbf{e}_i}$ by solving the equation $\ell\widehat{\mathbf{U}}^{\mathbf{e}_i} = \beta_{1,2}\widehat{\mathbf{V}}^{\mathbf{e}_i} + \beta_{1,1}$. Therefore, we must have (note that by definition, $\zeta_{\mathbf{e}_i,\mathbf{e}_i} = \beta_{1,2}$)

$$\ell\mathbb{E}\mathbf{x}_i - \ell\widehat{U}^{\mathbf{e}_i} = \beta_{1,2}\Big(\sum_{j\in[\ell]}\mathbf{v}_i^j - \widehat{V}^{\mathbf{e}_i}\Big)$$

$$\implies \ell\Phi_{\mathbf{e}_i} \geq \beta_{1,2}\epsilon_{\mathbf{e}_i} = \beta_{1,2}\zeta_{\mathbf{e}_i,\mathbf{e}_i}.$$

This completes the proof of the base case. Now, from definition, (recall that $\zeta_{\mathbf{z},\mathbf{u}} = \prod_{i\in\mathcal{C}}\beta_{\mathbf{z}_{\pi(i)},\mathbf{u}_{\pi(i)}+1}$), we have $\zeta_{\mathbf{e}_i,\mathbf{e}_i} = \beta_{1,2}$ which completes the proof of the base case. Now suppose for all vectors $\mathbf{u} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u} \leq \mathbf{t}$, the lemma statement is true. Consider another vector $\mathbf{z} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$ such that there exists an index $j \in |\mathcal{C}|$ for which $\mathbf{z}_j = \mathbf{t}_j + 1$ and $\mathbf{z}_i = \mathbf{t}_i$ for all $i \neq j$. From the statement of Lemma 8, we know that

$$\ell\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(i)}} - \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^j)^{\mathbf{u}_{\pi(i)}}\Big)$$
$$= \zeta_{\mathbf{z},\mathbf{z}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^j)^{\mathbf{z}_{\pi(i)}}\Big).$$

Hence, we must have

$$\Big(\ell\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(i)}} - \ell\widehat{U}^{\mathbf{z}}\Big) - \Big(\sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^j)^{\mathbf{u}_{\pi(i)}}$$
$$- \widehat{V}^{\mathbf{u}}\Big)\Big) = \zeta_{\mathbf{z},\mathbf{z}}\cdot\Big(\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^j)^{\mathbf{z}_{\pi(i)}} - \widehat{V}^{\mathbf{z}}\Big)$$
$$\implies \zeta_{\mathbf{z},\mathbf{z}}\epsilon_{\mathbf{z}} \leq \ell\Phi_{\mathbf{z}} + \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\epsilon_{\mathbf{u}}.$$

Now, by using our induction hypothesis, we must have

$$\zeta_{\mathbf{z},\mathbf{z}}\epsilon_{\mathbf{z}} \leq \ell\Phi_{\mathbf{z}} + \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\Bigg(\frac{\ell\Phi_{\mathbf{u}}}{\zeta_{\mathbf{u},\mathbf{u}}}$$
$$+ \sum_{\mathbf{v}<\mathbf{u}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{u},\mathbf{v})}\frac{\ell\Phi_{\mathbf{v}}\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Bigg)$$
$$\implies \epsilon_{\mathbf{z}} \leq \frac{\ell\Phi_{\mathbf{z}}}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\zeta_{\mathbf{z},\mathbf{u}}\Bigg(\frac{\ell\Phi_{\mathbf{u}}}{\zeta_{\mathbf{z},\mathbf{z}}\zeta_{\mathbf{u},\mathbf{u}}}$$
$$+ \sum_{\mathbf{v}<\mathbf{u}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{u},\mathbf{v})}\frac{\ell\Phi_{\mathbf{v}}\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\zeta_{\mathbf{z},\mathbf{z}}\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Bigg)$$
$$\implies \epsilon_{\mathbf{z}} \leq \frac{\ell\Phi_{\mathbf{z}}}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\Phi_{\mathbf{u}}\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}.$$

This completes the proof of the claim. □

Hence, for fixed $\Phi_{\mathbf{z}} = \Phi$ for all $\mathbf{z} \leq 2\ell\mathbf{1}_{|\mathcal{C}|}$, we get

$$\epsilon_{\mathbf{z}} \leq \Phi\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big).$$

For a fixed Φ, let us write ϵ to denote the following quantity:

$$\epsilon \triangleq \max_{\mathbf{z}\leq 2\ell\mathbf{1}_{|\mathcal{C}|}}\Phi\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{Q\in\mathcal{Q}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in Q}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(Q)}\zeta_{\mathbf{r},\mathbf{r}}}\Big)$$

Consider a fixed subset of indices $\mathcal{C} \subseteq [n]$ and a fixed vector $\mathbf{t} \in (\mathbb{Z}^+)^{|\mathcal{C}|}$. Using the fact $\max_{\mathbf{v}\in\mathcal{V},i\in[n]}\mathbf{v}_i^2 \leq R^2$, we have that

$$\frac{1}{\ell}\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p \leq R^{2p|\mathcal{C}|}$$

and

$$\mathsf{A}_{\mathcal{C},t} = \sum_{\substack{\mathcal{C}'\subseteq[\ell]\\ |\mathcal{C}'|=t}}\prod_{\substack{i\in\mathcal{C}\\ j\in\mathcal{C}'}}(\mathbf{v}_i^{(j)})^2 \leq \binom{\ell}{t}R^{2(t+|\mathcal{C}|)} \leq 2^\ell R^{2(t+|\mathcal{C}|)}.$$

We can compute an estimate $\widehat{\mathsf{A}}_{\mathcal{C},t}$ of $\mathsf{A}_{\mathcal{C},t}$ by using $\widehat{V}^{2p\mathbf{1}_{|\mathcal{C}|}}$ in the following set of recursive equations

$$t\widehat{\mathsf{A}}_{\mathcal{C},t} = \sum_{p=1}^{t}(-1)^{p+1}\widehat{\mathsf{A}}_{\mathcal{C},t-p}\widehat{V}^{2p\mathbf{1}_{|\mathcal{C}|}}.$$

Claim 2:

$$\Big|\widehat{\mathsf{A}}_{\mathcal{C},t} - \mathsf{A}_{\mathcal{C},t}\Big| \leq \epsilon\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(t-1)}t!$$

for all $t \in [\ell]$.

Proof: We will prove this claim by induction. For the base case i.e. $t = 1$, notice that

$$\Big|\widehat{\mathsf{A}}_{\mathcal{C},1} - \mathsf{A}_{\mathcal{C},1}\Big| \leq \Big|\widehat{V}^{2\mathbf{1}_{|\mathcal{C}|}} - \sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big| \leq \epsilon.$$

Now, suppose for all $t \leq k$, the following holds true:

$$\Big|\widehat{\mathsf{A}}_{\mathcal{C},t} - \mathsf{A}_{\mathcal{C},t}\Big| \leq \epsilon\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{t-1}t!.$$

For ease of notation, let us denote $a = 3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})$. In that case, for $t = k+1$, we must have

$$t\Big|\widehat{\mathsf{A}}_{\mathcal{C},t} - \mathsf{A}_{\mathcal{C},t}\Big|$$
$$\leq \sum_{p\leq t}\Big|\widehat{\mathsf{A}}_{\mathcal{C},t-p}\widehat{V}^{2p\mathbf{1}_{|\mathcal{C}|}} - \mathsf{A}_{\mathcal{C},t-p}\cdot\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p\Big|$$
$$\leq \Big|\widehat{V}^{2\mathbf{1}_{|\mathcal{C}|}} - \sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^{(k+1)}\Big| + \sum_{p\leq t-1}\Big|\epsilon a^{t-2}(t-1)!$$
$$\cdot\sum_{\mathbf{v}\in\mathcal{V}}\Big(\prod_{i\in\mathcal{C}}\mathbf{v}_i^2\Big)^p + \epsilon\cdot\mathsf{A}_{\mathcal{C},t-p} + \epsilon^2a^{t-2}(t-1)!\Big|$$
$$\leq \epsilon + \sum_{p\leq t-1}\Big|\epsilon a^{t-2}(t-1)!\ell R^{2\ell|\mathcal{C}|}$$
$$+ \epsilon\cdot 2^\ell R^{2(\ell+|\mathcal{C}|)} + \epsilon^2a^{t-2}(t-1)!\Big|$$
$$\leq \epsilon + \sum_{p\leq t-1}\epsilon a^{t-1}(t-1)! \leq \epsilon a^{(t-1)}t!.$$

Hence, $\Big|\widehat{\mathsf{A}}_{\mathcal{C},t} - \mathsf{A}_{\mathcal{C},t}\Big| \leq \epsilon a^{t-1}t!$ thus proving our claim. □

Hence, to identify $t^\star$ correctly, we must have

$$\epsilon\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(\ell-1)}\ell! \le \frac{\delta^{2\ell|\mathcal{C}|}}{2}$$

$$\implies \Phi \le \frac{\delta^{2\ell|\mathcal{C}|}}{2\Big(3\max(\ell R^{2\ell|\mathcal{C}|}, 2^\ell R^{\ell+|\mathcal{C}|})\Big)^{(\ell-1)}\ell!}$$

$$\cdot\left(\max_{\mathbf{z}\le 2p\mathbf{1}_{|\mathcal{C}|}} \frac{1}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})} \frac{\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\right)^{-1}$$

where we inserted the definition of Φ. Therefore, for every vector $\mathbf{z}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z}\le 2\ell\mathbf{1}_{|\mathcal{C}|}$, in order to compute $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ such that $\left|\widehat{U}^{\mathbf{z}} - \mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}\right| \le \Phi$, the number of samples that is sufficient with probability $1-\gamma$ is going to be

$$O\Big(\log(\gamma^{-1}(2\ell)^{|\mathcal{C}|})\frac{\max_{\mathbf{z}\le 2\ell\mathbf{1}_{|\mathcal{C}|}}\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}\Big).$$

□

Theorem (Restatement of Theorem 1): Let $\mathcal{V}$ be a set of ℓ unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1. Let $\mathcal{F}_m = \mathcal{Q}_1([n])\cup\mathcal{Q}_m(\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v}))$ and

$$\Phi_m = \frac{\delta^{2\ell m}}{2\Big(3\ell\max(R^{2\ell m}, 2^\ell R^{\ell+m})\Big)^{(\ell-1)}\ell!}\cdot$$

$$\left(\max_{\mathbf{z}\le 2\ell\mathbf{1}_m}\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\right)^{-1}$$

$$f_{\ell,\mathcal{V}} = \max_{\substack{\mathbf{z}\le 2\ell\mathbf{1}_{\log\ell+1}\\ \mathcal{C}\in\mathcal{F}_{\log\ell+1}}}\frac{\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2_{\log\ell+1}}$$

where $f_{\ell,\mathcal{V}}$ is a constant that is independent of sparsity k and ambient dimension n but depends on ℓ. Then, there exists an algorithm (see Algorithm 6 and 2) that achieves Exact Support Recovery with probability at least $1-\gamma$ using $O\Big(\log(\gamma^{-1}(2\ell)^{\log\ell+1}(n+(\ell k)^{\log\ell+1}))f_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_n$.

Proof: The proof follows directly from Corollary 1 and Lemma 6. □

Corollary (Restatement of Corollary 3): Consider the mean estimation problem where $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}_n}[\mathbf{x}_i \mid t=j] = \mathbf{v}_i^{(j)}$. Let $\mathcal{V}$ be a set of $\ell = O(1)$ unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1 and $f_{\ell,\mathcal{V}}$ be as defined in Theorem 2. Then, there exists an algorithm (see Algorithm 6 and 2) that with probability at least $1-\gamma$, achieves Exact Support Recovery using $O\Big(\mathrm{poly}\log(n\gamma^{-1})\mathrm{poly}(\delta R^{-1})f_{\ell,\mathcal{V}}\Big)$ samples generated according to $\mathcal{P}_n$.

Proof: We can re-scale the samples (dividing them by R) so that Assumption 1 will be satisfied with $\delta' = \delta/R$ and $R'\le 1$. Since ℓ is a constant, $\Phi_{\log\ell} = O(\mathsf{poly}(\delta R^{-1}))$. Therefore, the corollary follows from Theorem 1. □

lemma (Restatement of Lemma 7): Suppose Assumption 1 is true. Fix any set $\mathcal{C}\subseteq[n]$. Let

$$\Phi \triangleq \max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}}\frac{\delta^{2|\mathcal{C}|}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}$$

$$h_{\ell,\mathcal{V}} \triangleq \frac{\max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}}\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi^2}$$

where $h_{\ell,\mathcal{V}}$ is a constant independent of k and n but depends on ℓ. There exists an algorithm (see Algorithm 7) that can compute if $\left|\bigcap_{i\in\mathcal{C}}\mathcal{S}(i)\right| > 0$ correctly for the set $\mathcal{C}$ with probability at least $1-\gamma$ using $O(h_{\ell,\mathcal{V}}\log\gamma^{-1})$ samples generated according to $\mathcal{P}_n$.

Proof: For a fixed ordered set $\mathcal{C}\subseteq[n]$, consider the statistic $\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2$. If $\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2 > 0$, then $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| > 0$ and otherwise, if $\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2 = 0$, then $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| = 0$. Hence it suffices to estimate correctly if $\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2 > 0$ or not. From Lemma 9, we know that for each set $\mathcal{C}\subseteq[n]$, we can compute $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^2$ provided for all $\mathbf{u}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{u}\le 2\mathbf{1}_{|\mathcal{C}|}$, the quantity $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{u}_{\pi(\mathcal{C},i)}}$ is pre-computed.

Suppose, for every vector $\mathbf{z}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}$, we compute an estimate $\widehat{U}^{\mathbf{z}}$ of $\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ such that $\left|\widehat{U}^{\mathbf{z}} - \mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{\mathbf{z}_{\pi(\mathcal{C},i)}}\right| \le \Phi$ where Φ is going to be determined later. Using the computed $\widehat{U}^{\mathbf{z}}$'s, we can compute an estimate $\widehat{V}^{\mathbf{z}}$ of $\sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}$ for all $\mathbf{z}\in(\mathbb{Z}^+)^{|\mathcal{C}|}$ satisfying $\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}$. As before, let us denote the error in estimation by $\epsilon_{\mathbf{z}}$ i.e. we have $\left|\widehat{V}^{\mathbf{z}} - \sum_{j\in[\ell]}\prod_{i\in\mathcal{C}}(\mathbf{v}_i^{(j)})^{\mathbf{z}_{\pi(\mathcal{C},i)}}\right| \le \epsilon_{\mathbf{z}}$. Note that we showed in Lemma 10 that for fixed Φ, we get for all $\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}$,

$$\epsilon_{\mathbf{z}} \le \Phi\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big).$$

Note that the minimum value of $\sum_{\mathbf{v}\in\mathcal{V}}\prod_{i\in\mathcal{C}}\mathbf{v}_i^2$ is at least $\delta^{2|\mathcal{C}|}$ and therefore, it suffices $\epsilon_{\mathbf{z}}$ to be less than $\delta^{2|\mathcal{C}|}/2$. Hence, it is sufficient if

$$\Phi \le \max_{\mathbf{z}\le 2\mathbf{1}_{|\mathcal{C}|}}\frac{\delta^{2|\mathcal{C}|}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}.$$

Now, we use Lemma 14 to complete the proof of the lemma (similar to Lemma 10)

□

thmu (Restatement of Theorem 2): Let $\mathcal{V}$ be a set of unknown vectors in $\mathbb{R}^n$ satisfying Assumption 1. Let $\mathcal{F}_m = \mathcal{Q}_1([n])\cup\mathcal{Q}_m(\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v}))$ and

$$\Phi_m = \max_{\mathbf{z}\le 2\mathbf{1}_m}\frac{\delta^{2m}}{2}\Big(\frac{\ell}{\zeta_{\mathbf{z},\mathbf{z}}} + \sum_{\mathbf{u}<\mathbf{z}}\sum_{\mathsf{M}\in\mathcal{M}(\mathbf{z},\mathbf{u})}\frac{\ell\prod_{(\mathbf{r},\mathbf{s})\in\mathsf{M}}\zeta_{\mathbf{r},\mathbf{s}}}{\prod_{\mathbf{r}\in\mathcal{T}(\mathsf{M})}\zeta_{\mathbf{r},\mathbf{r}}}\Big)^{-1}$$

$$h'_{\ell,\mathcal{V}} \triangleq \max_{\substack{\mathbf{z}\le 2\mathbf{1}_\ell\\ \mathcal{C}\in\mathcal{F}_\ell}}\frac{\mathbb{E}\prod_{i\in\mathcal{C}}\mathbf{x}_i^{2\mathbf{z}_{\pi(\mathcal{C},i)}}}{\Phi_\ell^2}$$

where $h'_{\ell,\mathcal{V}}$ is a constant independent of k and n but depends on ℓ. Accordingly, there exists an algorithm (see Algorithm 7 and 4) that achieves maximal support recovery with probability at least $1-\gamma$ using $O\Big(h'_{\ell,\mathcal{V}}\log(\gamma^{-1}(n+(\ell k)^\ell))\Big)$ samples generated from $\mathcal{P}_n$.

Proof: The proof follows from Lemma 7 and Corollary 2. □

APPENDIX B
MISSING PROOFS FROM SECTION III

Proof of Lemma 2 when $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ is provided: Suppose we are given $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all sets $\mathcal{C}\subseteq[n]$ satisfying $|\mathcal{C}|\le s$. Notice that the set $\cap_{i\in\mathcal{C}}\mathcal{S}(i)$ is equivalent to the set $\mathsf{occ}(C,\mathbf{1}_{|C|})$ or the number of unknown vectors in $\mathcal{V}$ whose restriction to the indices in $\mathcal{C}$ is the all one vector and in particular, $\mathsf{occ}((i),1)=\mathcal{S}(i)$. Recall the principle of inclusion and exclusion - for any family of t sets $\mathcal{A}_1,\mathcal{A}_2,\dots,\mathcal{A}_t$, we must have

$$\left|\bigcup_{i=1}^{t}\mathcal{A}_i\right|=\sum_{u=1}^{t}(-1)^{u+1}\sum_{1\le i_1<i_2<\dots<i_u\le t}\left|\bigcap_{b=1}^{u}\mathcal{A}_{i_b}\right|.$$

We now show using induction on s that the quantities $\left\{\left|\bigcup_{i\in\mathcal{S}}\mathsf{occ}((i),1)\right|\ \forall\ \mathcal{T}\subseteq[n],|\mathcal{T}|\le s\right\}$ are sufficient to compute $|\mathsf{occ}(C,\mathbf{a})|$ for all subsets C of indices of size at most s, and any binary vector $\mathbf{a}\in\{0,1\}^{\le s}$.

Base case ($t=1$):

The base case follows since we can infer $|\mathsf{occ}((i),0)|=\ell-|\mathsf{occ}((i),1)|$ from $|\mathsf{occ}((i),1)|$ for all $i\in[n]$.

Inductive Step: Let us assume that the statement is true for $r<s$ i.e., we can compute $|\mathsf{occ}(\mathcal{C},\mathbf{a})|$ for all subsets $\mathcal{C}$ satisfying $|\mathcal{C}|\le r$ and any binary vector $\mathbf{a}\in\{0,1\}^{\le r}$ from the quantities $\left\{\left|\bigcup_{i\in\mathcal{S}}\mathsf{occ}((i),1)\right|\ \forall\ \mathcal{T}\subseteq[n],|\mathcal{T}|\le r\right\}$ provided as input. Now, we prove that the statement is true for $r+1$ under the induction hypothesis. Note that we can also rewrite $\mathsf{occ}(\mathcal{C},\mathbf{a})$ for each set $\mathcal{C}\subseteq[n],\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$ as

$$\mathsf{occ}(\mathcal{C},\mathbf{a})=\bigcap_{j\in\mathcal{C}'}\mathcal{S}(j)\bigcap_{j\in\mathcal{C}\setminus\mathcal{C}'}\mathcal{S}(j)^c$$

where $\mathcal{C}'\subseteq\mathcal{C}$ corresponds to the indices in $\mathcal{C}$ for which the entries in $\mathbf{a}$ is $\mathtt{1}$. Fix any set $i_1,i_2,\dots,i_{r+1}\in[n]$. Then we can compute $\left|\bigcap_{b=1}^{r+1}\mathcal{S}(i_b)\right|$ using the following equation:

$$(-1)^{r+3}\left|\bigcap_{b=1}^{r+1}\mathcal{S}(i_b)\right|=\sum_{u=1}^{r}(-1)^{u+1}\cdot\sum_{\substack{j_1,j_2,\dots,j_u\in\{i_1,i_2,\dots,i_{r+1}\}\\ j_1<j_2<\dots<j_u}}\left|\bigcap_{b=1}^{u}\mathcal{S}(j_b)\right|-\left|\bigcup_{b=1}^{r+1}\mathcal{S}(i_b)\right|.$$

Finally for each proper subset $\mathcal{Y}\subset\{i_1,i_2,\dots,i_{r+1}\}$, we can compute $\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)^c\right|$ using the following set of equations:

$$\begin{aligned}&\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)^c\right|\\ &=\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\Big(\bigcup_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)\Big)^c\right|\\ &=\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\right|-\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\Big(\bigcup_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)\Big)\right|\\ &=\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\right|-\left|\bigcup_{i_b\in\mathcal{Y}}\Big(\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\mathcal{S}(i_b)\Big)\right|.\end{aligned}$$

The first term is already pre-computed and the second term is again a union of intersection of sets. for each $j_b\in\mathcal{Y}$, let us define $\mathcal{H}(j_b):=\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap\mathcal{S}(j_b)$. Therefore we have

$$\left|\bigcup_{j_b\in\mathcal{Y}}\mathcal{H}(j_b)\right|=\sum_{u=1}^{|\mathcal{Y}|}(-1)^{u+1}\sum_{\substack{j_1,j_2,\dots,j_u\in\mathcal{Y}\\ j_1<j_2<\dots<j_u}}\left|\bigcap_{b=1}^{u}\mathcal{H}(j_b)\right|.$$

We can compute $\left|\bigcup_{j_b\in\mathcal{Y}}\mathcal{H}(j_b)\right|$ because the quantities on the right hand side of the equation have already been pre-computed (using our induction hypothesis). Therefore, the lemma is proved.

□

Proof of Lemma 2 when $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ is provided: Suppose we are given $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all sets $\mathcal{V}\subseteq[n]$ satisfying $|\mathcal{V}|\le s$. We will omit the subscript $\mathcal{V}$ from hereon for simplicity. As in Lemma 2, the set $\cap_{i\in\mathcal{C}}\mathcal{S}(i)$ is equivalent to the set $\mathsf{occ}(C,\mathbf{1}_{|C|})$ or the number of unknown vectors in $\mathcal{V}$ whose restriction to the indices in $\mathcal{C}$ is the all one vector and in particular, $\mathsf{occ}((i),1)=\mathcal{S}(i)$. We will re-use the equation that for t sets $\mathcal{A}_1,\mathcal{A}_2,\dots,\mathcal{A}_t$, we must have

$$\left|\bigcup_{i=1}^{t}\mathcal{A}_i\right|=\sum_{u=1}^{t}(-1)^{u+1}\sum_{1\le i_1<i_2<\dots<i_u\le t}\left|\bigcap_{b=1}^{u}\mathcal{A}_{i_b}\right|.$$

We now show using induction on s that the quantities $\left\{\left|\bigcap_{i\in\mathcal{S}}\mathsf{occ}((i),1)\right|\ \forall\ \mathcal{T}\subseteq[n],|\mathcal{T}|\le s\right\}$ are sufficient to compute $|\mathsf{occ}(C,\mathbf{a})|$ for all subsets C of indices of size at most s, and any binary vector $\mathbf{a}\in\{0,1\}^{\le s}$.

Base case ($t=1$):

The base case follows since we can infer $|\mathsf{occ}((i),0)|=\ell-|\mathsf{occ}((i),1)|$ from $|\mathsf{occ}((i),1)|$ for all $i\in[n]$.

Inductive Step: Let us assume that the statement is true for $r<s$ i.e., we can compute $|\mathsf{occ}(\mathcal{C},\mathbf{a})|$ for all subsets $\mathcal{C}$ satisfying $|\mathcal{C}|\le r$ and any binary vector $\mathbf{a}\in\{0,1\}^{\le r}$ from the quantities $\left\{\left|\bigcap_{i\in\mathcal{S}}\mathsf{occ}((i),1)\right|\ \forall\ \mathcal{T}\subseteq[n],|\mathcal{T}|\le r\right\}$ provided as input. Now, we prove that the statement is true for $r+1$ under the induction hypothesis. Note that we can also rewrite $\mathsf{occ}(\mathcal{C},\mathbf{a})$ for any set $\mathcal{C}\subseteq[n],\mathbf{a}\in\{0,1\}^{|\mathcal{C}|}$ as

$$\mathsf{occ}(\mathcal{C},\mathbf{a})=\bigcap_{j\in\mathcal{C}'}\mathcal{S}(j)\bigcap_{j\in\mathcal{C}\setminus\mathcal{C}'}\mathcal{S}(j)^c$$

where $\mathcal{C}'\subseteq\mathcal{C}$ corresponds to the indices in $\mathcal{C}$ for which the entries in $\mathbf{a}$ is $\mathtt{1}$. Fix any set $i_1,i_2,\dots,i_{r+1}\in[n]$. Then we can compute $\left|\bigcup_{b=1}^{r+1}\mathcal{S}(i_b)\right|$ using the following equation:

$$\left|\bigcup_{b=1}^{r+1}\mathcal{S}(i_b)\right|=\sum_{u=1}^{r+1}(-1)^{u+1}\sum_{\substack{j_1,j_2,\dots,j_u\in\{i_1,i_2,\dots,i_{r+1}\}\\ j_1<j_2<\dots<j_u}}\left|\bigcap_{b=1}^{u}\mathcal{S}(j_b)\right|.$$

Finally for any proper subset $\mathcal{Y}\subset\{i_1,i_2,\dots,i_{r+1}\}$, we can compute $\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)^c\right|$ using the following set of equations:

$$\left|\bigcap_{i_b\notin\mathcal{Y}}\mathcal{S}(i_b)\bigcap_{i_b\in\mathcal{Y}}\mathcal{S}(i_b)^c\right|$$

$$= \Big|\bigcap_{i_b\notin\mathcal{Y}} \mathcal{S}(i_b) \bigcap \Big(\bigcup_{i_b\in\mathcal{Y}} \mathcal{S}(i_b)\Big)^c\Big|$$
$$= \Big|\bigcap_{i_b\notin\mathcal{Y}} \mathcal{S}(i_b)\Big| - \Big|\bigcap_{i_b\notin\mathcal{Y}} \mathcal{S}(i_b) \bigcap \Big(\bigcup_{i_b\in\mathcal{Y}} \mathcal{S}(i_b)\Big)\Big|$$
$$= \Big|\bigcap_{i_b\notin\mathcal{Y}} \mathcal{S}(i_b)\Big| - \Big|\bigcup_{i_b\in\mathcal{Y}} \Big(\bigcap_{i_b\notin\mathcal{Y}} \mathcal{S}(i_b) \bigcap \mathcal{S}(i_b)\Big)\Big|.$$

The first term is already pre-computed and the second term is again a union of intersection of sets. For any $i_b \in \mathcal{Y}$, let us define $\mathcal{H}(j_b) := \bigcap_{i_b\notin\mathcal{Y}} \mathcal{S}(i_b) \bigcap \mathcal{S}(j_b)$. Therefore we have

$$\Big|\bigcup_{j_b\in\mathcal{Y}} \mathcal{H}(j_b)\Big| = \sum_{u=1}^{|\mathcal{Y}|} (-1)^{u+1} \sum_{\substack{j_1,j_2,\dots,j_u\in\mathcal{Y}\\ j_1<j_2<\dots<j_u}} \Big|\bigcap_{b=1}^{u} \mathcal{H}(j_b)\Big|.$$

We can compute $\Big|\bigcup_{j_b\in\mathcal{Y}} \mathcal{H}(j_b)\Big|$ because the quantities on the right hand side of the equation have already been pre-computed (using our induction hypothesis). Therefore, the lemma is proved. □

Proof of Corollary 1: We know that all vectors $\mathbf{v} \in \mathcal{V}$ satisfy $\|\mathbf{v}\|_0 \le k$ as they are k-sparse. Therefore, in the first stage, by computing $|\mathcal{S}(i)|$ for all $i \in [n]$, we can recover the union of support of all the unknown vectors $\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v})$ by computing $\mathcal{T} = \{i \in [n] \mid \mathcal{S}(i) > 0\}$. The probability of failure in finding the union of support exactly is at most $n\gamma$. Once we recover $\mathcal{T}$, we compute $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C} \subseteq \mathcal{T}, |\mathcal{C}| \le \log\ell + 1$ (or alternatively $|\cap_{i\in\mathcal{C}} \mathcal{S}(i)|$ for all $\mathcal{C} \subseteq \mathcal{T}, |\mathcal{C}| \le \log\ell+1$). The probability of failure for this this event $(\ell k)^{\log\ell+1}\gamma$. From Lemma 1, we know that computing $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C} \subseteq [n], |\mathcal{C}| \le \log\ell + 1$ (or alternatively $|\cap_{i\in\mathcal{C}} \mathcal{S}(i)|$ for all $\mathcal{C} \subseteq \mathcal{T}, |\mathcal{C}| \le \log\ell + 1$) exactly will allow us to recover the support of all the unknown vectors in $\mathcal{V}$. However $|\cup_{i\in\mathcal{C}}\mathcal{S}(i)| = 0$ for all $\mathcal{C} \subseteq [n] \setminus \mathcal{T}$ provided $\mathcal{T}$ is computed correctly. Therefore, we can recover the support of all the unknown vectors in $\mathcal{V}$ with $\mathsf{T}\log\gamma^{-1}$ samples with probability at least $1 - ((\ell k)^{\log\ell+1} + n)\gamma$. Rewriting the previous statement so that the failure probability is γ leads to the statement of the lemma. □

Proof of Corollary 2: Again, we know that all vectors $\mathbf{v} \in \mathcal{V}$ satisfy $\|\mathbf{v}\|_0 \le k$ as they are k-sparse. Therefore, in the first stage, by computing if $|\mathcal{S}(i)| > 0$ for all $i \in [n]$, we can recover the union of support of all the unknown vectors $\cup_{\mathbf{v}\in\mathcal{V}}\mathsf{supp}(\mathbf{v})$ by computing $\mathcal{T} = \{i \in [n] \mid \mathcal{S}(i) > 0\}$. The probability of failure in finding the union of support correctly is at most $n\gamma$. Once we recover $\mathcal{T}$ correctly, we compute $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C} \subseteq \mathcal{T}, |\mathcal{C}| \le \ell$. The probability of failure for this event $(\ell k)^{\ell}\gamma$. From Lemma 5, we know that computing $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)|$ for all $\mathcal{C} \subseteq [n], |\mathcal{C}| \le \ell$ exactly will allow us to recover the support of all the unknown vectors in $\mathcal{V}$. On the other hand, we will have $|\cap_{i\in\mathcal{C}}\mathcal{S}(i)| = 0$ for all $\mathcal{C} \subseteq [n] \setminus \mathcal{T}$ provided $\mathcal{T}$ is computed correctly. Therefore, we can achieve maximal support recovery of all the unknown vectors in $\mathcal{V}$ with $\mathsf{T}\log\gamma^{-1}$ samples with probability at least $1 - ((\ell k)^{\ell} + n)\gamma$. Rewriting, so that the failure probability is γ leads to the statement of the lemma. □

Proof of Lemma 5: Consider the special case when $|\mathsf{Maximal}(\mathcal{V})| = 1$ i.e. there exists a particular vector $\mathbf{v}$ in $\mathcal{V}$ whose support subsumes the support of all the other unknown vectors in $\mathcal{V}$. In that case, for each set $\mathcal{C} \subseteq A \in \mathsf{Maximal}(\mathcal{V})$, $|\mathcal{C}| \le \ell$, we must have that $\big|\bigcup_{i\in\mathcal{C}} \mathcal{S}(i)\big| > 0$ (as there is only a single set in $\mathsf{Maximal}(\mathcal{V})$). On the other hand, if $|\mathsf{Maximal}(\mathcal{V})| \ge 2$, then we know that $\mathsf{Maximal}(\mathcal{V})$ is $(\ell-1)$-good and therefore, for each set $A \in \mathsf{Maximal}(\mathcal{V})$, there exists an ordered set $\mathcal{C}$ and an index $j \subseteq \cup_{A'\in\mathsf{Maximal}(\mathcal{V})}A'$, $|\mathcal{C}| \le \ell - 1$ such that $\mathcal{C} \subseteq A$ but $\mathcal{C} \not\subseteq A'$ for any other set A'; hence $\big|\bigcap_{i\in\mathcal{C}} \mathcal{S}(i)\big| > 0$ but $\big|\bigcap_{i\in\mathcal{C}\cup\{j\}} \mathcal{S}(i)\big| = 0$. In other words, there exists a set of size ℓ that is a subset of the union of sets in $\mathsf{Maximal}(\mathcal{V})$ but there does not exist any unknown vector that has 1 in all the indices indexed by the aforementioned set. Again, Algorithm 3 precisely checks this condition and therefore this completes the proof. □

Appendix C
Proof of Lemma 1 (Theorem 1 in [23])

We will start with a few additional notations and definitions: For a set of unknown vectors $\mathcal{V} \equiv \{\mathbf{v}^1, \mathbf{v}^2, \dots, \mathbf{v}^\ell\}$, let $\mathbf{A} \in \{0,1\}^{n\times\ell}$ denote the support matrix corresponding to $\mathcal{V}$ where each column vector $\mathbf{A}_i \in \{0,1\}^n$ represents the support of the i^{th} unknown vector $\mathbf{v}^i$.

Definition 4 (p-identifiable): The i^{th} column $\mathbf{A}_i$ of a binary matrix $\mathbf{A} \in \{0,1\}^{n\times\ell}$ with all distinct columns is called p-identifiable if there exists a set $S \subset [n]$ of at most p-indices and a binary string $\mathbf{a} \in \{0,1\}^p$ such that $\mathbf{A}_i|_S = \boldsymbol{a}$, and $\mathbf{A}_j|_S \ne \boldsymbol{a}$ for all $j \ne i$.

A binary matrix $\mathbf{A} \in \{0,1\}^{n\times\ell}$ with all distinct columns is called p-identifiable if there exists a permutation $\sigma : [\ell] \to [\ell]$ such that for all $i \in [\ell]$, the sub-matrix $\mathbf{A}^i$ formed by deleting the columns indexed by the set $\{\sigma(1), \sigma(2), \dots, \sigma(i-1)\}$ has at least one p-identifiable column.

Let $\mathcal{V}$ be set of ℓ unknown vectors in $\mathbb{R}^n$, and $\mathbf{A} \in \{0,1\}^{n\times\ell}$ be its support matrix. Let $\mathbf{B}$ be the matrix obtained by deleting duplicate columns of $\mathbf{A}$. The set $\mathcal{V}$ is called p-identifiable if $\mathbf{B}$ is p-identifiable.

Theorem (Theorem 2 in [23]): Any $n \times \ell$, (with $n > \ell$) binary matrix with all distinct columns is p-identifiable for some $p \le \log\ell$.

Proof: Suppose $\mathbf{A}$ is the said matrix. Since all the columns of $\mathbf{A}$ are distinct, there must exist an index $i \in [n]$ which is not the same for all columns in $\mathbf{A}$. We must have $|\mathsf{occ}((i), a)| \le \ell/2$ for some $a \in \{0,1\}$. Subsequently, we consider the columns of $\mathbf{A}$ indexed by the set $\mathsf{occ}((i), a)$ and can repeat the same step. Evidently, there must exist an index $j \in [n]$ such that $|\mathsf{occ}((i), \mathbf{a})| \le \ell/4$ for some $\mathbf{a} \in \{0,1\}^2$. Clearly, we can repeat this step at most $\log\ell$ times to find $C \subset [n]$ and $\mathbf{a} \in \{0,1\}^{\le\log\ell}$ such that $|\mathsf{occ}(C, \mathbf{a})| = 1$ and therefore the column in $\mathsf{occ}(C, \mathbf{a})$ is p-identifiable. We denote the index of this column as $\sigma(1)$ and form the sub-matrix $\mathbf{A}^1$ by deleting the column. Again, $\mathbf{A}^1$ has $\ell - 1$ distinct columns and by repeating similar steps, $\mathbf{A}^1$ has a column that is $\log(\ell-1)$ identifiable. More generally, $\mathbf{A}^i$ formed by deleting the columns indexed in the set $\{\sigma(1), \sigma(2), \ldots, \sigma(i-1)\}$, has a column that is $\log(\ell - i)$ identifiable with the index (in $\mathbf{A}$) of the column having the unique sub-string (in $\mathbf{A}^i$) denoted by $\sigma(i)$. Thus the lemma is proved. □

A. How Algorithm 1 Works

Next, we present an algorithm (see Algorithm 1) for support recovery of all the ℓ unknown vectors $\mathcal{V} \equiv \{\boldsymbol{v}^1, \ldots, \boldsymbol{v}^\ell\}$ when $\mathcal{V}$ is p-identifiable. In particular, we show that if $\mathcal{V}$ is p-identifiable, then computing $|\mathsf{occ}(C, \mathbf{a})|$ for every subset of p and $p+1$ indices is sufficient to recover the supports.

The proof follows from the observation that for any subset of p indices $C \subset [n]$, index $j \in [n] \setminus C$ and $\mathbf{a} \in \{0,1\}^p$, $|\mathsf{occ}(C, \mathbf{a})| = |\mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 1))| + |\mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 0))|$. Therefore if one of the terms in the RHS is 0 for all $j \in [n] \setminus C$, then all the vectors in $\mathsf{occ}(C, \mathbf{a})$ share the same support.

Also, if some two vectors $\mathbf{u}, \mathbf{v} \in \mathsf{occ}(C, \mathbf{a})$ do not have the same support, then there will exist at least one index $j \in [n] \setminus C$ such that $\mathbf{u} \in \mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 1))|$ and $\mathbf{v} \in \mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 0))$ or the other way round, and therefore $|\mathsf{occ}(C \cup \{j\}, (\mathbf{a}, 1))| \notin \{0, |\mathsf{occ}(C, \mathbf{a})|\}$. Algorithm 1 precisely checks for this condition. The existence of some vector $\mathbf{v} \in \mathcal{V}$ (p-identifiable column), a subset of indices $C \subset [n]$ of size p, and a binary sub-string $\mathbf{b} \in \{0,1\}^{\leq p}$ follows from the fact that $\mathcal{V}$ is p-identifiable. Let us denote the subset of unknown vectors with distinct support in $\mathcal{V}$ by $\mathcal{V}^1$.

Once we recover the p-identifiable column of $\mathcal{V}^1$, we mark it as $\mathbf{u}^1$ and remove it from the set (if there are multiple p-identifiable columns, we arbitrarily choose one of them). Subsequently, we can modify the $|\mathsf{occ}(\cdot)|$ values for all $S \subseteq [n], |S| \in \{p, p+1\}$ and $\mathbf{t} \in \{0,1\}^p \cup \{0,1\}^{p+1}$ as follows:

$$\left|\mathsf{occ}^2(S, \mathbf{t})\right| \triangleq |\mathsf{occ}(S, \mathbf{t})| - |\mathsf{occ}(C, \mathbf{b})| \times \mathbf{1}[\mathsf{supp}(\mathbf{u}^1)|_S = \mathbf{t}]. \tag{3}$$

Notice that, Equation 3 computes $\left|\mathsf{occ}^2(S, \mathbf{t})\right| = \left|\{\mathbf{v}^i \in \mathcal{V}^2 \mid \mathsf{supp}(\mathbf{v}^i)|_S = \mathbf{t}\}\right|$ where $\mathcal{V}^2$ is formed by deleting all copies of $\mathbf{u}^1$ from $\mathcal{V}$. Since $\mathcal{V}^1$ is p-identifiable, there exists a p-identifiable column in $\mathcal{V}^1 \setminus \{\mathbf{u}^1\}$ as well which we can recover. More generally for $q > 2$, if $\mathbf{u}^{q-1}$ is the p-identifiable column with the unique binary sub-string $\mathbf{b}^{q-1}$ corresponding to the set of indices C^{q-1}, we will have for all $S \subseteq [n], |S| \in \{p, p+1\}$ and $\mathbf{t} \in \{0,1\}^p \cup \{0,1\}^{p+1}$

$$|\mathsf{occ}^q(S, \mathbf{t})| \triangleq \left|\mathsf{occ}^{q-1}(S, \mathbf{t})\right| - \left|\mathsf{occ}^{q-1}(C^{q-1}, \mathbf{b}^{q-1})\right| \times \mathbf{1}[\mathsf{supp}(\mathbf{u}^{q-1})|_S = \mathbf{t}]$$

implying $|\mathsf{occ}^q(S, \mathbf{t})| = \left|\{\mathbf{v}^i \in \mathcal{V}^q \mid \mathsf{supp}(\mathbf{v}^i)|_S = \mathbf{t}\}\right|$ where $\mathcal{V}^q$ is formed deleting all copies of $\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^{q-1}$ from $\mathcal{V}$. Applying these steps recursively and repeatedly using the property that $\mathcal{V}$ is p-identifiable, we can recover all the vectors present in $\mathcal{V}$.

Appendix D
Technical Lemmas

Lemma 11 (Hoeffding's Inequality for Bounded Random Variables): Let $X_1, X_2, \ldots, X_m$ be independent random variables strictly bounded in the interval $[a, b]$. Let $\mu = m^{-1} \sum_i \mathbb{E} X_i$. In that case, we must have

$$\Pr\Big(\Big|\frac{1}{m}\sum_{i=1}^m X_i - \mu\Big| \geq t\Big) \leq 2\exp\Big(-\frac{2mt^2}{(b-a)^2}\Big).$$

Lemma 12 (Gaussian Concentration Inequality): Consider a random variable Z distributed according to $\mathcal{N}(0, \sigma^2)$. In that case, we must have $\Pr(|Z| \geq t) \leq 2\exp(-t^2/2)$ for any $t > 0$.

Lemma 13 (Gaussian Anti-Concentration Inequality): Consider a random variable Z distributed according to $\mathcal{N}(0, \sigma^2)$. In that case, we must have $\Pr(|Z| \leq t) \leq \sqrt{\frac{2}{\pi}} \cdot \frac{t}{\sigma}$ for any $t < \sigma\sqrt{\pi}/\sqrt{2}$.

Proof: By simple calculations, we can have

$$\Pr(|Z| < t) \leq \int_{-t}^{t} \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi\sigma}} dx \leq \sqrt{\frac{2}{\pi}} \cdot \frac{t}{\sigma}.$$

□

Finally, we will also use the following well-known lemma stating that we can compute estimates of the expectation of any one-dimensional random variable with only a few samples similar to sub-gaussian random variables.

Lemma 14: For a random variable $x \sim \mathcal{P}$, there exists an algorithm (see Algorithm 5) that can compute an estimate u of $\mathbb{E}x$ such that $|u - \mathbb{E}x| \leq \epsilon$ with $O(\log \gamma^{-1} \mathbb{E}x^2/\epsilon^2)$ with probability at least $1 - \gamma$.

Proof of Lemma 14: Suppose we obtain m independent samples $x^{(1)}, x^{(2)}, \ldots, x^{(m)} \sim \mathcal{P}$. We use the median of means trick to compute u, an estimate of $\mathbb{E}x$. We will partition m samples obtained from $\mathcal{P}$ into $B = \lceil m/m' \rceil$ batches each containing m' samples each. In that case let us denote S^j to be the sample mean of the j^{th} batch i.e.

$$S^j = \sum_{s \in \text{Batch } j} \frac{x^{(s)}}{m'}.$$

We will estimate the true mean $\mathbb{E}x$ by computing u where $u \triangleq \mathsf{median}(\{S^j\}_{j=1}^B)$. For a fixed batch j, we can use Chebychev's inequality to say that

$$\Pr\Big(\left|S^j - \mathbb{E}x\right| \geq \epsilon\Big) \leq \frac{\mathbb{E}x^2}{t\epsilon^2} \leq \frac{1}{3}$$

for $t = O(\mathbb{E}x^2/\epsilon^2)$. Therefore for each batch j, we define an indicator random variable $Z_j = \mathbf{1}[\left|S^j - \mathbb{E}x\right| \geq \epsilon]$ and from our previous analysis we know that the probability of Z_j being 1 is less than $1/3$. It is clear that $\mathbb{E}\sum_{j=1}^B Z_j \leq B/3$ and on the other hand $|u - \mathbb{E}x| \geq \epsilon$ iff $\sum_{j=1}^B Z_j \geq B/2$. Therefore, due to the fact that Z_j's are independent, we can use Chernoff bound to conclude the following:

$$\Pr\Big(|u - \mathbb{E}x| \geq \epsilon\Big) \leq \Pr\Big(\Big|\sum_{j=1}^B Z_j - \mathbb{E}\sum_{j=1}^B Z_j\Big| \geq \frac{\mathbb{E}\sum_{j=1}^B Z_j}{2}\Big)$$
$$\leq 2e^{-B/36}.$$

Hence, for $B = 36\log\gamma^{-1}$, the estimate u is at most ϵ away from the true mean $\mathbb{E}x$ with probability at least $1-\gamma$. Therefore the sufficient sample complexity is $m = O(\log\gamma^{-1}\mathbb{E}x^2/\epsilon^2)$. □

References

[1] S. Pal and A. Mazumdar, "On learning mixture models with sparse parameters," in *Int. Conf. Artif. Intell. Statist.*, 2022, pp. 9182–9213.

[2] F. Moosman and D. Peel, *Finite Mixture Models*, vol. 3. Hoboken, NJ, USA: Wiley, 2000, p. 4.

[3] D. M. Titterington, A. F. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. Hoboken, NJ, USA: Wiley, 1985.

[4] S. Dasgupta, "Learning mixtures of Gaussians," in *Proc. 40th Annu. Symp. Found. Comput. Sci.*, Aug. 1999, pp. 634–644.

[5] D. Achlioptas and F. McSherry, "On spectral learning of mixtures of distributions," in *Proc. Conf. Learn. Theory*, Jan. 2005, pp. 458–469.

[6] A. T. Kalai, A. Moitra, and G. Valiant, "Efficiently learning mixtures of two Gaussians," in *Proc. 42nd ACM Symp. Theory Comput.*, Jun. 2010, pp. 553–562.

[7] M. Belkin and K. Sinha, "Polynomial learning of distribution families," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, Jul. 2010, pp. 103–112.

[8] A. Sanjeev and R. Kannan, "Learning mixtures of arbitrary Gaussians," in *Proc. 33rd Annu. ACM Symp. Theory Comput.*, Jul. 2001, pp. 247–257.

[9] A. Moitra and G. Valiant, "Settling the polynomial learnability of mixtures of Gaussians," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, Oct. 2010, pp. 93–102.

[10] M. Sandler, R. Odonnell, and R. A. Servedio, "Learning mixtures of product distributions over discrete domains," in *Proc. 46th Annu. IEEE Symp. Found. Comput. Sci. (FOCS05)*, May 2008, pp. 501–510.

[11] S.-O. Chan, I. Diakonikolas, R. A. Servedio, and X. Sun, "Efficient density estimation via piecewise polynomial approximation," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, May 2014, pp. 604–613.

[12] J. Acharya, I. Diakonikolas, J. Li, and L. Schmidt, "Sample-optimal density estimation in nearly-linear time," in *Proc. 28th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, Jan. 2017, pp. 1278–1289.

[13] S. B. Hopkins and J. Li, "Mixture models, robustness, and sum of squares proofs," in *Proc. 50th Annu. ACM SIGACT Symp. Theory Comput.*, Jun. 2018, pp. 1021–1034.

[14] I. Diakonikolas, D. M. Kane, and A. Stewart, "List-decodable robust mean estimation and learning mixtures of spherical Gaussians," in *Proc. 50th Annu. ACM SIGACT Symp. Theory Comput.*, Jun. 2018, pp. 1047–1060.

[15] P. K. Kothari, J. Steinhardt, and D. Steurer, "Robust moment estimation and improved clustering via sum of squares," in *Proc. 50th Annu. ACM SIGACT Symp. Theory Comput.*, Jun. 2018, pp. 1035–1046.

[16] M. Hardt and E. Price, "Tight bounds for learning a mixture of two Gaussians," in *Proc. 47th Annu. ACM Symp. Theory Comput.*, Jun. 2015, pp. 753–760.

[17] N. Verzelen and E. Arias-Castro, "Detection and feature selection in sparse mixture models," *Ann. Statist.*, vol. 45, no. 5, pp. 1920–1950, Oct. 2017.

[18] E. Arias-Castro and X. Pu, "A simple approach to sparse clustering," *Comput. Statist. Data Anal.*, vol. 105, pp. 217–228, Jan. 2017.

[19] M. Azizyan, A. Singh, and L. Wasserman, "Minimax theory for high-dimensional Gaussian mixtures with sparse mean separation," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2013, pp. 2139–2147.

[20] I. M. Johnstone and A. Y. Lu, "On consistency and sparsity for principal components analysis in high dimensions," *J. Amer. Statist. Assoc.*, vol. 104, no. 486, pp. 682–693, Jan. 2009.

[21] G. Bresler, G. H. Chen, and D. Shah, "A latent source model for online collaborative filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Oct. 2014, pp. 1–11.

[22] V. Gandikota, A. Mazumdar, and S. Pal, "Recovery of sparse linear classifiers from mixture of responses," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2020, pp. 1–9.

[23] V. Gandikota, A. Mazumdar, and S. Pal, "Support recovery of sparse signals from a mixture of linear measurements," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 19082–19094.

[24] D. Hsu and S. M. Kakade, "Learning mixtures of spherical gaussians: Moment methods and spectral decompositions," in *Proc. 4th Conf. Innov. Theor. Comput. Sci.*, Jan. 2013, pp. 11–20.

[25] A. Feller, E. Greif, N. Ho, L. Miratrix, and N. Pillai, "Weak separation in mixture models and implications for principal stratification," 2016, *arXiv:1602.06595*.

[26] N. Ho and X. Nguyen, "Convergence rates of parameter estimation for some weakly identifiable finite mixtures," *Ann. Statist.*, vol. 44, no. 6, pp. 2726–2755, Dec. 2016.

[27] T. Manole and N. Ho, "Uniform convergence rates for maximum likelihood estimation under two-component Gaussian mixture models," 2020, *arXiv:2006.00704*.

[28] P. Heinrich and J. Kahn, "Strong identifiability and optimal minimax rates for finite mixture estimation," *Ann. Statist.*, vol. 46, no. 6, pp. 2844–2870, Dec. 2018.

[29] A. Krishnamurthy, A. Mazumdar, A. McGregor, and S. Pal, "Algebraic and analytic approaches for parameter learning in mixture models," in *Proc. 31st Int. Conf. Algorithmic Learn. Theory (ALT)*, Jan. 2020, pp. 468–489.

[30] S. Arora, R. Ge, R. Kannan, and A. Moitra, "Computing a nonnegative matrix factorization–Provably," *SIAM J. Comput.*, vol. 45, no. 4, pp. 1582–1611, Jan. 2016.

[31] D. L. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, Dec. 2003, pp. 1141–1148.

[32] M. Slawski, M. Hein, and P. Lutsik, "Matrix factorization with binary components," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, Dec. 2013, pp. 3210–3218.

Arya Mazumdar (Senior Member, IEEE) received the Ph.D. degree from the University of Maryland, College Park, in 2011. He is currently a Professor in data science with the University of California at San Diego. In 2015 and 2021, he was an Assistant followed by an Associate Professor with the College of Information and Computer Sciences, University of Massachusetts at Amherst. Prior to that, he was a Faculty Member with the University of Minnesota-Twin Cities (2013–2015) and a Post-Doctoral Researcher with Massachusetts Institute of Technology (2011–2012). He was a recipient of multiple awards, including a Distinguished Dissertation Award for his Ph.D. thesis (2011), the NSF CAREER Award (2015), an EURASIP Best Paper Award (2020), and the IEEE ISIT Jack K. Wolf Student Paper Award (2010). He is a Distinguished Lecturer of the IEEE Information Theory Society (2023–2024). He currently serves as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION THEORY and as an Area Editor for *Foundation and Trends in Communication and Information Theory Series* (Now Publishers). His research interests include coding theory, information theory, statistical learning, and distributed optimization.

Soumyabrata Pal received the B.Tech. degree from Indian Institute of Technology Kharagpur, India, in 2016, and the Ph.D. degree from the College of Information and Computer Sciences, University of Massachusetts at Amherst, advised by Prof. Arya Mazumdar. He is currently a Research Scientist with Adobe Research, Bengaluru, India. Prior to this, he was a Post-Doctoral Researcher with Google Research, Bengaluru. His research interests include theoretical machine learning, applied statistics, and information theory.